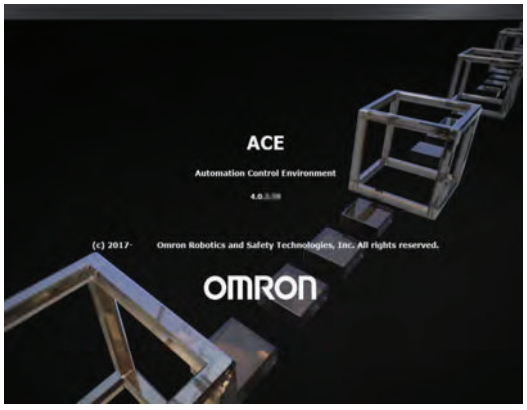
OMRON

Automation Control Environment (ACE) Version 4

User's Manual



I633-E-11

NOTE

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.
No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
Company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

Introduction

This manual is OMRON's original instructions describing how to use Automation Control Environment (ACE) software.
Please read this manual and make sure you understand the ACE software features, functions, and concepts before attempting to use it.

Intended Audience

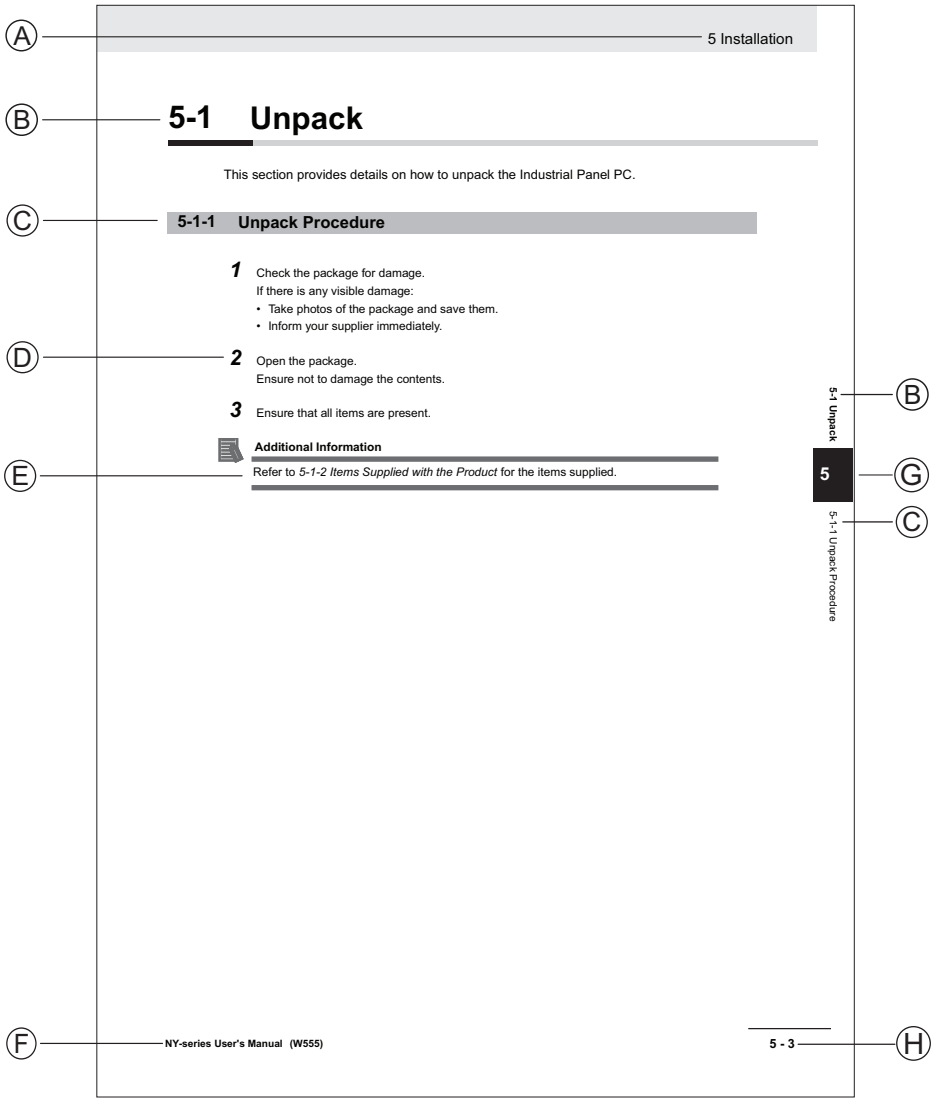
This manual is intended for the following personnel.

- Personnel in charge of introducing Factory Automation systems.
- Personnel in charge of designing Factory Automation systems.
- Personnel in charge of installing and maintaining Factory Automation systems.
- Personnel in charge of managing Factory Automation systems and facilities.

Manual Information

Page Structure

The following page structure is used in this manual.



Note: This illustration is provided as a sample. It will not literally appear in this manual.

Item	Explanation	Item	Explanation
A	Level 1 heading	E	Special Information
B	Level 2 heading	F	Manual name
C	Level 3 heading	G	Page tab with the number of the main section
D	Step in a procedure	H	Page number

Special Information

Special information in this manual is classified as follows:



Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.



Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.



Additional Information

Additional information to read as required.
This information is provided to increase understanding or make operation easier.



Version Information

Information on differences in specifications and functionality between different versions.

Sections in this Manual

1 Introduction

2 Installation and Uninstallation

3 Creating a Project

4 Online Connections to SmartControllers

5 Understanding the ACE User Interface

6 Programming with ACE Software

7 Robot Controller Configuration and Setup

8 Application Manager Configuration and Setup

9 Troubleshooting

A Appendices

1 2 3 4 5 6 7 8 9 A

CONTENTS

Introduction 1
Intended Audience 1

Manual Information 2
Page Structure 2
Special Information 2

Sections in this Manual 5

Terms and Conditions Agreement 12
Warranty and Limitations of Liability 12
Application Considerations 12
Disclaimers 13

Safety Precautions 14
Definition of Precautionary Information 14
Symbols 14
Warnings 14
Caution 15

Precautions for Safe Use 17

Precautions for Correct Use 18

Related Manuals 19

Revision History 20

Section 1 Introduction

1-1 Software Overview 1-2

1-2 Configuration Wizards 1-3
1-2-1 Wizard Elements 1-3
1-2-2 Wizard Functionality in Emulation Mode 1-5

1-3 Licenses 1-6
1-3-1 PC Licenses 1-7

1-4 Software Features 1-9
1-4-1 Emulation Mode 1-9
1-4-2 Application Samples 1-12

1-5 Robot Concepts 1-15
1-5-1 Coordinate Systems 1-15
1-5-2 Calibrations 1-22

1-6 Basic Robot Motion 1-27
1-6-1 Speed, Acceleration, Deceleration 1-27
1-6-2 Approach and Depart 1-27
1-6-3 Arm Configuration 1-28
1-6-4 Location Precision 1-29
1-6-5 Continuous Path Motion 1-29
1-6-6 Joint-Interpolated Motion vs. Straight-Line Motion 1-29
1-6-7 Performance Considerations 1-30

1-7 Understanding Belts (Conveyors) 1-31
1-7-1 Indexing versus Tracking Conveyors 1-31

Section 2 Installation and Uninstallation

2-1 Installing the Software....2-2
2-1-1 System Requirements....2-2
2-1-2 ACE Installation Instructions....2-2
2-1-3 Usage Considerations....2-3
2-1-4 Other Installed Items and Software....2-4
2-2 Uninstalling the Software....2-7

Section 3 Creating a Project

3-1 Starting and Exiting the ACE Software....3-2
3-1-1 Starting ACE Software....3-2
3-1-2 Starting Application Manager....3-3
3-1-3 Automatic Project Launch on System Boot Up Procedure....3-3
3-1-4 Clear Application Manager Memory....3-6
3-1-5 Exiting the ACE Software....3-7
3-2 Creating A Project File....3-9
3-3 Saving the Project File....3-12
3-4 Saving a Project File Under a Different Name....3-13
3-5 Creating a Project from an Online Connection....3-14
3-6 Closing a Project and Returning to the Start Page....3-15
3-7 Opening a Project File....3-16
3-8 Importing a Project File....3-18
3-8-1 Importing from the Start Page....3-18
3-8-2 Importing from the Menu Bar in the Application Window....3-18
3-8-3 Importing an .awp2 ACE Project File....3-18
3-8-4 Importing from Version Control....3-19
3-9 Exporting a Project File....3-20
3-9-1 Exporting from the Start Page....3-20
3-9-2 Exporting from the Application Window....3-20
3-9-3 Saving an Exported Project File....3-21
3-10 Password Protection for Project Files....3-23
3-10-1 Setting Password Protection for Project Files....3-23
3-10-2 Removing Password Protection for Project Files....3-23
3-11 Project Version Numbers....3-25

Section 4 Online Connections to Robot Controllers

4-1 Going Online with a Robot Controller....4-2
4-1-1 Going Online from the Start Page....4-2
4-1-2 Detect and Configure Controller....4-4
4-1-3 Going Online from an Open Project....4-5
4-2 Going Offline with a Robot Controller....4-6
4-3 Memory Comparison....4-7

Section 5 Understanding the ACE User Interface

5-1 The Main Application Window....5-3
5-1-1 Main Operation Window....5-3
5-1-2 Menu Command Structure....5-4

5-1-3 Toolbar Icons 5-9
5-1-4 Operation Toolbar Icons 5-10
5-1-5 Multiview Explorer 5-10
5-1-6 Filter Pane 5-12
5-1-7 Project Shortcut View 5-13
5-1-8 Status Bar 5-14
5-1-9 Edit Pane 5-14
5-1-10 Multifunction Tabs 5-15
5-1-11 Special Tools 5-15
5-1-12 Other Application Window Functions 5-16

5-2 Toolbox 5-23

5-3 3D Visualizer 5-24
5-3-1 3D Visualizer Basic Features 5-24
5-3-2 Creating a 3D Workspace 5-25
5-3-3 3D Visualizer Window 5-25
5-3-4 Graphical Collision Detection 5-29

5-4 Output Tab Page 5-31

5-5 Cross Reference Tab Page 5-32

5-6 Build Tab Page 5-33

5-7 Event Log 5-34

5-8 V+ Jog Control 5-35
5-8-1 Robot Section 5-35
5-8-2 Current Position Section 5-36
5-8-3 Jog Control Section 5-36
5-8-4 World, Joint, Tool Selection 5-37
5-8-5 Location Section 5-37

5-9 Task Status Control 5-40
5-9-1 Online/Offline 5-40
5-9-2 Robot Power Control 5-40
5-9-3 Monitor Speed Setting 5-40
5-9-4 Open the Monitor Window 5-41
5-9-5 Task Manager 5-41
5-9-6 I/O Watcher 5-43
5-9-7 V+ File Browser 5-43
5-9-8 Virtual Front Panel 5-45
5-9-9 Fieldbus Status 5-46
5-9-10 Profiler 5-48
5-9-11 Application Manager Control 5-49
5-9-12 Control Buttons 5-49

5-10 Vision Window 5-50
5-10-1 Zoom Level 5-50

5-11 V+ Watch 5-52
5-11-1 Adding Variables to the V+ Watch Window 5-52

5-12 Search and Replace 5-53
5-12-1 Search Options 5-53
5-12-2 Button Functions 5-53

5-13 User Management 5-55
5-13-1 Passwords 5-56
5-13-2 Engineer Accessible Items 5-56
5-13-3 Technician Accessible Items 5-56
5-13-4 Operator Accessible Items 5-56

5-14 Project Options 5-57
5-14-1 Color Theme 5-57
5-14-2 Project Settings 5-58
5-14-3 Window 5-58
5-14-4 V+ Programs 5-59
5-14-5 V+ Variables 5-60
5-14-6 Uninterruptible Power Supply (UPS) 5-62

5-14-7 Connection Settings 5-64
5-14-8 3D Visualizer 5-65

Section 6 Programming with ACE Software

6-1 Introduction 6-2

6-2 Robot Controller Programming 6-3
6-2-1 V+ Module and Program Structure 6-3
6-2-2 Creating a New V+ Module 6-4
6-2-3 Save a V+ Module to the PC 6-4
6-2-4 Show Global Variable References 6-4
6-2-5 Show Program Call Commands 6-5
6-2-6 Pull V+ Memory 6-5
6-2-7 V+ Editor 6-5
6-2-8 V+ Program Debugging 6-10

6-3 V+ Variables 6-11
6-3-1 V+ Variable Types 6-11

6-4 V+ Variable Editor 6-13
6-4-1 V+ Variable Names 6-13
6-4-2 V+ Variable Properties 6-13
6-4-3 Creating a New V+ Variable 6-15
6-4-4 Editing an Existing V+ Variable 6-16
6-4-5 Filtering by Types and Categories 6-16
6-4-6 Other V+ Variable Editor Functions 6-17

6-5 Application Manager Programming 6-19
6-5-1 Creating a New C# Program 6-19
6-5-2 C# Program Editor 6-20

6-6 C# Variable Objects 6-26
6-6-1 C# Variable Object Types 6-26
6-6-2 Creating a New C# Variable Object 6-26
6-6-3 Other C# Variable Object Function Keywords 6-26
6-6-4 Editing C# Variable Objects 6-26

6-7 Operation Programming 6-28
6-7-1 Automatically Create an Operation 6-29
6-7-2 Manually Create an Operation 6-30
6-7-3 Manually Create a Process Operation 6-32
6-7-4 Add Skills to an Operation 6-33
6-7-5 Skill Icons 6-34
6-7-6 Edit an Operation 6-35
6-7-7 Operation Skills 6-36
6-7-8 Building Custom Skills 6-38

Section 7 Robot Controller Configuration and Setup

7-1 Overview 7-2

7-2 Controller Settings 7-5
7-2-1 Configuration 7-5
7-2-2 Parameters 7-6
7-2-3 Control 7-7
7-2-4 Upgrade 7-9
7-2-5 Configure 7-14
7-2-6 Backup/Restore 7-31
7-2-7 Encoders 7-33

7-3 Save Configuration 7-35

7-4 Monitor Window 7-37
7-4-1 Using the Monitor Window 7-37

7-4-2 Monitor Command Entry 7-38
7-4-3 Monitor Command Keyword Parameters 7-38
7-4-4 Canceling a Monitor Command 7-39
7-4-5 Executing a Monitor Command 7-39

7-5 Robot Objects 7-40
7-5-1 Configure Robots 7-42
7-5-2 I/O EndEffectors 7-43
7-5-3 3D Visualization 7-46
7-5-4 Configuration 7-47
7-5-5 End-Effector 7-50
7-5-6 Location 7-51
7-5-7 Object 7-51
7-5-8 Configure 7-51
7-5-9 Control 7-56

Section 8 Application Manager Configuration and Setup

8-1 Remote Application Manager 8-3
8-1-1 Remote Application Manager Set Up 8-3
8-1-2 ACE Server Instance 8-6
8-1-3 ACE Client Instance 8-7
8-1-4 Synchronization 8-8
8-1-5 Remote User Management 8-9

8-2 3D Visualization 8-10
8-2-1 Adding Shapes 8-10
8-2-2 Adding and Configuring CAD Data 8-11
8-2-3 Update 3D Shapes 8-14
8-2-4 Connection Points 8-14

8-3 Robot Vision Manager 8-18
8-3-1 Robot Vision Manager Object Configuration 8-19
8-3-2 Saving and Loading Calibration Data 8-19
8-3-3 Belt Latch Calibration 8-19
8-3-4 Camera Calibration 8-21
8-3-5 Belt Calibration 8-23
8-3-6 Gripper Offset Table 8-26
8-3-7 Vision Sequence 8-27
8-3-8 Overlap Tool 8-33
8-3-9 Communication Tool 8-34
8-3-10 Robot Frame 8-38
8-3-11 Robot Location 8-39
8-3-12 Robot Precision Point 8-41
8-3-13 Robot Transfer Path 8-42

8-4 Camera Objects 8-46
8-4-1 Virtual Cameras 8-46
8-4-2 Emulation Cameras 8-54
8-4-3 Physical Cameras 8-56
8-4-4 Custom Devices 8-57
8-4-5 Visualizer Capture Device 8-58

8-5 Configuration 8-61
8-5-1 Controller Connection Startup 8-61
8-5-2 Data Mapper 8-63
8-5-3 Note Object 8-64
8-5-4 OPC Container 8-65
8-5-5 Program System Startup 8-68
8-5-6 Recipe Overview 8-69
8-5-7 Recipe Manager 8-70
8-5-8 Recipe Editor 8-74
8-5-9 Recipe Manager Script 8-80

8-6 Feeders 8-82
8-6-1 AnyFeeder Object 8-82

8-6-2 I/O Feeder Object 8-88
8-6-3 iPF Object 8-90

8-7 Process Objects 8-97
8-7-1 Part Buffer Object 8-98
8-7-2 Part Target Object 8-101
8-7-3 Part Object 8-104
8-7-4 Belt Object 8-108
8-7-5 Process Manager Object 8-117
8-7-6 Allocation Script Object 8-160
8-7-7 Pallet Object 8-161
8-7-8 Vision Refinement Station Object 8-167
8-7-9 Process Manager Customization 8-170
8-7-10 Process Operation Object 8-189

8-8 Vision Tools 8-190
8-8-1 Adding Vision Tools 8-192
8-8-2 Vision Tool Editor 8-193
8-8-3 Region of Interest 8-194
8-8-4 Color Spaces 8-197
8-8-5 Finder Tools 8-199
8-8-6 Inspection Tools 8-247
8-8-7 Reader Tools 8-280
8-8-8 Calculation Tools 8-295
8-8-9 Image Process Tools 8-304
8-8-10 Custom Tools 8-323

Section 9 Troubleshooting

9-1 Event Log 9-2
9-1-1 Accessing the Event Log 9-2
9-1-2 Display Messages by Event Type 9-3
9-1-3 Sorting Messages 9-3
9-1-4 Selecting and Copying Messages 9-3

9-2 Clearing the Event Log 9-4

9-3 Error Log 9-5
9-3-1 Accessing the Error Log 9-5

9-4 Device Node Event Log 9-7
9-4-1 Accessing and Clearing the Device Node Event Log 9-7
9-4-2 Device Node Configuration Conflicts 9-8

9-5 Diagnostic Summary 9-9
9-5-1 Saving a Core Dump File 9-9

Appendices

A-1 Latch Signal Test A-2
A-1-1 Latch Signal Test Procedure A-2

A-2 Uninterruptible Power Supply (UPS) Installation A-3
A-2-1 UPS Software Required A-3
A-2-2 AC-AC Hardware Configuration A-3
A-2-3 DC-DC Hardware Configuration A-7

A-3 Commissioning Physical Hardware A-13

A-4 Version Control Introduction A-14
A-4-1 Git Configuration A-14
A-4-2 Multiple User Configuration A-15

Terms and Conditions Agreement

Warranty and Limitations of Liability

Warranty

- **Exclusive Warranty**
  Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, expressed or implied.
- **Limitations**
  OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.
  Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.
- **Buyer Remedy**
  Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitations of Liability

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY. Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability for Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.
NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

- Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.
- Omron Companies shall not be responsible for the operation of the user accessible operating system (e.g. Windows, Linux), or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Safety Precautions

Definition of Precautionary Information

The following notation is used in this manual to provide precautions required to ensure safe usage of the robot. The safety precautions that are provided are extremely important to safety.
Always read and heed the information provided in all safety precautions.
The following notation is used.

⚠ **DANGER**	Identifies an imminently hazardous situation which, if not avoided, is likely to result in serious injury, and might result in fatality or severe property damage.
⚠ **WARNING**	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.
⚠ **CAUTION**	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

Symbols

	The circle and slash symbol indicates operations that you must not do. The specific operation is shown in the circle and explained in text. This example indicates prohibiting disassembly.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for electric shock.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a general precaution.
	The filled circle symbol indicates operations that you must do. The specific operation is shown in the circle and explained in text. This example shows a general precaution for something that you must do.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for high temperatures.

Warnings

⚠ WARNING

Cybersecurity

To maintain the security and reliability of the system, a robust cybersecurity defense program should be implemented, which may include some or all of the following:

Anti-virus protection

- Install the latest commercial-quality anti-virus software on the computer connected to the control system and keep the software and virus definitions up-to-date.
- Scan USB drives or other external storage devices before connecting them to control systems and equipment.

Security measures to prevent unauthorized network access

- Install physical controls so that only authorized personnel can access control systems and equipment.
- Reduce connections to control systems and equipment via networks to prevent access from untrusted devices.
- Install firewalls to block unused communications ports and limit communication between systems. Limit access between control systems and systems from the IT network.
- Control remote access and adopt multifactor authentication to devices with remote access to control systems and equipment.
- Set strong password policies and monitor for compliance frequently.

Data input and output protection

- Backup data and keep the data up-to-date periodically to prepare for data loss.
- Validate backups and retention policies to cope with unintentional modification of input/output data to control systems and equipment.
- Validate the scope of data protection regularly to accommodate changes.
- Check validity of backups by scheduling test restores to ensure successful recovery from incidents.
- Safety design, such as emergency shutdown and fail-soft operations in case of data tampering and incidents.



Additional recommendations

- When using an external network environment to connect to an unauthorized terminal such as a SCADA, HMI or to an unauthorized server may result in network security issues such as spoofing and tampering.
- You must take sufficient measures such as restricting access to the terminal, using a terminal equipped with a secure function, and locking the installation area by yourself.
- When constructing network infrastructure, communication failure may occur due to cable disconnection or the influence of unauthorized network equipment.
- Take adequate measures, such as restricting physical access to network devices, by means such as locking the installation area.
- When using devices equipped with an SD Memory Card, there is a security risk that a third party may acquire, alter, or replace the files and data in the removable media by removing or unmounting the media.
- Please take sufficient measures, such as restricting physical access to the Controller or taking appropriate management measures for removable media, by means of locking and controlling access to the installation area.
- Educate employees to help them identify phishing scams received via email on systems that will connect to the control network.

Caution

⚠ CAUTION

Always physically disconnect the robot from the EtherCAT network before disabling its EtherCAT communication.



Precautions for Safe Use

- Ensure adequate safety measures are taken if the safety timeout is disabled.
- Robot maintenance and troubleshooting procedures require the safety timeout to be enabled. The robot's flashing high power indicator is used to verify that the lamp is not burned out and this should be considered before disabling the safety timeout feature.
- The user is responsible to connect and install the appropriate visual indicator(s) as per the requirements of UL 1740. This function is required to be enabled to meet UL 1740 requirements.
- Fieldbus communications are not secure. Additional measures must be taken if compliance with security standards is required.

Precautions for Correct Use

- ACE software version 3.8 (and older) cannot connect to a controller that ACE software version 4.2 (or higher) has connected to previously until a zero memory or reboot have been executed.
- If you overwrite the ACE project V+ memory with the contents of the controller, there is no way to recover previous project data unless a backup copy has been previously created. Consider selecting **Save As...** from the File Menu to save a new backup copy of the project before pushing the program to the Controller memory.
- You cannot execute a program with an error present.
- You cannot execute a Process Manager when an error is present on an associated object.
- Using the **Jog Appro** Button will cause straight-line motion to occur. Monitor the robot during this movement to avoid collisions with obstacles between the starting location and the destination location.
- A user signed in with Technician or Operator access cannot edit users or access the User Manager. Only a user with Engineer access can edit users with the User Manager.
- The User Manager implements a basic level of user-access security. For applications that require a higher level of security, implement a security scheme within a custom user-interface. This could be based on the network login credentials from Windows or other similar access-control methods.
- If you are inexperienced with the object-oriented C# programming language, it is recommended that you explore Microsoft's documentation and tutorials to become familiar before attempting to create or edit these types of programs.
- No other tasks should be running while a V+ upgrade is underway.
- The system can become corrupted if power is lost while a V+ upgrade is underway.
- If a device node network configuration is modified and saved, the physical connections will be identified and restored accordingly on the next boot.
- I/O EndEffector settings are not saved on the controller with the Save Configuration function.
- Improper editing of robot joints, option bits, and kinematic parameters can cause the robot to malfunction or become inoperable. Editing must be performed by qualified personnel only.
- Disconnecting EtherCAT communications while the robot is transitioning between control states may cause an invalid fieldbus state. Cycle power to the robot to recover from this condition.

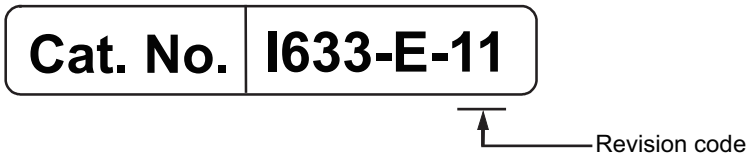
Related Manuals

Use the following related manuals for reference.

Manual Title	Description
Robot Safety Guide (Cat. No. I590)	Provides safety information for OMRON industrial robots.
T20 Pendant User's Manual (Cat. No. I601)	Provides information for the setup and use of the T20 Pendant.
eV+ Language User's Guide (Cat. No. I604)	Provides information about the eV+ language.
eV+ Language Reference Guide (Cat. No. I605)	Provides references to eV+ language and functionality.
Robot Vision Manager User's Manual (Cat. No. I667)	Provides information necessary for understanding the integration of cameras and V+ programs.
V+ Module Reference Manual (Cat. No. I668)	Provides references to V+ language and functionality.
V+ User's Manual (Cat. No. I671)	Provides references to V+ language and functionality.
V+ Keyword Reference Manual (Cat. No. I672)	Provides references to V+ Keyword use and functionality.
Automation Control Environment (ACE) Version 4 Camera Configuration User's Manual (Cat. No. I679)	Provides installation and operating instructions for ACE cameras.
S8BA-series Uninterruptible Power Supply (UPS) User's Manual (Cat. No. U726)	Provides installation and operating instructions for the S8BA UPS.
Sysmac Studio for Project Version Control Function Operation Manual (Cat. No. W589)	Provides version control information to properly save, import, export and manage projects in multi-user environments.
Sysmac Studio for Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)	Provides information that is necessary to use the robot control function of the NJ-series CPU Unit.
FH/FHV Series Vision System Operation Manual for Sysmac Studio (Cat. No. Z343)	Provides information for the integration and operation of the FH camera with Sysmac Studio.
FH/FHV Series Vision System User's Manual (Cat. No. Z365)	Provides information for proper operation of the FH camera within a network.
FH Series Vision System Hardware Setup Manual (Cat. No. Z366)	Contains information to install and properly wire the FH cameras.
Anyfeeder User's Guide (18876-000)	Provides information about the operation of the AnyFeeder unit.
Automation Control Environment (ACE) Version 4 Camera Configuration User's Manual (Cat. No. I679)	Provides information about the configuration of cameras used with ACE software.
Automatic Shutdown Software for Omron Uninterruptible Power Supply (UPS) PowerAttendant Lite for Windows Users Manual (K1L-D-18004B)	Provides information about the installation and configuration PowerAttendant Lite used with ACE software.
Industrial Robot Fieldbus Configuration User's Guide (Cat. No. M130)	Provides fieldbus configuration information for OMRON industrial robots.
Robot User's Manual	User Manual for specific (fieldbus supporting) robot types.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Revision code	Date	Revised content
11	June, 2026	Minor revisions and updates
10	November, 2025	Addition of EtherCAT SubDevice functionality for v4.8 Minor revisions and updates
09	June, 2023	Addition of new features and functions for v4.7 Minor revisions and updates
08	August, 2022	Addition of new features and functions for v4.6 Format update.
07	January, 2022	Added FH Camera and minor updates for v4.4
06	February, 2021	Added 3D Pick Manager, minor updates
05	September, 2020	Minor updates and additions, v4.2
04	August, 2020	Minor revisions and restructuring
03	July, 2020	Revisions for an upgrade to version 4
02	September, 2019	Style and content update
01	April, 2019	Original production

1

Introduction

This section describes ACE software features, functions, and concepts.

1-1 Software Overview 1-2

1-2 Configuration Wizards 1-3
1-2-1 Wizard Elements 1-3
1-2-2 Wizard Functionality in Emulation Mode 1-5

1-3 Licenses 1-6
1-3-1 PC Licenses 1-7

1-4 Software Features 1-9
1-4-1 Emulation Mode 1-9
1-4-2 Application Samples 1-12

1-5 Robot Concepts 1-15
1-5-1 Coordinate Systems 1-15
1-5-2 Calibrations 1-22

1-6 Basic Robot Motion 1-27
1-6-1 Speed, Acceleration, Deceleration 1-27
1-6-2 Approach and Depart 1-27
1-6-3 Arm Configuration 1-28
1-6-4 Location Precision 1-29
1-6-5 Continuous Path Motion 1-29
1-6-6 Joint-Interpolated Motion vs. Straight-Line Motion 1-29
1-6-7 Performance Considerations 1-30

1-7 Understanding Belts (Conveyors) 1-31
1-7-1 Indexing versus Tracking Conveyors 1-31

1-1 Software Overview

ACE is a PC-based software package that easily manages OMRON's entire portfolio of robots, controls, vision, and feeding systems.
ACE software provides an easy-to-use environment to program and deploy applications ranging from simple pick & place to multi-robot belt-tracking lines. It allows you to increase productivity while streamlining configuration setup.
The software can be used as a development tool or as an Application Manager runtime execution on an IPC Application Controller. The following advanced features are available when it is used as an Application Manager.

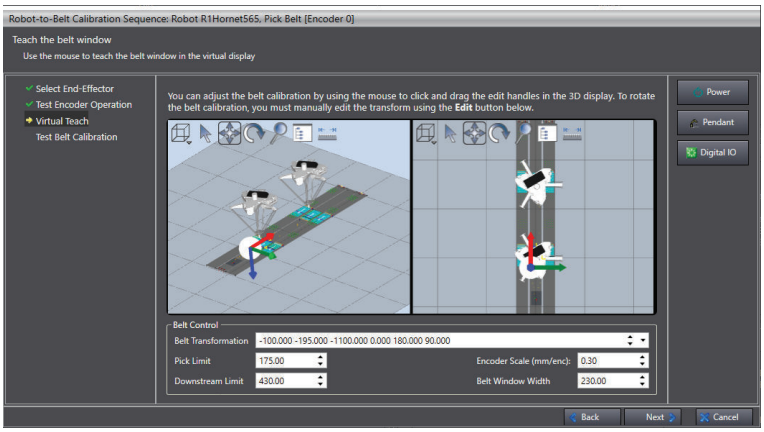
- The PackManager module enables advanced programming capabilities for packaging applications with simple step-by-step guidance without scripting.
- The Robot Vision Manager module enables vision guided conveyor tracking and part feeder integration with the help of configuration wizards. It also includes vision inspection tools to improve quality assurance and traceability.
- The Process Manager optimizes system resources, decreases idle time, and maximizes the amount of parts processed by the robot.

Additional features are detailed below.

- Emulation and 3D visualization provides a quick and robust proof of concept approach.
- Recipe Manager simplifies management of manufacturing process and handles a range of product variations, ideal for flexible automation to reduce changeover time.
- Operation programming provides an alternate method to create applications with a flowchart style approach. Advanced robot operations and sequences can be developed with a drag-and-drop method that automatically generates V+ programming code.

1-2 Configuration Wizards

Many of the ACE software components are configured using wizards. These wizards provide a series of screens to guide users through a detailed configuration process.
Selections and steps in the wizards will vary depending on the application and because of this, each wizard is not fully detailed in this document. Use the information provided in the wizard steps to understand the selections that are required.
An example robot-to-belt calibration wizard is shown below.



1-2-1 Wizard Elements

Many of the wizards share common elements such as buttons, fields, etc. The following information describe common wizard interface items.

Navigation

Use the following table to understand wizard navigation.

Item	Description
Back Button	Opens the previous screen in the wizard. For certain procedures, the **Back** button is not available making it impossible to repeat the previous screen.
Next Button	Opens the next screen in the wizard. The **Next** button will not be available until the current screen is completed.
Cancel Button	Cancels the operation and closes the wizard.

Dialog-access Controls

Use the following table to understand wizard dialog-access controls.

Item	Description
Pendant Button	Opens the V+ Jog control.
Power Button	Toggles high power for the robot.

Robot Position Controls

Use the following table to understand wizard robot positioning 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** Button () 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 (percent of full speed) for the robot movements.
Move	Moves the robot to the recorded (taught) position using the speed specified in Monitor Speed.
Taught Position	Displays the taught (recorded) position.

Conveyor Belt Controls

Use the following table to understand wizard conveyor belt controls.



Additional Information

These controls require the following conditions.
- The belt is under active control by the 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.

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

Vision Controls

Use the following table to understand vision controls.

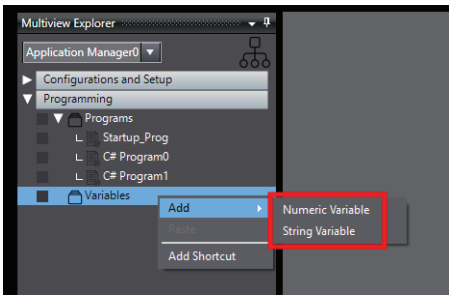
Item	Description
Edit	Opens the Vision Tools Properties window that is used to edit various parameters of the vision tool.
Live	Displays a live image from the camera input.
Picture	Acquires a still image from the camera input.
Stop	Stops the currently-running vision tool or process (this is only active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

1-2-2 Wizard Functionality in Emulation Mode

When Emulation Mode is enabled, some of the ACE software wizards contain differences from their operation in standard mode. This section describes those differences.

Calibration Wizards in Emulation Mode

When performing a belt calibration or sensor calibration in Emulation Mode, those wizards include special interactive 3D Visualizer windows that allow users to interactively position the elements being calibrated. This feature shows what is being changed and how the change affects the calibration. An example is shown in the following figure.
When multiple robots are present that access the same belt in the workspace, if a belt has not been taught it is not displayed in the 3D teach processes.
For Emulation Mode calibrations, the belt controls in the Calibration wizards will allow belt operation, even when the Active Control option of the Belt object is not enabled.



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

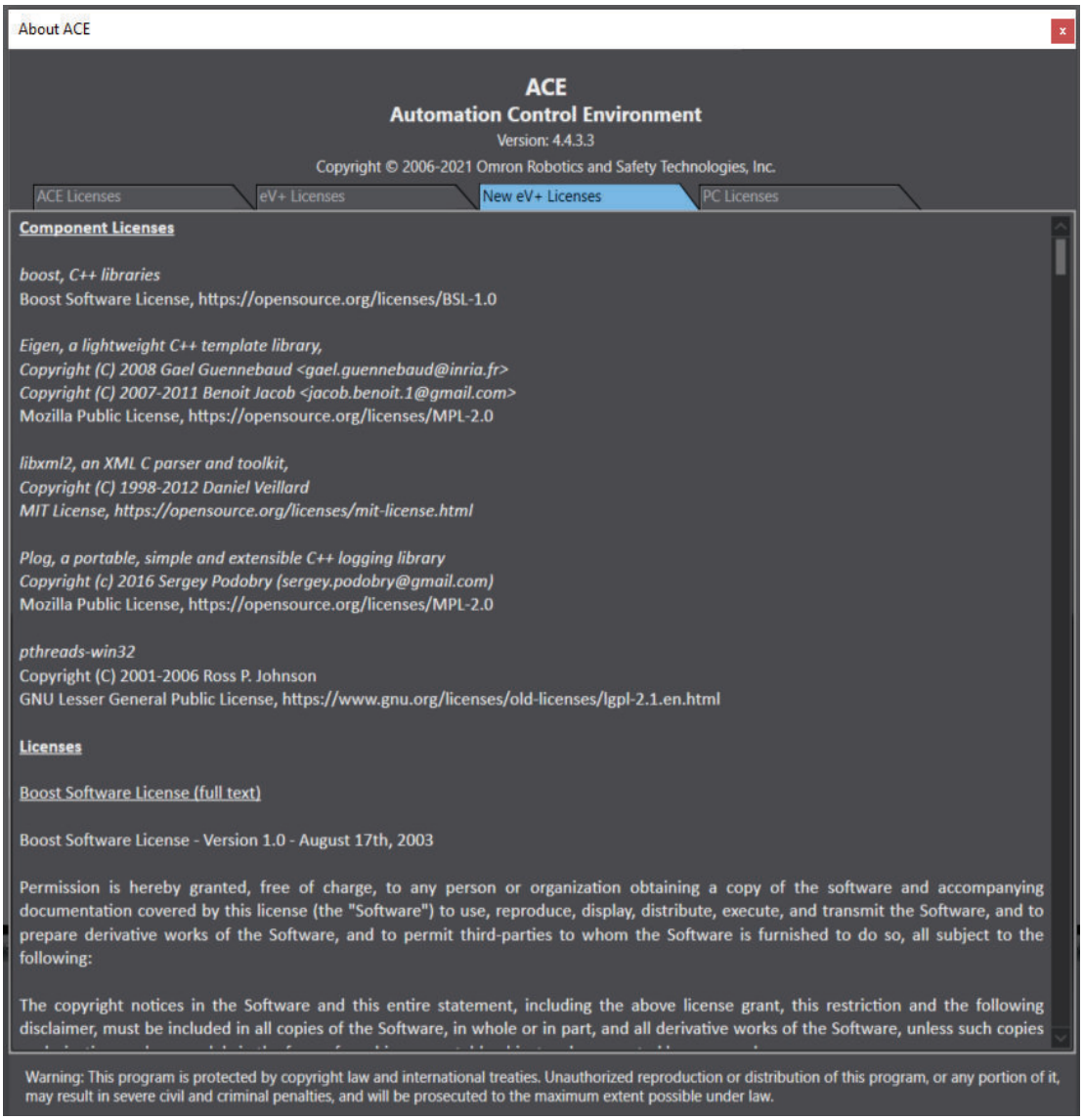
- Use the interactive 3D windows to drag the elements to the desired positions. After positioning the elements, the changes to the values are visible in the fields below the 3D windows.
- Use the fields below the interactive 3D windows to enter the values. After entering the values, the changes show in the 3D windows.

1-3 Licenses

To enable full functionality, the ACE software requires the V+ controller licenses and PC licenses supplied on the USB hardware key (dongle) as described below. For details on obtaining these items, please contact your local Omron representative.

To view the licenses installed on the dongle, access the *Help* menu item and then select **About...** This will open the *About ACE* Dialog Box. Choose the *PC Licenses* tab to view the installed licenses, including the V+ Licenses, as shown below.





1-3-1 PC Licenses

The following licenses are available for the PC running ACE software. The PC licenses are supplied on the USB hardware key (dongle, sold separately). Contact your local Omron representative for more information.

- Robot Vision Manager
- Pack Manager
- 3D Pick Manager

The ACE software will still work without licensing in place, but the following restrictions will apply.

Emulation Mode (Not Connected to Physical Devices)

When licensing is not activated, Emulation Mode will have full functionality for two hours . After two hours, the ACE software must be restarted to continue.

While Connected to Physical Devices (Not in Emulation Mode)

Execution of Pack Manager or Robot Vision functionality is restricted.

1-4 Software Features

This section provides details about the following ACE software features.

- Emulation Mode
- Application Samples

1-4-1 Emulation Mode

The ACE software can operate in an Emulation Mode. This mode provides a virtual environment that emulates the physical hardware, allowing the application to be programmed and operated with no connection to physical hardware.

Although the Emulation Mode is an optional operating mode, it behaves as though the ACE software is working in the standard operating mode. Once Emulation Mode is enabled, it is possible to create and program an ACE application in the same manner as when connected to physical hardware. This provides a seamless user experience that is nearly identical to running with real, physical hardware.

Emulation Mode can run multiple, simultaneous instances of controllers and robots on the same PC, at the same time. This includes the handling of network ports and multiple file systems. This feature allows the design, programming, and operation of a real multi-controller / robot application.

This section details the start up, features, and limitations of Emulation Mode.

Emulation Mode Features

Emulation Mode provides the following features and benefits.

The term offline implies that no connection to physical hardware is present.

- Create applications offline
  If no hardware is available, and offline application can be created. Emulation Mode allows a project to be configured with robots, belts, feeders, and other hardware. When the physical hardware becomes available, the transition from the virtual hardware to the physical hardware is easily done.
- Program offline
  Open and edit existing ACE projects. Edit V+ programs and C# programs.
- Operate applications offline
  Using a project, the application can open and run without the physical hardware. The application runs in the Emulation Mode using the 3D Visualizer feature to simulate the behavior of the system.
- Experimentation with products and hardware
  Using Emulation Mode, the application is created with virtual hardware, making it possible to experiment with different robot cell designs and layouts before purchasing the physical hardware.
- Training without products or hardware
  The emulated environment provides a convenient and low-risk method for training technicians, operators, and other system users.

Emulation Mode Differences

Emulation Mode has the following differences when compared to operating while connected to physical hardware.

Additional Information

Other minor Emulation Mode differences not listed below exist. These are noted in the related sections of this document.

- Saving data to the emulated controller is not supported.
  When Emulation Mode is enabled, ACE software creates a new emulated-controller file system in a temporary folder. When ACE software is shut down, that file system is deleted which means the contents of any user-created folders, files, or data will also be deleted. Save the data in a PC folder and/or with the ACE project to avoid loss of data.
- 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. It prompts a selection for the robot(s) connected to the controller instead.
- Some controller and robot configuration items are disabled.
  When Emulation Mode is enabled, certain Controller Settings and Robot Object configuration items are not available. Those items are dimmed or hidden to indicate that they are not available. Refer to *7-2 Controller Settings* on page 7-5 and *7-5 Robot Objects* on page 7-40 for more information.
- I/O signals are managed differently.
  In Emulation Mode, the Digital I/O window can be used to set input signals. Refer to *5-9 Task Status Control* on page 5-40 for more information.
- Robot-to-hardware calibrations use the 3D Visualizer.
  In Emulation Mode, the robot-to-hardware calibrations and project referencing procedures are different. When Emulation Mode is enabled, these procedures use the 3D Visualizer display to allow for offline calibration and configuration. Refer to *8-7 Process Objects* on page 8-97 for more information.
- The Belt object control, speed, and latch settings are different.
  When Emulation Mode is enabled wizards allow use of belt control signals even if Active Control is disabled, fast and slow speed settings are used, and the Latch Period generates a new latch at each distance interval of belt travel. Refer to *8-7-4 Belt Object* on page 8-108 for more information.
- The Virtual Camera configuration is different.
  When Emulation Mode is enabled, the Virtual Camera object uses an Emulation Configuration parameter that is used to specify the operating mode. Refer to *8-4 Camera Objects* on page 8-46 for more information.
- Cycle times are not identical.
  When Emulation Mode is enabled, the cycle times will not exactly match those obtained when connected to physical hardware.
- Enable power and calibration are managed differently.
  When Emulation Mode is enabled, the robot power is enabled and the robot is automatically calibrated when loading a project, creating a controller, rebooting a controller, and when changing a Quattro platform.

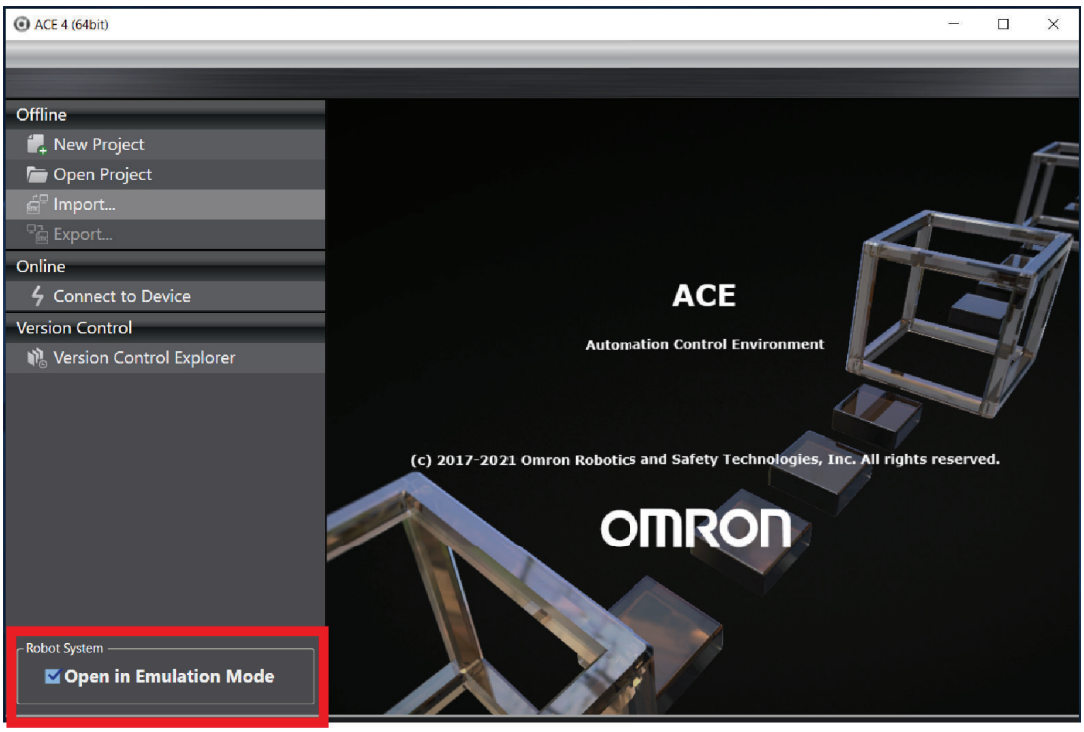
Enabling and Disabling Emulation Mode

Emulation Mode can be enabled or disabled by clicking the **Enable Emulation Mode** Icon ( ) or the **Disable Emulation Mode** Icon ( ), respectively. Any unsaved data must be saved before doing this. Note that the icons are disabled if they do not apply. For example, if Emulation Mode is enabled, the *Enable Emulation Mode* Icon cannot be clicked.

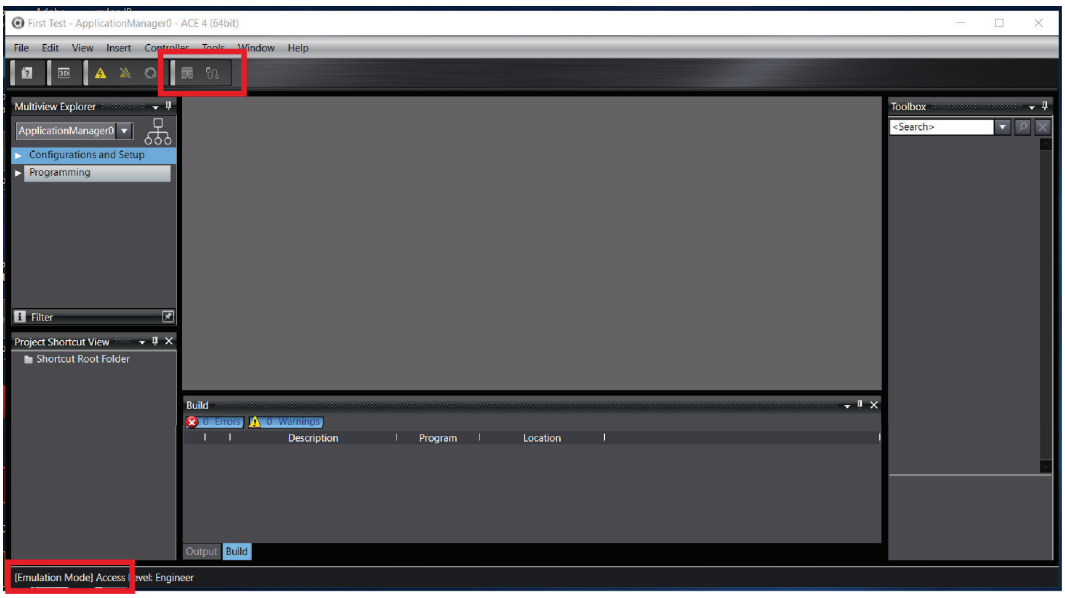
Alternatively, Emulation Mode can be enabled when opening a project with the following procedure:

1. Start the ACE software. The ACE Start page will be displayed.

2. Select the **Open in Emulation Mode** option.



3. Make other selections for opening a new or existing project, or connecting to an emulated device, refer to *Section 3 Creating a Project* on page 3-1 for more information. After these selections are made, the ACE project will open, using guided steps if necessary. Emulation Mode will be indicated in the Status bar at the bottom of the ACE software, as shown in the following figure. The Enable and Disable Emulation icons are indicated on the ACE 4 Toolbar.



4 After the ACE project is open and Emulation Mode is active, the procedure is complete. The same procedure can be done while deselecting the **Open in Emulation Mode** option to disable Emulation Mode.

1-4-2 Application Samples

The ACE software provides Application Samples to assist in the development of robotic applications. When an Application Sample is created, a wizard is launched to collect preliminary information about the application. When the wizard is completed, an ACE project is created with the basic objects and configurations for the application. This new project can be used as a starting point for many types of robotic applications.
There are two types of Application Samples that are currently offered with the ACE software: Robot Vision and Pack Manager. Refer to *Robot Vision Application Sample* on page 1-12 and *Pack Manager Application Sample* on page 1-13 on page 1 for more information.
Application Samples are not intended to develop an entire application.
Canceling a wizard before completion can lead to an ACE project with partial or no functionality. Completing the wizard is recommended.
Users must have Engineer access and either be connected to a physical controller or in Emulation Mode to use the ACE Application Sample functions.
All information collected during the wizard can be modified within the project later.
The Application Sample wizard follows the basic steps listed below. These may vary based on the Application Sample type and selections made during the wizard, but generally follow this sequence.

1 Select robots to install on the controller (Emulation Mode only).

2 Identify the pick and place configuration.

3 Teach the pick operation.

4 Teach the place operation.

5 Specify the robot used in the application sample.

6 Specify the end effector configuration.

7 Teach the safe robot position.

8 Teach the part pick position.

9 Teach the part placement position.

Robot Vision Application Sample

Robot Vision Application Samples can be used to create example V+ programs and Robot Vision objects for integrating Robot Vision with robots, belts, feeders, and more. When using Robot Vision with V+ programs, users are responsible for writing the V+ programs that drive robot motion and other activity in the robot cell.

Robot Vision sample wizards can be used for example application structure, but are not intended to provide all V+ program code required for the application.

Pack Manager Application Sample

Pack Manager Application Samples can be used to create single-robot packaging application projects with Pack Manager. These single-robot samples can later be expanded for multi-robot applications. The ACE software provides a point-and-click interface to develop many packaging applications without writing V+ programs. If the default behavior does not meet the needs of the application, V+ programs can be customized. Pack Manager uses a Process Manager to manage run-time control of the application including allocation of part and target instances in multi-robot packaging lines, visualization of resources, and statistics tracking.

Additional Information

Refer to *8-7-3 Part Object* on page 8-104 and the *8-7-5 Process Manager Object* on page 8-117 for more information.

Creating an Application Sample

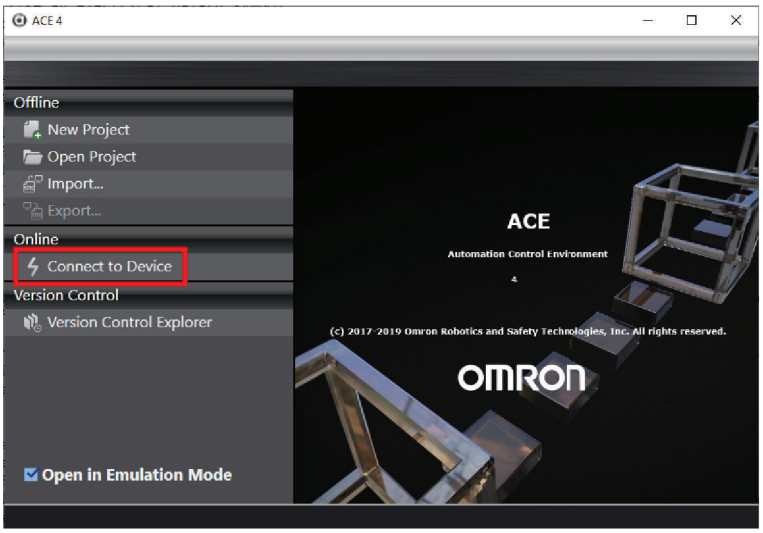
There are two methods used to create an Application Sample as described below.

● Create an Application Sample from the Start Page

Use the following procedure to create an Application Sample from the Start Page

1 Start the ACE software.

2 Select **Connect to Device** from the Start Page.



3 When creating the application while connected to a physical controller, make the appropriate Connect to Device settings. Refer to *Section 4 Online Connections to Robot Controllers* on page 4-1 for more information. If not connected to a physical controller, select the check box for **Open in Emulation Mode**.

4 Select the **Create Application Sample**.

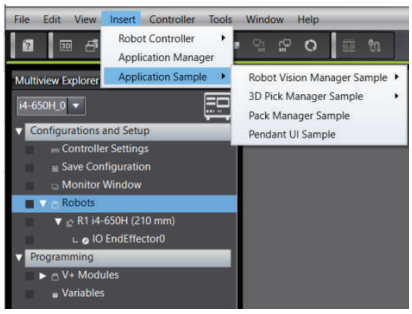
5 Then choose **Robot Vision Manager Sample** or **Pack Manager Sample**.

6 Then, click the **Connect** Button. This will create a new ACE project with the selected Application Sample and launch the Application Sample wizard.

7 Select robots to install on the controller (if running in Emulation Mode) and finish all wizard steps to complete the procedure. A new ACE project will be created according to the collected wizard information.

● Create an Application Sample from an ACE Project

To create an Application Sample from an ACE project, select **Insert** from the menu bar, select **Application Sample**, and then make a selection appropriate for your application.



A wizard will appear to guide you through the configuration steps. When the wizard is completed, an Application Manager will be present in the Multiview Explorer.

Additional Information

A robot controller device must be selected and online to access the Application Sample wizard.

1-5 Robot Concepts

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

1-5-1 Coordinate Systems

The ACE software uses multiple coordinate frames to define the locations of elements. These are often positioned in reference to other objects or origins. Each coordinate system is briefly described in the following table.

The coordinates in each system are measured in terms of (X, Y, Z, Yaw, Pitch, Roll) unless otherwise specified, where Yaw, Pitch, and Roll are defined as:

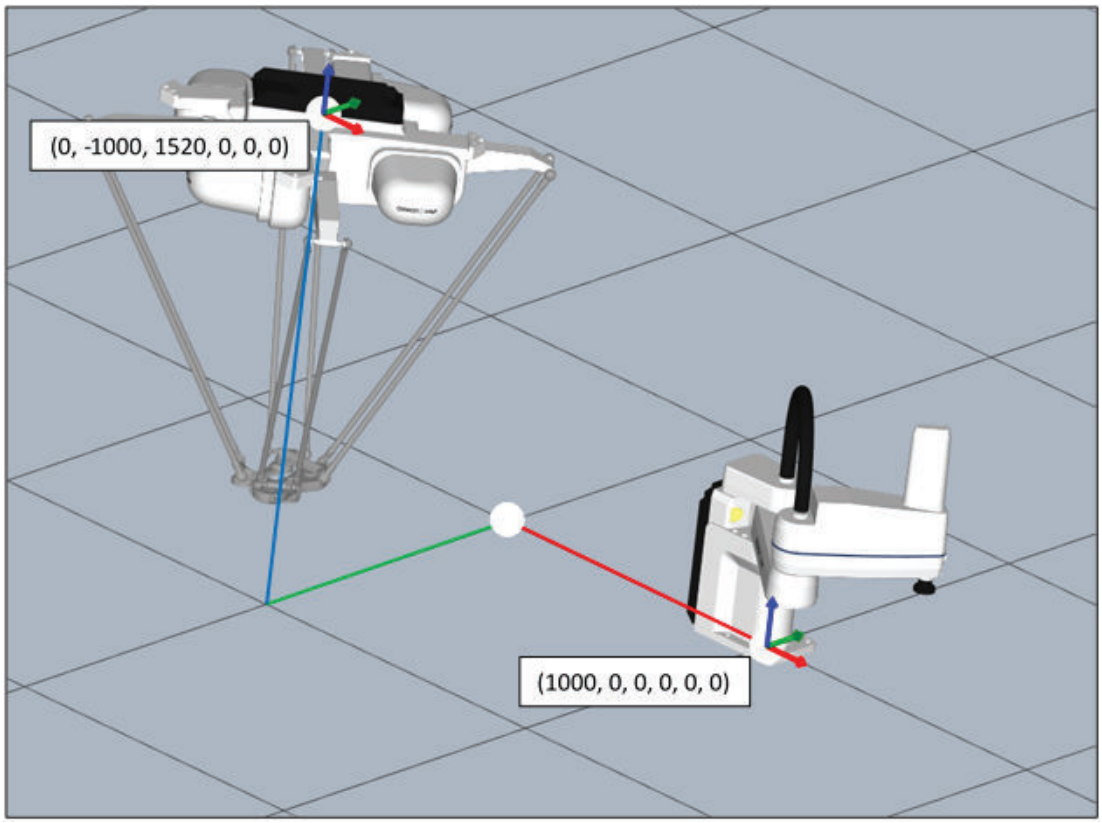
- Yaw: Rotation about the local reference frame Z-axis.
- Pitch: Rotation about the local reference frame Y-axis.
- Roll: Rotation about the local reference frame Z-axis after Yaw and Pitch have been calculated.

Coordinate System	Overview
Workspace	This is the global coordinate system of the 3D Visualizer in ACE software. Workspace Coordinates are used to define the position of objects. Refer to *Workspace Coordinates* on page 1-16 for more information.
Robot - World	Each robot has a world coordinate system. The X-Y plane of this coordinate system is the robot mounting surface. The Z-axis and origin are defined for each robot model and can be viewed by enabling the **Edit Workspace Position** Button in the 3D Visualizer. V+ Location variables that record the current robot position will be defined relative to this coordinate system. Refer to *Robot Coordinates - World* on page 1-16 for more information.
Robot - Joint	Each robot has a joint coordinate system based on the orientation of each individual joint. Each element of a coordinate is the angular position of the joint. V+ Precision Point variables will be defined relative to this coordinate system for each robot. Refer to *Robot Coordinates - Joint* on page 1-17 for more information.
Robot - Tool	This is the coordinate system based on the robot tool. The origin is positioned at the tool flange with the Z-axis oriented away from the tool flange when a null tool offset is applied. Refer to *Robot Coordinates - Tool* on page 1-18 for more information.
Belt	This is the coordinate system describing the direction and orientation of a conveyor belt. Each robot may have a different reference to belt coordinates for the same physical conveyor belt. Refer to *Belt Coordinates* on page 1-19 for more information.
Camera	This is the coordinate system defining the coordinates within the camera field of view. 2D vision primarily uses only the X, Y, and Roll components of a coordinate. The origin is located at the center of the field of view. Refer to *Camera Coordinates* on page 1-21 for more information.

Workspace Coordinates

The workspace coordinate system is a global reference frame for all object positions in the 3D Visualizer. The workspace origin is not visible, but it is positioned at the center of the tile grid, as shown in the figure below.

Workspace coordinates are primarily used for positioning robots and other features in the workspace. Allocation of belt-relative Part and Part Target instances during run time depends on the relative position of robots along a process belt object, therefore robot positions cannot be changed while a Process Manager is active.

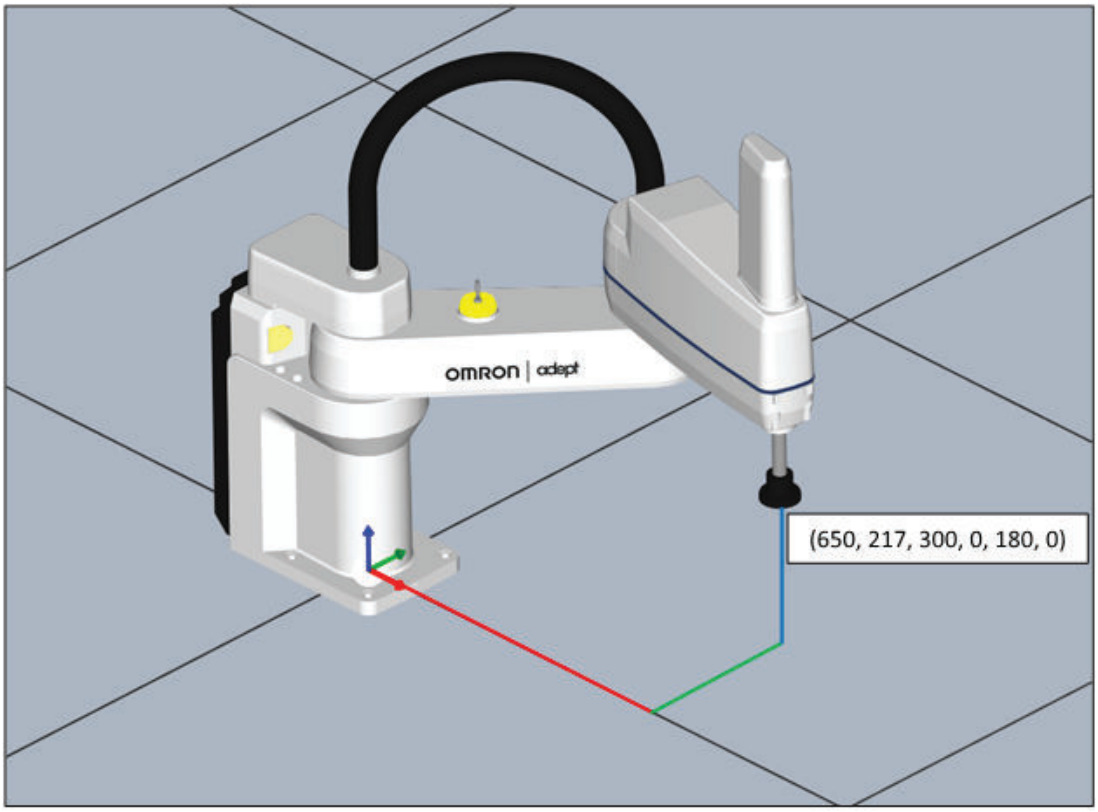


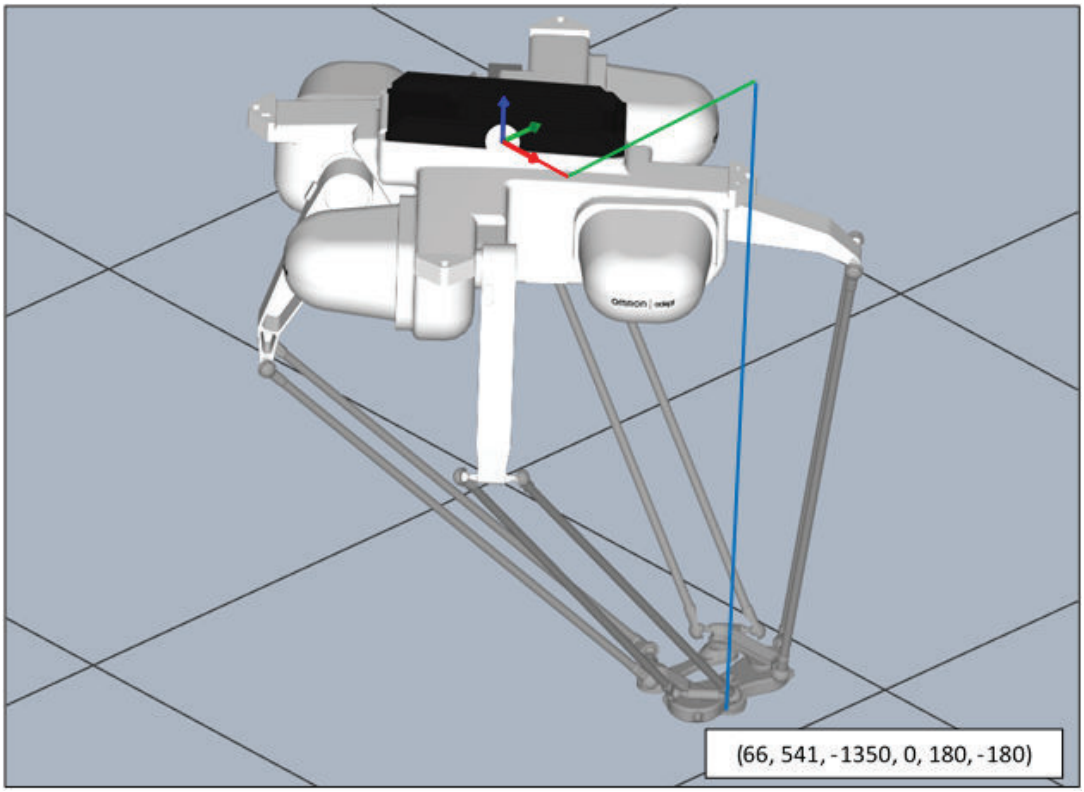
Robot Coordinates - World

The robot world coordinate system is a frame of reference for all transformations recorded by a specific robot. It is primarily used to define points with respect to the robot itself. Each robot model has a unique base frame, but the X-Y plane is typically located at the robot mounting surface. For example, the position markers of the robots shown in the figures below are also the robot origin in each robot world coordinate system.

This coordinate system is used when a program defines a transformation-type location. Whenever a position is taught or motion executed, it is usually done with respect to this coordinate system.

A position defined as a location variable is not unique and depends on applicable arm configuration parameters, including **lefty-righty**, **above-below**, and **flip-noflip**. Refer to the *eV+ Language Reference Guide (Cat. No. I605)* or the *V+ Keyword Reference Manual (Cat. No. I672)* for details on these keywords.

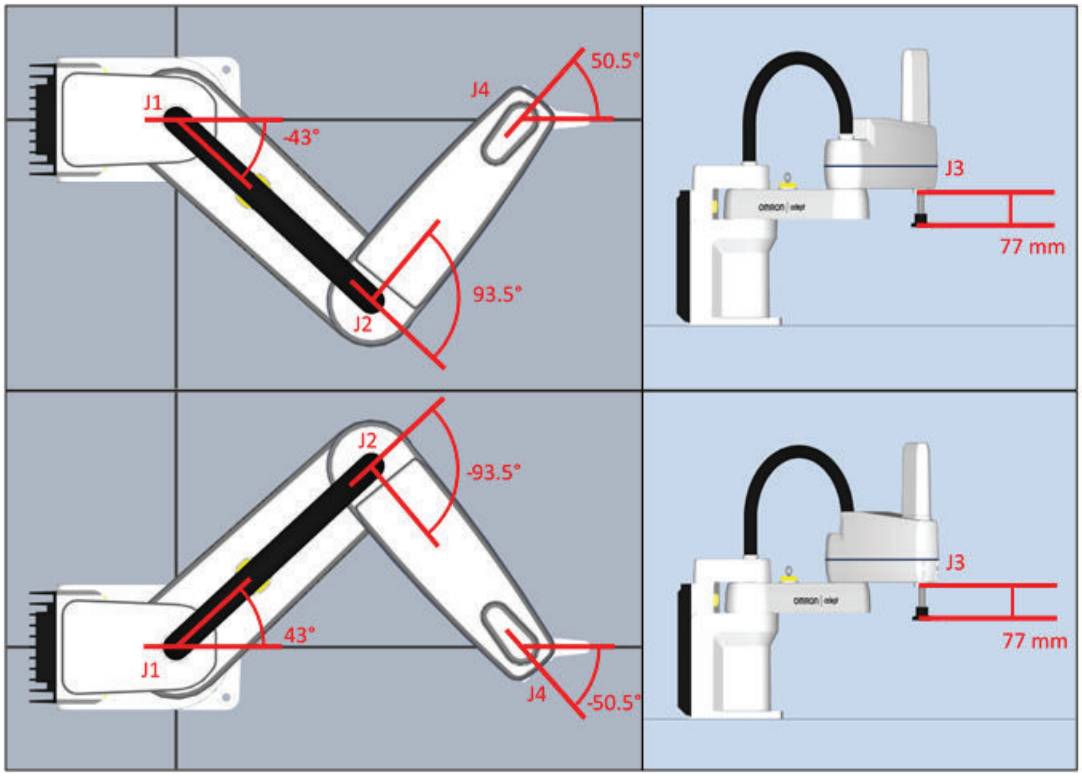




Robot Coordinates - Joint

The joint coordinate system is used to define the position of each individual joint. Each coordinate has as many elements as there are joints in the robot. For example, a Cobra would have four elements in a coordinate while a Viper would have six elements.

Joint coordinates become useful when defining a point that can be reached in multiple orientations. For example, the two configurations shown in the figure below have the gripper in the same position (550, 0, 317, 0, 180, 180) as defined by robot world coordinates. However, the robot arm can be oriented in two different ways to reach this position. The top configuration in the figure shows joint coordinates (-43, 93.5, 77, 50.5) and the bottom configuration shows joint coordinates (43, -93.5, 77, -50.5). The size of the tool is exaggerated in the figures to clearly demonstrate the orientation of J4.



A location based on joint coordinates instead of world coordinates is called a precision point. These are useful in cases where one orientation would cause the robot to strike an obstacle.

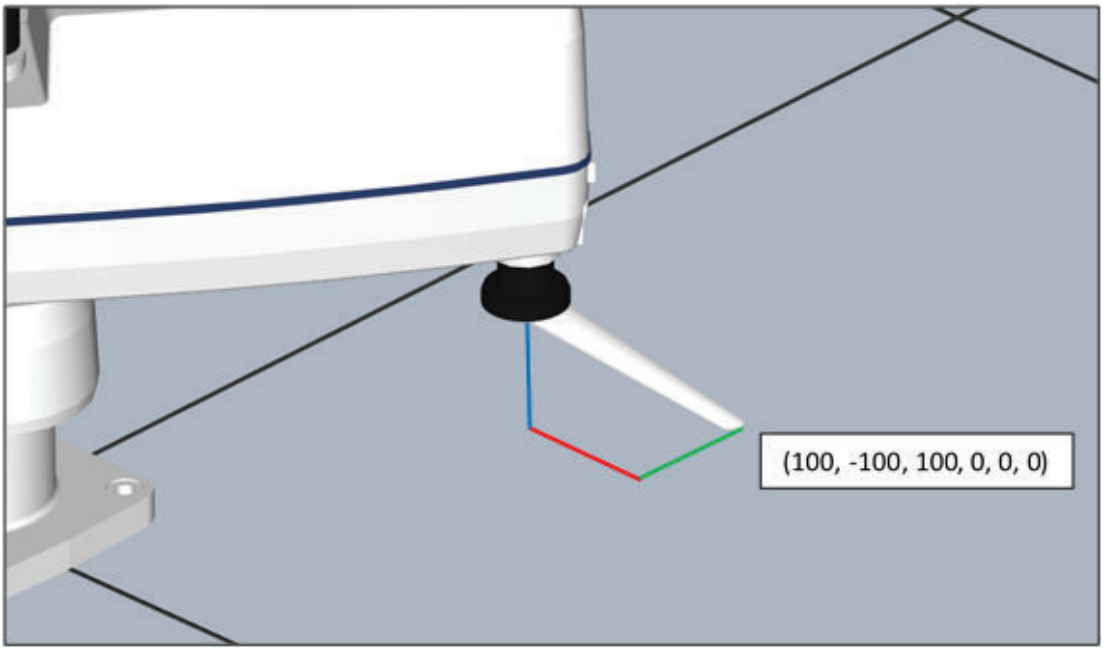
A precision point guarantees that a robot will always move to the correct orientation. This is most commonly seen in Cobra and Viper robots, since locations can be reached from multiple positions. Precision points can be defined for parallel robots such as the Hornet and Quattro, but because each location can only be reached while the joints are in one position, joint coordinates and precision points usually are not used with these robots. Joint coordinate jogging is also not allowed for parallel arm robots.

The orientation of the servo is important when considering joint coordinates. For example, in the figure above, the J4 orientation convention is in the opposite direction of the other two rotational joints. This is because the joint 4 servo internal mounting position is inverted.

Robot Coordinates - Tool

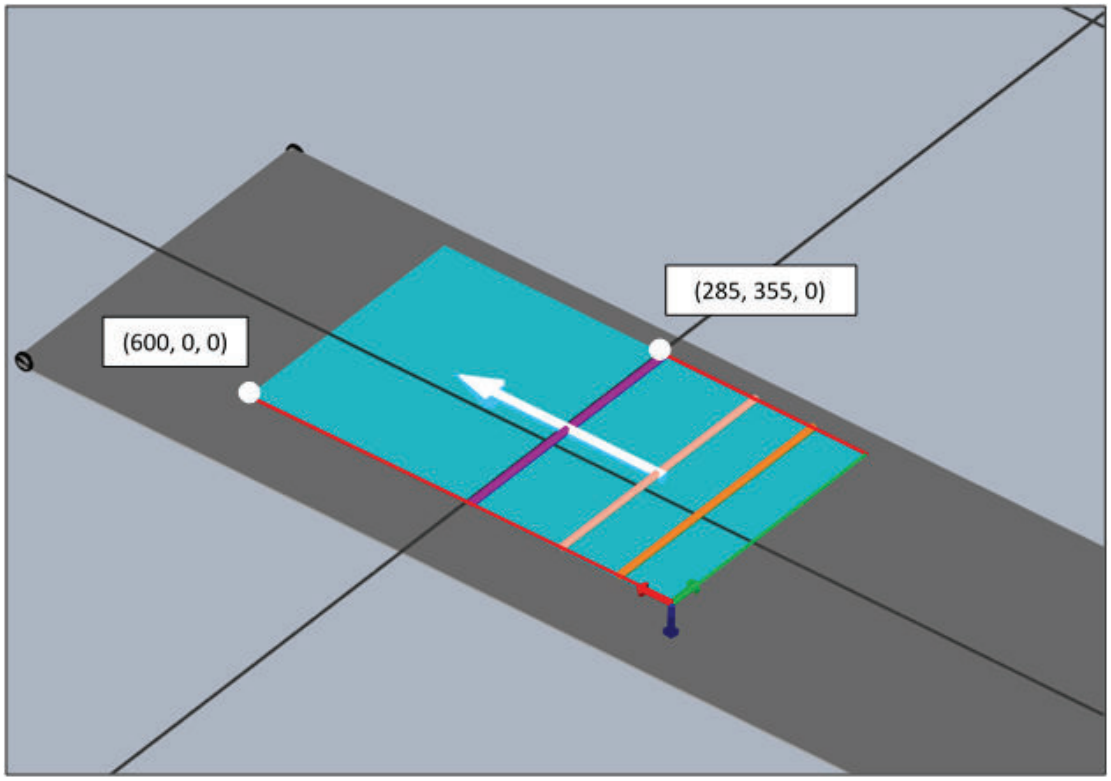
The tool coordinate system is used to define the position of tool tips. Its frame of reference is positioned at the tool flange itself. The tool Z-axis points opposite the other frames. This is because the main purpose of this system is to define the offset of tool tips. For example, a tool tip with coordinates (0, 0, 100, 0, 0, 0) is an offset of 100 mm in the negative Z-axis of the workspace and robot world coordinate systems.

The reversal of the Z-axis does not affect the controls in the V+ Jog Control. The down arrow still moves the tool along the negative Z-axis with respect to the robot world coordinates.



Belt Coordinates

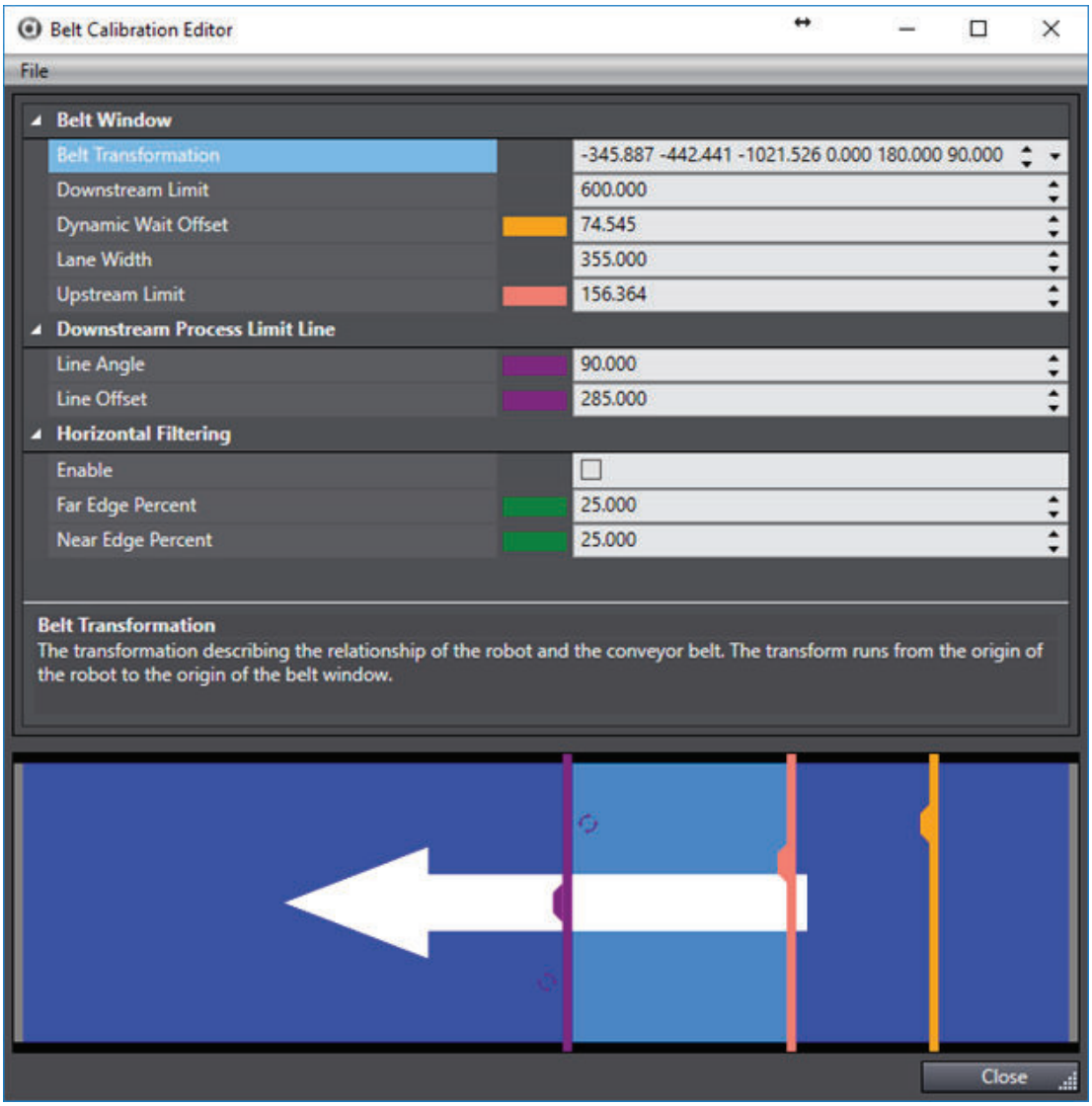
The belt coordinate system is used to define positions on a belt window. Its frame of reference is at one of the upstream corners of the belt. The axes are oriented so the positive X-axis is in the direction of the belt vector and the Y-axis is along the belt width. The belt is typically positioned so that the Z-axis of the belt frame aligns with the tool Z-axis, but it can be reversed if necessary.
This coordinate system is primarily used to provide part locations on a belt to a robot and to verify that a belt-relative location is within the belt window before commanding a motion to the location. When an instance is located, the identified belt coordinate is converted to a robot world coordinate. This means that a belt-to-robot calibration must be done before any belt coordinates are recorded.



The belt coordinate system is also used to set the various allocation limits in the belt window for a Process Manager. The various limits are set using X-coordinates and, for the Downstream Process Limit Line, an angle. In this case, the angle translates to both X and Y-coordinates to determine when an instance crosses that line. The various coordinates in the previous image are based on the numbers shown in the Belt Calibration Editor in the figure below. Refer to *Belt Configuration* on page 8-109 for more information.

Additional Information

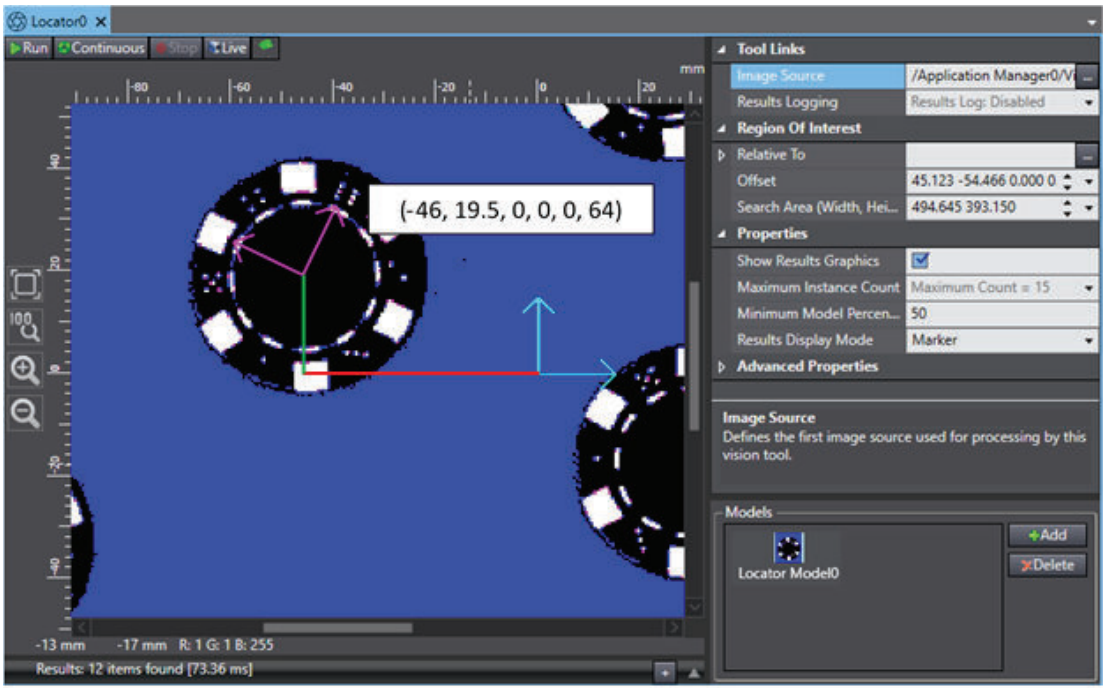
Belt coordinates do not apply to a Belt object created in the Process area of the Multiview Explorer (that is separate from a belt window). Belt objects are used to record information about the belt itself, such as encoders and signals, and provides a representation of a belt in the 3D Visualizer. Their location in the workspace is set by their Workspace Location parameter that uses workspace coordinates. Conversely, belt windows regard the positioning of the robot gripper with respect to the belt and use belt coordinates to determine instance locations.



Camera Coordinates

The camera coordinate system is used to define positions relative to a camera. Vision tools return positional data on detected instances or points in camera coordinates. 2D vision only requires the X, Y, and Roll components. Since the positions are still returned and used as 6-element transformations, the resulting locations are in the form of (X, Y, 0, 0, 0, Roll).

Camera coordinates must be interpreted into a different coordinate system before they can be practically used in an application. A robot-to-camera calibration is required to translate vision results to locations a robot can move to.



1-5-2 Calibrations

Calibrations are used to define relationships between coordinate frames. The calibration method may differ depending on whether the application uses Robot Vision Manager or a Process Manager, but the function of calibration is the same. In applications using Process Manager, the calibrations can be found in their respective sections in the Process Manager edit pane. The Process Manager will show calibrations required for defined processes.

When Robot Vision Manager is used, the calibrations are found by right-clicking **Robot Vision** in the Multiview Explorer, clicking **Add Calibration**, and selecting the appropriate calibration object.

Verify that any necessary tool tip offsets have been defined before beginning calibration.

Calibration Order

There are two types of hardware calibrations used in the ACE software. Most applications will use at least one, but if more than one are necessary, the calibrations should always be performed in the following order:

- Robot-to-Belt Calibration
- Robot-to-Sensor Calibration (these include a wide range of different calibrations including Robot-to Camera and Robot-to-Latch calibrations)

This is important because calibrations are dependent on previously defined locations. For example, a robot-to-camera calibrations in an application with a belt utilizes a belt vector to define the locations of instances detected by the camera. If the camera is calibrated first, then the camera location will not be recorded properly and will need to be re-calibrated once the belt has been defined.

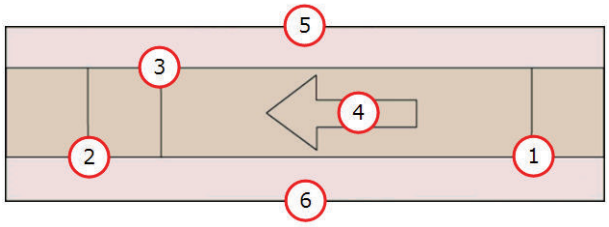
This assumes robot hardware calibrations were performed before doing the calibrations shown above. If robot hardware calibration changes, the other calibrations may need to be performed again.

Robot-to-Belt Calibration

This calibration translates positional information from the belt coordinate system to the robot world coordinate system. This is required whenever a belt is used in an application. One calibration needs to be performed for each encoder input associated with a robot.

Robot-to-belt calibration will require three points to be defined on the surface of the belt, shown in order in the figure below. Use the following procedure to execute a robot-to-belt calibration.

The three points in this calibration also define other values, such as the upstream and downstream allocation limits, also shown in the figure below. However, these do not directly affect the calibration of the belt and can later be changed to fit the needs of the application. For Robot Vision Manager and V+ programs, the pick limits will be defined in a V+ program. For applications with a Process Manager, refer to *8-3-5 Belt Calibration* on page 8-23 for more information.



Item	Description
1	Upstream Limit
2	Downstream Limit
3	Downstream Pick Limit
4	Belt Direction
5	Robot Side
6	Far Side

1 Place a calibration target on the belt at the farthest point upstream that the robot can reach. Verify that the robot can reach that belt position across the entire width of the belt.

2 Position the robot at the calibration target and record the position. This saves the robot location and the belt encoder position.

3 Lift the robot and advance the belt to move the calibration target to the farthest downstream position that the robot can reach. Again, verify the robot can reach across the entire width of the belt to ensure that the entire belt window remains within the work envelope.
It is important to ensure the calibration target does not move relative to the belt while advancing the belt.

4 Reposition the robot at the calibration target and record the position. The combination of recorded robot world coordinates and the belt encoder positions of these two points define the belt vector, which is the X-axis of the belt transformation, and the millimeter-per-count scale factor (mm/count).

5 Remove the calibration target from the belt and reposition on the opposite side of the belt at the farthest downstream position the robot should pick a part. Record its position in the same way as the other two points. This defines the belt pitch or Y-axis of the belt transformation. The

Z-axis of the belt transformation is defined based on the right-hand rule. After completing this step, the robot-to-belt calibration procedure is finished.

Robot-to-Camera Calibration

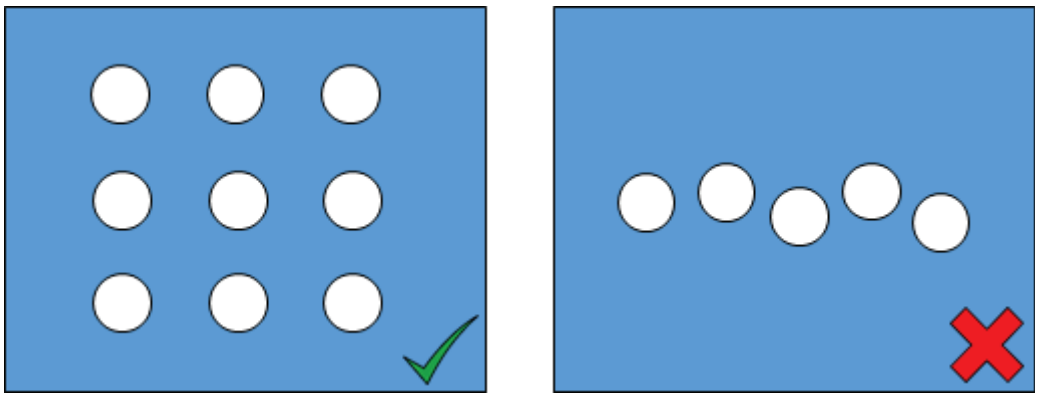
This calibration orients a camera frame of reference relative to a robot world coordinate system. It is used to translate positional information of vision results from the camera coordinate system to the robot world coordinate system. One of these is required for each association between a robot and a camera.

To perform this calibration, a grid calibration must be active in the Virtual Camera. If it is not, perform a grid calibration before proceeding. Refer to *8-4-1 Virtual Cameras* on page 8-46 for more information.

The process of calibrating the camera is dependent on the type of pick application in which it is involved. Generally, there are three categories with which the application could be associated:

- Fixed-mounted
- Arm-mounted
- Belt-relative

In all cases, the robot tool tip will be used to record various positions on the pick surface to associate it with the robot world coordinates. At least four points are required to generate the calibration, but a grid of 3x3 is recommended. The accuracy of the calibration increases with the distribution of the targets across the field of view (refer to the figure below). The configuration on the left would result in an accurate calibration while the configuration on the right could yield inaccurate results.



When the camera is fixed-mounted, a calibration target is recorded as a Locator Model and the target is placed in a defined region of interest of a Locator included in the calibration wizard. The target then must be repositioned in several different points on the pick surface. For each one, the camera detects the instance and records the position in camera coordinates, and then the position is taught to the robot by moving the gripper to the instance. The combination of the recorded data teaches the robot where the pick surface and the camera are relative to the robot, thus translating camera data to robot world coordinates.

If the application has a belt, the calibration is effectively the same, but it must be executed in two phases since the robot cannot reach the camera field of view. In the first phase, the targets are placed on the belt underneath the camera and their positions are recorded with respect to the camera. Then, the belt is advanced to where the robot can touch the targets and record their position in the robot world coordinate system. These locations and the associated belt encoder positions are used to define the location of the camera with respect to the robot world coordinate system.

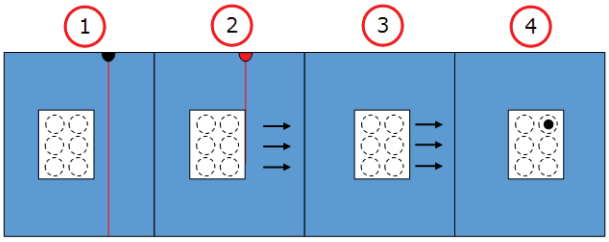
Robot-to-Latch Calibration

This calibration positions a latch sensor relative to a belt coordinate system. It is used to translate latch detection results to belt coordinate positions. One of these is required for each association between a robot and a belt with a latch sensor.

The robot-to-latch calibration is similar to the robot-to-camera calibration when a belt is present. However, instead of using a camera to detect the location of the target, the calibration is used to determine a part detection point, relative to a sensor signal.

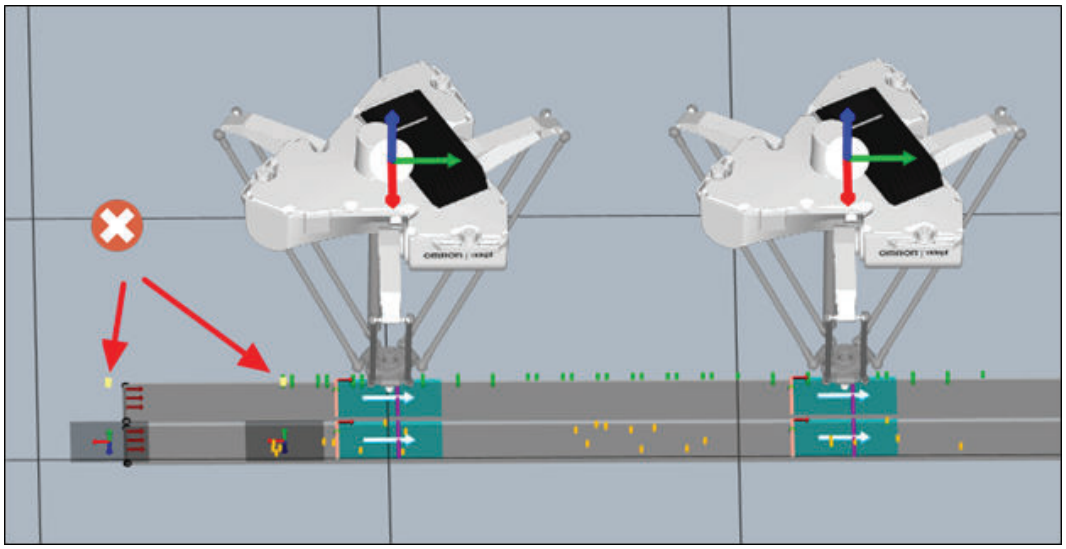
The target and the associated object are placed upstream of the latch sensor. When the belt is advanced past the sensor, the belt encoder position is recorded. Then, the belt is advanced to where the robot can touch the part. The recorded location combined with the belt encoder position indicates where the part will be detected by the sensor relative to the latched belt encoder position. The figure below shows an example using a pallet with slots for six parts.

In the following figure, the blue field represents the belt with the arrows indicating the direction of belt travel. The numbered sections represent the different steps of the calibration, explained as follows.

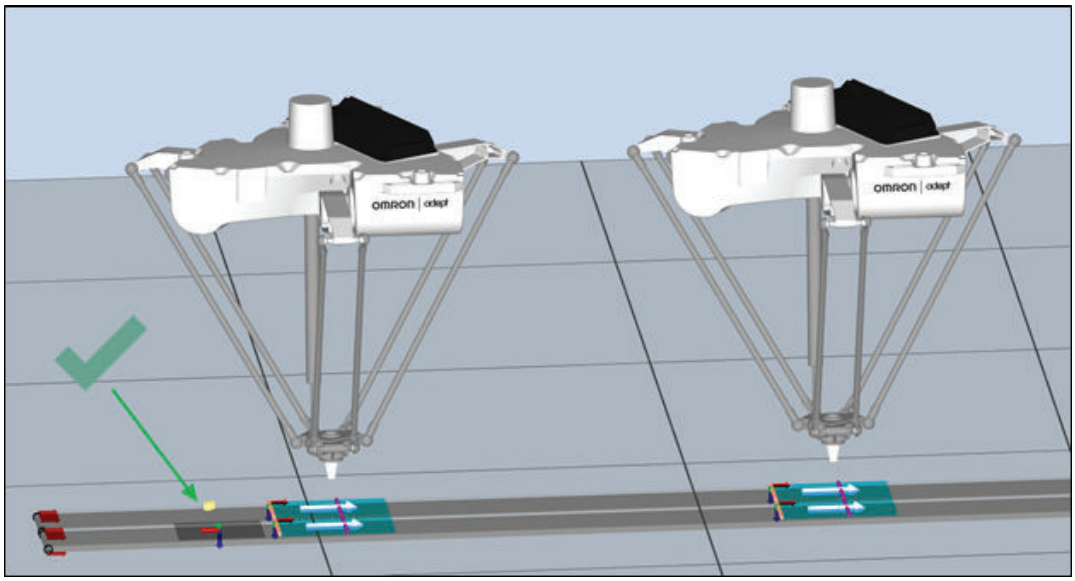


Step	Description
1	The pallet is positioned upstream of the latch sensor and the belt is not in motion.
2	The belt is advanced and the pallet is detected by the latch sensor. The belt encoder position at this position is recorded.
3	The belt is advanced to a position where the pallet is within the robot range of motion.
4	The robot tool tip (marked by a black circle) is positioned where the first part will be placed. The current belt encoder position is recorded and compared to the latched belt encoder position. This difference and the position of the robot along the belt vector are used to position the upstream part detection point for the latch sensor.

When calibrating multiple robots to a single sensor, ensure the initial position of the calibration object is identical for each robot calibration to avoid large a deviation in part placement relative to a latched position. There should not be a large deviation in sensor position for a single detection source, as shown in the figure below.



Instead, the sensors should be close together, as shown in the following figure. It is normal for there to be a small deviation due to differences between physical assemblies and ideal positions in 3D visualization.



Additional Information

If large deviations are caused by incorrect robot object Offset from Parent Properties, adjust this property to match the hardware system as closely as possible.

1-6 Basic Robot Motion

Use the information in this section to understand basic robot motion parameters to optimize overall system performance.

1-6-1 Speed, Acceleration, 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. The result obtained by the speed value depends on the operating mode of the robot (joint-interpolated versus straight-line). Refer to *1-6-6 Joint-Interpolated Motion vs. Straight-Line Motion* on page 1-29 for more information.

Whether in joint-interpolated mode or straight-line mode, the maximum speed is restricted by the slowest moving joint during the motion, since all the joint are required to start and stop at the same time. For example, if a given motion requires that the tool tip is rotated on a SCARA robot (Joint 4), that joint could limit 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.

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.

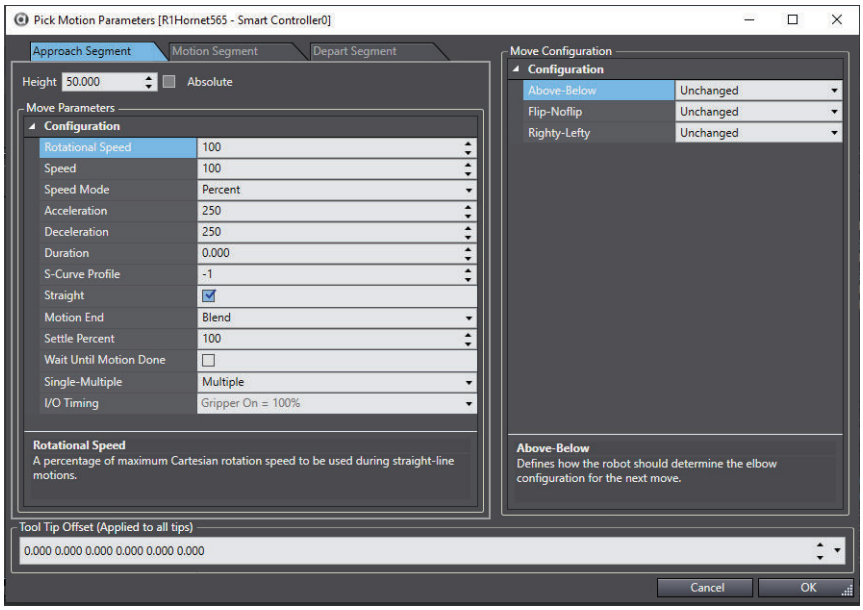
You can use the acceleration parameter to control the rate at which the robot reaches its designated speed and stops. Like speed, the acceleration / deceleration rate is specified as a percentage of the normal acceleration / deceleration rate. To make the robot start or stop smoothly using lower acceleration 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 lower 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 do not always result in faster cycle times due to positioning overshoot that may occur.

1-6-2 Approach and Depart

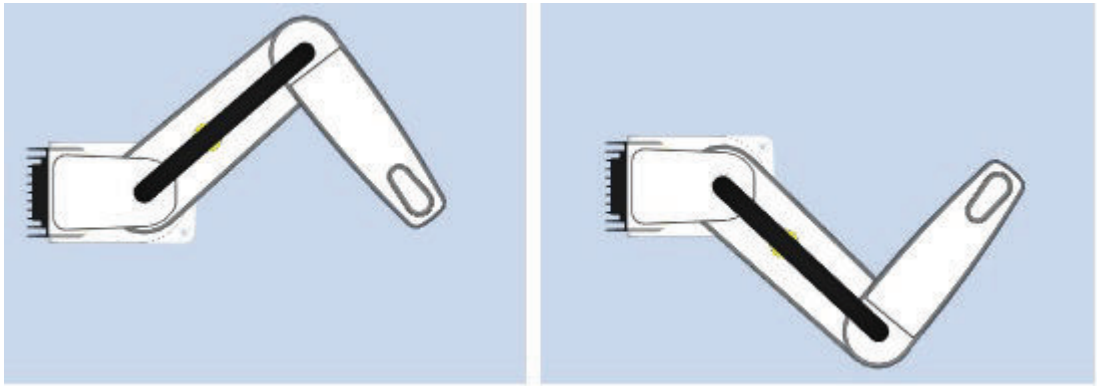
Approach and depart heights are used to make sure that the robot approaches and departs from a location without running into any other objects or obstructions in the robot envelope. Approaches and departs are always parallel to the Z-axis of the tool coordinate system. Approach and depart heights are typically specified for pick and place locations. The approach segment parameters are shown in the following figure.



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.
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 if the application requires this.

1-6-3 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 Cobra robots), but the above / below option does not apply to those robots.
The arm configuration can be specified in the Configuration Items of the Process Manager object or in a V+ program. Refer to *Configuration Items* on page 8-141 or the
or the *eV+ Language Reference Guide (Cat. No. I605)* or the *V+ Keyword Reference Manual (Cat. No. I672)* for more information
The following figure illustrates how a SCARA robot can reach a point with a lefty or righty arm configuration.



1-6-4 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* parameter (*Settle Fine* / *Settle Coarse*). If **Settle Coarse** is selected, the robot will spend less time attempting to reach the exact location. In many cases, this setting will be adequate and will improve robot cycle times.

1-6-5 Continuous Path Motion

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.

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*

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. Refer to *1-6-6 Joint-Interpolated Motion vs. Straight-Line Motion* on page 1-29 for more information.

1-6-6 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 (refer to *1-6-1 Speed, Acceleration, Deceleration* on page 1-27 for more information).

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.

The joint-interpolated or straight-line motion is set using the **Straight** parameter in the **Pick Motion Parameters** or **Place Motion Parameters** dialogs in the Process Manager or in V+ programs.

1-6-7 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 of the mounting surface will degrade performance. Therefore, it is recommended that you carefully follow the robot-mounting procedures described in your robot user's manual.
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 lighter materials and removing material that is not needed on tooling.

1-7 Understanding Belts (Conveyors)

This section describes basic belt (conveyor) concepts.



Additional Information

Conveyors are referred to as belts in the ACE software. Refer to *8-3-5 Belt Calibration* on page 8-23 for more information.

1-7-1 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 then it 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 non-vision or vision. With a non-vision 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 that reports the movement of the belt and distance moved to the ACE software. Tracking conveyor systems are configured as either non-vision or vision.

With a non-vision 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, the vision system detects parts that are randomly positioned and oriented on the belt. A fixed-mount camera takes pictures of the moving parts and based on the distance the conveyor travels, returns the location of parts. These part locations are queued and accessed by the robot.

2

Installation and Uninstallation

This section describes installation and uninstallation details for the ACE software.

2-1 Installing the Software 2-2
2-1-1 System Requirements 2-2
2-1-2 ACE Installation Instructions 2-2
2-1-3 Usage Considerations 2-3
2-1-4 Other Installed Items and Software 2-4
2-2 Uninstalling the Software 2-7

2-1 Installing the Software

This section describes the ACE software installation procedure and other software installation details.

2-1-1 System Requirements

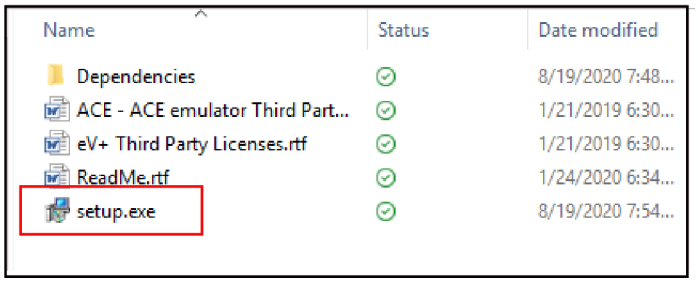
ACE software has the following system requirements.

- Windows operating system running Windows 10 or Windows 11.
- Installation and use of the Photoneo 3D camera requires a Windows 10 or Windows 11, 64 bit OS.
- An external graphics card, supporting OpenGL v 2.0 for operating the Photoneo camera.
- It is recommended to have more than 8 GB of RAM, a minimum 2 GB for the application and a minimum of 4 GB free disk space.
- Microsoft .NET Framework 4.6.1. The .NET Framework 4.6.1 will be installed automatically, if required. This will require a valid network connection to the internet.
- DirectX 11 compatible graphics card.
- The SmartController EX or compatible device running eV+ 2.1 B8 or above. The controller should also have a CompactFlash card containing up-to-date utility software.
- Ethernet communication between the PC and the controller.

2-1-2 ACE Installation Instructions

If there is an existing version of ACE on the computer, it should be uninstalled prior to installing the latest version.

Once installation media is loaded, you can access the installation media contents from a Windows Explorer window, and double click on the file **setup.exe**.



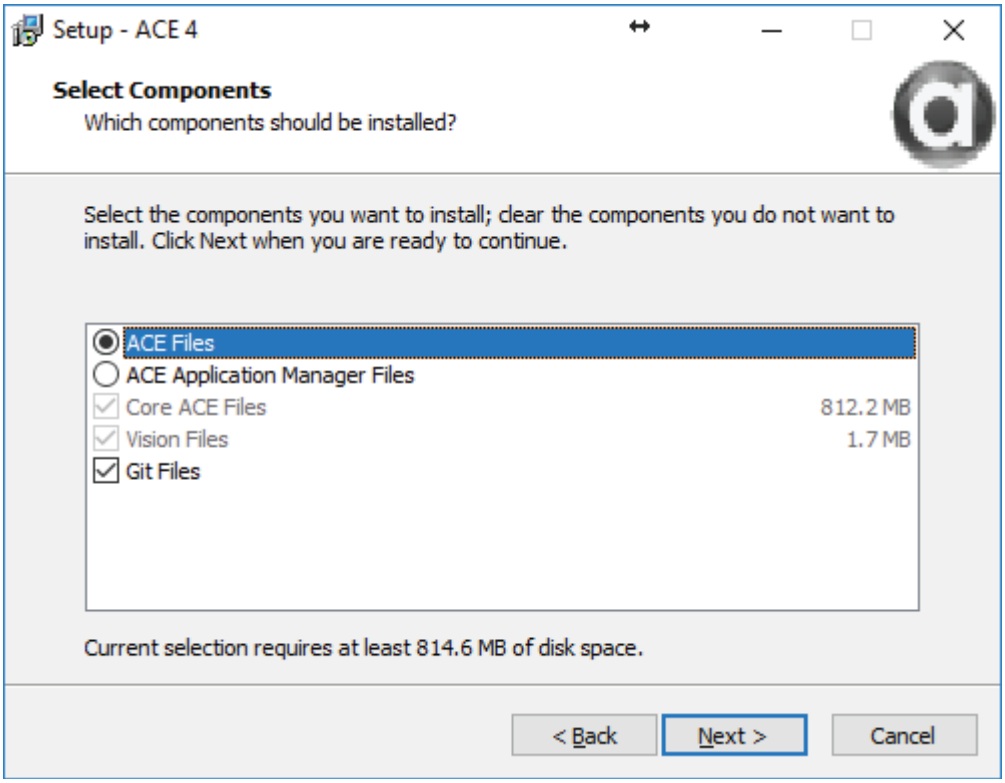
The installation will be performed in two distinct phases. The first phase checks for prerequisites on the PC. The second phase installs the ACE software on the computer.

In the first phase, the installer checks for the following prerequisites:

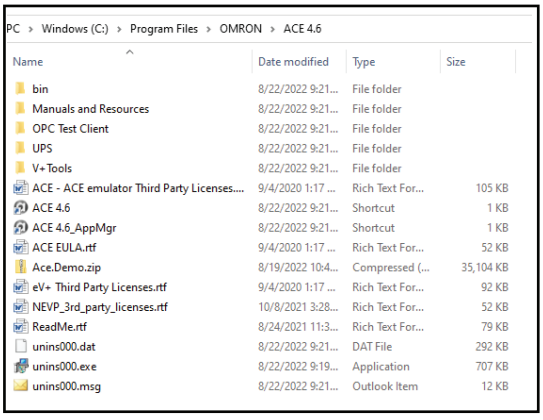
- Microsoft .NET Framework 4.6.1
- Basler Pylon
- Git
- Sentinel Protection Installer
- OPC Redistributables

If these required packages are not on the computer, they will be installed. If the Microsoft .NET Framework 4.6.1 is missing, the installer will attempt to download the files from the internet. If the computer does not have internet connectivity, you will need to manually download and install the Microsoft .NET Frameworks 4.6.1 from the Microsoft download site.

If any packages are already on the computer, they will be grayed out and not selectable from the installation GUI and they will not be installed.
The ACE 4 Setup wizard provides options depending on the required use of ACE. Select the **ACE Files** option to install the standard version of ACE. Select the **ACE Application Manager Files** option to install a version of ACE that is designed to act as a server instance. Refer to *8-1 Remote Application Manager* on page 8-3 for more information. Additionally, *Git Files* can be selected to install the latest version.
If the installer detects that a version of Git is installed, the option in the window will be unchecked and disabled.



When the installation completes the directory will be similar to that shown in the following figure. There will be two executables within the installation folder.



2-1-3 Usage Considerations

The following network ports are used by the ACE software.
PC network ports:

- UDP port 69: TFTP file transfer
- UDP ports 1993: Controller scan
- UDP 1994 and 1996: Startup communications
- TCP port 43434: Remote connections to ACE
- UDP port 1990: Robot Vision Manager V+ communications

Controller network ports:

- UDP port 1992: Controller scan
- UDP ports 1994-1997: ActiveV
- TCP port 1999: AdeptWindows
- TCP port 43234: ACE communication

Controllers must be set to auto-boot when rebooting from ACE; otherwise the connect attempt will fail.
If other programs are running in the V+ controller while ACE is running, certain operations (such as configuring device nodes or rebooting the controller) will fail.
Do not run the DC_SETUP or SPEC utility programs while ACE is running. They may delete variables required by ACE for proper operation.
Not all functions work properly if the hardware front-panel keyswitch for the controller is set to Manual mode.
When the ACE software connects to a controller, it will use a certain number of V+ tasks. It will use two tasks for general system functions and an additional task for each robot configured on the controller. In applications using the Pack Manager module, the process manager will allocate additional tasks based on the specific process configuration.

2-1-4 Other Installed Items and Software

Other items are installed with ACE software to supplement the application as described below. Access these items in the ACE software installation directory.

- Third Party and End User License Agreements
  Copies of software license agreements are provided in the installation directory.
- Basler Pylon Suite
  A Basler Pylon software suite is installed for support and configuration of Basler cameras.
- Sentech ST Utilities
  The ST Utilities are used from the configuration of the Sentech cameras.
- Git Repository Access (optional)
  Git repository resources are installed for use with project version control.
- OPC Test Client
  An OPC test client is included with the ACE software installation. Refer to *8-5-4 OPC Container* on page 8-65 for more information.
- Offline Visualizer
  Software used for playback of 3D Visualizer recording files (.awp3d). Refer to *5-3 3D Visualizer* on page 5-24 for more information.
- Uninterruptible Power Supply Script File Example
  Find an uninterruptible power supply script file example in the ACE software installation directory in the UPS folder. Refer to *5-14-6 Uninterruptible Power Supply (UPS)* on page 5-62 for more information.
- ACE User's Manual (This Document)
  A PDF file of this document is included in the ACE software installation directory.
- Dot Pitch Calibration Targets

Two PDF files are included to assist in vision system grid calibrations. Refer to *8-4-4 Custom Devices* on page 8-57 for more information.

- ACE Software Language Variations
  When installing ACE software, the following language variations are included. Access the different language variants with the Windows Start Menu group or in the following default installation directory (French, German, Italian, Spanish, Korean, Simplified Chinese, Traditional Chinese).
  C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Omron\ACE 4.6\Localized Shortcuts

Third Party and End User License Agreements

Copies of software license agreements are provided in the installation directory.

Basler Pylon Suite

A Basler Pylon software suite is installed for support and configuration of Basler cameras.

Sentech ST Utilities

The ST Utilities are used from the configuration of the Sentech cameras.

Git Repository Access

Git repository resources are installed (optional) for use with project version control.

OPC Test Client

An OPC test client is included with the ACE software installation. Refer to *8-5-4 OPC Container* on page 8-65 for more information.

Offline Visualizer

Software used for playback of 3D Visualizer recording files (.awp3d). Refer to *5-3 3D Visualizer* on page 5-24 for more information.

Uninterruptible Power Supply Script File Example

Find an uninterruptible power supply script file example in the ACE software installation directory in the UPS folder. Refer to *5-14-6 Uninterruptible Power Supply (UPS)* on page 5-62 for more information.

ACE User's Manual (This Document)

A PDF file of this document is included in the ACE software installation directory.

Dot Pitch Calibration Targets

Two PDF files are included to assist in vision system grid calibrations. Refer to *8-4-4 Custom Devices* on page 8-57 for more information.

ACE Software Language Variations

When installing ACE software, the following language variations are included. Access the different language variants with the Windows Start Menu group or in the following default installation directory.
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Omron\ACE 4.6\Localized Shortcuts

● Language Variants

- French
- German
- Italian
- Spanish
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

2-2 Uninstalling the Software

Use the following procedure to uninstall the ACE software and all associated items and other software.

1. Open the Control Panel from the Windows Start Menu and then select **Programs and Features** (**Apps & features**).The *Uninstall or change a program* Dialog Box is displayed.
2. Select **ACE 4** and then click the **Uninstall** Button.
3. Proceed with any confirmation messages to uninstall the software.
4. Click the **Finish** Button to complete the uninstallation of the ACE software.

3

Creating a Project

This section provides details associated with creating an ACE project.

3-1 Starting and Exiting the ACE Software 3-2
- 3-1-1 Starting ACE Software 3-2
- 3-1-2 Starting Application Manager 3-3
- 3-1-3 Automatic Project Launch on System Boot Up Procedure 3-3
- 3-1-4 Clear Application Manager Memory 3-6
- 3-1-5 Exiting the ACE Software 3-7

3-2 Creating A Project File 3-9

3-3 Saving the Project File 3-12

3-4 Saving a Project File Under a Different Name 3-13

3-5 Creating a Project from an Online Connection 3-14

3-6 Closing a Project and Returning to the Start Page 3-15

3-7 Opening a Project File 3-16

3-8 Importing a Project File 3-18
- 3-8-1 Importing from the Start Page 3-18
- 3-8-2 Importing from the Menu Bar in the Application Window 3-18
- 3-8-3 Importing an .awp2 ACE Project File 3-18
- 3-8-4 Importing from Version Control 3-19

3-9 Exporting a Project File 3-20
- 3-9-1 Exporting from the Start Page 3-20
- 3-9-2 Exporting from the Application Window 3-20
- 3-9-3 Saving an Exported Project File 3-21

3-10 Password Protection for Project Files 3-23
- 3-10-1 Setting Password Protection for Project Files 3-23
- 3-10-2 Removing Password Protection for Project Files 3-23

3-11 Project Version Numbers 3-25

3-1 Starting and Exiting the ACE Software

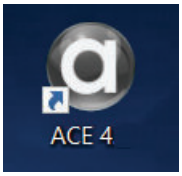
Use the following precautions when you start the ACE software.

- Exit all applications that are not necessary to use the ACE software. If the ACE software startup or operations are affected by virus checking or other software, take measures to remove ACE software from the scope.
- If any hard disks or printers that are connected to the computer are shared with other computers on a network, isolate them so that they are no longer shared.
- With some notebook computers, the default settings do not supply power to the USB port or Ethernet port to save energy. There are energy-saving settings in Windows, and also sometimes in utilities or the BIOS of the computer. Refer to the user documentation for your computer and disable all energy-saving features.

3-1-1 Starting ACE Software

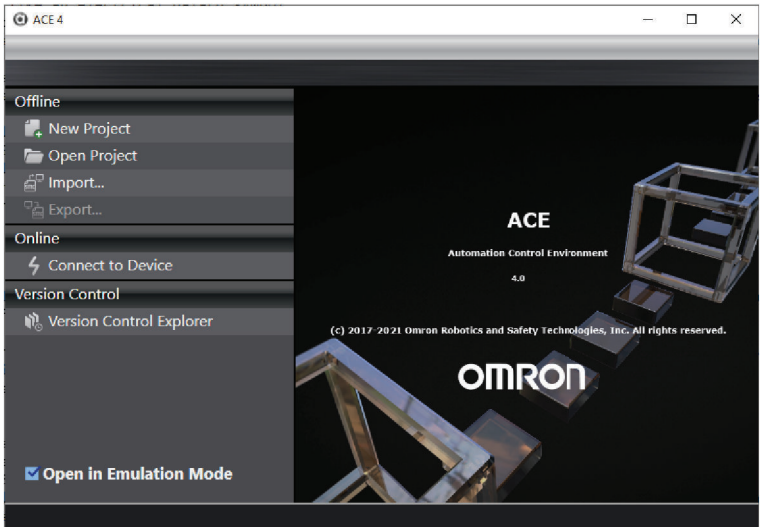
The installation includes ACE Application Manager and ACE, where ACE is the server instance that can integrate with clients. Use one of the following methods to start the ACE software.

- Double-click the ACE software short cut icon on your desktop that was created during the installation process.

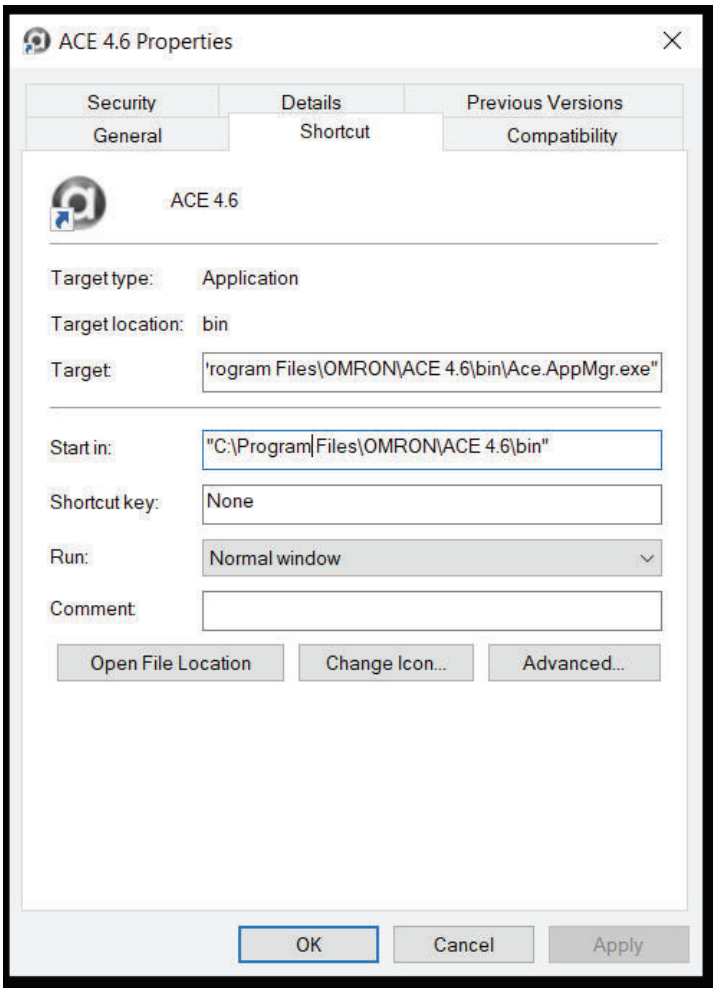


- From the Windows Start Menu, select **Omron - ACE 4** in the desired language.

The ACE software starts and the following window is displayed. This window is called the Start page. The language displayed on this page will be the one selected at start up.



Right-click the desktop icon and select **Properties** to show the path to the installed application. The path for this instance is: C:\Program Files\OMRON\ACE 4.6\bin\Ace.AppMgr.exe.



3-1-2 Starting Application Manager

To change the short cut to start the ACE Application Manager, the properties file needs to be edited. With the Properties panel opened and the Properties line highlighted, add startserver to the end, so that it now reads: C:\Program Files\OMRON\ACE 4.7\bin\Ace.AppMgr.exe startserver. When done, click **Apply** and then click **OK**.

An alternative to this is to go to the installation folder (refer to *2-1-2 ACE Installation Instructions* on page 2-2) and create a desktop short cut from the ACE 4.7_AppMgr file. Using this method allows you to have both an Application Manager and ACE desktop short cut.

3-1-3 Automatic Project Launch on System Boot Up Procedure

The ACE software can be configured to automatically launch and open a specific project when the system boots up. Use the following procedure to configure automatic project launch on system boot up.

You may need to disable Windows Login to allow automatic launch of the ACE project. Contact your system administrator for more information.

Depending on the application, it may be necessary to configure items such as an Auto Start program, a Program System Startup object, and a Controller Connection Startup object. Refer to the following sections for more information.

- *Auto Start* on page 7-25

- *8-5-5 Program System Startup* on page 8-68
- *8-5 Configuration* on page 8-61

Identify the preferred language variant to use with the ACE project. Take note of the language code. This will be used later in this procedure.

Language	Code
German	de-DE
English	en-US
Spanish	es-ES
French	fr-FR
Italian	it-IT
Japanese	ja-JP
Korean	ko-KR
Simplified Chinese	zh-CN
Traditional Chinese / Taiwanese	zh-TW



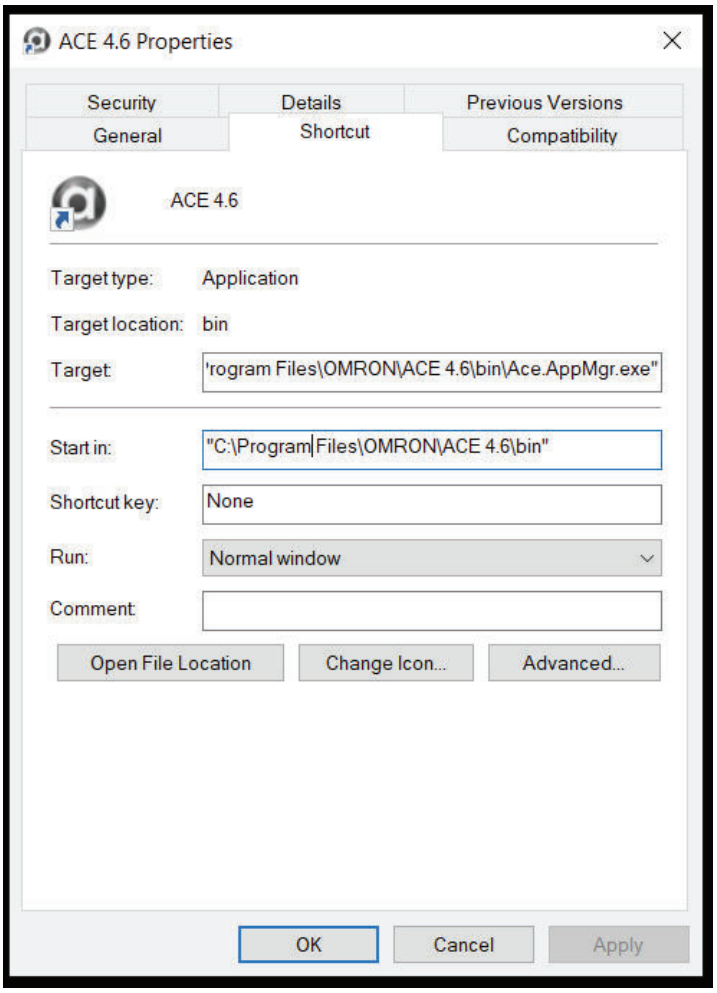
Additional Information

The following procedure assumes a default ACE software installation directory. Modify the file paths below accordingly if an alternate path was chosen during the ACE software installation.

1 Locate the ACE software executable file. The default software installation directory where the ACE software executable file is normally located can be found here:
C:\Program Files\Omron\ACE 4.6\bin\Ace.AppMgr.exe

2 Create a short cut to the Ace.AppMgr.exe file in new location that can be accessed in the next step (on the desktop, for example).

3 Access the new short cut to the Ace.AppMgr.exe file. Open the short cut properties by right-clicking the short cut and selecting **Properties**. Then, view the short cut tab of the *Properties* Dialog Box.



4 Modify the file name at the end of the Target path by replacing \Ace.AppMgr.exe with the following.

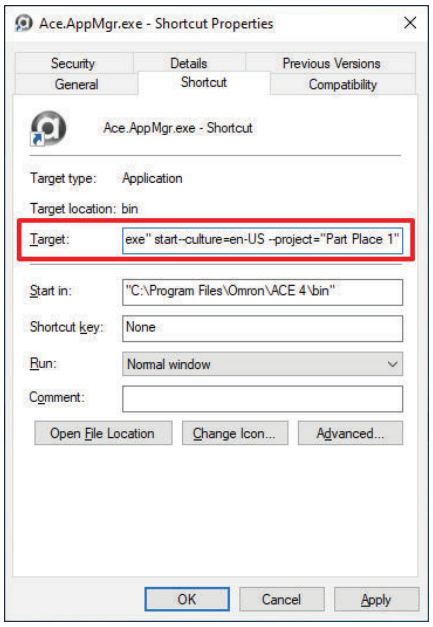
```
\Ace.AppMgr.exe" start --culture=xx-XX --project="Project Name
```

- Substitute xx-XX with the preferred language code.
- Substitute Project Name with the name of the project to automatically open.

An example Target path would appear as shown below for the Ace.AppMgr.exe shortcut that was created. The example below uses the preferred language of English and the project name of Part Place 1.

```
C:\Program Files\Omron\ACE 4.6\bin\Ace.AppMgr.exe" start --culture=en-US --project="Part Place 1"
```

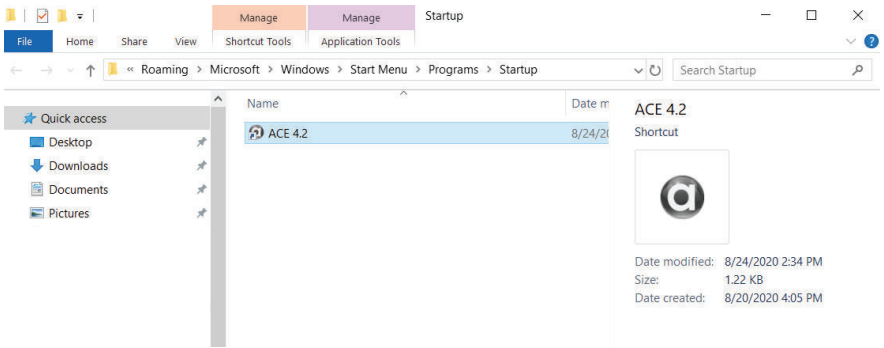
Before assigning the project to the Autostart, review the project name list in the repository to confirm there is only one project with that name. Multiple names in a repository can create problems when attempting Autostart.



5 Confirm the function of the modified short cut Target path by double-clicking the short cut. The project should open in the ACE software to confirm the correct target command was used.

6 Access the Windows Startup Folder, found in C:\ProgramData\Microsoft\Windows\Start Menu \Programs\Startup.
The Windows Startup folder will open.

7 Move the modified short cut into the Windows Startup folder.

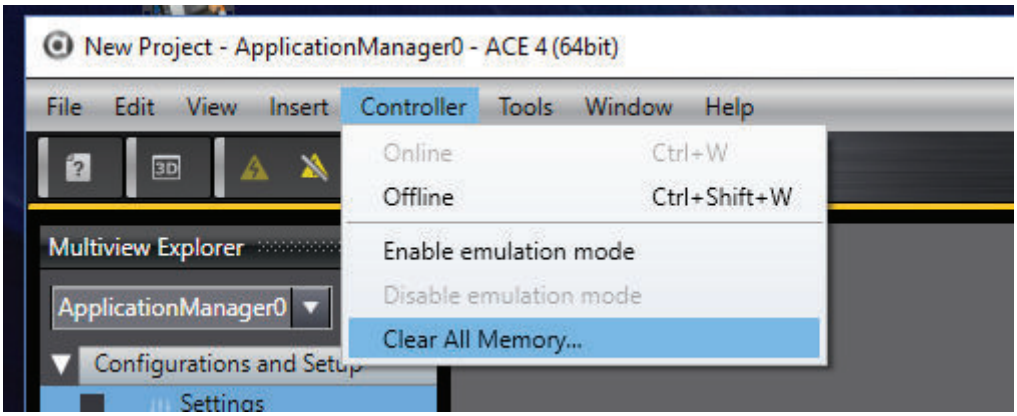


This will cause the ACE project to automatically open upon boot up.

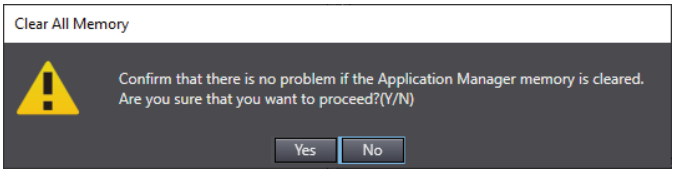
8 To complete this procedure, reboot the PC and confirm that the ACE project automatically launches.

3-1-4 Clear Application Manager Memory

When creating an Autostart process, there may be interference from server objects. This interference appears when attempting the synchronize process. To correct this issue and transfer objects, the Autostart process needs to be deleted using the *Clear All Memory...* function as described below.
With Application Manager opened, click **Controller** on the menu and then select **Clear All Memory**.



Click **Yes** to confirm that there is no problem if the Application Manager memory is cleared. The @autostart project will be deleted from the server instance.

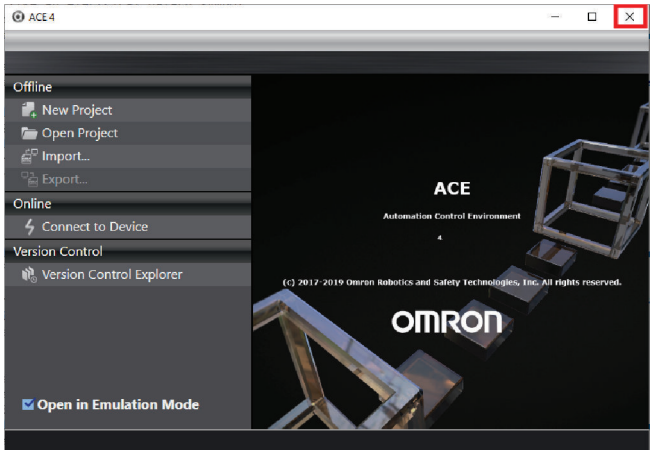


Perform the needed changes, synchronize the server, and then the project can be added to the server instance without interference from server objects.

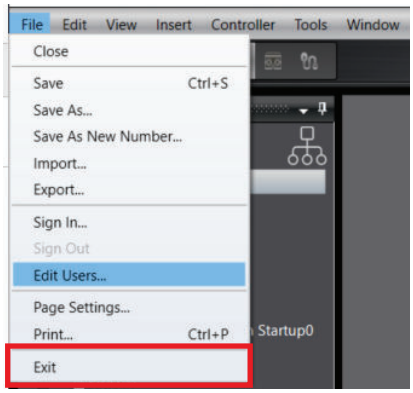
3-1-5 Exiting the ACE Software

Use one of the following methods to exit the ACE software.

- Click the **Close** Button in the title bar of the Start page.

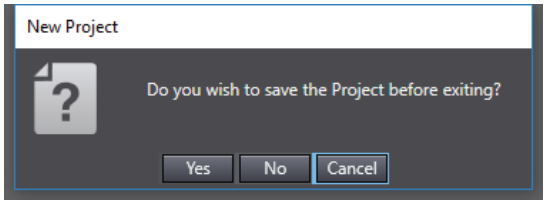


- Select **Exit** from the File Menu when working in an open project. The ACE application closes.



Additional Information

If there is unsaved data when you exit the ACE software, a dialog box will appear that asks if you want to save the data



Save the data if required. The ACE software will close after this operation.

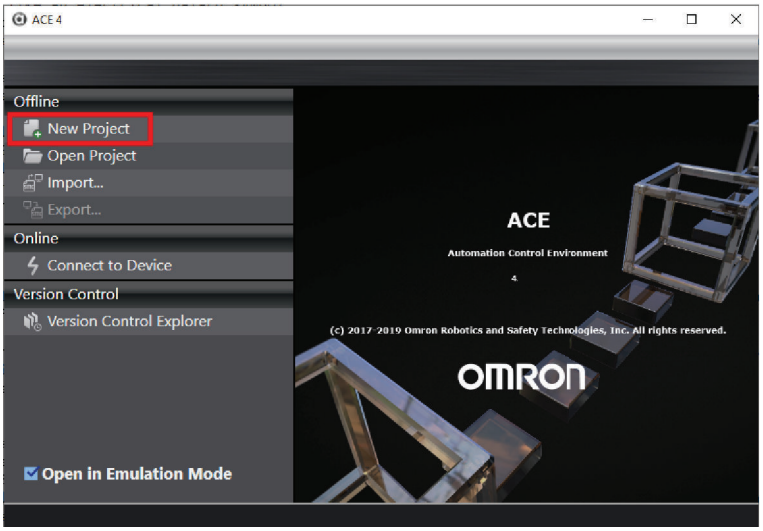
3-2 Creating A Project File

The following procedure describes how to create a project file from the start page.

Additional Information

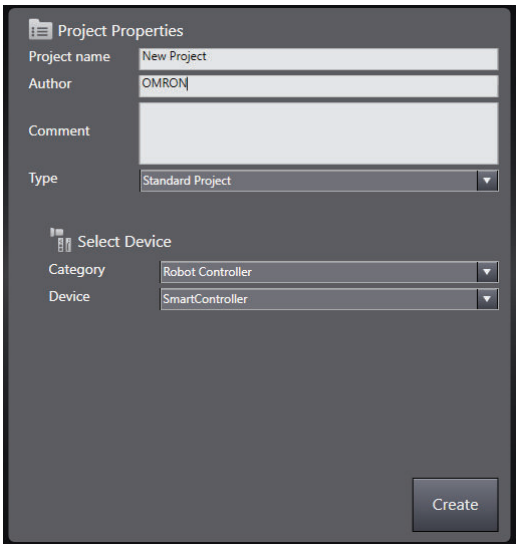
A project file can also be created with the **Connect to Device** method. Refer to *3-5 Creating a Project from an Online Connection* on page 3-14 for more information.

1 Click **New Project** on the Start page.



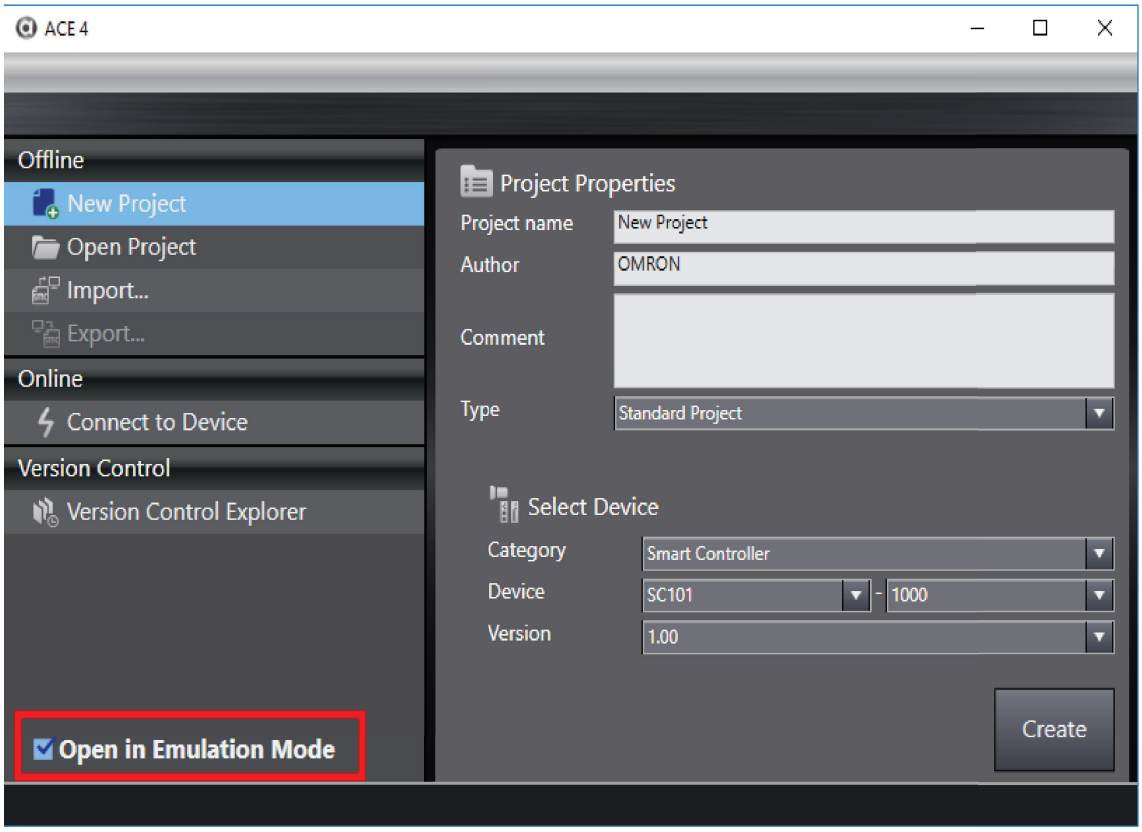
The *Project Properties* Screen is displayed.

2 Make selections in the *Project Properties* and *Select Device* area.

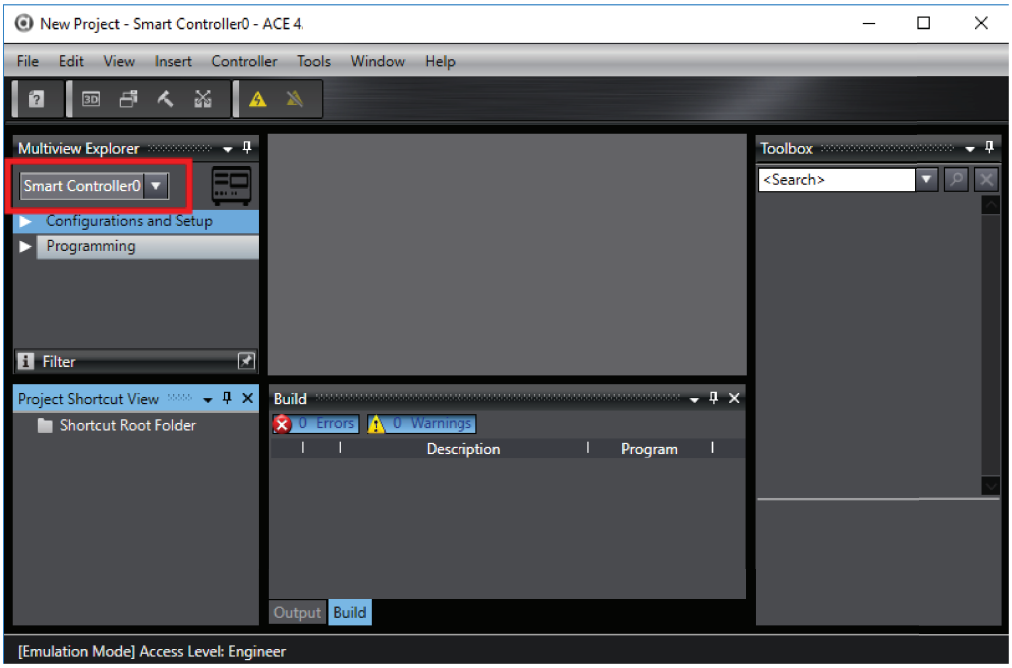


Item		Description
Project Properties	Project name	Provide a name of the project for identification purposes.
	Author	Provide an author for the project. The default author can be changed in the option settings. Refer to *5-14-2 Project Settings* on page 5-58 for more information.
	Comment	Add comments about the project for additional information.
	Type	Select the project type. • Standard Project This type is used for a normal project. • Library Project This type is used to create a library project (reserved for future use).
Select Device	Category	Select the device category. Select a Robot Controller or an Application Manager based on the application requirements.
	Device	Select the device details based on the chosen category and application requirements.

3 Make a selection for **Open in Emulation Mode**. Refer to *1-4-1 Emulation Mode* on page 1-9 for more information.

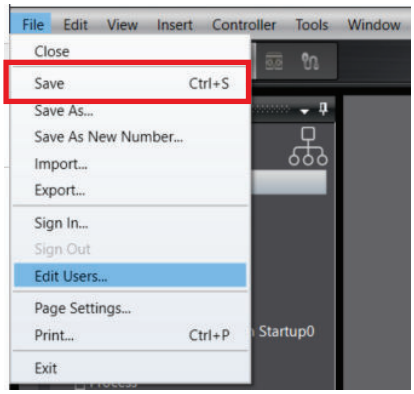


4 Click the **Create** Button. A project file is created and the following window is displayed with the specified device inserted.



3-3 Saving the Project File

This section describes how to save the project file.
ACE software project files are serialized and saved on the PC at C:\OMRON\Ace\Data\Solution. This directory can be backed up if necessary.
This directory applies when the default software installation path is used.
All project files that have been saved will be displayed in a list when the **Open Project** selection is made. Refer to *3-7 Opening a Project File* on page 3-16 for more information.
Do not modify data internal to the default save location. Data can be corrupted and project files can be lost as a result.
To save an existing project file, select **Save** from the File Menu (or press the **Ctrl + S** keys).

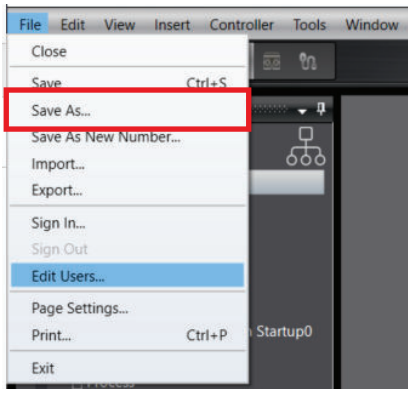


To save a new project, select **Save As**. Enter a specific project name, do not accept the default, this may create multiple projects with the same name but different serial numbers.
Projects with the same name may cause runtime issues when opened with AutoStart. Confirm each project has a unique name.
The new project file is saved. To use a project file on a different computer, export the project file as described in *3-9 Exporting a Project File* on page 3-20.
Refer to *6-2-6 Pull V+ Memory* on page 6-5 for information about saving controller memory contents to the project file.
For further information about version control, refer to *Sysmac Studio for Project Version Control Function Operation Manual (Cat. No. W589)*.

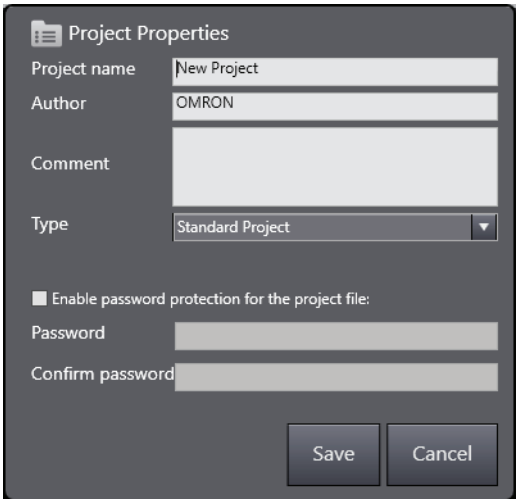
3-4 Saving a Project File Under a Different Name

This section describes how to save a project file under a different name.

Select **Save As** from the File Menu. This will display the Project Properties Dialog Box.



Change the project file name and any other project property details, and then click the **Save** Button.



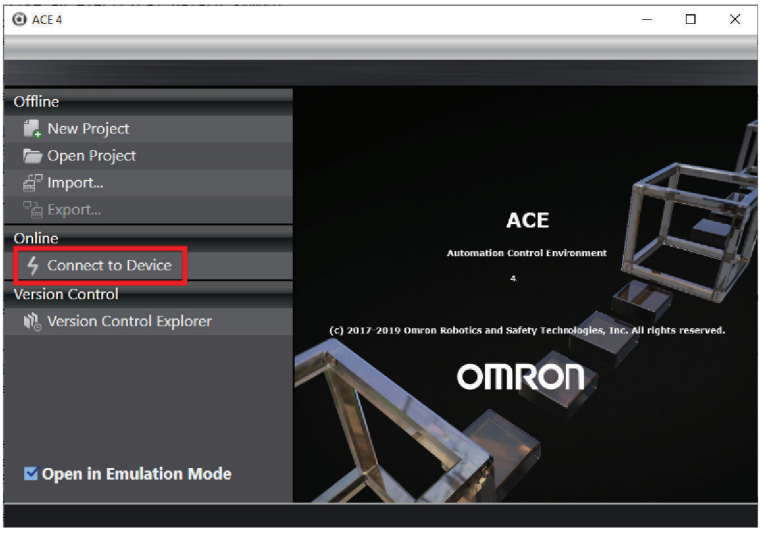
Additional Information

For information about project file password protection, refer to *3-10 Password Protection for Project Files* on page 3-23.

3-5 Creating a Project from an Online Connection

If you connect to a device without specifying a project, a new project is created automatically. Use the procedure below to connect to a device and create a new project after uploading from the robot controller memory.

1 Click **Connect to Device** from the Start page. This will open the Connect to Device Dialog Box



2 Make the appropriate connection settings in the Connect to Device Dialog Box and click the **Connect** Button (refer to *Section 4 Online Connections to Robot Controllers* on page 4-1 for more information).

3 After the connection is established, the Application window will be shown. A new project has been created from an online connection.
This new project gets a default project name. Select **Save as...** from the File Menu to save this project with a different name and adjust project properties if necessary.

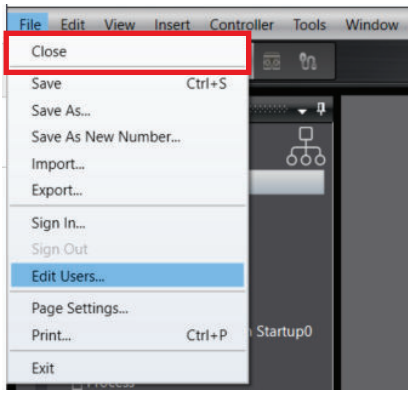
Additional Information

To capture the V+ memory into the new project, pull V+ memory from the robot controller. Refer to *6-2-6 Pull V+ Memory* on page 6-5 for more information

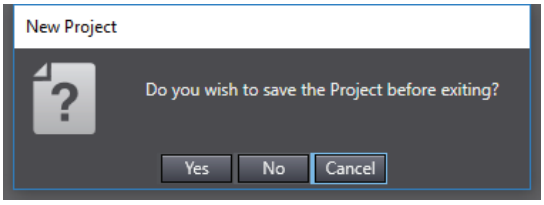
3-6 Closing a Project and Returning to the Start Page

Use the following procedure to close a project file and return to the Start page.

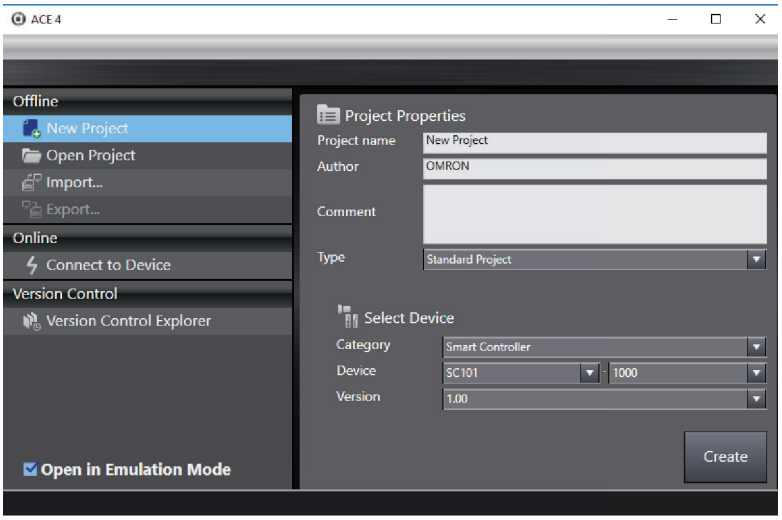
1 Select **Close** from the File Menu.



2 A dialog box is displayed to ask if you need to save the project. Click the **Yes** Button or **No** Button for the project saving choice.



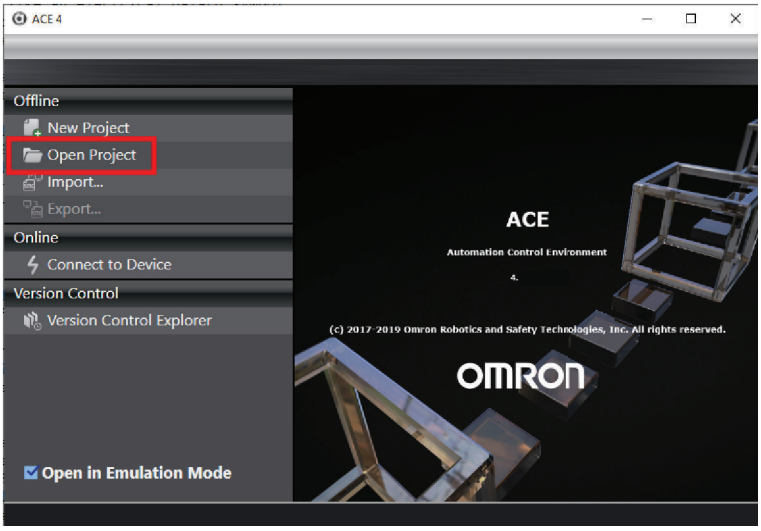
3 The Start page is displayed.



3-7 Opening a Project File

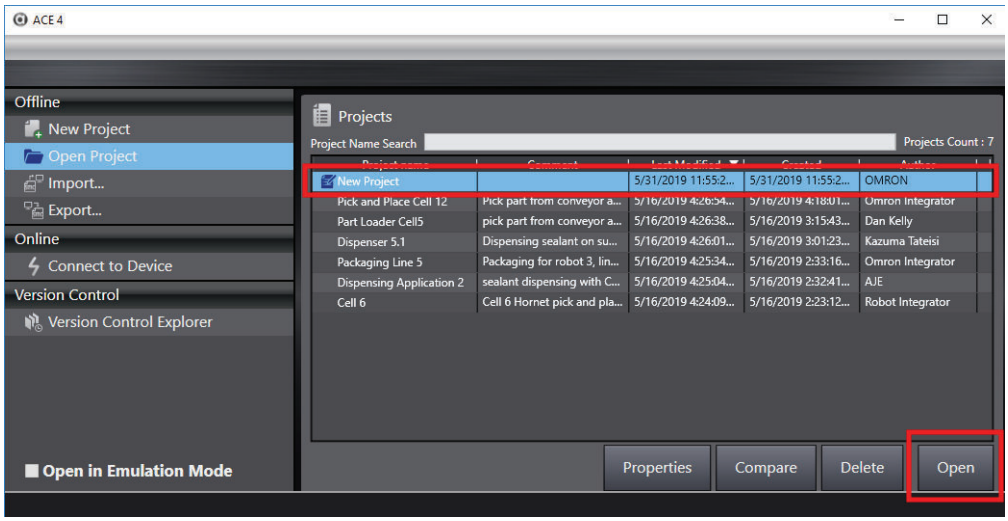
Use the following procedure to open an existing project file.

1 Click **Open Project** on the Start page. This will display the Project Screen.



2 Find a project by searching for its name or selecting it from the project list and click the **Open** Button. This will open the project. The following information is displayed to help with the selection of the correct project file to open.

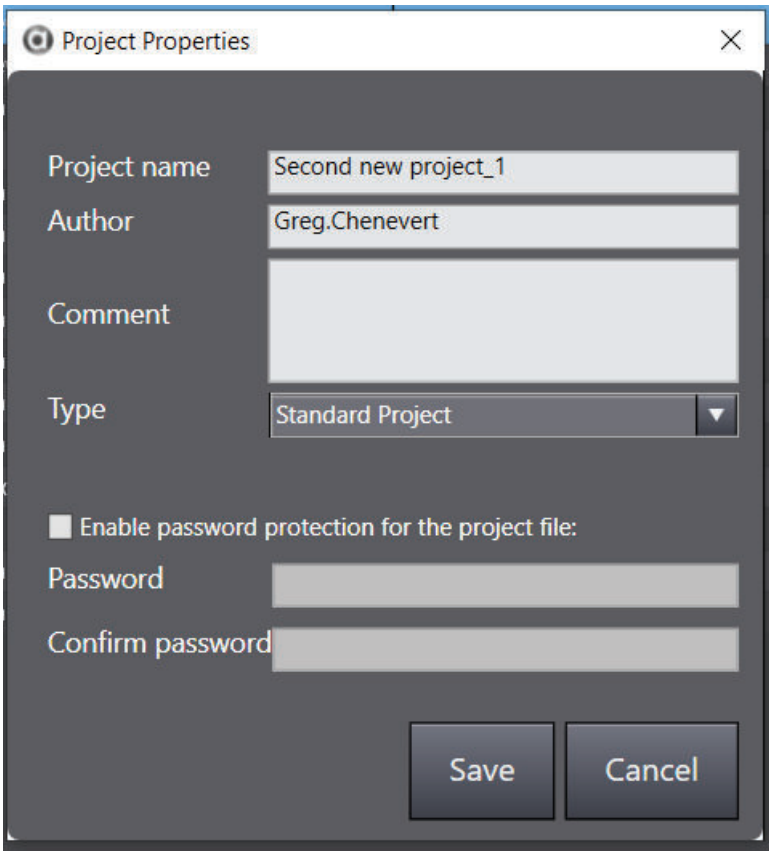
Item	Description
Project name	The project names that were entered when the projects were created are displayed. The default name is **New Project** when creating a new project. The default name is **Auto Connect Project** when **Connect to Device** is used.
Comment	The comment that was entered when the project was created.
Last Modified	The last date that the project was modified.
Created	The date and time that the project was created.
Author	The name that was entered when the project was created.



Additional Information

A project's properties can be accessed and edited by selecting the project first, and then clicking the edit icon ( ) or the **Properties** Button. A project can be deleted by selecting the project and then clicking the **Delete** Button.

The **Compare** Button is reserved for future use.
The Project Properties panel allows the project to be set with password security, Refer to *3-10 Password Protection for Project Files* on page 3-23 for more information.



To enable password protection, click the check box, then enter and confirm the password and then click **Save**.

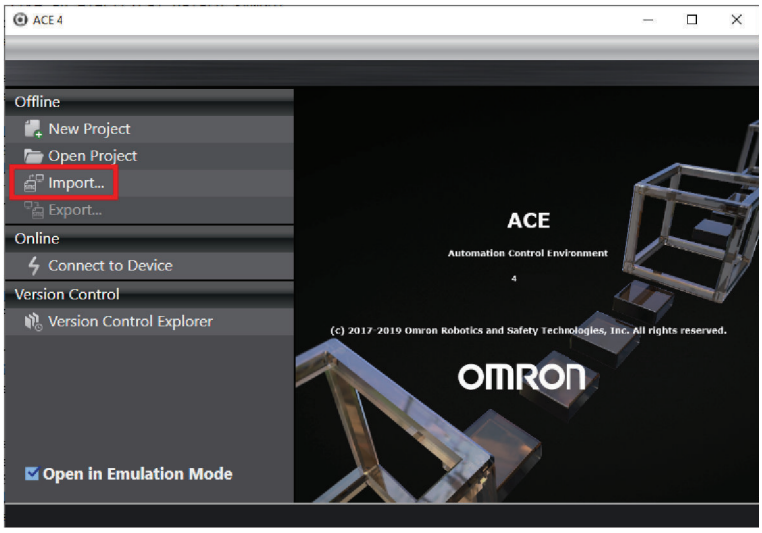
3-8 Importing a Project File

Project files can be imported with three methods described below.

- Import from the Start page
- Import from the Menu Bar in the Application Window
- Import an .awp2 ACE project file

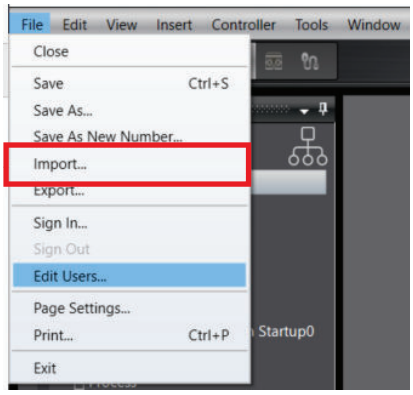
3-8-1 Importing from the Start Page

Select **Import** on the Start page and then browse to find the .awp2 ACE project file to import.



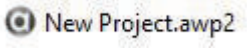
3-8-2 Importing from the Menu Bar in the Application Window

Select **Import** from the File Menu in the Application Window and then browse to find the .awp2 ACE project file to import.



3-8-3 Importing an .awp2 ACE Project File

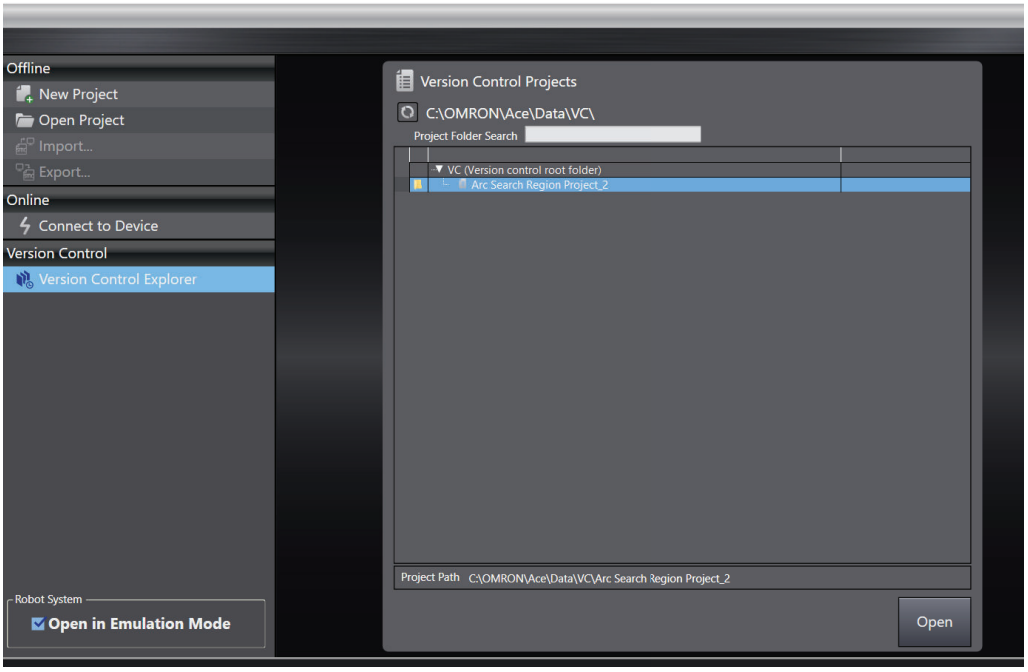
Browse to an .awp2 ACE project file with a file explorer:



Opening the ACE project file will automatically launch the ACE software, import a new project file, and open the project.
Repeated opening of a particular .awp2 ACE project file will result in duplicate copies of the project. This method should be used for projects not previously imported.

3-8-4 Importing from Version Control

With ACE opened, click the **Version Control Explorer** to open the Version Control Projects.



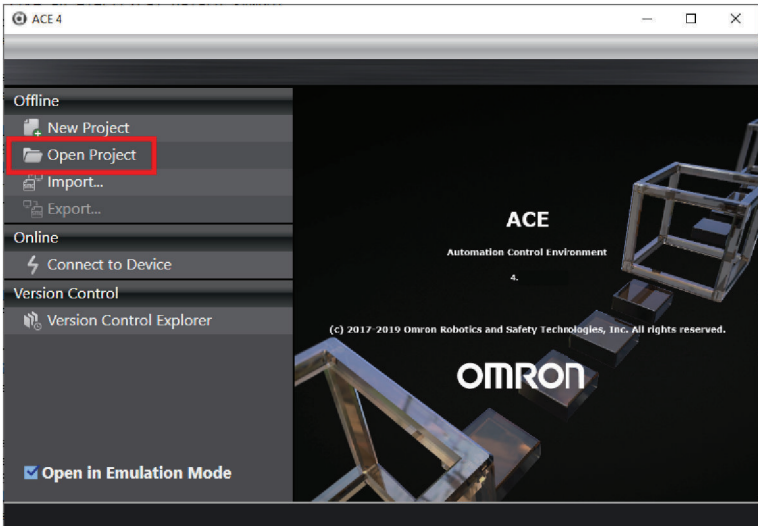
Select a project and click **Open**. The project opens in the Multiview Explorer.

3-9 Exporting a Project File

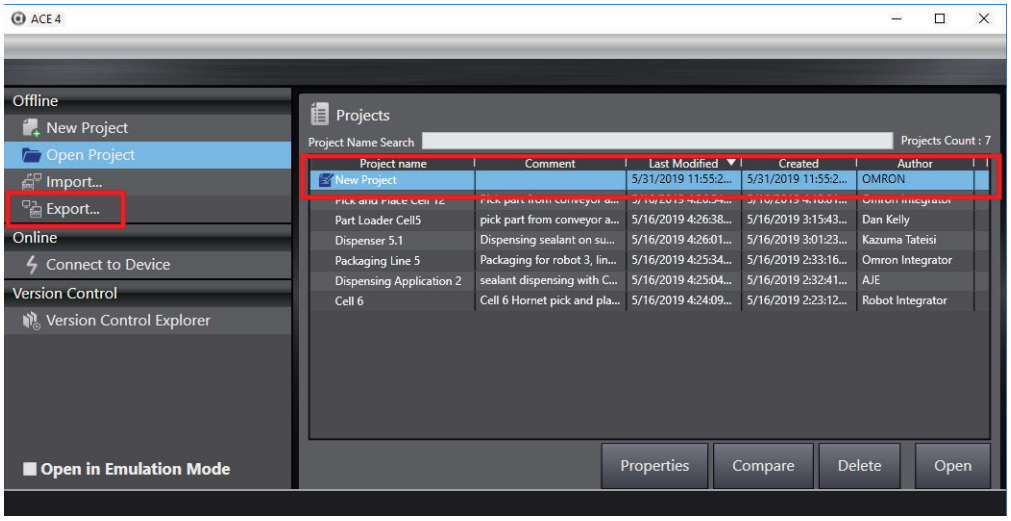
Project files can be exported from the Start page and also from the Application Window. Both procedures are described below.
The exported project file has the same contents as a project that was saved in the ACE software.

3-9-1 Exporting from the Start Page

1 Select **Open Project** on the Start Page.

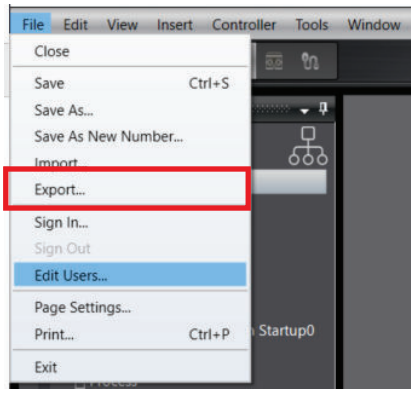


2 Select a project to export from the list of project names and click **Export**. The Export Project Dialog Box will be displayed.

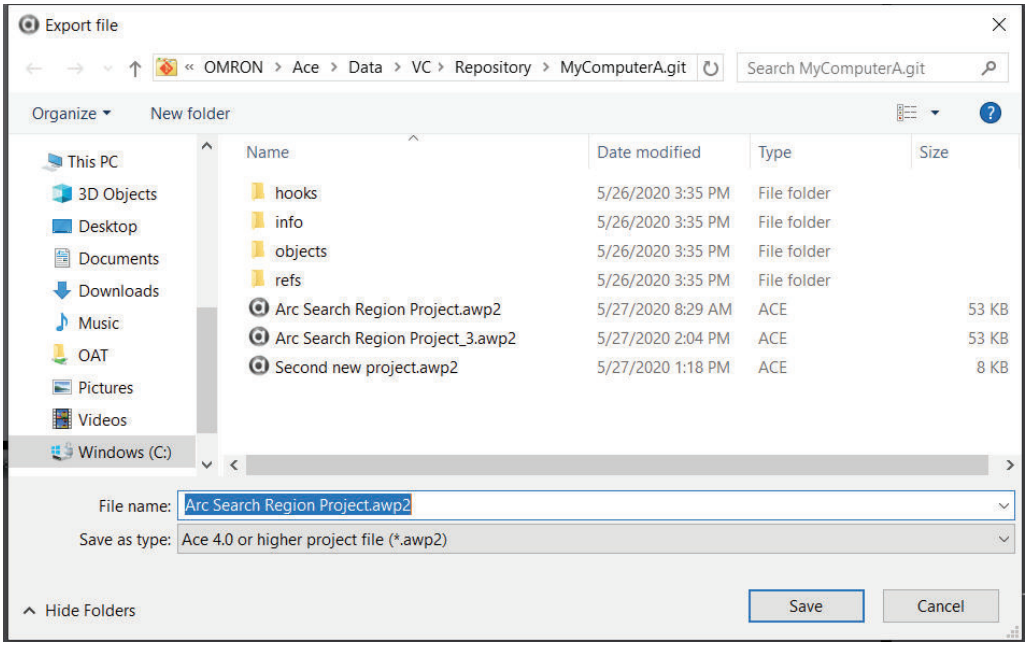


3-9-2 Exporting from the Application Window

Select **Export** from the File Menu in the Application Window. The *Export Project* Dialog Box will be displayed.



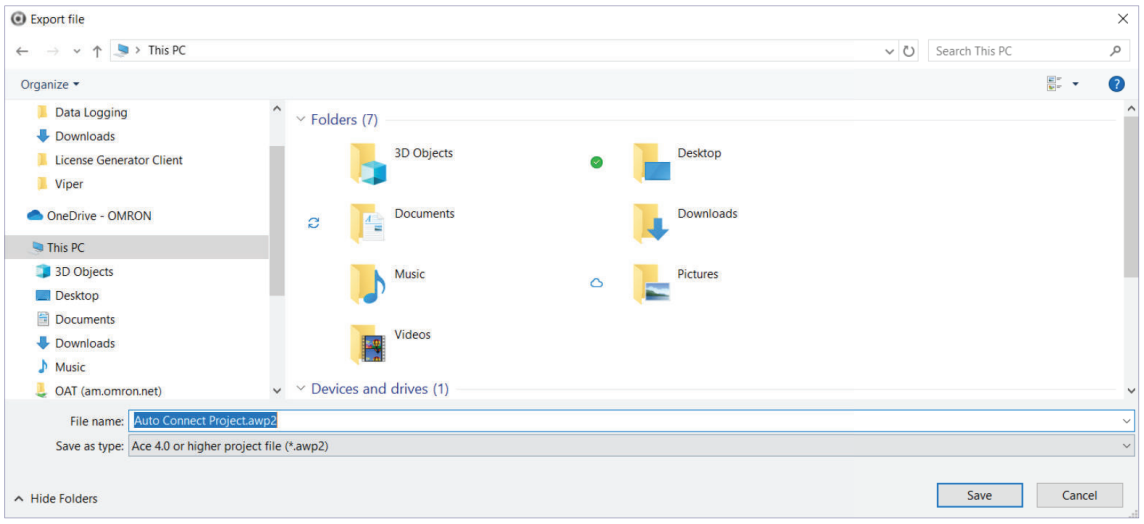
When Git is installed onto the computer, or the ACE application is connected to a remote repository, clicking **Export** opens the repository view.



Refer to *A-4-1 Git Configuration* on page A-14 for details about Git repositories.

3-9-3 Saving an Exported Project File

Specify a directory, enter a file name, select the file type, and then click the **Save** Button. This will export the project to a file.



Additional Information

You can export the contents of a project file for which password protection is set. Refer to *3-10 Password Protection for Project Files* on page 3-23 for the procedure to set password protection.

3-10 Password Protection for Project Files

You can set password protection for project files to protect your assets.
Passwords are case sensitive and must be a string of 8 to 32 alphanumeric characters.
Password protection for project files does not apply to data that is transferred to the SmartController
If password protection is set, a password is required for the following operations.

- Opening the project file
- Changing the properties of the project file
- Importing the contents of the project file

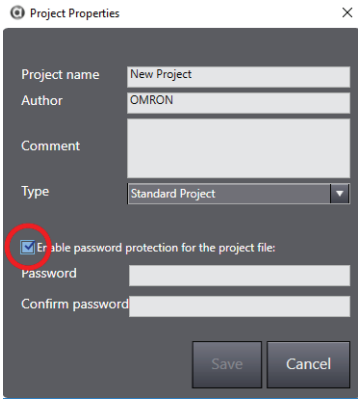
If you forget the password, you will no longer be able to open the project file, change the properties of the project file, or import the contents of the project file. Be sure to record the password in case you forget it.

3-10-1 Setting Password Protection for Project Files

Use the following procedure to set password protection for project files.
Click **Open Project** on the Start page. Then, click the **Properties** Button or click the Edit **Properties** icon

1 Click **Open Project** on the Start page. Then, click the **Properties** Button or click the **Edit Properties** icon (). This will display the Properties Dialog Box.

2 Select the **Enable password protection for the project file** Check Box.



3 Enter the password, confirm the password (these must match) and then click the **Save** Button.
The file is saved and password protection is set for this project.
The Edit Properties icon displayed in the project file list will indicate password protection.

No password protection set: 

Password Protection set: 

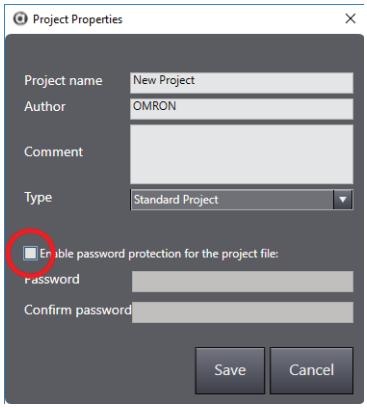
3-10-2 Removing Password Protection for Project Files

Use the procedure below to remove password protection for a project file.

1 Click **Open Project** on the Start page. Then, click the **Properties** Button or click the **Edit Properties** icon ( ). This will display the password entry prompt.

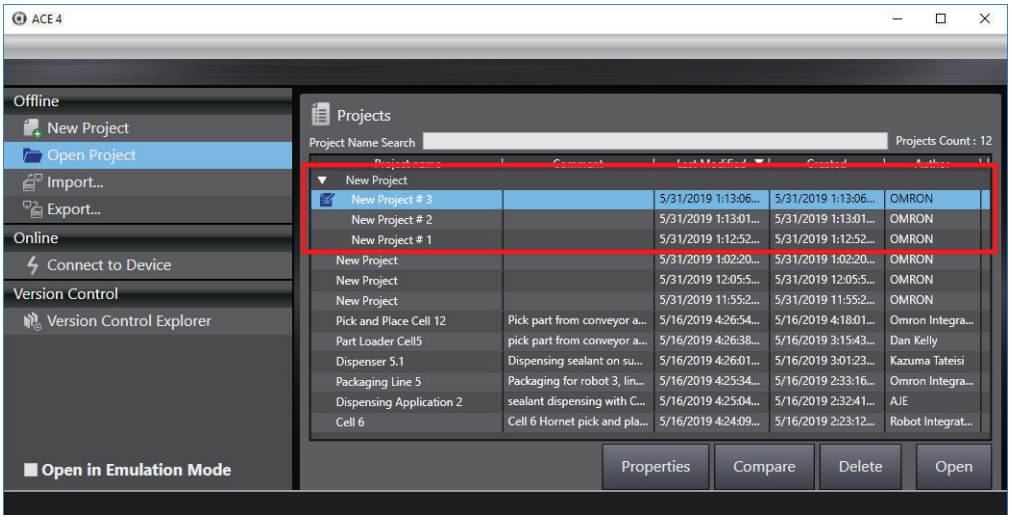
2 Enter the correct password for the project and click the **OK** Button. This will display the Project Properties Dialog Box.

3 Uncheck the **Enable password protection for the project file** and then click the **Save** Button. This will remove password protection for the project.

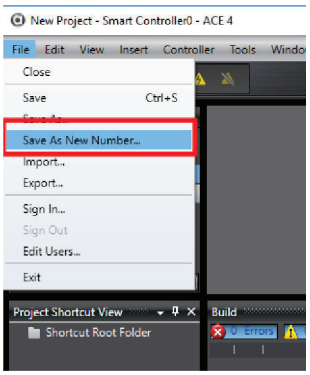


3-11 Project Version Numbers

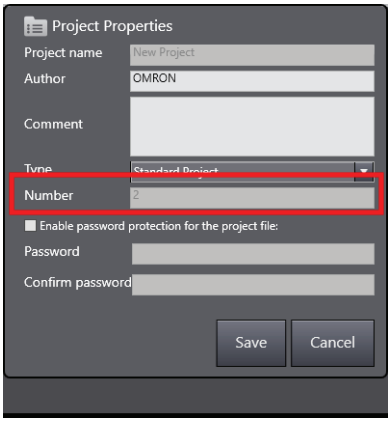
You can assign numbers to projects to manage the project history. The project numbers are displayed in a hierarchy in the Open Project Screen.
Once a new project number is created, the number cannot be changed.



Use the **Save As New Number...** from the File Menu to assign an incrementing number to an open project.



A new number is automatically added for the project number.



4

Online Connections to Robot Controllers

Use the information in this section to understand how to use the online connection functionality with the ACE software and Robot Controllers.

4-1 Going Online with a Robot Controller 4-2
4-1-1 Going Online from the Start Page 4-2
4-1-2 Detect and Configure Controller 4-4
4-1-3 Going Online from an Open Project 4-5
4-2 Going Offline with a Robot Controller 4-6
4-3 Memory Comparison 4-7

4-1 Going Online with a Robot Controller

An online connection must be established to allow communications between the ACE software and a Robot Controller. Online connections are also possible in Emulation Mode. Refer to *1-4-1 Emulation Mode* on page 1-9 for more information.
An online connection to a physical Robot Controller or an emulated controller can be established from the ACE software Start page or from within an existing open project.

Precautions for Correct Use

ACE software version 3.8 (and older) cannot connect to a controller that ACE software version 4.0 (or higher) has connected to previously until a zero memory or reboot have been executed.

Additional Information

You can simultaneously go online with more than one controller in a project from the ACE software. The operations that are described in this section apply to the controller currently selected in the Multiview Explorer. If there is more than one controller registered in the project, select the controller in the Multiview Explorer before attempting to go online.

If an online connection has been established, you can perform operations such as:

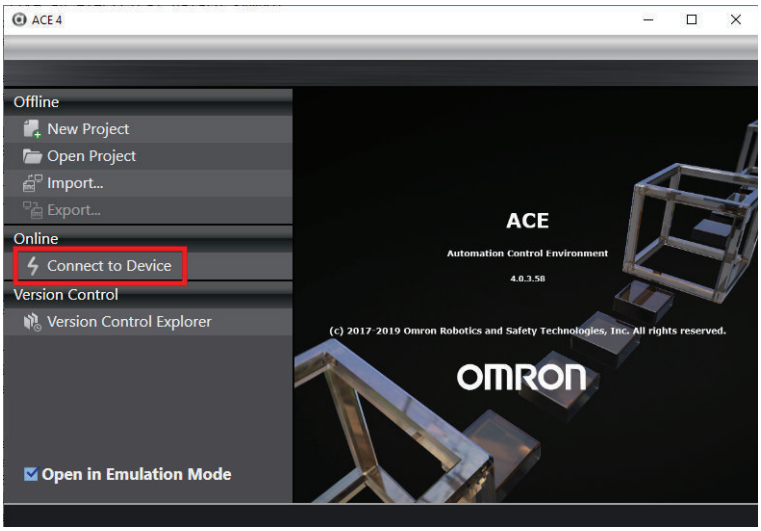
- Program debugging, transfer, and comparison
- Task management
- I/O monitoring
- File transfer
- Robot jogging
- Monitoring system performance

4-1-1 Going Online from the Start Page

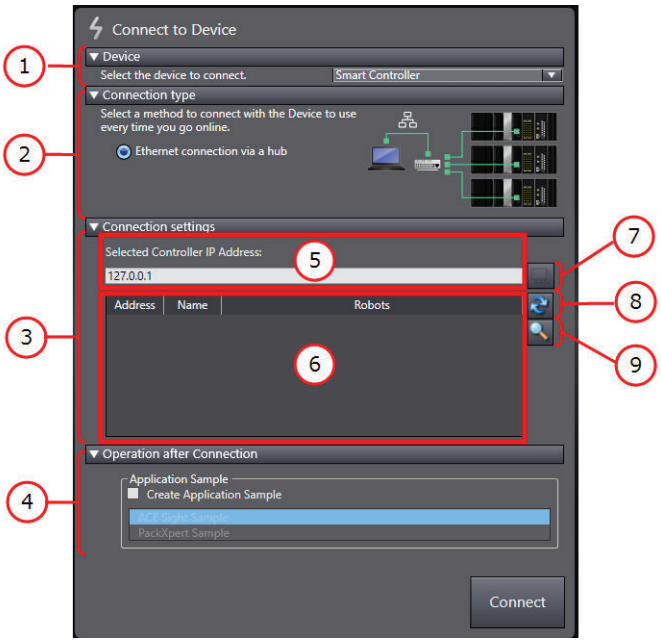
Clicking **Connect to Device** on the Start page will open the *Connect to Device* Dialog Box. The Connect to Device screen is used to prepare the online connection to a controller.

Additional Information

A successful connection to a controller with this method will automatically create a new project with the name *Auto Connect Project*. After the project is open, it is recommended to save the project with a more specific name using *Save As...* from the File menu. Closing without saving may result in an empty project.



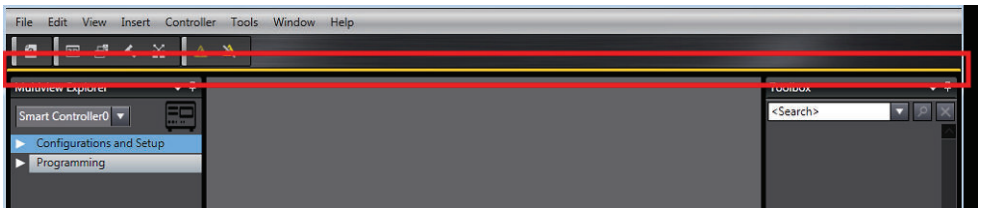
The main areas of the **Connect to Device** Dialog Box are described below.



Item	Name	Description
1	Device	Use this area to select the device to connect.
2	Connection Type	Use this area to select a connection method when going online.
3	Connection Settings	Use this area to detect and select controller IP addresses used for online connection.
4	Operation after Connection	Use this area to create an Robot Vision Manager or Pack Manager application sample after a connection is established.
5	Selected Controller IP Address	Used to manually enter a controller IP address (such as when connecting through a gateway) or displays the currently selected controller IP address. To create a project in Emulation Mode that uses a specific controller IP address, use this area to set the IP address before connecting. The controller created in Emulation Mode will have the IP address entered into this field.

Item	Name	Description
6	Detected Controller IP Addresses	After a controller has been detected, it will appear in this area for selection A controller configured to use a gateway will not appear in this area. If the network configuration is correct, the controller should be able to connect by manually entering the IP address of the desired controller. Refer to *4-1-2 Detect and Configure Controller* on page 4-4 for more information.
7	Monitor Window	Used for opening the Monitor Window for the selected controller. Refer to *7-4 Monitor Window* on page 7-37 for more information.
8	Refresh	Used to refresh the detected controller IP address area.
9	Detect and Configure Controller	Opens the *Controller IP Address Detection and Configuration* Dialog Box. Refer to *4-1-2 Detect and Configure Controller* on page 4-4 for more information.

If the Controller appears in the *Detected Controller IP Address* area, select it and click the **Connect** Button. This will initiate the online connection process. If the Controller does not appear, refer to *4-1-2 Detect and Configure Controller* on page 4-4 for more information.
The project workspace appears and a yellow bar is present under the toolbar when an online connection is present.



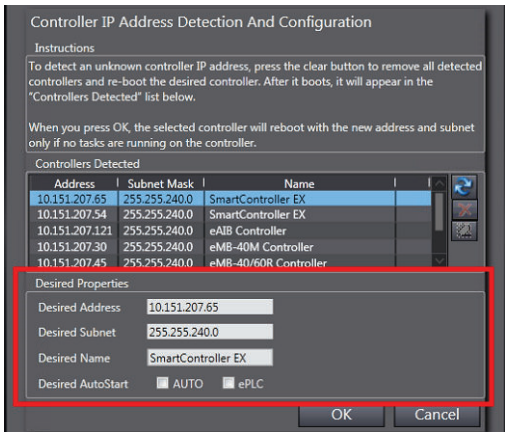
4-1-2 Detect and Configure Controller

This area is used to detect and configure controllers on a network. In this area, a detected controller's properties can be modified to establish an online connection. You can also change the Desired AutoStart behavior for a connected device in this area. Refer to *Configure System Settings* on page 7-23 for more information.
If a controller is detected, but resides on a different LAN and does not have a gateway configured, an online connection is not possible until one of the following actions are taken.

- The detected controller's IP address and/or subnet is changed for compatibility with the PC's LAN connection. Change the Desired Properties Address, Subnet, and Name accordingly in the area shown below.

Additional Information

Refer to *Configure System Settings* on page 7-23 to change the address, subnet, or name of a connected controller.



- The PC's IP address and LAN connection are changed to be compatible with the controller. Adjust the PCs network adapter properties accordingly.

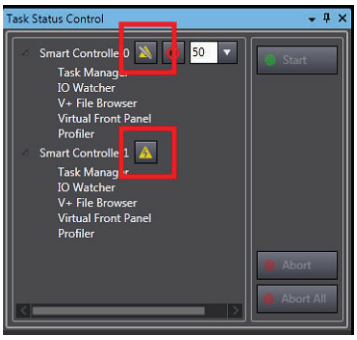
4-1-3 Going Online from an Open Project

There are three methods available for going online from an open project:

- Use the **Online** icon ( ) in the toolbar to go online with the controller selected in the Multiview Explorer.
- Select **Online** from the Controller Menu to go online with the controller selected in the Multiview Explorer.
- Click the **Online** icon ( ) in the Task Status Control area to go online with the associated controller.

  Refer to *5-9 Task Status Control* on page 5-40 for more information about Task Status Control.

Use the **Online** and **Offline** Buttons to control the online connections to each controller.



These methods rely on a correct IP assignment for the controller. Refer to *7-2 Controller Settings* on page 7-5 for more information.

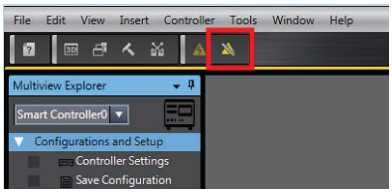
Additional Information

The Controller Connection Startup object allows the ACE software to automatically establish an online connection when the project opens. Refer to *8-5-1 Controller Connection Startup* on page 8-61 for more information.

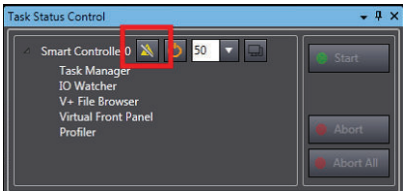
4-2 Going Offline with a Robot Controller

Use the offline function to close an active connection to a controller.
Use one of the following methods to go offline with a controller.

- Click the **Offline** Button in the toolbar.



- Click the **Offline** icon in the Task Status Control area (refer to *5-9 Task Status Control* on page 5-40 for more information).



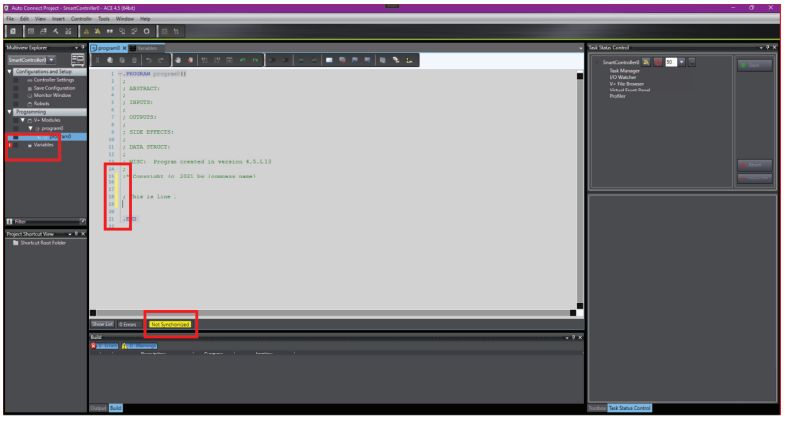
4-3 Memory Comparison

When an online connection is initiated, after opening a project, ACE makes a comparison between the controller memory and the ACE project memory operating on the PC. Upon a successful connection with a controller, ACE will try and synchronize its memory by comparing V+ modules, V+ programs and V+ Variables with the controller memory. If any of the above are different between ACE and the controller memory, ACE will indicate this to the user in the Multiview Explorer with red icons.

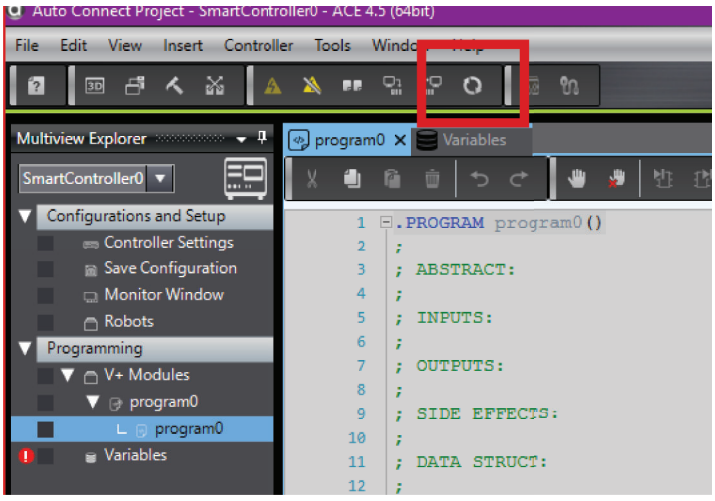
Additional Information

Refer to *5-14-7 Connection Settings* on page 5-64 for more information about default memory comparison settings.

With the V+ editor opened and synchronized between the PC and controller, any changes to a program are written to both the PC and controller. Any unsynchronized changes between the ACE project and the controller memory are reflected in the V+ Editor window with yellow color indicating the unsynchronized changes. The ACE editor shows the differences between the ACE project and the controller memory with three indicators: red icons next to the program, a yellow bar within the editor next to the unsaved V+ code and the yellow icon on the bottom tab, as shown in the following figure.

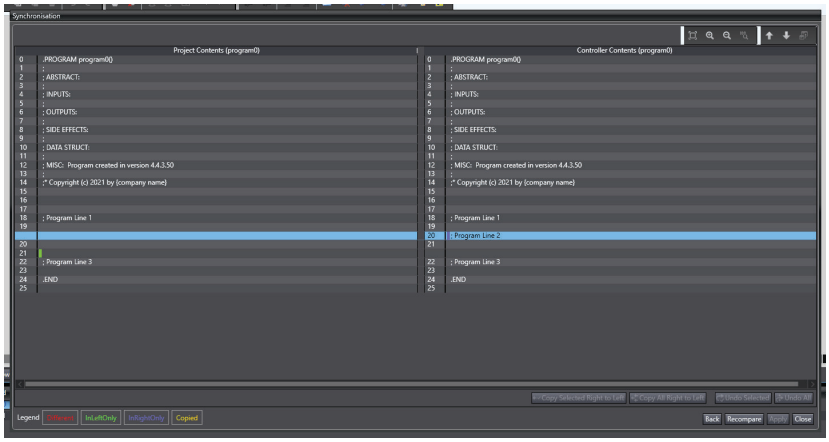


When updates to the V+ code are complete, click the **Synchronize** Button to compare differences between project data and the controller memory.

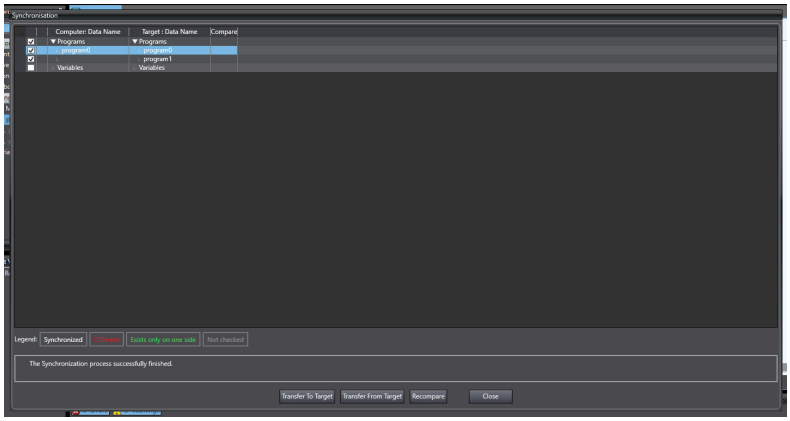


ACE opens a panel, showing a comparison of the controller memory and ACE project memory, as shown in the following figure. The Legend on the bottom left of the screen corresponds to the colored

bar on the left side of the Comparison Panel. Click, **Recompare** to show a comparison between a selected V+ program as it exists in the ACE project memory on the left hand side, and the V+ program as it exists in the controller memory on the right hand side, as shown in the following figure.



To synchronize programs that are different, either pull the unchanged controller program into the ACE project memory or push the ACE project memory to the controller. Pull from the controller memory by clicking the **Transfer From Target** Button. Push the changed program from your ACE project memory to the controller by clicking the **Transfer To Target** Button as shown in the following figure.



Precautions for Correct Use

The is no recovery if the ACE project V+ memory is overwritten with the contents of the controller. Consider selecting **Save As...** from the File Menu to save a new backup copy of the project before pushing or pulling the program. Use the backup to recover the ACE V+ memory.

5

Understanding the ACE User Interface

This section provides the names, functions, and basic arrangement of the ACE software user interface.

5-1 The Main Application Window 5-3
- 5-1-1 Main Operation Window 5-3
- 5-1-2 Menu Command Structure 5-4
- 5-1-3 Toolbar Icons 5-9
- 5-1-4 Operation Toolbar Icons 5-10
- 5-1-5 Multiview Explorer 5-10
- 5-1-6 Filter Pane 5-12
- 5-1-7 Project Shortcut View 5-13
- 5-1-8 Status Bar 5-14
- 5-1-9 Edit Pane 5-14
- 5-1-10 Multifunction Tabs 5-15
- 5-1-11 Special Tools 5-15
- 5-1-12 Other Application Window Functions 5-16

5-2 Toolbox 5-23

5-3 3D Visualizer 5-24
- 5-3-1 3D Visualizer Basic Features 5-24
- 5-3-2 Creating a 3D Workspace 5-25
- 5-3-3 3D Visualizer Window 5-25
- 5-3-4 Graphical Collision Detection 5-29

5-4 Output Tab Page 5-31

5-5 Cross Reference Tab Page 5-32

5-6 Build Tab Page 5-33

5-7 Event Log 5-34

5-8 V+ Jog Control 5-35
- 5-8-1 Robot Section 5-35
- 5-8-2 Current Position Section 5-36
- 5-8-3 Jog Control Section 5-36
- 5-8-4 World, Joint, Tool Selection 5-37
- 5-8-5 Location Section 5-37

5-9 Task Status Control 5-40
- 5-9-1 Online/Offline 5-40
- 5-9-2 Robot Power Control 5-40
- 5-9-3 Monitor Speed Setting 5-40
- 5-9-4 Open the Monitor Window 5-41

5-9-5 Task Manager 5-41
5-9-6 I/O Watcher 5-43
5-9-7 V+ File Browser 5-43
5-9-8 Virtual Front Panel 5-45
5-9-9 Fieldbus Status 5-46
5-9-10 Profiler 5-48
5-9-11 Application Manager Control 5-49
5-9-12 Control Buttons 5-49

5-10 Vision Window 5-50
5-10-1 Zoom Level 5-50

5-11 V+ Watch 5-52
5-11-1 Adding Variables to the V+ Watch Window 5-52

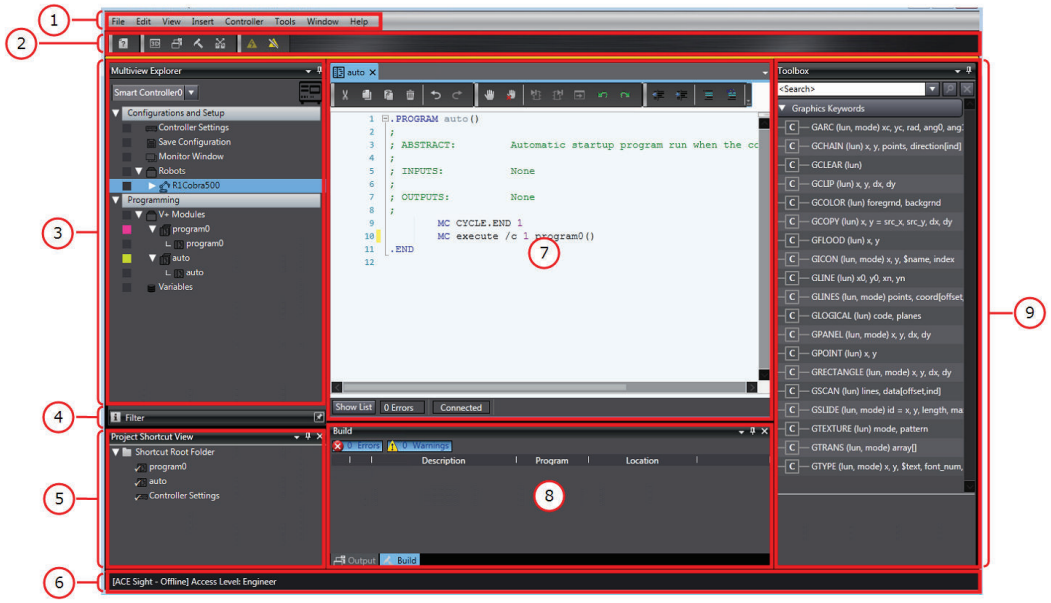
5-12 Search and Replace 5-53
5-12-1 Search Options 5-53
5-12-2 Button Functions 5-53

5-13 User Management 5-55
5-13-1 Passwords 5-56
5-13-2 Engineer Accessible Items 5-56
5-13-3 Technician Accessible Items 5-56
5-13-4 Operator Accessible Items 5-56

5-14 Project Options 5-57
5-14-1 Color Theme 5-57
5-14-2 Project Settings 5-58
5-14-3 Window 5-58
5-14-4 V+ Programs 5-59
5-14-5 V+ Variables 5-60
5-14-6 Uninterruptible Power Supply (UPS) 5-62
5-14-7 Connection Settings 5-64
5-14-8 3D Visualizer 5-65

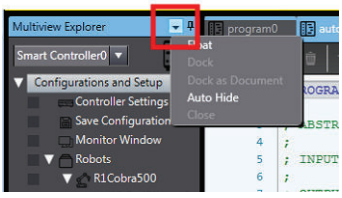
5-1 The Main Application Window

The default application window layout is described below.



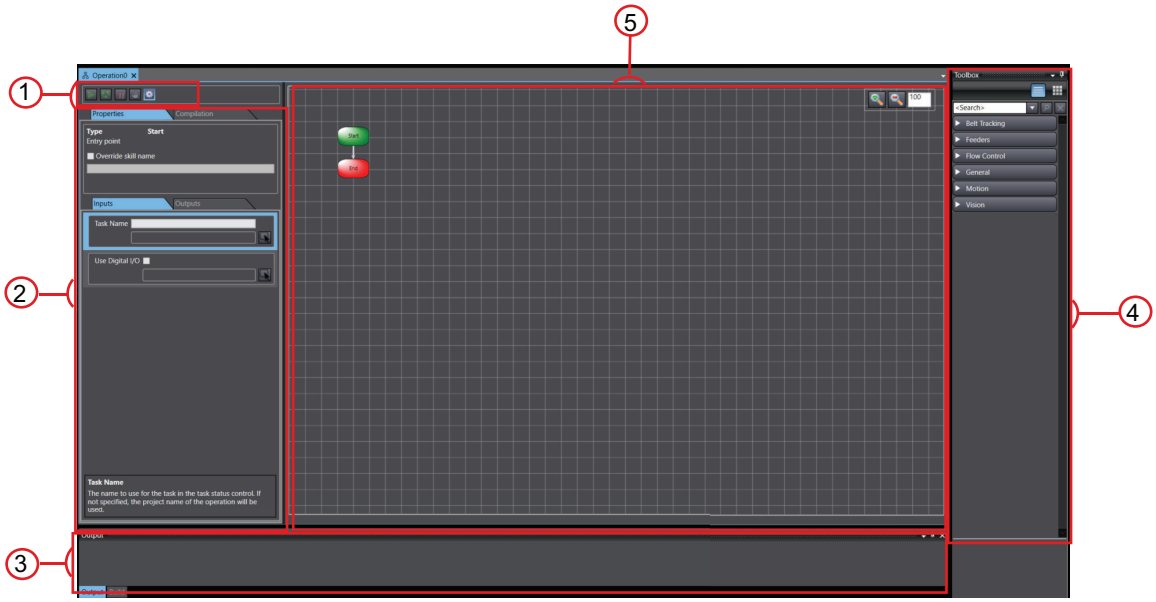
Item	Description
1	Menu Bar
2	Toolbar
3	Multiview Explorer
4	Filter pane
5	Project Shortcut View
6	Status Bar
7	Edit Pane
8	Multifunction Tabs
9	Multipurpose Tools

Float, Dock and Close functions are also accessible with the Options menu.



5-1-1 Main Operation Window

The default application window for Operation is described below.



Item	Description
1	Toolbar
2	Properties and Compilation
3	Output, Build and Event Log
4	Toolbox
5	Operation Grid

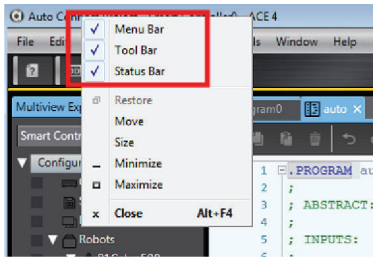
The initial view contains only a Start and End function on the Operation Grid. The Properties panel and Toolbox are also displayed on each side of the grid. Once an operation is made and active, it can be viewed in the 3D Visualizer. Refer to *5-3-3 3D Visualizer Window* on page 5-25 for more information. The Skills Library is located on the right side Toolbox. Refer to *6-7-7 Operation Skills* on page 6-36 for more information. The toolbar is located above the Properties panel.
Float, Dock and Auto-Hide functions are also enabled with the Toolbox and 3D Visualizer.

5-1-2 Menu Command Structure

The menu bar contains the following command structure.

Additional Information

Hide or show the menu bar, toolbar, and status bar by right-clicking the title bar in the main window and either selecting or deselecting the item.



File Menu

The File menu has the following contents.

Contents	Description
Close	Clears all objects and project data and returns to the ACE start screen.
Save	Saves the current project to the project library.
Save As...	Save the currently opened project with options to adjust project properties.
Save As New Number	Save the currently opened project with an incremented number.
Import...	Import an ACE project file. Refer to *3-8 Importing a Project File* on page 3-18 for more information.
Export...	Export an ACE project to a file. *3-9 Exporting a Project File* on page 3-20 for more information.
Sign In...	Sign in with a specific user access level. Refer to *5-13 User Management* on page 5-55 for more information.
Sign Out	Sign out of a specific user access level. Refer to *5-13 User Management* on page 5-55 for more information.
Edit Users...	Edit user access levels. Refer to *5-13 User Management* on page 5-55 for more information.
Page Settings	Opens page print setting.
Print	Prints the project's robot controller(s) and Application Manager(s) structure.
Exit	Exit the ACE software.

Edit Menu

The Edit menu has the following contents.

Contents	Description
Undo	Undo the last action.
Redo	Redo the last action.
Cut	Cut the selected item to the clipboard.
Copy	Copy the selected item to the clipboard.
Paste	Paste an item from the clipboard to the selected area.
Delete	Delete the selected item.
Select All	Select all items in an area or list.
Search and Replace	Open the Search and Replace tool. Refer to *5-12 Search and Replace* on page 5-53 for more information.
Find Previous	Find a previous instance of a search term.
Find Next	Find the next instance of a search term.
Jump...	Reserved for future use.
Jump to Multiview Explorer	Display the active item in the expanded Multiview Explorer.

View Menu

The View menu has the following contents.

Contents	Description
Multiview Explorer	Display the Multiview Explorer. Refer to *5-1-5 Multiview Explorer* on page 5-10 for more information.
Project Shortcut View	Display the Project Shortcut view. Refer to *5-1-7 Project Shortcut View* on page 5-13 for more information.
Toolbox	Display the Toolbox in the multipurpose tools area. Refer to *5-2 Toolbox* on page 5-23 for more information.
3D Visualizer	Display the 3D Visualizer in the multipurpose tools area. Refer to *5-3 3D Visualizer* on page 5-24 for more information.
Output Tab Page	Display the Output Tab. If the Output Tab is already displayed and selected, it will be hidden instead. This is only available when a robot controller device is selected. For more information, refer to *5-4 Output Tab Page* on page 5-31.
Cross Reference Tab Page	Display the Cross Reference Tab. If the Cross Reference Tab is already displayed and selected, it will be hidden instead. This is only available when a robot controller device is selected. Refer to *5-5 Cross Reference Tab Page* on page 5-32 for more information.
Build Tab Page	Display the Build Tab. If the Build Tab is already displayed and selected, it will be hidden instead. This is only available when a robot controller device is selected. Refer to *5-6 Build Tab Page* on page 5-33 for more information.
Event Log	Display the Event Log in the multifunction tabs area. Refer to *5-7 Event Log* on page 5-34 for more information.
Smart Project Search	Display the Smart Project Search tool as a pop up window. Refer to *Smart Project Search* on page 5-22 for more information.
Recently Closed Windows	Display the Recently Closed Windows as a pop up window. Refer to *Recently Closed Windows* on page 5-16 for more information.
Clear Recently Closed Windows History	Clear the Recently Closed Windows history. Refer to *Recently Closed Windows* on page 5-16 for more information
V+ Jog Control	Display the V+ Jog Control in the multipurpose tools area. Refer to *5-8 V+ Jog Control* on page 5-35 for more information.
Task Status Control	Display the Task Status Control in the multipurpose tools area. Refer to *5-9 Task Status Control* on page 5-40 for more information.
Vision Window	Display the Vision Window in the multipurpose tools area. Refer to *5-10 Vision Window* on page 5-50 for more information.
V+ Watch	Display the V+ Watch in the multifunction tabs area. Refer to *5-11 V+ Watch* on page 5-52 for more information.
System Monitor	Display the System Monitor in the multifunction tabs area. Refer to *5-1-12 Other Application Window Functions* on page 5-16 for more information.
Skill Library	Display the Skill Library with a list of available Skills. Refer to *6-7-7 Operation Skills* on page 6-36 for more information.
Reset Window Layout	Reset the window layout to the default view.

Insert Menu

The Insert menu has the following contents.

Contents	Description
Robot Controller	Scan for, or select and Insert a new controller into the project.

Contents	Description
Application Manager	Insert a new Application Manager in the project. Refer to *1-1 Software Overview* on page 1-2 for more information.
Application Sample	Insert application samples using configuration wizards to create a working project.

Controller Menu

The Controller menu has the following contents.

Contents	Description
Online	Go online with a robot controller. The Controller menu item is not available when the Application Manager device is selected.
Offline	Go offline with a robot controller. The Controller menu item is not available when the Application Manager device is selected.
Enable emulation mode	Enables project emulation
Disable emulation mode	Prevents project emulation
Robot Vision Manager	Enables, or disables monitoring
Clear All Memory	Clears project memory

Tools Menu

The Tools menu has the following contents.

Contents	Description
Customize Shortcut Keys...	Display the Shortcut Key Customization pop up window.
Option...	Display the project Option pop up window. Refer to *5-14 Project Options* on page 5-57 for more information.

Window Menu

The Windows menu has the following contents.

Contents	Description
Close	Close the active window.
Float	Float the active window.
Dock	Dock the active window.
New Horizontal Tab Group	Divide the Edit Pane into upper and lower sections. Only available when the Edit Pane has more than one tab open.
New Vertical Tab Group	Divide the Edit Pane into left and right sections. Only available when the Edit Pane has more than one tab open.

Contents	Description
Move to Next Tab Group	Move the selected tab page to the tab group below or to the right of the current tab group. Only available when there is a tab group below or to the right of the current tab group.
Move to Previous Tab Group	Move the selected tab page to the tab group above or to the left of the current tab group. Only available when there is a tab group above or to the left of the current tab group.
Close All But This	Close all tab pages and floating Edit Panes, except for the selected tab page.
Close All Except Active Device	Close all tab pages and floating Edit Panes for other devices, leaving those for the selected device.
Close All	Close all tab pages and floating Edit Panes, including the selected tab page.
Close Tab	Close the active Edit Pane tab.
Open Next Tab	Open the next Edit Pane tab. Only available when the Edit Pane has more than one tab open.
Open Previous Tab	Open the previous Edit Pane tab. Only available when the Edit Pane has more than one tab open.

Help Menu

The Help menu has the following contents.

Contents	Description
ACE Manuals	Access to the following resources: • Automation Control Environment (ACE) Version 4 User's Manual (Cat. No. I633) • Robot Vision Manager User's Manual (Cat. No. I667) • Automation Control Environment (ACE) Version 4 Camera Configuration User's Manual (Cat. No. I679) • V+ Module Reference Manual (Cat. No. I668) • ACE API Reference Manual • Process Manager Customization Manual
V+ Manuals	Access to the following resources when Standard Control is selected: • V+ User's Manual (Cat. No. I671) • V+ Keyword Reference Manual (Cat. No. I672) Access to the following resources when eV+2 is selected: • eV+ Language Reference Guide (Cat. No. I605) • eV+ Operating System Reference Guide (Cat. No. I607)
Diagnostic Summary	Displays project information and errors. Refer to *9-5 Diagnostic Summary* on page 9-9 for more information.
About...	Display information about the ACE software.

5-1-3 Toolbar Icons

Toolbar icons are described below.

Toolbar Icons Description

Icon	Name	Description
	Help	Opens ACE User's Manual.
	3D Visualizer	Opens the 3D Visualizer. For more information, refer to *5-3-1 3D Visualizer Basic Features* on page 5-24.
	Output Tab Page	Only available when a robot controller device is selected. Shows the Output Tab. If the Output Tab is already displayed and selected, it will be hidden instead. For more information, refer to *5-4 Output Tab Page* on page 5-31.
	Build Tab Page	Only available when a robot controller device is selected. Shows the Build Tab. If the Build Tab is already displayed and selected, it will be hidden instead. For more information, refer to *5-6 Build Tab Page* on page 5-33
	Cross Reference Tab Page	Only available when a robot controller device is selected. Shows the Cross Reference Tab. If the Cross Reference Tab is already displayed and selected, it will be hidden instead. For more information, refer to *5-5 Cross Reference Tab Page* on page 5-32
	Online	Opens communication between the IPC and the specified controller or between the Application Manager and the specified server, depending on which type of object is selected. This is disabled if the device is already online. Refer to *4-1 Going Online with a Robot Controller* on page 4-2 or more information. .
	Offline	Closes communication between the IPC and the specified robot controller or between the Application Manager and the specified server, depending on which type of object is selected. This is disabled if the device is already offline. Refer to *4-2 Going Offline with a Robot Controller* on page 4-6 or more information.
	Push V+ Memory	Only available when a robot controller device is selected. Pushes all V+ modules from ACE to the controller. This is disabled if there are no detected differences between the V+ code in ACE and the controller. For more information, refer to *4-3 Memory Comparison* on page 4-7.
	Pull V+ Memory	Only available when a robot controller device is selected. Pulls all V+ modules from the controller to ACE. This is disabled if there are no detected differences between the V+ code in ACE and the controller. For more information, refer to *4-3 Memory Comparison* on page 4-7.
	Synchronize	Opens the Synchronize window. If a robot controller is selected, the window will compare ACE and the controller. If an Application Manager is selected, the window will compare the client with the server. For more information, refer to *4-3 Memory Comparison* on page 4-7.
	Check V+ Memory	Only available when a robot controller device is selected. Compares the V+ Modules in the project to those in the controller. Differences are marked as errors in the Multiview Explorer. This is disabled if the device is not online. For more information, refer to *4-3 Memory Comparison* on page 4-7.
	Enable Emulation Mode	Closes and reopens the project with Emulation Mode enabled. If there is unsaved data, a dialog box will appear asking if the user wants to save before continuing. For more information, refer to *1-4-1 Emulation Mode* on page 1-9.

Icon	Name	Description
	Disable Emulation Mode	Closes and reopens the project with Emulation Mode disabled. This button enables a quick change from Emulation Mode to using physical hardware, including cameras and controller devices. If there is unsaved data, a dialog box opens with the option to save the project before continuing.

5-1-4 Operation Toolbar Icons

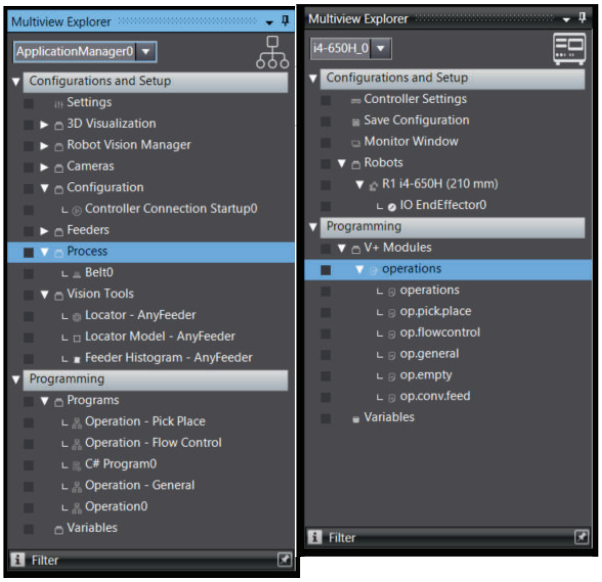
Operation has five icons in the toolbar. The icons are used to control the Operation process.

Icon	Name	Description
	Run	Starts the process completing a full cycle.
	Reset	Brings process to the start point, initializing the process.
	Pause	Pauses the cycle.
	Toggle Break	When selected, clicking the **Run** icon cycles the process a single step at a time.
	Compile	Pushes the code to the controller.

5-1-5 Multiview Explorer

Use the Multiview Explorer pane to access for all ACE project data. It is separated into *Configurations* and *Setup and Programming*.

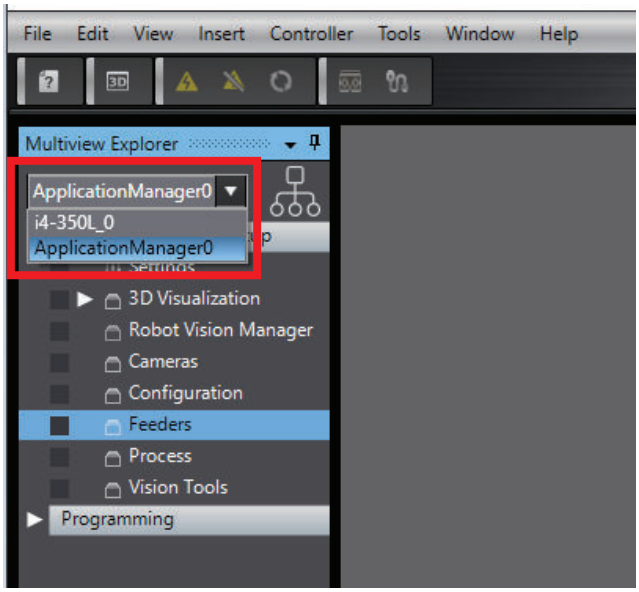
Click the icons (▶ or ▼) in front of each item in the tree to expand or collapse the tree.



Device List and Device Icon

An ACE project can include robot controller devices, Application Manager devices, or both.

The Device List contains all devices registered in the project. Use the Device List to select the active device and access the objects associated with that device. The active device and associated toolbar and menu items can automatically change based on the Edit Pane object that is currently selected.

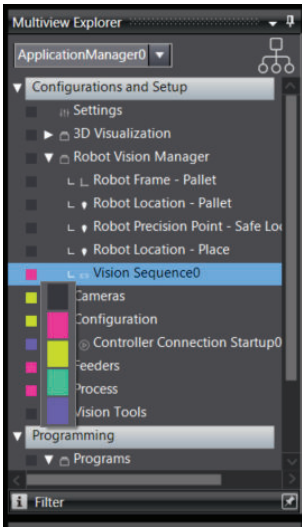


Right-click the icon (or) to Add Device, Change Device, Rename, and Delete a Device or Switch Device Control Display. Select **Switch Device Control Display** to switch from a drop down view (default) to a list view.

Color Codes

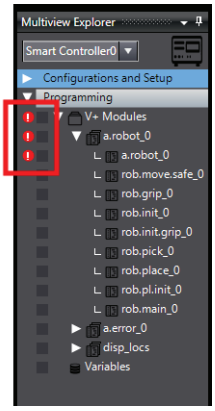
Left-click the square mark next to the object name in Multiview Explorer to view the color marks. Use color marks on the categories and objects of the Multiview Explorer to indicate associated objects within a device. The following example uses color codes to indicate which vision tools are used in each Vision Sequence.

These colors can also be used as filter conditions in the Filter Pane that is described later in this section.



Error Icons

Error icons in the Multiview Explorer indicate an object that is improperly configured or a program that is not executable.

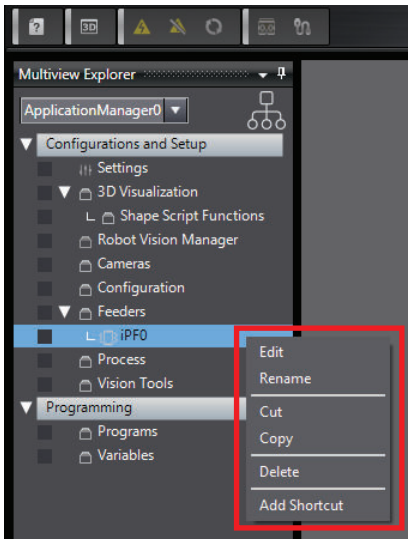


Precautions for Correct Use

- A program will not execute with an error present.
- A Process Manager will not execute when an error is present on an associated object.

Context Menu

A context menu is available for objects in the Multiview Explorer. Access the context menu by right-clicking an object. All available actions for that object will be displayed. An example is provided below.



5-1-6 Filter Pane

Click the Filter Pane Bar to display the Filter Pane. Use the Filter Pane to search for color codes and for items with an error icon. The results for the search condition are displayed in a list of items with a specific color code or items with error icons. Click any item in the search result to display the item in the Edit Pane.

The Filter Pane is hidden automatically if the mouse is not placed in it for more than five seconds at a time. Automatically hiding the Filter Pane can be canceled by clicking the **Pin** icon ( ).

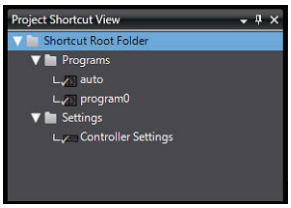


5-1-7 Project Shortcut View

The Project Shortcut View provides a convenient location for storing short cuts to Multiview Explorer items. Any item placed in this view can be quickly opened with a double-click. Use this area to group shortcuts to objects belonging to multiple devices for quick reference, without changing the selected device in the Multiview Explorer.

To add an item to the Project Shortcut View, right-click it in the Multiview Explorer and select **Add Shortcut**. An item can also be added by right-clicking a folder in the Project Shortcut View and then clicking **Add Shortcut of Selected Item** (any item that is currently selected in the Multiview Explorer will be added). A new shortcut to the item will be placed in the Project Shortcut View.

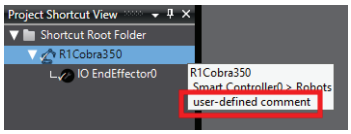
To access functions in the Project Shortcut View, right-click the item to open a menu. The functions of the Project Shortcut View are described below.



Item	Item Name	Description
Folder	Add Folder	Add a new folder.
	Add Shortcut of Selected Item	Add a shortcut of the item currently selected in the Multiview Explorer.
	Cut Folder	Cut a selected folder.
	Copy Folder	Copy a selected folder.
	Paste Folder	Paste a folder to the selected location.
	Delete Folder	Delete a selected folder.
	Rename Folder	Rename a selected folder.
Shortcut	Multiview Explorer Menu	Access the item's Multiview Explorer menu.
	Cut Shortcut	Cut a shortcut.
	Copy Shortcut	Copy a shortcut.
	Paste Shortcut	Paste a shortcut.
	Delete Shortcut	Delete a shortcut.
	Jump to Multiview Explorer	Highlight and locate the selected item in the Multiview Explorer.

Additional Information

Selecting **Edit Comment** will allow the addition of a comment for an item in the Project Shortcut View. This comment can be viewed in the tooltip for that item.

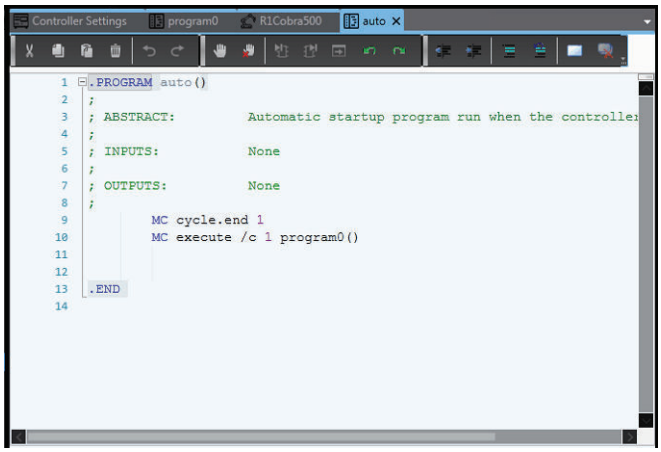


5-1-8 Status Bar

The status bar provides additional information about the project state and the user access level.

5-1-9 Edit Pane

The Edit Pane is used to display and edit the data for any of the items. Double-click an item in the Multiview Explorer to display details of the selected item in the Edit Pane.
The Edit Pane is displayed as tab pages. Switch the tab page to display in the Edit Pane by clicking the corresponding tab. Left-click and hold the Edit Pane to undock it from the main window.



Edit Pane Details

Change the number of tab pages in the Edit Pane that can be displayed at a time with the Option setting. The default setting is 10 and the maximum display number is 20. Refer to *5-14-3 Window* on page 5-58 for details on the settings.
Close the Edit Pane by right-clicking a tab and selecting one of the options from the pop-up menu. The following options are available.

Menu Command	Description
Close	Select this command to close the currently selected edit window.
Close All But This	Select this command to close all tab pages and floating edit windows, except for the selected tab page.
Close All Except Active Device	Select this command to close all tab pages and floating edit windows for other devices, leaving those for the active device open.
Close All	Select this command to close all tab pages and floating edit windows, including the selected tab page.

Tab Groups

Use tab groups to organize multiple tab pages in the Edit Pane .

Create a tab group by right-clicking any of the tabs and selecting an option from the pop-up menu. More than one tab must be open in the Edit Pane to use the following functions.

Menu Command	Description
New Horizontal Tab Group	Select this command to divide the Edit Pane into upper and lower sections and move the selected tab page to the tab group that is newly created.
New Vertical Tab Group	Select this command to divide the Edit Pane into left and right sections and move the selected tab page to the newly created tab group.
Move to Next Tab Group	Select this command to move the selected tab page to the tab group below or to the right of the current tab group.
Move to Previous Tab Group	Select this command to move the selected tab page to the tab group above or to the left of the current tab group.

Additional Information

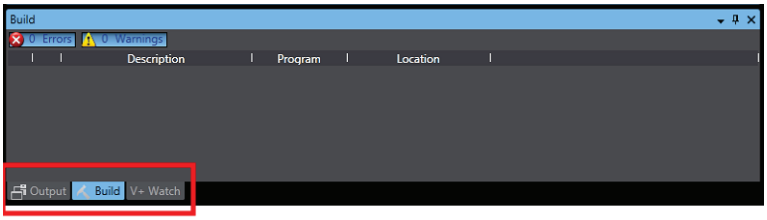
Tab pages are moved between tab groups by left-clicking the page and dragging it to the new group.

5-1-10 Multifunction Tabs

The Multifunction Tabs area contains several objects that are described in this section.

- Output Tab Page (reserved for future use)
- Cross Reference Tab Page (reserved for future use)
- Build Tab Page (refer to *5-6 Build Tab Page* on page 5-33 for more information)
- Event Log (refer to *5-7 Event Log* on page 5-34 for more information)
- V+ Watch (refer to *5-11 V+ Watch* on page 5-52 for more information)

Multiple objects can be open in the Multifunction Tabs area. Use the tab selections to switch between active objects.



5-1-11 Special Tools

Multiple objects can be open in the Special Tools area.

- Toolbox (refer to *5-2 Toolbox* on page 5-23 for more information)
- 3D Visualizer (refer to *5-3 3D Visualizer* on page 5-24 for more information)
- V+ Jog Control (refer to *5-8 V+ Jog Control* on page 5-35 for more information)
- Task Status Control (refer to *5-9 Task Status Control* on page 5-40 for more information)
- Vision Window (refer to *5-10 Vision Window* on page 5-50 for more information)

Use the tab selections to switch between active objects. Dock objects in other areas of the software by dragging and dropping them to different locations.

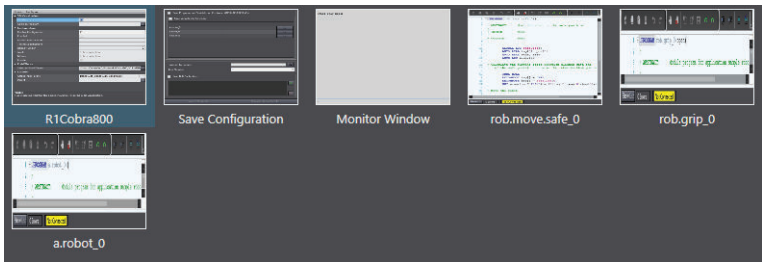


5-1-12 Other Application Window Functions

Other application window functions are described below. These can be accessed in the View menu item.

Recently Closed Windows

Previously displayed in the Edit Pane can be displayed again through a thumbnail index by selecting **Recently Closed Windows** from the View Menu. The windows previously displayed in the Edit Pane are displayed in the thumbnail index of recently closed windows. Double-click the window to display.



Up to 20 thumbnails of the most recently closed windows are displayed in the order they were closed, the upper left is the most recent. Open multiple windows by clicking the individual windows while holding down the **Shift** or **Ctrl** Key. This is also done by pressing the **Arrow** Keys while holding down the **Shift** or **Ctrl** Key.

Additional Information

The history of recently closed windows is retained for each device.

Clear Recently Closed Windows History

Delete the history of recently closed windows from a project by selecting **Clear Recently Closed Windows History** from the View Menu, and then clicking the **Yes** Button in the confirmation dialog box that is displayed. All of the histories are deleted.

Reset Window Layout

Select **Reset Window Layout** from the View Menu to restore the window layout to the ACE software default layout.

System Monitor

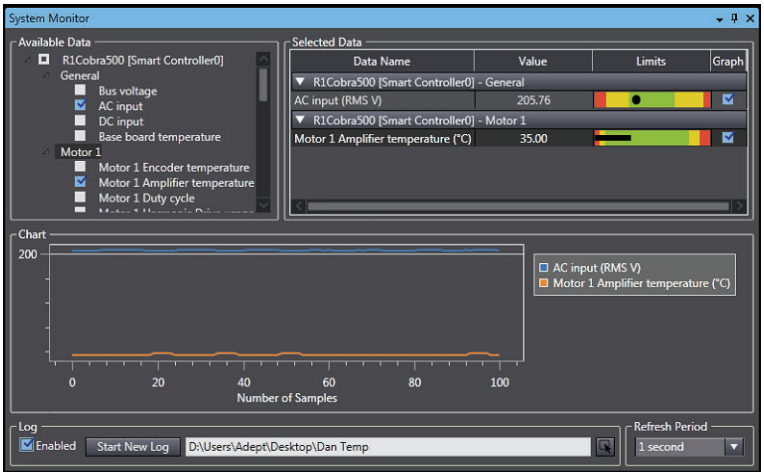
The System Monitor can be used to perform real-time monitoring of robot hardware performance and Process Manager objects (when present in the Application Manager device). The data can be used to identify inefficient part allocation, for example.

It is important to recognize that Process Manager statistics reflect the defined processes and represent exactly what is happening in the Process Manager. Although it may appear that the statistics are not accurate, the data needs to be interpreted appropriately for each system.

Consider a system containing one belt camera and three robots. If a part identified by the camera is allocated to robot one, not picked, then allocated to robot two, not picked, then allocated to robot three and picked, there would be two parts not processed because the instance was allocated and not picked by the first two robots. To understand the parts not processed count for this system, examine the Parts Not Processed for Robot 3 (the most downstream robot).

If a system requires customized statistics processing, this can be achieved using a C# program and the API provided in the StatisticsBin method. Refer to the ACE API Reference Manual (Help Menu) for more information.

Connection to a physical controller is required when using the System Monitor (Emulation Mode is not supported).



Use the following tables to understand System Monitor data.

Parameter Name	Description
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)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.
Peak torque (% max torque)	The peak torque, as a percentage based on maximum torque, for the selected motor. If this is frequently exceeded, consider reducing acceleration, deceleration, or speed parameters or changing s-curve profile to reduce peak torque required for the motion. This should operate below the specified max 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).

Process Manager Belt Parameter Descriptions

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.

Parameter	Description
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 displays 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.

Process Manager Robot and Process Parameter Descriptions

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 is actively processing part or part target instances.)
Average Total Time (ms)	For the Process Manager group only. The *average* total time for all robots. Other fields, such as Parts Processed/Not Processed and Targets Processed/Not Processed, show a summation for all robots.
Parts Per Minute	The average number of parts per minute. When viewing the Process Manager group, this is a summation for all robots.
Targets Per Minute	The number of targets per minute. When viewing the Process Manager group, this is a summation 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 summation 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 summation 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 summation 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 summation for all robots.

Process Manager Process Strategy Parameter Descriptions

Parameter	Description
Average Allocation Time (ms)	The average time it takes to run the allocation algorithm for allocating all parts and part targets.

● Robot Parameter Descriptions

Robot parameter descriptions are provided below.

- 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.
- Peak torque (% max torque)
  The peak torque, as a percentage based on maximum torque, for the selected motor. If this is frequently exceeded, consider reducing acceleration, deceleration, or speed parameters or changing s-curve profile to reduce peak torque required for the motion. This should operate below the specified max 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).

● Process Manager Belt Parameter Descriptions

Process Manager Belt parameter descriptions are provided below.

- 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 displays 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.

● Process Manager Robot and Process Parameter Descriptions

Process Manager Robot and Process parameter descriptions are provided below

- 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 is actively processing part or part target instances.)
- Average Total Time (ms)
  For the Process Manager group only. The average total time for all robots. Other fields, such as Parts Processed/Not Processed and Targets Processed/Not Processed, show a summation for all robots.
- Parts Per Minute
  The average number of parts per minute. When viewing the Process Manager group, this is a summation for all robots.
- Targets Per Minute
  The number of targets per minute. When viewing the Process Manager group, this is a summation 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 summation 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 summation 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 summation 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 summation for all robots.

● Process Manager Process Strategy Parameter Description

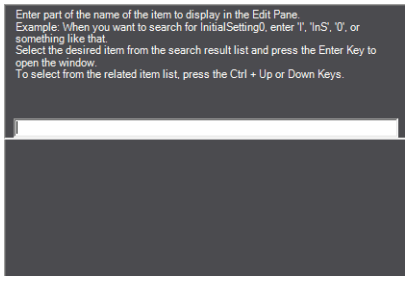
The Process Manager Process Strategy parameter description is provided below.

- Average Allocation Time (ms)
  The average time it takes to run the allocation algorithm for allocating all parts and part targets.

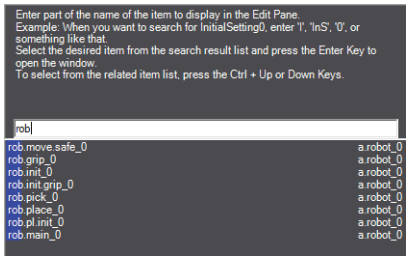
Smart Project Search

Use the Smart Project Search feature to quickly find items in the Multiview Explorer. For example, if there are a large number of programs or sections present in the project, locate the desired program or section with the Smart Project Search. The search is performed only within the project active device. Use the following procedure for the Smart Project Search function.

1 Select **Smart Project Search** from the View Menu to display the Smart Project Search Dialog Box.



2 Enter part of the name of the item in the Search Box The items in the Multiview Explorer or menus that contain the entered text string are displayed on the left side of the search result list. The search results are displayed from the top in the order they are displayed in the Multiview Explorer. On the right side of the search result list, the level that is one higher than the item that resulted from the search is displayed.



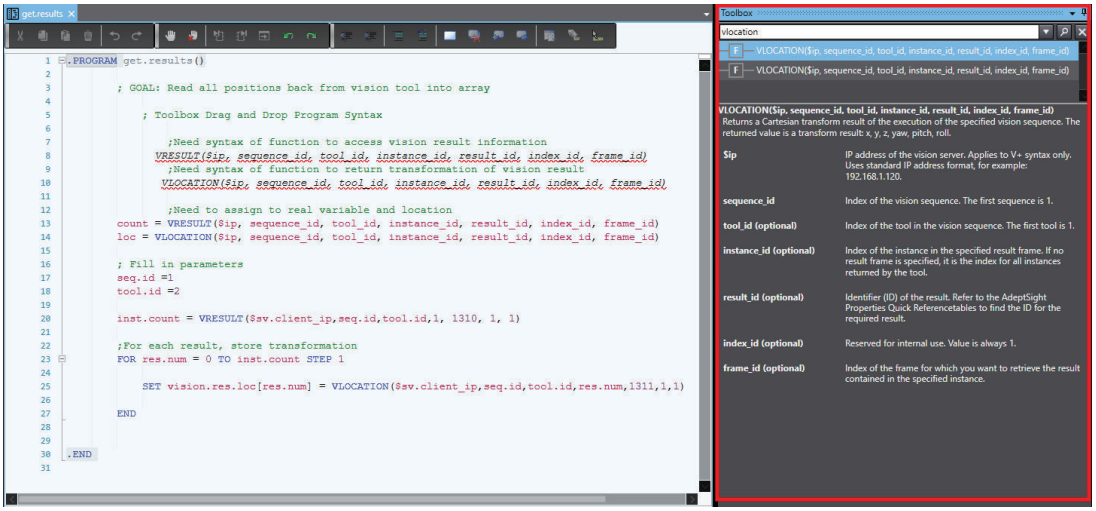
3 Double-click the desired item in the search result list or press the **Enter** key after making a selection. The Search Dialog Box is closed and the selected item is displayed in the Edit Pane. To close the Search Dialog Box without selecting an item, click in a pane other than the Search Dialog Box or press the **Esc** key.

Additional Information

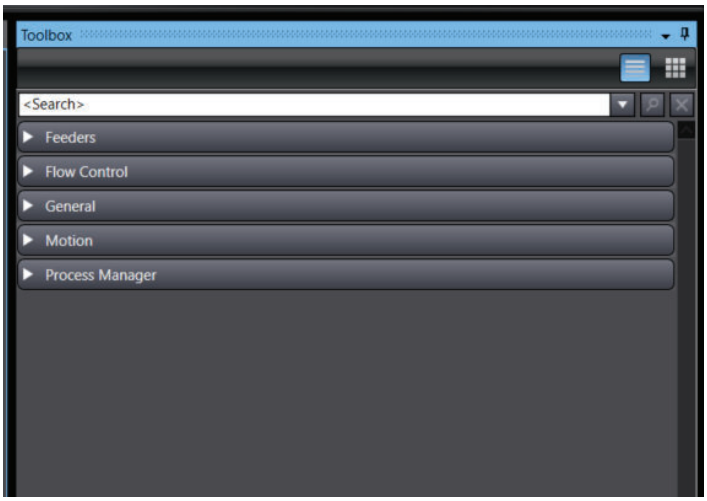
Enter English characters into the Search Box to search for item names that contain words starting with the entered characters in capitals. Example:Entering *is0* in the Search Box, displays an item named InitialSetting0 in the search result list.

5-2 Toolbox

The Toolbox pane is a dynamic area that provides selectable items that can be used in the current editor pane. For example, when a V+ program is open in the Edit pane, the Toolbox will display a list of available V+ Keywords. Drag from the Toolbox and Drop in the editor to implement specific Toolbox items.



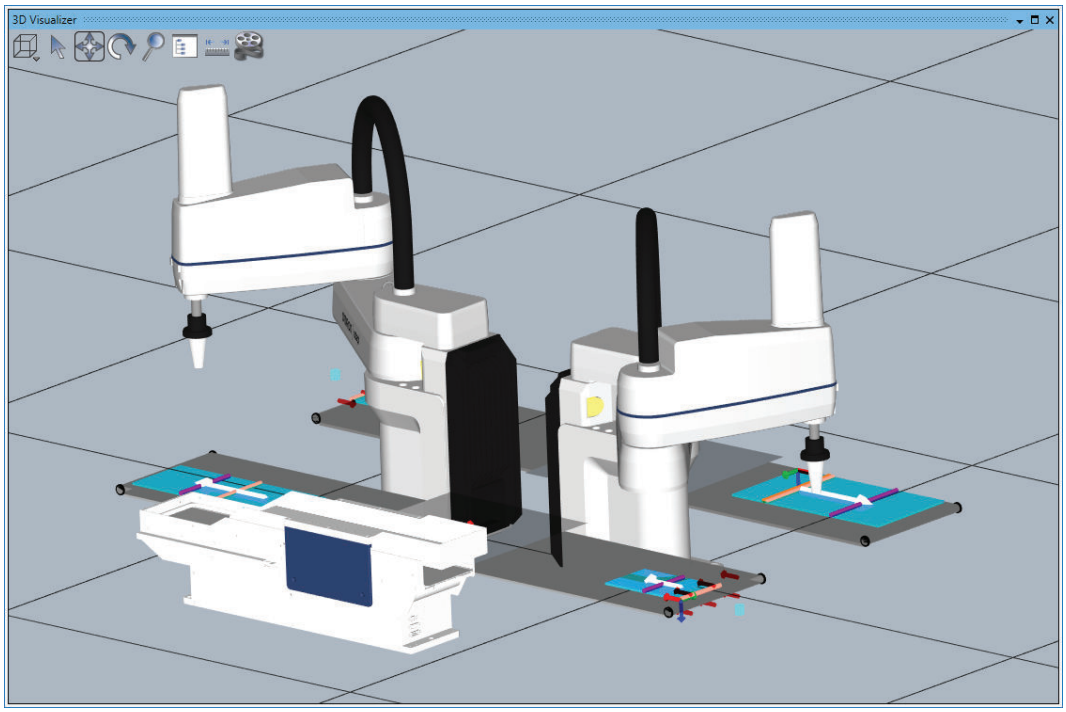
The Toolbox for an Application Manager Operation shows the categories of Skills that can be expanded to drag and drop skills into an operation.



5-3 3D Visualizer

The 3D Visualizer shows both simulated and real 3D motion for robots and other Process Manager items such as belts and parts. The 3D Visualizer window displays the following information.

- Graphical representation of all belts, cameras, robots, fixed objects, and obstacles in the workspace
- Robot belt windows and allocation limits
- Robot work envelopes
- Teach points and other part locations
- Belt lane widths
- Location of latch sensors
- Field of view for cameras
- Process Manager objects and Part / Part Target instances



Additional Information

Refer to *7-5-3 3D Visualization* on page 7-46 for more information about Application Manager objects.

5-3-1 3D Visualizer Basic Features

The 3D Visualizer has several functions to enable easy development and testing. The 3D Visualizer tool is available in both Emulation Mode and also while connected to physical controllers. The 3D Visualizer has the following features to help develop and test robot applications.

- 2D and 3D recording for playback in the ACE Offline Visualizer
- Collision detection in Emulation Mode
- Measurements
- Robot jogging
- Location visualization and editing
- Integration of work cell objects such as belts, parts, end effectors, and cameras

5-3-2 Creating a 3D Workspace

To use the 3D Visualizer, an accurate 3D workspace must be create to represent all objects of concern in the application. Use the following procedure to create a 3D workspace.

1. Configure robots by selecting robot types and offset positions. Refer to *7-5 Robot Objects* on page 7-40 for more information.
2. Add static objects such as boxes, cylinders, and imported CAD shapes if necessary.
3. Add cameras if necessary.
4. Add feeders if necessary.
5. Add Process Manager items if necessary (part buffer, part target, part, belt, pallet, .etc).

Additional Information

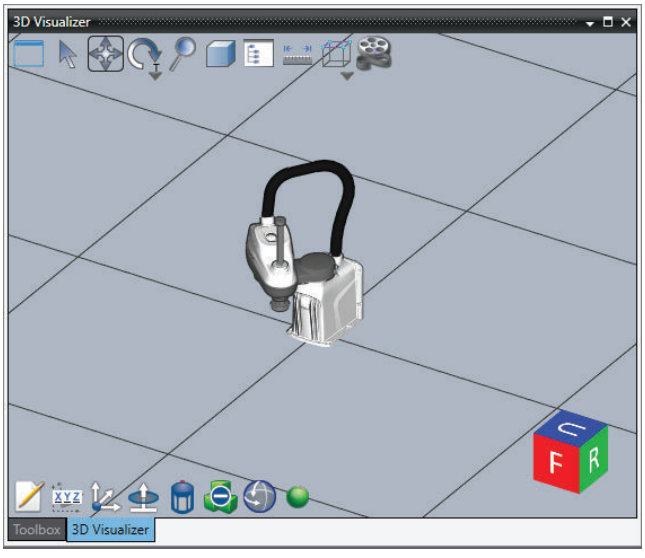
Static objects such as boxes, cylinders, and imported CAD shapes are stored in the ACE project, not the controller. Refer to *8-2 3D Visualization* on page 8-10 for more information about steps 2 through 5 above.

5-3-3 3D Visualizer Window

Access the 3D Visualizer Window with the main toolbar icon ( ) or by selecting **3D Visualizer** from the View Menu.
The 3D Visualizer Window has the following control icons.

Additional Information

The use of the term camera in this section refers to the perspective at which the 3D Visualizer is viewed and not a camera configured in the Application Manager (unless specified).



The following table provides information for the icons in the 3D Visualizer.

Additional Information

Many of the functions in the following table can be accessed by right-clicking an object in the 3D Visualizer view.

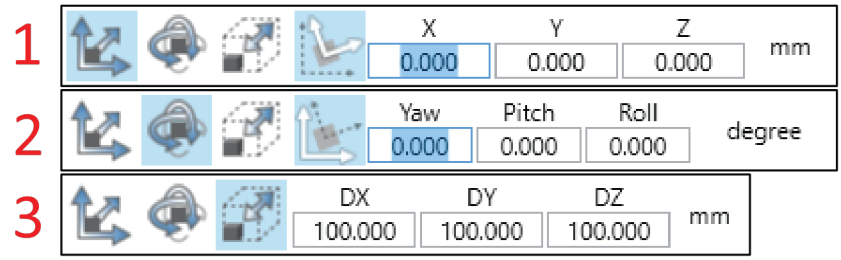
Icon	Description
	Displays the Projection Mode for the 3D cube. Left-click and drag the icon, moving the projection with the changes. Use this feature to quickly change user perspective of the visualization area to a different orientation. F (front): the side facing the user. B (back): the back face. R (right): the right face. L (left): the left face. U (up): the upper face. D (down): the side opposite to the upper face.
	Edit Opens the selected object's settings in the Edit Pane.
	Edition Control Manager Opens the Edition Control Manager. Refer to *Edition Control Manager* on page 5-29 for more information .
	Edit Workspace Position Displays a coordinate icon for relocating the X, Y, and Z location of the selected object. Hover over an axis of the coordinate icon to see the cursor change and then left-click and drag the object to a new position. Hovering the cursor in the white circle portion of the coordinate icon will allow free movement of the object when clicked and dragged. The new position will be reflected in the Offset from the Parent value of that object
	Edit Workspace Orientation Displays a coordinate icon for manipulating the orientation of an object. Hover over any of the axis rings to see the cursor change and then click and drag to rotate the object to a new orientation. Hovering the cursor in the white circle portion of the coordinate icon will allow free movement of the object when clicked and dragged. The new position will be reflected in the Offset from Parent value of that object. Note that a different number of axis rings will be available depending on the selected object.
	Jog mode Click to toggle the jog mode between world, tool, and joint. Jog icons will appear. Use the jog icons to manually control the selected robot's position. This is available for robot objects only.
	Show Obstacles Toggles visibility of obstacles that are present. Refer to *7-5 Robot Objects* on page 7-40 for more information. This is only available for robot objects.
	Show Robot Work Envelope Toggles visibility of the selected robot work envelope. This is only available for robot objects. The displayed work envelope applies to the robot tool flange and does not account for any applied tool offsets.
	Teach Point Adds a new location variable at the robot's current position. This is only available for robot objects.
	Show/Hide Mount Points Toggles visibility of mount points on the selected object. This is only available for Box, Cylinder, and CAD Data objects.
	Delete Delete a selected position from the variables list. This is only available when a point is selected.

Icon	Description
	Jog To Jog to the selected position. This is only available when a point is selected.
	Approach Height Toggle the approach height for the Jog To command between 0, 10, and 25 mm for the selected position. This is only available when a point is selected.
	Jog Speed Toggle the jog speed between 100%, 50%, 25%, and 5% for the selected position. This is only available when a point is selected.
	Clear All Instances Clear all instances being tracked by the selected Process Manager. This is only available when a point is selected.
	Show Camera Image Display camera images for a selected camera configured in the Application Manager. This is only available when a point is selected. This is typically used during setup and troubleshooting and keeping this enabled during run time is not recommended.
	Split Window: Click this icon to open a new dialog window that allows splitting the visualizer window into multiple views. This allows the user to view the workspace from multiple positions at the same time.
	Selection: Select an object in the view for editing.
	Translate (pan): Move the camera position without rotation. Click with the third mouse button and drag as an alternative. Use the **Alt** key + the left mouse button as an alternative.
	Rotate: the camera position without translation. Click the right mouse button and drag as an alternative. Click the arrow beneath this icon to choose between Tumbler and Turntable rotation.
	Zoom Move: the camera position closer or farther from the workspace. Use the mouse scroll wheel as an alternative.
	Scene Graph: Opens the Scene Graph window. The Visibility tab allows the user to set the usability of each object shown in the Visualizer. The Collision Filter tab configures collision sources. Refer to *5-3-4 Graphical Collision Detection* on page 5-29 for more information.
	Measurement Ruler: Add a ruler to the 3D Visualizer to make a measurement. Once enabled, the ruler will appear and can be selected to change the position on the visualizer for individual measurements. Selecting the ruler will display the Select Point 1 () and Select Point 2 () icons, which can be used to adjust the ruler position. Total X, Y, and Z distances will be displayed. It is often easiest to position the ruler using camera angles that are aligned with global coordinate planes.
	Clicking the down arrow in the icon opens an option pane with four choices. Use this to snap the view to a specific location. Refer to *Connecting Links and Mounts* on page 8-16 for more information.

Icon	Description
	Record: Begin a recording for playback in ACE Offline Visualizer. The icon will change while recording (flashing ). Click the flashing icon to stop recording. A Save As window will appear for saving the recording to the PC as a .awp3d file. Opening this file will launch an Offline Visualizer window for playback. This is only available on PCs with ACE software installed.
	Split Window Click this icon to open a new dialog window that allows splitting the visualizer window into multiple views. This allows the user to view the workspace from multiple positions at the same time.
	Selection Select an object in the view for editing.
	Translate (pan) Move the camera position without rotation. Click with the third mouse button and drag as an alternative. Use the **Alt** key + the left mouse button as an alternative.
	Rotate the camera position without translation. Click the right mouse button and drag as an alternative. Click the arrow beneath this icon to choose between Tumbler and Turntable rotation.
	Zoom Move: the camera position closer or farther from the workspace. Use the mouse scroll wheel as an alternative.
	Scene Graph Opens the Scene Graph window. The Visibility tab allows the user to set the usability of each object shown in the Visualizer. The Collision Filter tab configures collision sources. Refer to *5-3-4 Graphical Collision Detection* on page 5-29 for more information.
	Measurement Ruler Add a ruler to the 3D Visualizer to make a measurement. Once enabled, the ruler will appear and can be selected to change the position on the visualizer for individual measurements. Selecting the ruler will display the Select Point 1 () and Select Point 2 () icons, which can be used to adjust the ruler position. Total X, Y, and Z distances will be displayed. It is often easiest to position the ruler using camera angles that are aligned with global coordinate planes.
	Clicking the down arrow in the icon opens an option pane with 2 choices. Using this the view will snap to the Face by clicking the () icon .The view will snap to the Edge by clicking the () icon.
	Record Begin a recording for playback in ACE Offline Visualizer. The icon will change while recording (flashing ). Click the flashing icon to stop recording. A Save As window will appear for saving the recording to the PC as a .awp3d file. Opening this file will launch an Offline Visualizer window for playback. This is only available on PCs with ACE software installed.

Edition Control Manager

Double click the Edition Control Manager and use it to set certain editing parameters for the 3D Visualizer Window. The parameters are set in either millimeters or degrees depending upon the coordinate system.



The icons in the manager are explained in the following table.

Icon	Description
	Edit Workspace Position Changes the fields to X, Y, and Z, as shown by item (1) of the above figure. Changing the values of these fields will translate the object in the Visualizer.
	Edit Workspace Orientation Changes the fields to Yaw, Pitch, and Roll, as shown by item (2) of the above figure. Changing the values of the fields will rotate the object in the Visualizer.
	Edit Size Changes the fields to DX, DY, and DZ for a selected Box object or changes them to Radius and Height for a selected Cylinder object, as shown by item (3) of the above figure. Changing these fields will change the object size. This is only available for Box and Cylinder objects.
	Object Coordinates/World Coordinates Toggles between the Object Coordinates and World Coordinates modes. When Object Coordinates is selected, the icon will appear as it does to the left, and any adjustments will be with respect to the objects parent. When World Coordinates is selected, the icon will change () and any adjustments will ignore parent constraints. This icon is hidden when Edit Size is selected.

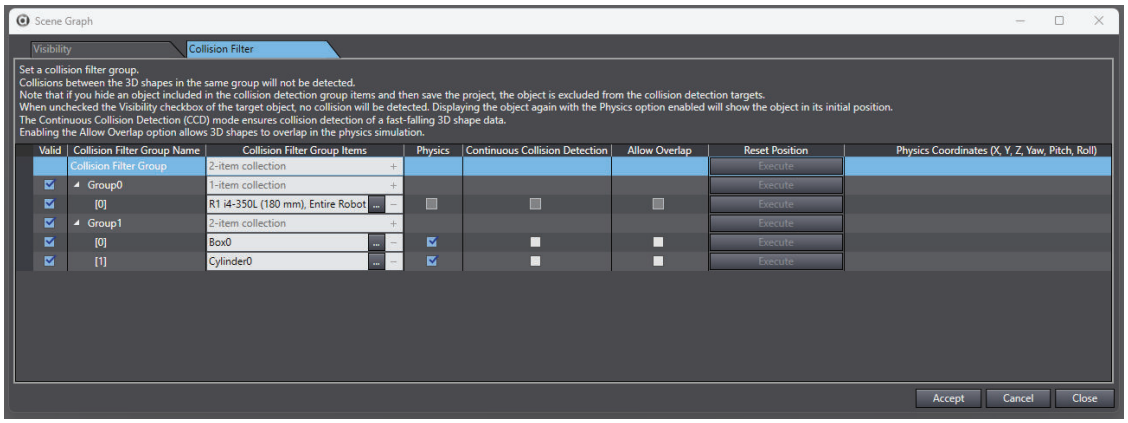
5-3-4 Graphical Collision Detection

The 3D Visualizer can be configured to detect collisions between objects. When a collision is detected in the 3D Visualizer, any C# Collision program(s) that are associated with the objects in the collision are called.

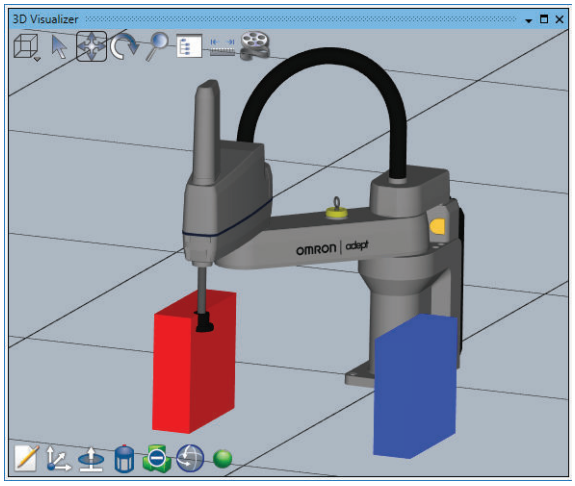
This should not be confused with obstacles that are configured in robot objects. Refer to *Obstacles* on page 7-54 for more information.

Configuring Collisions in the 3D Visualizer

Use the Scene Graph icon () in the 3D Visualizer to configure collisions. This will open the Scene Graph dialog box. Refer to *5-3 3D Visualizer* on page 5-24 for more information.



Use the Collision Filter tab in the Scene Graph dialog box to add or remove object sources for collision definition(s). The following is a list of suggested object sources for collision: robot base, robot arm (overall), link 1, link 2, end effector, box, cylinder, user provided 3D CAD data, conveyor object, part feeder etc. Use the **Add** button ( ) to create a new collision definition between two objects (Source 1 and Source 2). Use the **Delete** button ( ) to remove a selected collision definition. Select or deselect the **Valid** check box to enable or disable a collision definition.
If a collision is detected between two objects, they will become shaded in the 3D Visualizer as shown below.



Additional Information

The Average Time value in the Collisions tab of the Scene Graph represents the processing time to detect a collision.

5-4 Output Tab Page

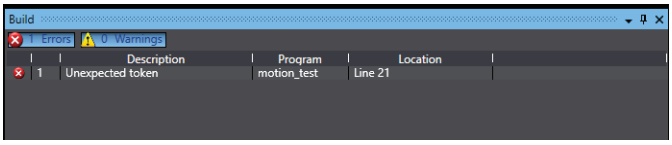
The Output Tab Page is reserved for future use.

5-5 Cross Reference Tab Page

The Cross Reference Tab Page is reserved for future use.

5-6 Build Tab Page

The Build Tab Page shows the output from the ACE V+ program parser to identify potential issues with program syntax or control structures.
Double-click items in the Build Tab Page to directly access the program location where the issue occurs.



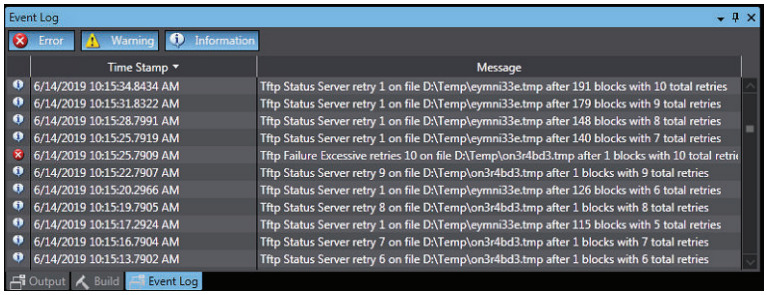
5-7 Event Log

The Event Log displays a log of controller events that have occurred since the ACE software was started. Events are categorized as Error, Warning, and Information and can be used for troubleshooting and diagnostics.

Event Log messages can be displayed by event type, time stamp, and message. Use the following functions to adjust the Event Log display.

Past information that is displayed in the logs may not reflect the current status of the controller or programs.

Refer to *Section 9 Troubleshooting* on page 9-1 for more information.



5-8 V+ Jog Control

V+ Jog Control provides an interface for positioning and monitoring the position of the selected robot. It is typically used for teaching robot locations.
Use the 3D Visualizer to see robot motion when the V+ Jog Control is used in Emulation Mode. The V + Jog Control functions and operation is described below.
The V+ Jog Control works for both emulated and physical robots.
Many jog commands and setting are disabled while a robot is under program control. Refer to *5-8-2 Current Position Section* on page 5-36 for more information.

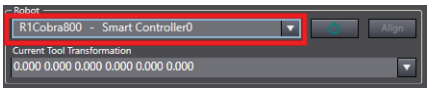


5-8-1 Robot Section

The robot section provides the following functionality.

Robot

Select a robot in the ACE project to control with the V+ Jog Control. All robots in the project will be accessible in the drop-down selection area.

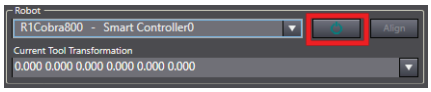


Robot Power

The *Power* button toggles the robot high power ON and OFF and calibrates the selected robot. Robot power must be ON to allow jog control.

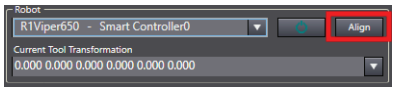
Additional Information

Turning the robot high power ON for the first time after system power up executes the CALIBRATE() keyword to load joint calibration offsets into memory. This does not perform a full robot hardware calibration.



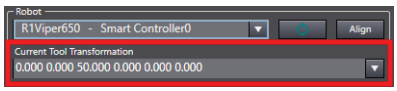
Align

The **Align** button aligns the robot tool Z-axis with the nearest World axis (six-axis robots only).



Current Tool Transformation

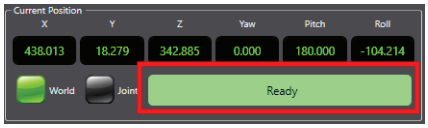
This displays the current tool transformation applied to the robot. The dropdown can be used to clear the tool transformation or choose a tool transformation provided by an I/O EndEffector tip.



5-8-2 Current Position Section

This section displays the status and current position of the robot in world or joint coordinates. Click the **World** Button to display coordinates in world mode. Click the **Joint** Button to display coordinates in joint mode.

Jogging is only possible when *Ready* is displayed in the status area.



If a robot is under program control, *Robot under program control* will be displayed in this area and jogging is not possible. The task controlling the robot must be stopped before jogging functions are enabled.

5-8-3 Jog Control Section

The jog control section is used to manually position the robot.

Move Axis Buttons

After all jog control settings are made, use the move axis buttons to move the selected axis in the positive () or negative () direction.



Speed, Increment Selection Buttons

Jogging is possible at a preset speed or in incremental distances, to allow greater positioning precision.

When the *Speed* button is active, use the slider or input a value between 0 and 100% to set the jog speed when a move axis button is pressed.

When the *Increment* button is active, use the slider or input a value between 0 and 10 mm to set the movement distance when a move axis button is pressed.



5-8-4 World, Joint, Tool Selection

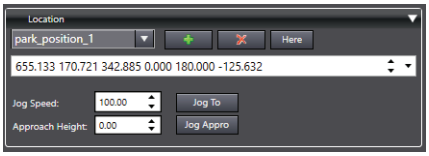
Choose world, joint or tool for the jog control mode.



- 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 the RX, RY, or theta axes in the world coordinate system.
- Joint - Enables the jog control to move the selected robot joint.
- Tool - Enables the jog control to move the robot in the selected direction: X, Y, or Z axes of the tool frame of reference or rotated around the RX, RY, or theta axes in the tool coordinate system.

5-8-5 Location Section

The Location Section is used to view, teach, remove, and jog to robot locations. Refer to *6-3 V+ Variables* on page 6-11 for more information.



Jog to a Robot Location

Use the following procedure to jog to a robot location.

A robot location must exist and be selected to use this function.

1 Select a location with the drop-down menu.

2 Make selections for the *Jog Speed* and *Approach Height* fields.

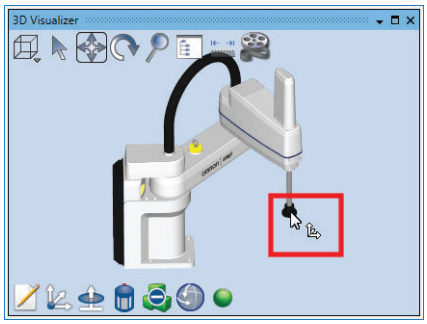
3 Click and hold the **Jog To** Button (Jog To) to make the robot jog to the specified location or click and hold the **Jog Appro** Button (Jog Appro) to make the robot jog to the specified location at the approach height specified.

Precautions for Correct Use

Using the **Jog Appro** Button will cause straight-line motion to occur. Monitor the robot during this movement to avoid collisions with obstacles between the starting location and the destination location.

Teach Robot Locations

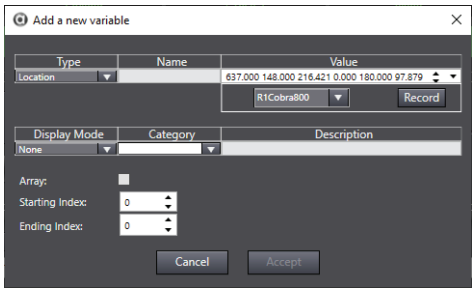
Before teaching a location, move the robot to the desired location (either by jogging or powering OFF and physically moving the robot) and then click the **Here** Button (Here). Clicking the **Here** Button will put the robot's current axis positions into the display field for use in the following teach procedure. While in Emulation Mode, the robot tool tip position in the 3D Visualizer is changed with the mouse cursor. Hover over the tool tip until the mouse pointer changes () and then left-click and drag to the new position.



Additional Information

Refer to *6-4 V+ Variable Editor* on page 6-13 for other robot position teach functions.

1 Click the **Plus** Button (). This opens the Add a new variable Dialog Box.



2 Select a variable type (location or precision point), provide a new name, and verify the value. If the robot dropdown selection is changed, click the **Record** Button (Record) to update the value for that robot accordingly.

3 Choose the display mode, category, and provide an optional description.

4 Make a selection for array, starting and ending index when applicable.

5 Click the **Accept** Button to create the new robot location variable.

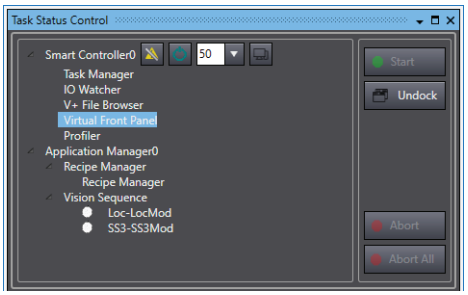
Remove Robot Locations

To remove an existing robot location, select the location from the dropdown menu and then click the **Delete** Button (). A confirmation dialog box will appear. Click **Yes** to remove the robot location variable.

5-9 Task Status Control

The Task Status Control provides a multi-device monitoring interface to view all robot related activity in the project, and allows access to all controller connection and power statuses, monitor speeds, and execution status of Vision Sequences, C# programs, Process Managers, and more.
All controllers in a project will be displayed in the Task Status Control interface. If an Application Manager Device is present in the project, items such as Robot Vision Manager Sequences, C# programs, Process Managers, and Recipe Managers may also be displayed in the Task Status Control interface.
Task Status Control functions are described below.

- Online / Offline
- Robot High Power Control
- Monitor Speed Setting
- Open the Monitor Window
- Task Manager
- I/O Watcher
- V+ File Browser
- Virtual Front Panel
- Fieldbus Status
- Profiler
- Application Manager Control (when applicable)



5-9-1 Online/Offline

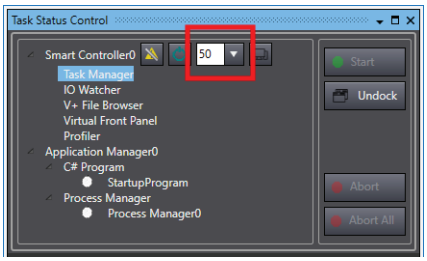
Use the **Online/Offline** Buttons (/) to control the connection status to a controller.

5-9-2 Robot Power Control

Use the **ON/OFF** Button (/) to control the robot power state. This button is only available while online with the controller. The **Robot Power** Button in the Virtual Front Panel have the same functionality.
Enabling robot power is not possible in an E-Stop state.

5-9-3 Monitor Speed Setting

The monitor speed setting is used to adjust the monitor speed for the associated controller. This is a multi-robot speed scaling parameter for each controller that is used to decrease the overall speed of the system without modifying programs. This is typically used while debugging programs.

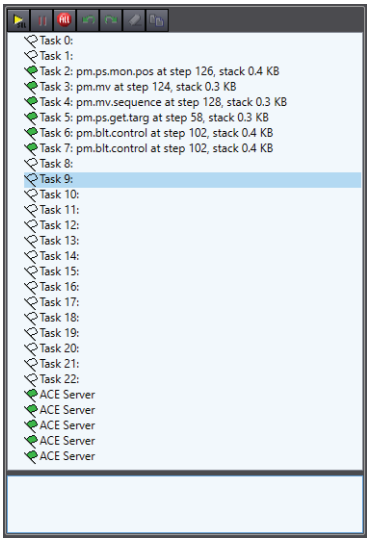


5-9-4 Open the Monitor Window

Use the **Monitor Window** Button () to open the Monitor Window in the Edit Pane. Refer to *7-4 Monitor Window* on page 7-37 for more information.

5-9-5 Task Manager

The Task Manager displays and controls activity on user tasks 0 to 27. The ACE software uses two tasks plus one task per robot, counting down from 27. The remaining tasks (0 to 21, or more if fewer than four robots) are available for the execution of user-created V+ programs. This includes programs started by a Process Manager as shown below.
If a program is paused the task can be expanded to view the current program stack.
Use the following descriptions to understand Task Manager controls.



Task Manager Toolbar Items

Item	Icon	Description
Execute Task		If a stopped program is selected, this button will execute that program on the task.
		If a task is selected with no program, this button will open a dialog box for program selection. Selecting a program name and clicking **OK** will execute the program on the selected task.
Pause Task		The selected task execution is paused at the next command.

Item	Icon	Description
Stop All Tasks		Stops the execution of all running tasks.
Retry Step		If the selected task was paused or stopped due to an error, this button attempts to re-execute the current step and continue execution.
Proceed Task		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.
Kill Task		Clears the selected task of any programs. AUTO variables or calling arguments cannot be changed while a program is in a task stack.
Copy Stack to Windows Clipboard		Copies the contents of the selected task stack to the Windows clipboard. If a program terminates with an error, this is used to copy and paste the stack contents for troubleshooting. The robot ID is also recorded in this operation.

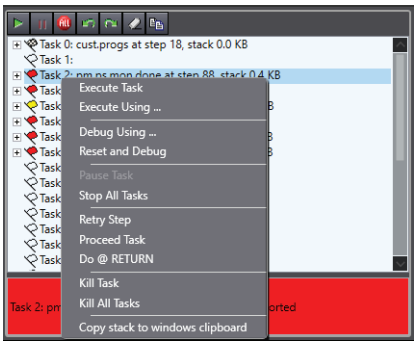
Task Manager List Area

The flag icon next to each task in the list area represents the task state. Use the following table to understand different task states.

Task Flag Icon	Description
	Task is idle or primed.
	Task is executing.
	Task is paused or at a breakpoint. Dragging program's task flag icon onto a task to prime it change the icon to yellow.
	Task has an execution error or program execution was manually aborted.
	Task execution has completed.

Other Functions

Right-clicking a task in the task list will open a menu with other functions not described above. Use the following descriptions to understand the other functions.

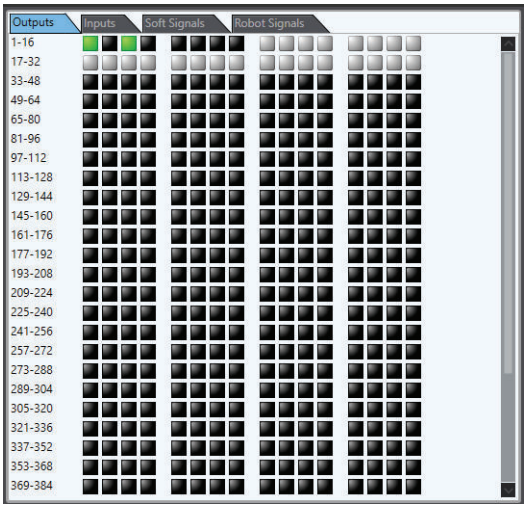


Other Function	Description
Execute Using...	Prompts for the name of the program to execute on the selected task.
Debug Using...	Prompts for the name of a program to debug, primes the specified program, and opens the V + program in the Edit Pane.
Reset and Debug	Resets the program and open the V+ program in the Edit Pane for the selected task.
Do @ RETURN	Execute the task when the *Return Key* is pressed.
Kill All Tasks	Clears all tasks that do not have running programs.

5-9-6 I/O Watcher

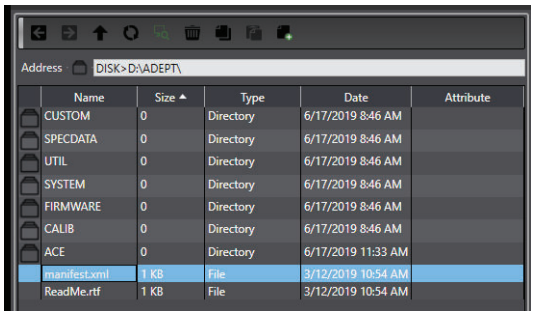
Select I/O Watcher to display an interface for monitoring the state of digital I/O signals (inputs, outputs, soft signals, and robot signals) on the connected controller. Digital output signals and soft signals can be turned ON and OFF manually by clicking on the **Signal** Button (■/■).

When Emulation Mode is enabled, digital input signals can be manipulated.



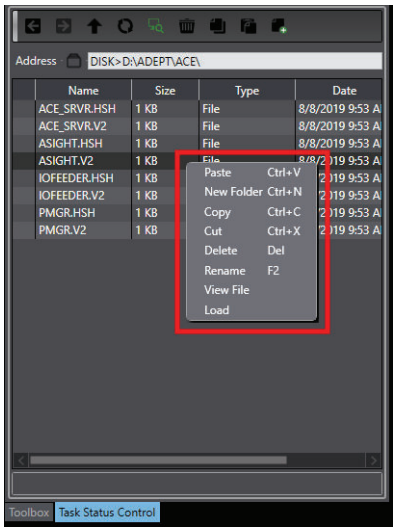
5-9-7 V+ File Browser

The V+ File Browser open a files and folders view on the controller. This is only possible while online with the controller.



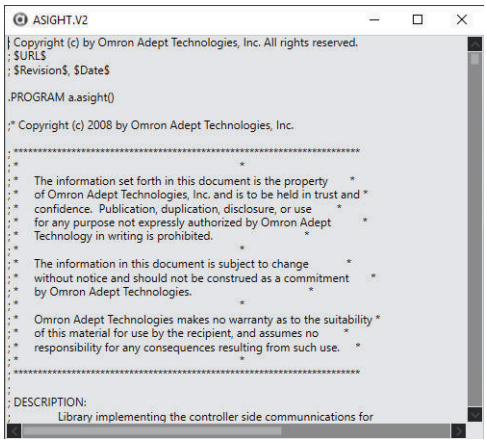
The V+ File Browser works with the Windows clipboard, enabling easy transfer of files to and from controllers.

Use the icons in the V+ File Browser toolbar to perform common file browser functions such as navigation, creating new folders, rename, delete, cut, copy, and paste. Right-clicking a file or folder will also display a menu with common file browser functions and other items described below.



View File

Select **View File** to open the file in a quick-view window without the need for transferring the file to a PC. This is available for program, variable, and text files.



Load

Select **Load** to transfer the contents of the selected file from disk to system memory.

5-9-8 Virtual Front Panel

The Virtual Front Panel provides a simulated front panel for testing robot behavior when mode selection, robot power, and E-stop conditions are changed.
In Emulation Mode, full use of the Virtual Front Panel is possible.
When connected to a physical controller, Virtual Front Panel functions are read-only.
Virtual Front Panel functions are described below.



Mode Selection

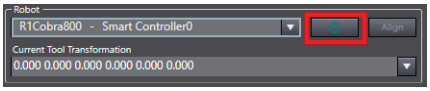
Switches between Manual ( ) and Automatic ( ) mode. In Automatic mode, executing programs control the robot, and the robot can run at full speed. In Manual mode, the system limits robot speed and torque so that an operator can safely work in the cell. Manual mode initiates software restrictions on robot speed, commanding no more than 250 mm/sec. There is no high speed mode in manual mode. Refer to the robot's user's manual for more information.

Robot Power

The **Power** button toggles the robot high power ON and OFF and calibrates the selected robot. Robot power must be ON to allow jog control.

Additional Information

Turning the robot high power ON for the first time after system power up executes the CALIBRATE() keyword to load joint calibration offsets into memory. This does not perform a full robot hardware calibration.



E-Stop

E-Stop behavior can be tested and monitored with the **E-Stop** button on the Virtual Front Panel. Use the ESTOP Channel area to simulate various E-Stop system functions.



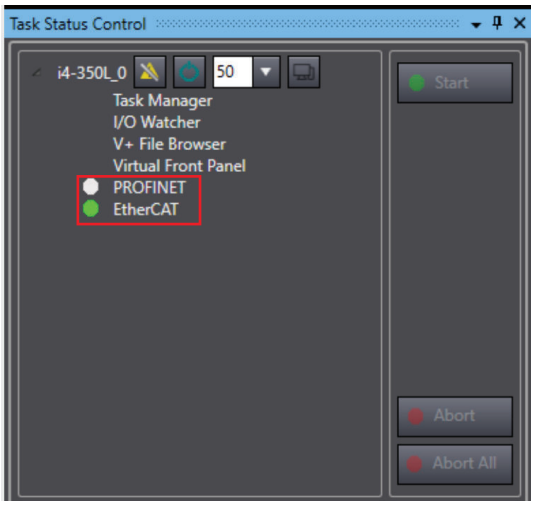
Refer to the robot user's manual for more information about E-Stop functions.

5-9-9 Fieldbus Status

Fieldbus Status provides an indication of the current fieldbus state.
This information is available when an online connection is present.

Additional Information

This information can also be acquired using the FB.STATE keyword. Refer to *V+ Keyword Reference Manual (Cat. No. I672)* for more information.



The displayed color provides the following fieldbus status information.

Color	Fieldbus Status	
	PROFINET	EtherCAT
White	Not enabled.	
Red	Enabled but inactive.	

Color	Fieldbus Status	
	PROFINET	EtherCAT
Yellow	Enabled and connected, but there is no data exchange. Refer to the *Industrial Robot Fieldbus Configuration User's Guide (Cat. No. M130)* for more information.	Enabled and connected, but with limited data exchange. The EtherCAT SubDevice (robot) is in PreOp or SafeOp state. Refer to the *Industrial Robot Fieldbus Configuration User's Guide (Cat. No. M130)* for more information.
Green	Enabled, connected, and exchanging data.	

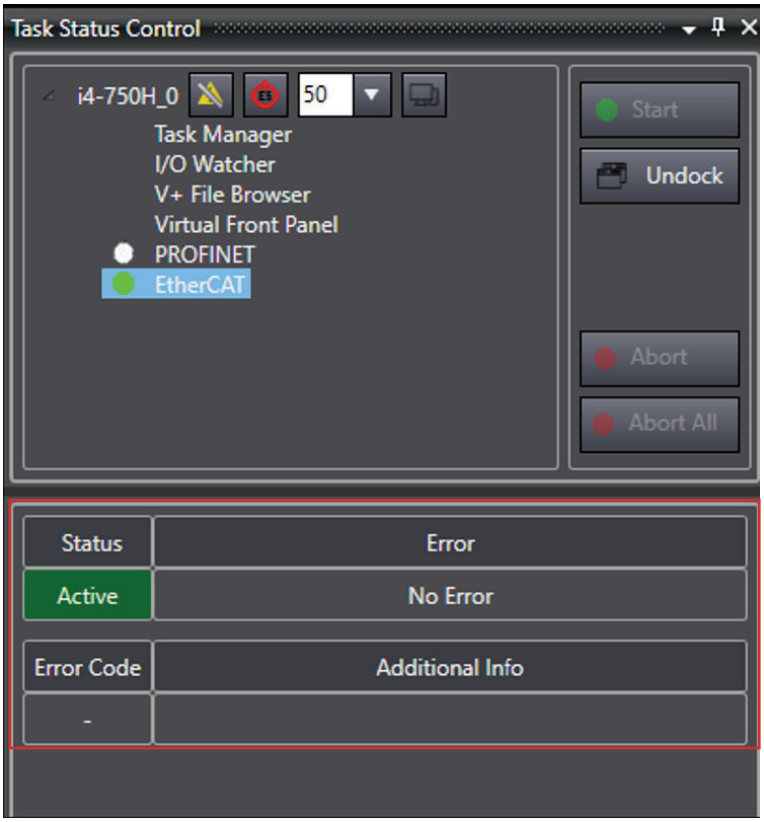
Precautions for Correct Use

Disconnecting EtherCAT communications while the robot is transitioning between control states may cause an invalid fieldbus state. Cycle power to the robot to recover from this condition.

Additional Information

PROFINET and EtherCAT fieldbus states are disabled (white) by default.

Single-click the fieldbus label or status indicator color in the *Task Status Control* window to access the fieldbus diagnostic table. This table will not appear if the fieldbus is disabled or when offline.



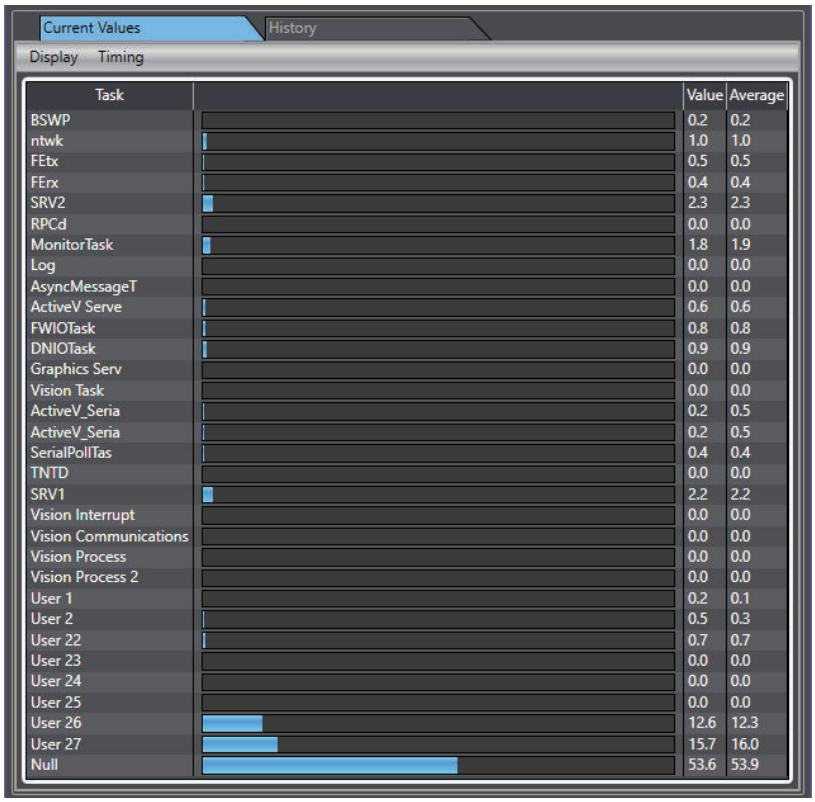
Double-click the fieldbus Status label or status indicator color to access the fieldbus configuration settings area. Refer to *Configure PROFINET Settings* on page 7-26 or *Configure EtherCAT Settings* on page 7-27 for more information.

5-9-10 Profiler

The Profiler is available for each controller in the project. It is used to provide a graphical display of controller processor usage for diagnostic purposes. There are two tabs in the Profiler view as described below.
The Profiler function is not available in Emulation Mode.

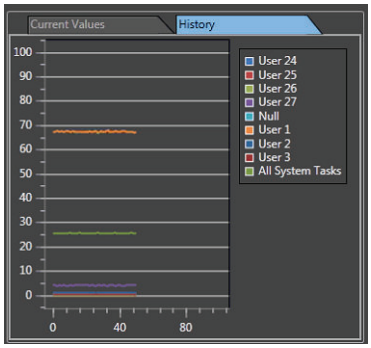
Current Values Tab

The Current Values tab shows a list of tasks and their respective processor usage. Use the Display and Timing menu items to adjust the listed items and the update rate.
Select **All User Tasks** to display all the user tasks available to the connected system. If **All User Tasks** is not selected, only tasks with a program on the execution stack are displayed.



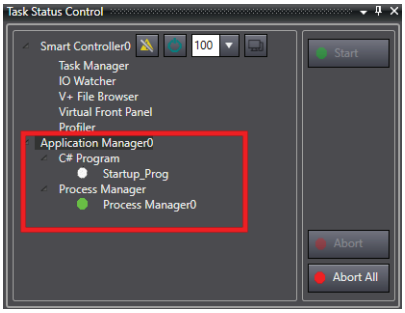
History Tab

Viewing the History tab displays a line plot history of CPU load over time for each task.



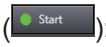
5-9-11 Application Manager Control

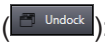
Use the Application Manager Control area to examine information about, and control activity associated with the Application Manager. Application Manager objects such as Robot Vision Manager sequences, C# programs, Process Managers, and Recipe Managers that require user interaction for execution or run time monitoring are displayed in this area.
Selecting a task will provide additional information in the areas below. More information about the Task Status Control functions for Application Manager items can be found in the associated sections of this document.

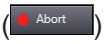


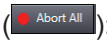
5-9-12 Control Buttons

Selecting a particular Application Manager object 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.

- **Start** Button (

): Executes the selected object.
- **Undock** Button (

): Undocks the selected Task Status Control item.
- **Abort** Button (

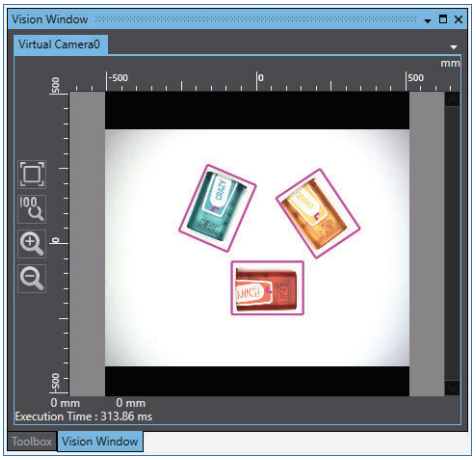
): Aborts the selected, running object.
- **Abort All** Button (

): Aborts all objects.

5-10 Vision Window

The Vision Window displays the input for each image source defined in the system. Click a tab for each source, to view various cameras and tools configured in the project. Most vision tools can display their results in this window, which enables troubleshooting, testing, or monitoring the system performance during run time. In addition, like the Edit Pane in the ACE main application window, the tabs in the Vision Window can be rearranged for better observation. Each window also includes rulers on the side and top of the image to show the scale of the items in the image. Refer to *8-8 Vision Tools* on page 8-190 for more information.

Use the following information to understand the Vision Window functions.



5-10-1 Zoom Level

Use the icons on the left of the window to adjust the zoom level. These icons are described below.

Fit to Screen

Click the fit to screen icon ( ) to change the zoom level so the entire image fits in the Vision Window.

Zoom to 100%

Click the zoom to 100% icon ( ) to change the zoom level to the default size of the acquired image.

Zoom in or out

Click the zoom in ( ) and zoom out ( ) icons to manually adjust the zoom level.

Calibration Scale

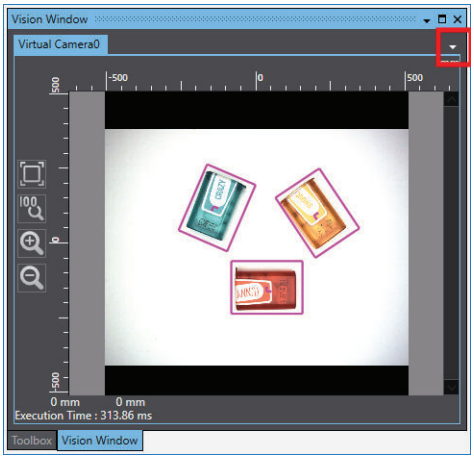
The left and top axis values represent the calibration scale setting in mm/pixel that is present in the Virtual Camera settings. Refer to *8-4-1 Virtual Cameras* on page 8-46 for more information.

Execution Time

The Vision Sequence execution time is displayed in the lower left area of the Vision Window.

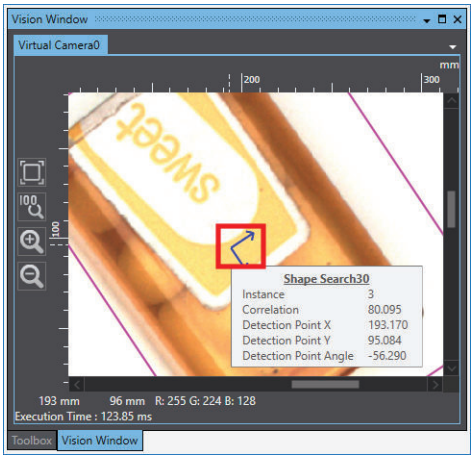
Camera Selection

Use the dropdown arrow to select from all available Image Sources defined in the system.



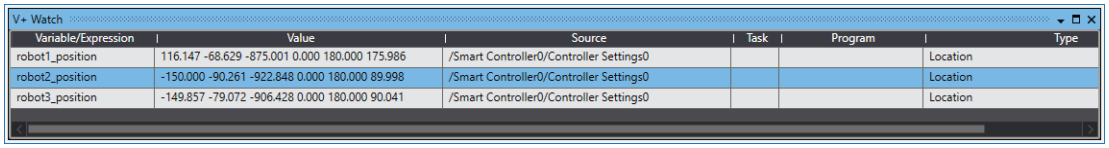
Cursor Information

Moving the cursor in the field of view portion of the Vision Window will reveal additional information about the inspection results.
The X-Y coordinates are displayed at the bottom of the Vision Window for the current cursor position.
Color / gray scale values are also displayed when applicable.
Hover over the coordinate icon in the field of view to display inspection results as shown below.



5-11 V+ Watch

V+ Watch is used to monitor specified variables while developing or debugging V+ programs. Variables can be added to the V+ Watch window in several different ways as described below. The V + Watch window contents will be saved with the project. Refer to *6-3 V+ Variables* on page 6-11 for more information.



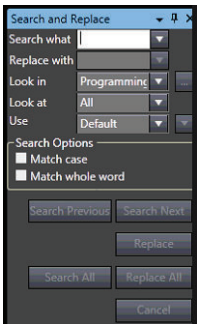
5-11-1 Adding Variables to the V+ Watch Window

Use one of the following methods to add variables to the V+ Watch window.

- Select one or more variables in the Variable Editor and right-click. Selecting **Add to Watch** will place these variables in the V+ Watch window.
- Right-click in the V+ Watch window area and select **New**. Type the name of the variable to add it to the V+ Watch window.
- Right-click the variable name in the V+ Editor. Select **Add to Watch** to place this variable in the V + Watch window.

5-12 Search and Replace

Use the Search and Replace tool to search for and replace text strings in modules and programs within the project. Access the Search and Replace tool in the Edit Menu list or with the **CTRL**+**F** Keys.
If more than one device is registered in the project, the target of a search and replace is the currently active device only. Confirm the active device is in the device list for the project before performing a search and replace.



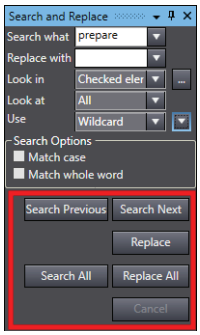
Item	Description	Details
Search what	Use this field to enter a string for a search.	Use the drop down arrow to select from previous search strings.
Replace with	Enter a string to replace the search string with.	Use the drop down arrow to select from previous replace strings.
Look in	Specify a range to search by selecting from the following: • Programming: all of the programs of the project's currently active device. • Checked elements: the items selected in the Select search and replace scope Dialog Box are searched. • Current view: the active V+ Editor tab is searched.	When selecting **Checked elements**, click the **More Options** Button (...) to display the Select search and replace scope Dialog Box.
Look at	All: Searches all text strings.	
Use	Specify the characters used. • Default: Do not use wildcard characters. • Wildcard: Use wildcard characters.	Use wildcard characters by clicking the (▼) button to the right to view a list of characters used for wildcard characters. Select any of these characters to enter them in the search string.

5-12-1 Search Options

Item	Description
Match case	When this option is selected, searches are case sensitive.
Match whole word	When selected, only exact string matches are returned.

5-12-2 Button Functions

Use the following table to understand the functions of the buttons in the Search and Replace tool.



Item	Description
Search Next	Performs a search according to the selected options.
Search Previous	Performs a search in the reverse order of **Search Next**.
Search All	Searches all items and displays the results in the Search and Replace Results Tab Page displayed in the Multifunction Tabs area.
Replace	Performs a replace according to the selected options.
Replace All	Replaces all items and displays the results in the Search and Replace Results Tab Page.
Cancel	Cancels the current search and replace operation.

5-13 User Management

The User Manager provides an interface to add, remove, and edit existing users. All settings in the User Manager are stored with the ACE project and are accessed with the **Edit Users...** selection in the **File** menu item.

User access levels allow software feature and function access control for designated users and assign a specific access level to each one within an ACE project. Password protection for each user level is also an option.

One of the following three access levels can be assigned to a user.

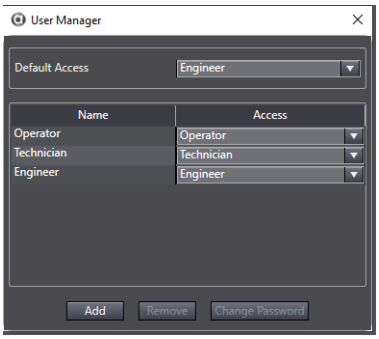
- Engineer (highest level): full access to all items
- Technician: limited access to some items
- Operator (lowest level): read-only access

Precautions for Correct Use

- A user signed in with Technician or Operator access cannot edit users or access the User Manager. Only a user with Engineer access can edit users with the User Manager.
- The User Manager implements a basic level of user-access security. For applications that require a higher level of security, implement a security scheme within a custom user-interface. This could be based on the network login credentials from Windows or other similar access-control methods.

Use the following details to understand the User Manager interface shown below.

Passwords are not mandatory for new and existing users. Refer to *5-13-1 Passwords* on page 5-56 for more information.



Item	Description
Default Access	Designate the default access level when opening a project or signing out.
Name	This is the User Name when signing in. The Name field can be edited for each user. Default names of Operator, Technician, and Engineer are provided for all new ACE projects.
Access	Designate the access level for each Name in the User Manager list (Operator, Technician, or Engineer).
Add	Click the **Add** Button to add a new name with a specific access level to the user list. A new user will be added and sign in will be possible with the User Name and Password (if specified).
Remove	Click the **Remove** Button to remove a selected name from the User Manager list. The user will be deleted and sign in will not be possible with the previously stored credentials.

Item	Description
Change Password	Click the **Change Password** Button to create a unique password for each user listed. Clicking this button will open the Change Password Dialog Box. Refer to *5-13-1 Passwords* on page 5-56 for more information.
	Designate the default access level when opening a project or signing out.

5-13-1 Passwords

Passwords for each User Name can be specified, but are not mandatory. Passwords can include symbols, letters, and numbers and are case sensitive.

If a password has not been specified for a user, the following password omissions are present.

- The Old Password field can be left blank when changing a password.
- The Password field can be left blank when signing in as this user.

5-13-2 Engineer Accessible Items

All features and functions in the ACE software are accessible to a user with the Engineer access level. A user signed in with Technician or Operator access or cannot edit users or access the User Manager. Only a user with Engineer access can edit users with the User Manager.

5-13-3 Technician Accessible Items

The following features and functions are accessible to a user with the Technician access level. The Technician access level cannot add any new items to the ACE project and can only view and edit the following items if they already exist in the workspace.

- Save Controller Configuration
- AnyFeeder Objects
- Application Manager Settings
- Pack Manager Belt Objects
- Robot Vision Manager Objects
- 3D / CAD Data Objects
- Data Mapper Objects
- Emulation Camera Images
- I/O EndEffector
- I/O Feeder Objects
- OPC Container
- Process Manager
- Recipe Manager
- System Startup
- V+ and C# Variables
- Vision Tools

5-13-4 Operator Accessible Items

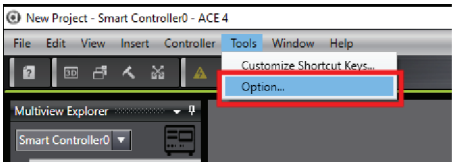
The notes function is editable for a user with the Operator access level only when a note was already created by another user with Technician or Engineer access. All other features and functions are either read-only or inaccessible.

5-14 Project Options

Project options can be access in the **Tools** menu item. The following project option settings are detailed below.

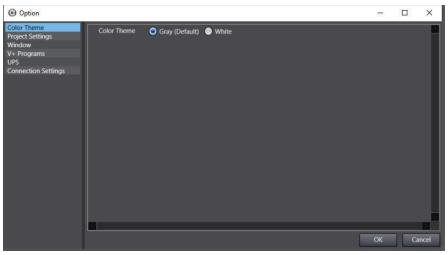
- Color Theme
- Project Settings
- Window
- V+ programs
- V+ Variables
- UPS
- Connection Settings
- 3D Visualizer

To access the project options, click the Tools menu bar item and then select **Option**... The Option window will open.

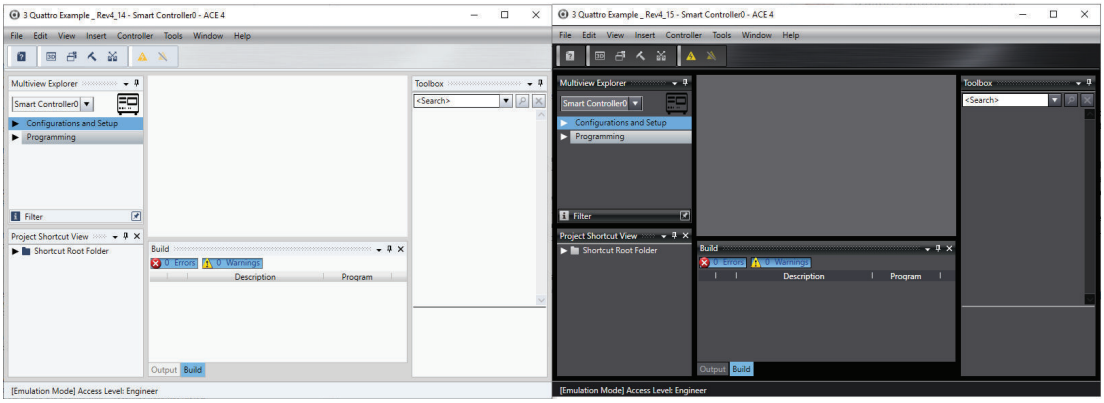


5-14-1 Color Theme

To access the color theme setting, select **Color Theme** from Option Dialog Box. The color theme settings will be displayed.

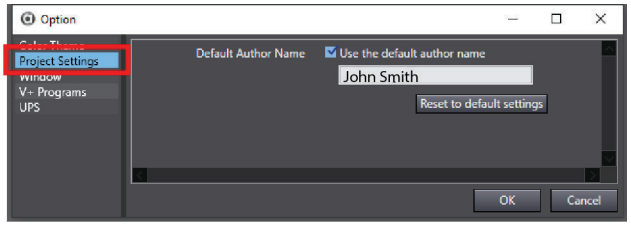


Choose Gray (default) or White as the color theme for the ACE software. Changing this setting will require a software restart to see the result. Both color themes are shown below.

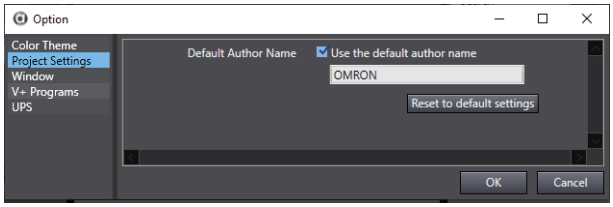


5-14-2 Project Settings

To access the project settings, select **Project Settings** from Option Dialog Box. The project settings will be displayed.



The default project author name can be specified in the Project Settings area. The following settings are available.



Use the Default Author Name

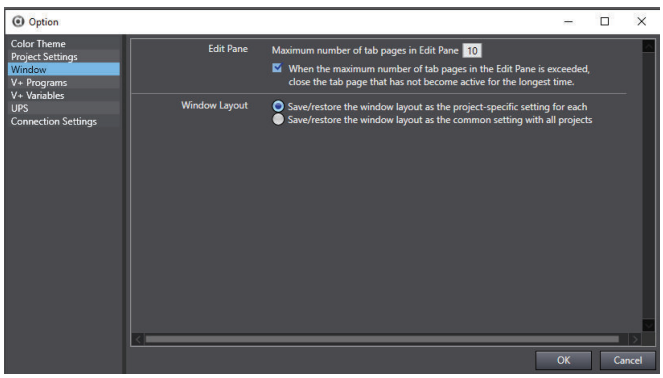
Selecting this will use the name specified when saving or creating a new ACE project. A restart of the ACE software is required to see this change.
If this is not selected, the author field for saving or creating a new ACE project will be blank.

Reset to Default Settings

Clicking the **Reset to default settings** Button will set the project's author name to the Windows user name. A restart of the ACE software is required to see this change.

5-14-3 Window

To access the Window settings, select **Window** from Option Dialog Box. The Window settings will be displayed.



Enter the maximum number of tab pages to view in the Edit Pane. This is used to set the number of tab pages in the Edit Pane and floating edit windows that can be displayed at a time.

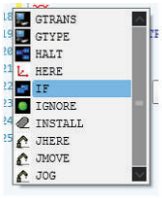
If the check box is selected for *When the maximum number of tab pages in the Edit Pane is exceeded, close the tab page that has not become active for the longest time* is active, the Edit Pane that was opened first will automatically close when attempting to open new tab that exceeds the set amount. If this is not selected, a warning is displayed and new Edit Pane tabs cannot be opened until old tabs are closed.

Header for New V+ Programs

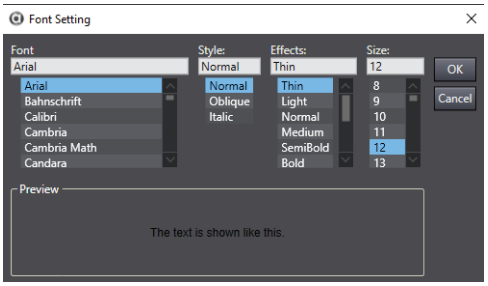
Text can be specified as the header for all new V+ programs in an ACE project when this is enabled. If no text is specified, a new V+ program will consist of only .PROGRAM and .END statements.

Other Parameters

If the *Allow automatic inteliprompt pop-ups in the V+ editor* option is selected, typing commands in the V+ editor will trigger suggestions with matching text, as shown below.

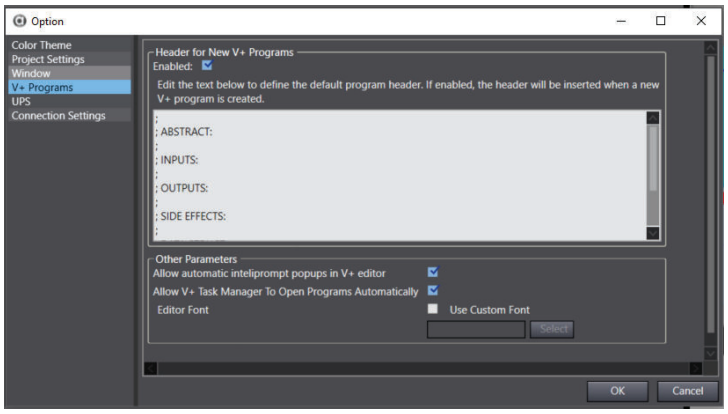


When the *Use Custom Font* option is selected, a system font for the V+ program editor is selected by clicking the **Select** Button. Then choose from the Font Setting Dialog Box and click **OK**. The new font setting will be used in the V+ program editor.

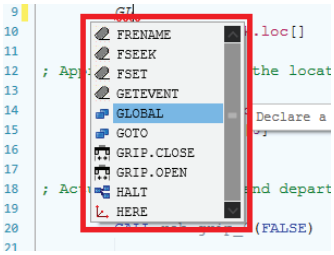


5-14-4 V+ Programs

To access the V+ programs settings, select **V+ Programs** from *Option* Dialog Box. The V+ program settings will be displayed.



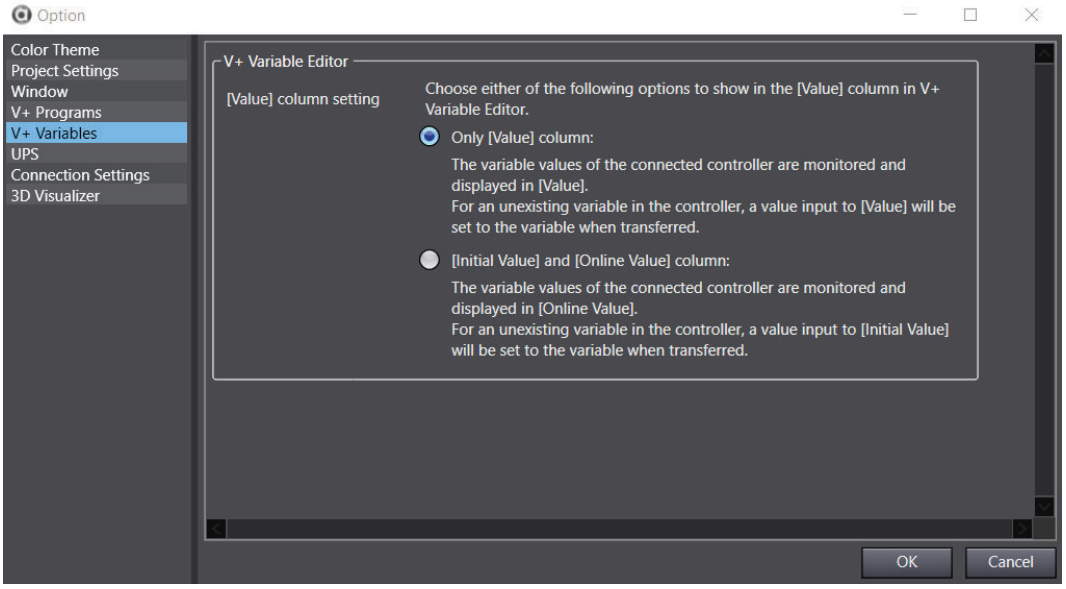
The following settings are available.

Setting	Description
Header for New V+ Programs	Enabling this will place text comments at the top of new V + programs by default. This text can be customized.
Allow automatic inteliprompt pop-ups in V+ editor	Enabling this will turn ON inteliprompts in the V+ editor. 
Allow V+ Task Manager to Open Programs Auto-matically	When this is enabled, the V+ editor will automatically open a program with an execution error.
Editor Font	Enable this to select a custom font for the V+ editor.

If there is a conflict between V+Programs settings and the controller memory, ACE will try and synchronize its memory by comparing V+ modules, V+ programs and V+ Variables with the controller memory.

5-14-5 V+ Variables

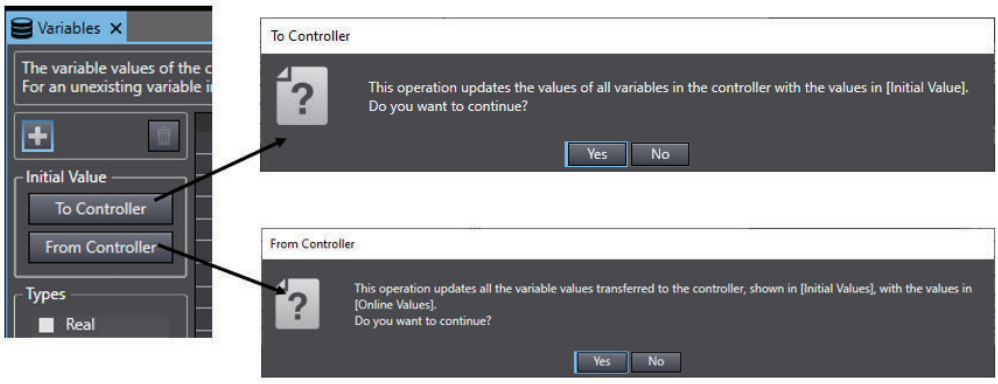
To access the V+ variables settings, select *V+ Variables* from the *Option* Dialog Box. The V+ Variables settings will be displayed.



The following settings are available.

Setting	Description
Only [Value] column	Enabling this will display only the *Value* column. For an unexisting variable in the controller, a value input to the *Value* column will be set to the variable when transferred.
[Initial Value] and [Online Value] column	Enabling this will display the *Initial Value* and the *Online Value* columns. The variable values of the connected controller are monitored and displayed in the*Online Value* column. For an unexisting variable in the controller, a value input to the *Initial Value* column will be set to the variable when transfered

When the *[Initial Value] and [Online Value] column* setting is enabled there will be two options for updating the controller variables.



These options are not available when the *Only [Value] column* setting is enabled.

5-14-6 Uninterruptible Power Supply (UPS)

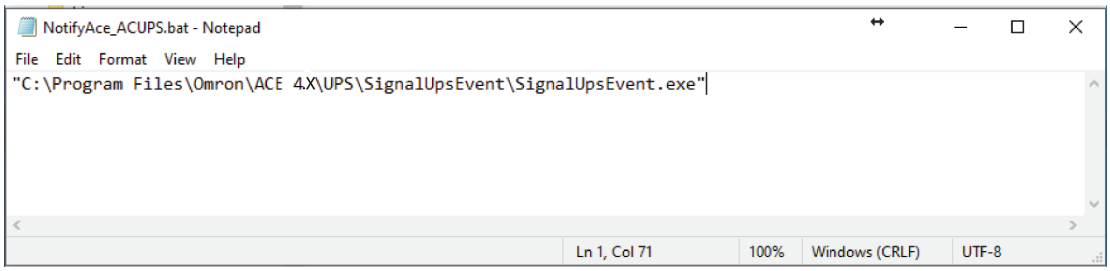
The ACE software can communicate with an Omron S8BA-Series UPS to provide a controlled shut-down during run-time if a power abnormality or other such event occurs (referred to as a UPS event). There is a choice of using either an AC-AC or DC-DC UPS, depending upon the facility requirements. Using the script command function described in this section, it is possible to perform the following functions in a systematic fashion when a UPS event occurs.

- Save the ACE project.
- Disable high power on all connected controllers.
- Close the ACE software.
- Shut down the operating system.

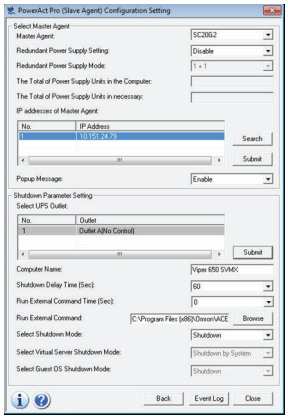
Refer to *A-2 Uninterruptible Power Supply (UPS) Installation* on page A-3 for UPS installation procedures.

Add ACE Batch File

ACE reacts to the UPS shutdown through a .bat file within the Slave Unit Client. Create the .bat and save it with a unique name in the directory, C:\Program Files\Omron\ACE 4.6\UPS. Copy and paste the following into the .bat file: "C:\Program Files\Omron\ACE 4.6\UPS\SignalUpsEvent\SignalUpsEvent.exe". Include the quote marks, " ",in the string.

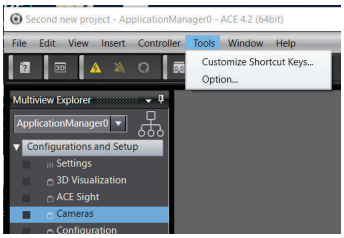


After the .bat file is created and saved, open the PowerAct Pro on the Industrial PC and go to the Configuration Setting, shown below. Refer to *AC-AC UPS Software Configuration* on page A-5 for more information using PowerAct Pro.



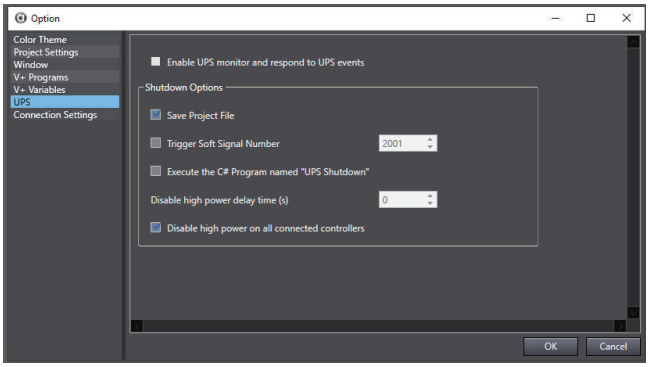
Enable ACE Software UPS Response

1 Open ACE and select **Tools** and then **Options** from the toolbar, to access the UPS settings in the ACE software.



2 When the window opens, select **UPS** from Option Dialog Box. The UPS settings will be displayed.

3 Select **Enable UPS monitor and respond to UPS events, Save Project File and Disable High Power on all connected controllers**. Click **OK**



Item	Description
Enable UPS Monitor and Respond to UPS Events	This option enables or disables the UPS functionality in the ACE software. The functions are described below.
Shutdown Options	• Save Project File Once the changes are made, select **Save Project File** from the menu. • Trigger Soft Signal Number ACE will trigger a soft signal on all connected controllers. This can be used with the REACTI keywords in a V+ program to create a controlled shutdown sequence for a power loss event. • Execute the C# Program names *UPS Shutdown* Triggers the C# program named *UPS Shutdown*. This enables a C# script to execute on a power event. The C# program must be named exactly as UPS Shutdown. • Disable High Power Delay Time (s) If selected, this specifies the amount of time in seconds that ACE will wait before saving and closing. Please verify that this time is long enough for a REACTI subroutine in V+ or the optional *UPS Shutdown* C# program to complete before ACE closes. • Disable High Power on All Controllers If this option is enabled, it disables High Power on all controllers connected to the instance of ACE. An alternative is to use a V+ program to bring the robot to a controlled stop in a safe location, before high power is disabled. This will stop the robots as quickly as possible immediately after the shutdown signal is received.

Add ACE Batch File Steps

1. Within the Configuration Setting, select **SC20G2** as the Master Unit.
2. Click the **Search** Button and select the IP of the UPS.
3. Click the upper **Submit** Button.
4. Select the outlet the robot is connected to.
5. Set the Shutdown Delay Time to 120(sec).
6. For Run External Command, click **Browse** and select the previously identified .bat file.
7. For Shutdown Mode, select **Shutdown.**

Script Command Function

The script command function is used to specify an executable when a UPS event occurs. The SignalUPSEvent.exe executable runs when a UPS event occurs and this will trigger the actions set in the ACE Software UPS Settings area described below. Refer to *DC-DC Set Up Power Attendant Lite* on page A-9 for more information.

The NotifyAce.usc script file must be placed in a specific directory of the PC running the ACE software. This file is included with an ACE software installation and can be found in the ACE software installation directory in the UPS folder. The default path to find this script file is provided below.

C:\Program Files\Omron\ACE 4.6\UPS

The NotifyAce.usc script must be copied to the following location on the PC running the ACE software.

C:\Users\Public\Public Documents\OMRON\PAL



Additional Information

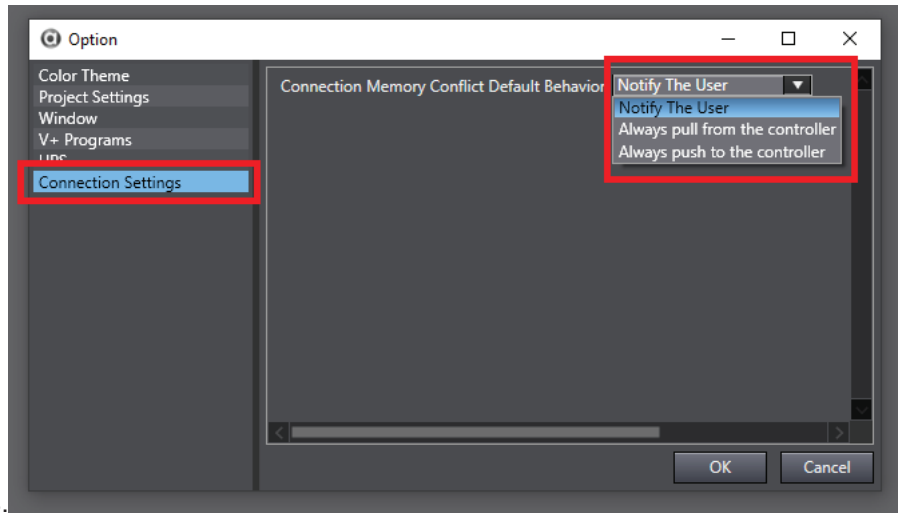
The PowerAttendant Lite software must be installed on the PC running the ACE software.

5-14-7 Connection Settings

The default memory comparison function can be set to one of the following options.

- Notify The User
- Always pull from the controller
- Always push to the controller

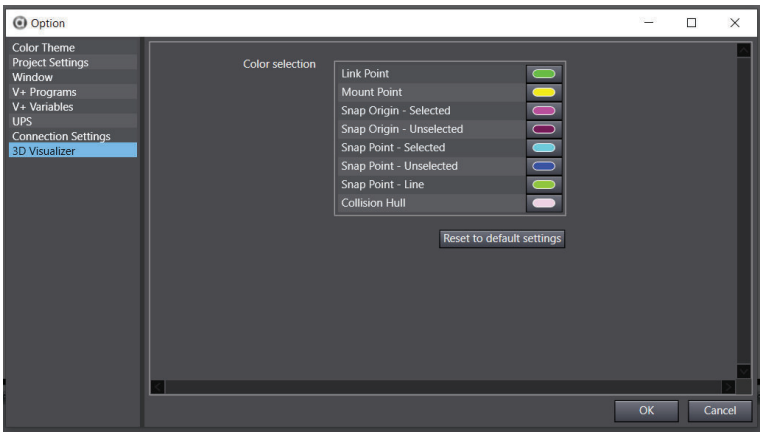
Use the ACE menu to select **Tools** > **Options**. When the Option panel opens, select **Connection Settings**.



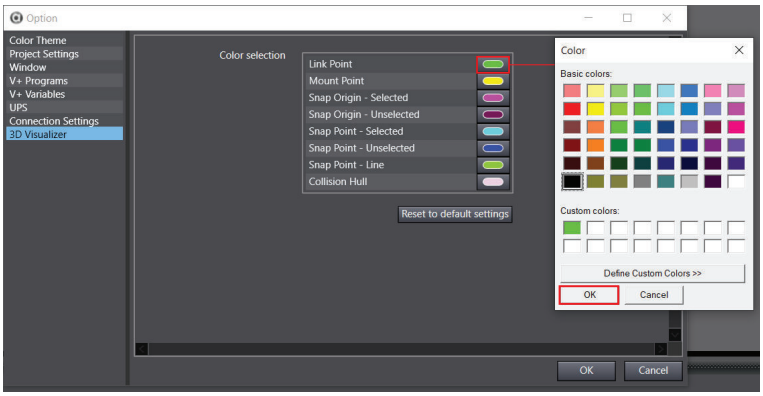
Use the drop-down options to select the default behavior and then click the **OK** Button.

5-14-8 3D Visualizer

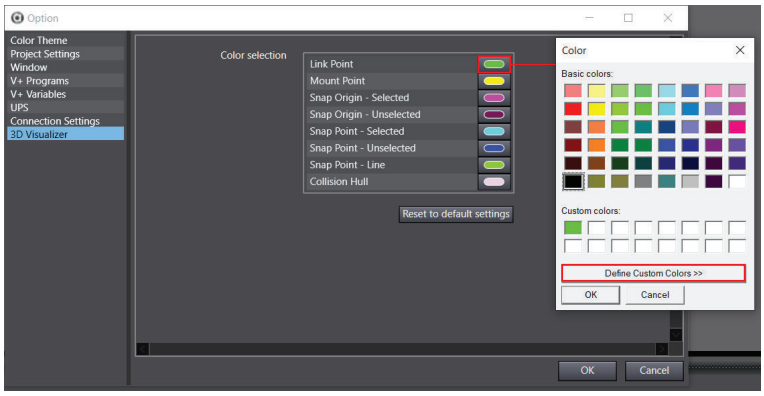
To access the 3D Visualizer color settings, select **3D Visualizer** from the *Option* Dialog Box. The connection point **Color selection** settings will be displayed.

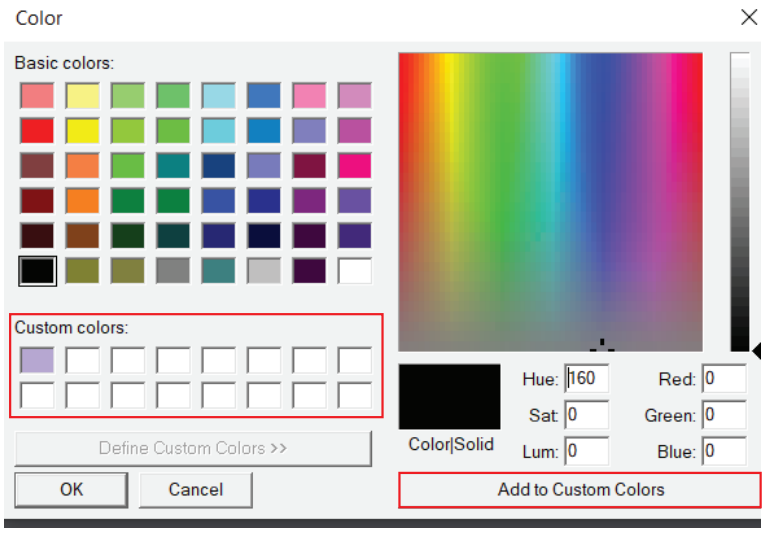


To change the color of a connection point select the button next to the connection point you want to change. Select a color and click *Ok* to apply the color to the connection point.

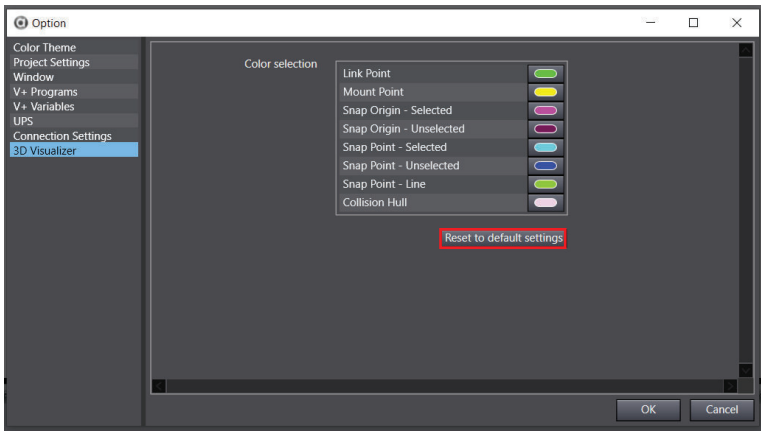


To define a custom color select **Define Custom Colors >>** and use the color wheel or enter numerical values to define a color. Click **Add to Custom Colors** to add your color to the **Custom colors** swatches panel.





To reset the connection point colors to the default settings, select **Reset to default settings**.



6

Programming with ACE Software

The information in this section describes programming with ACE Software.

6-1 Introduction 6-2

6-2 Robot Controller Programming 6-3

6-2-1 V+ Module and Program Structure 6-3
6-2-2 Creating a New V+ Module 6-4
6-2-3 Save a V+ Module to the PC 6-4
6-2-4 Show Global Variable References 6-4
6-2-5 Show Program Call Commands 6-5
6-2-6 Pull V+ Memory 6-5
6-2-7 V+ Editor 6-5
6-2-8 V+ Program Debugging 6-10

6-3 V+ Variables 6-11

6-3-1 V+ Variable Types 6-11

6-4 V+ Variable Editor 6-13

6-4-1 V+ Variable Names 6-13
6-4-2 V+ Variable Properties 6-13
6-4-3 Creating a New V+ Variable 6-15
6-4-4 Editing an Existing V+ Variable 6-16
6-4-5 Filtering by Types and Categories 6-16
6-4-6 Other V+ Variable Editor Functions 6-17

6-5 Application Manager Programming 6-19

6-5-1 Creating a New C# Program 6-19
6-5-2 C# Program Editor 6-20

6-6 C# Variable Objects 6-26

6-6-1 C# Variable Object Types 6-26
6-6-2 Creating a New C# Variable Object 6-26
6-6-3 Other C# Variable Object Function Keywords 6-26
6-6-4 Editing C# Variable Objects 6-26

6-7 Operation Programming 6-28

6-7-1 Automatically Create an Operation 6-29
6-7-2 Manually Create an Operation 6-30
6-7-3 Manually Create a Process Operation 6-32
6-7-4 Add Skills to an Operation 6-33
6-7-5 Skill Icons 6-34
6-7-6 Edit an Operation 6-35
6-7-7 Operation Skills 6-36
6-7-8 Building Custom Skills 6-38

6-1 Introduction

There are three methods of programming ACE applications depending upon the type of application being developed. The available methods are summarized below and described in more detail in this section.

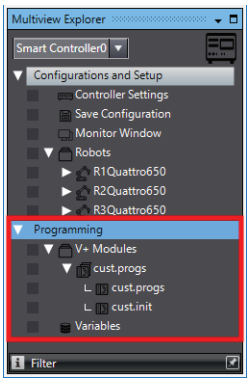
- Robot controller programming Overview: The V+ Editor described in this section is the main tool used to develop programs for a robot controller.
- Application Manager Programming: The C# Editor described in this section is the main tool used to develop C# programs for an Application Manager.
- Operation Programming: Create a flowchart with skills to develop programs for a robot controller.

6-2 Robot Controller Programming

Creating robot controller programs can be accomplished by working with V+ Modules, V+ programs, and variables. These items can be accessed in the Programming section of the Multiview Explorer for a robot controller device as shown below.

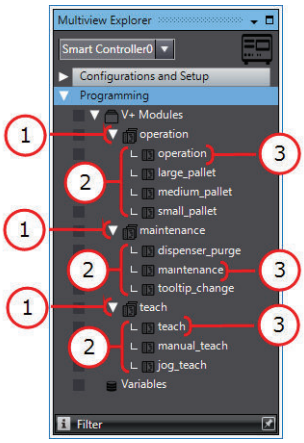
Additional Information

When an ACE project is saved, all V+ Modules, V+ programs, and Variables are saved with the project.



6-2-1 V+ Module and Program Structure

V+ Modules are used to organize and group related programs. Group all similar or related V+ programs inside a specific V+ Module for easy and clear program referencing.
Each V+ Module needs to have a designated Module Program. A Module Program is the primary V + program that ACE software uses for Module naming and other internal functions.
When adding a new V+ Module, a new V+ Module Program will be inserted with the default name of program0. To change the name of the V+ Module, edit the name of the V+ Module Program. To designate a different V+ program as the Module Program, right-click a selected program and click **Set as Module Program**.
All V+ programs are created inside V+ Modules. V+ programs will be displayed under a V+ Module as shown below.



Item	Description
1	V+ Modules
2	V+ programs
3	V+ Module Programs

V+ Module and Program Names

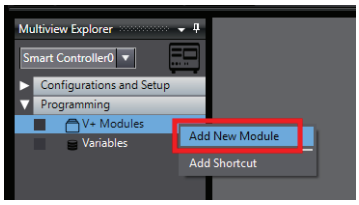
Rename V+ programs by right-clicking the program and selecting **Rename**. Renaming the V+ Module Program will also rename the parent V+ Module. The V+ Module name will always become the name of the V+ program that is designated as the Module Program.

Use the following rules when naming a V+ Module or Program.

- Names must begin with a letter and can be followed by any sequence of letters, numbers, periods, and underscores.
- There is a 15 character limit for names.
- Names are not case sensitive and always default to lower-case letters.

6-2-2 Creating a New V+ Module

To create a new V+ Module, right-click **V+ Modules** in the *Multiview Explorer* and select **Add New Module**. A new V+ Module and default V+ program will be created.

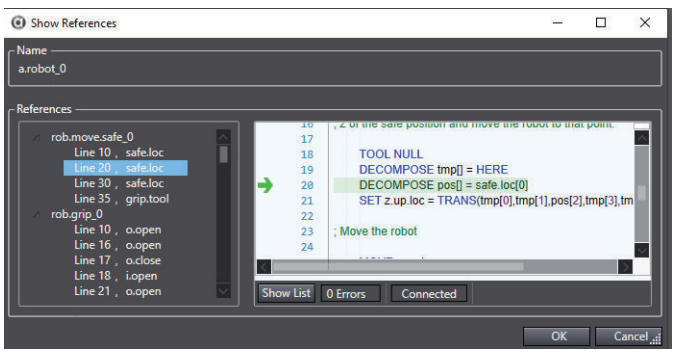


6-2-3 Save a V+ Module to the PC

V+ Modules and their programs can be saved to the PC. Right-click a module and select **Save to PC File**. A *Save As* Dialog Box will be displayed for saving a V+ File (*.pg) to the selected PC file.

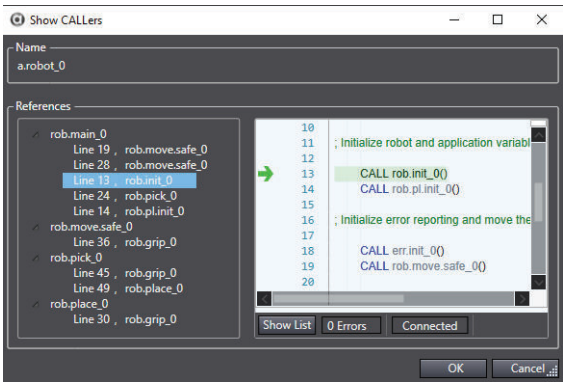
6-2-4 Show Global Variable References

All global V+ variable references can be displayed for a particular V+ Module. Right-click a module and select **Show Global Variables**. A *Show References* Dialog Box will be displayed. This displays all V+ programs and all V+ Variables that are used. Clicking on a V+ Variable in the list will display its use in the program.



6-2-5 Show Program Call Commands

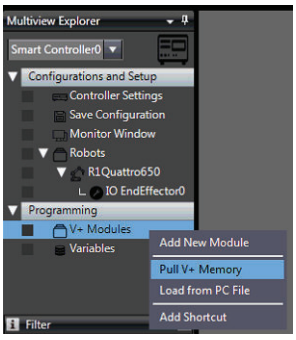
All program CALL commands can be listed for a particular V+ Module. Right-click a module and select **Show CALLers**. A *Show CALLers* Dialog Box will be displayed to provide a list of all CALL commands used throughout the V+ Module's programs. Clicking on an item in the list will display its use in the program.



6-2-6 Pull V+ Memory

Use the Pull V+ memory function to ensure the ACE software has the full contents of a controller available to the user interface. This function will upload V+ Modules, V+ programs, and Variables from the controller to the ACE project.

To upload V+ programs and Modules, right click any V+ Module and select **Pull V+ Memory**. This will upload all controller settings, V+ Modules and Programs to the ACE project.



6-2-7 V+ Editor

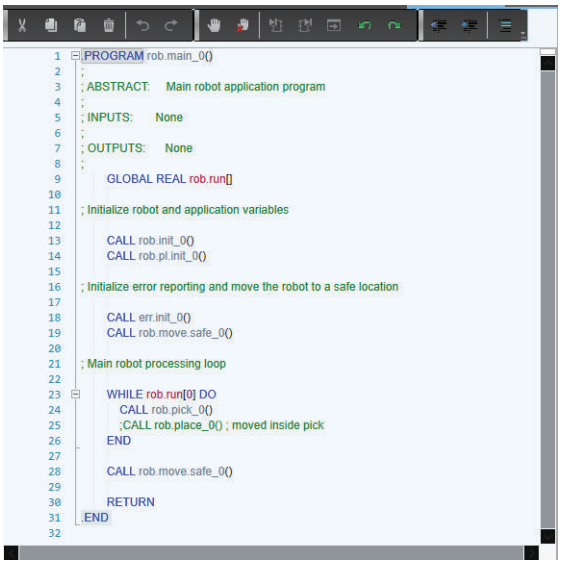
The ACE V+ Editor is an online, interactive editor that checks syntax and formatting while programming.

When the ACE software is online with a controller an opened program will be read from the controller memory. If a program is opened while offline, the program is read from the ACE project memory.

During the ACE online connection process, the controller and project memory are compared. Refer to *4-3 Memory Comparison* on page 4-7 for more info.

To access the V+ Editor, double-click a program in the *Multiview Explorer*. The V+ Editor will open in the *Edit* Pane.

The V+ Editor appearance can be adjusted. Refer to *5-14 Project Options* on page 5-57 for more information.



```
PROGRAM rob.main_0()
;
; ABSTRACT:    Main robot application program
;
; INPUTS:      None
;
; OUTPUTS:     None
;
        GLOBAL REAL rob.run[]

; Initialize robot and application variables

        CALL rob.init_0()
        CALL rob.pl.init_0()

; Initialize error reporting and move the robot to a safe location

        CALL err.init_0()
        CALL rob.move.safe_0()

; Main robot processing loop

        WHILE rob.run[0] DO
            CALL rob.pick_0()
            ;CALL rob.place_0() ; moved inside pick
        END

        CALL rob.move.safe_0()

        RETURN
.END
```

V+ Editor Functions

The Editor Tool has the following functions.

● Copy/Paste

Allows users to copy and paste ASCII text from other programs or other sources.

● Inteliprompt Popups

Type the significant letters of a V+ keyword into the editor and then press the Enter key. The editor attempts to complete the keyword.

- If the keyword is successfully completed, it is displayed in blue text.
- If there is an error, it is displayed in red text.

Inteliprompt Popups can be disabled in the project options area. Refer to *5-14 Project Options* on page 5-57 for more information.

● Tool Tip Syntax

Hover the cursor over a statement, to display a tool tip with the syntax and short description for that statement.

● Formatting and Syntax Checking

As each line of program is entered it is processed by ACE. This processing performs the formatting and checking, reports back the resulting format, and then the editor is updated to reflect this. If there is a problem with the entry, an error appears in the Error List tab below the Edit Pane.

● Drag and Drop

The editor supports drag and drop of ASCII text from another Windows program onto an open area of the editor. This feature is also used to move lines of code within the editor. To move text, highlight the text and then left-click and drag it to the new position.

● Colorization

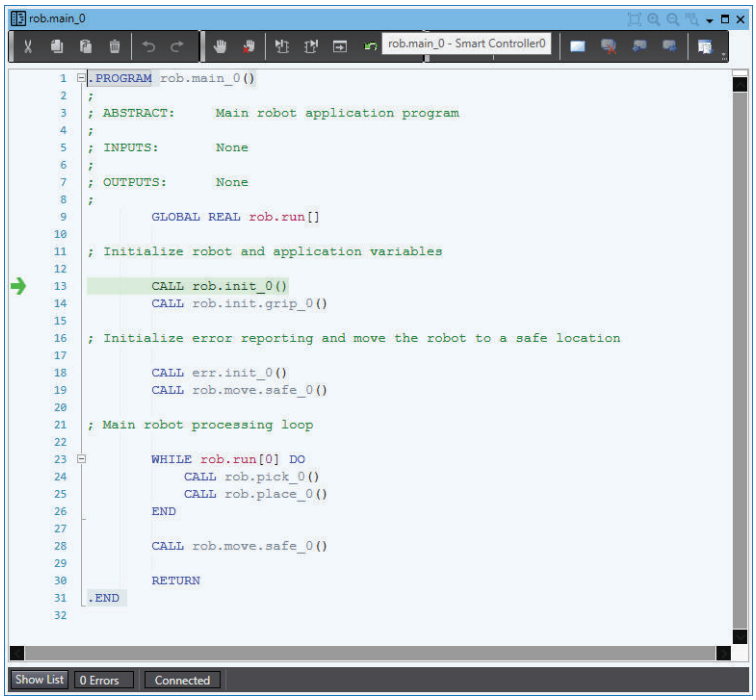
The code lines and text are colored to help visually distinguish between actual code, comments, and errors.

Code lines can have the following colors.

- No Color: code line has no change.
- Green: code line has a saved edit.
- Yellow: code line has an unsaved edit.

Text can have the following colors.

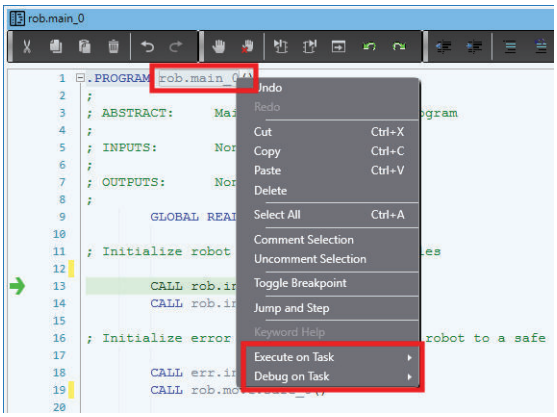
- Green: text used as comments.
- Blue: valid V+ keywords.
- Red underline: misplaced declaration statement or unknown command.
- Gray: object member.



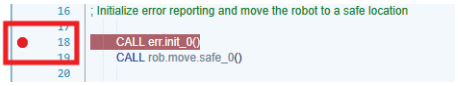
● Right-click Menu Functions

Right-clicking in the V+ Editor opens a menu with several functions. In a addition to basic editing commands (cut, copy, paste, delete, and select all), the following functions are also available.

- Comment/Uncomment Selection: change a code line to/from commented text.
- Execute on task/Debug on task: execute or debug a V+ program on a selected task number. Right-click the program name in the editor and select **Execute on task** or **Debug on task** as shown below.



- Toggle Breakpoint: insert a breakpoint to pause a program at a specific line (used for trouble-shooting and diagnostics). A breakpoint is indicated with a red dot to the left of the code line as shown below.



● Step Into/Over

Use the following Step Into and Step Over functions during troubleshooting and diagnostics.

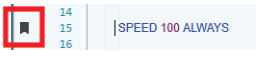
- Step Into: Single step operation that will enter a program call and single-step through the program. After the last line of the program has been executed, it returns to the step following the program call.
- Step Over: Single step operation that skips a program call. When the execution pointer is positioned at a CALL or CALLS keywords, typing F10 will cause the entire subroutine to be executed and execution pauses at the step following the program call.

● Comments

Add comments to a program for explanation and annotation. Commented program lines are not executed.

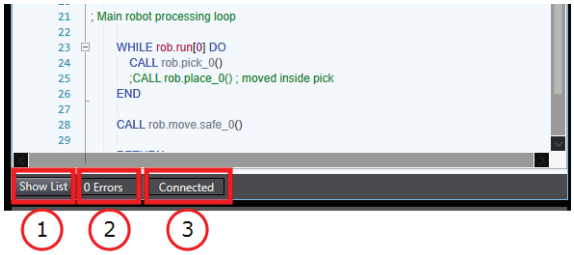
● Bookmarks

Add bookmarks to a program for quick access to specific program lines. Bookmarks do not affect program execution. A bookmarked program line will be indicated with the following symbol.



Status Bar

The area below the V+ Editor window has the following functions.



Item	Name	Description
1	Show List Button	Open the Build Tab Page and display any errors that are present for any programs in the project.
2	Error List	Displays the total count of errors present in the program.
3	Connection State	Displays the connection state of the ACE project.

Toolbar Items

The V+ Editor has the following buttons on the toolbar.

Icon	Name	Description
	Cut	Cut the selection and store on the clipboard.
	Copy	Copy the selection to the clipboard.
	Paste	Paste an item from the clipboard.
	Delete	Delete the selection.
	Undo	Undo the last action.
	Redo	Redo the last action.
	Toggle breakpoint at current line	Add a breakpoint to the currently selected program line.
	Clear all breakpoints	Remove all breakpoints from the active program.
	Step into (V+ program debugging)	Single step operation that will enter a program call and single-step through the program. After the last line of the program has been executed, it returns to the step following the program call.
	Step over (V+ program debugging)	Single step operation that skips a program call. When the execution pointer is positioned at a CALL or CALLS keyword, typing F10 will cause the entire program to be executed and execution pauses at the step following the program call.
	Jump to current line and step (V+ program debugging)	Jump to the currently selected line and then single-step through the program.
	Retry line (V+ program debugging)	Retries the current line
	Proceed execution (V+ program debugging)	Continues execution of the task until the next breakpoint or the program terminates.

Icon	Name	Description
	Outdent	Outdents the current line.
	Indent	Indents the current line.
	Comment section	Adds a semicolon to the beginning of a selected line to make it a program comment.
	Uncomment section	Removes a semicolon from the beginning of a selected line to make it an executable statement.
	Toggle bookmark	Add a bookmark to the selected line.
	Clear bookmark	Remove all bookmarks from the program.
	Previous bookmark	Move the cursor to the previous bookmark in the program.
	Next bookmark	Move the cursor to the next bookmark in the program.
	Display an object member list	Based on the cursor location, this displays a list of the available object members.
	Display parameter info	While the cursor is on a command, this will display that command's parameter info.
	Display quick info	Displays the tooltip info for the cursor location (the same as the tooltip when the cursor hovers).

6-2-8 V+ Program Debugging

The V+ Editor provides debugging tools when an online connection to the controller is present. This 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 V+ Editor will automatically step into that program. Breakpoints in the code can be added or removed while debugging. This provides as many active debugging sessions as there are tasks.



Additional Information

The ACE project must be online to use the V+ Editor program debugging functions.

- Use one of the following methods to access the V+ Editor debugging functions for a program.
- Right-click the program in the *Multiview Explorer* and select **Debug on Task**. Select a task and the program will run with the V+ debugging functions activated.
- Right-click a stopped task in the *Task Manager* and select **Reset and Debug**. The program will reset and run with the V+ debugging functions activated.

A green arrow (➔) indicates the current program line where the stepping function occurs.

6-3 V+ Variables

V+ Variables are values used for storing various types of data. They can be created and edited with the V+ Variable Editor for use in V+ programming. Use the information below to understand how to create and use V+ Variables with the ACE software.



Additional Information

C# Variable objects can also be created and edited, but these are used with C# programs only. Refer to *6-5 Application Manager Programming* on page 6-19 for more information.

6-3-1 V+ Variable Types

The different V+ Variable types are described below.



Additional Information

All V+ Variables that are created with the V+ Variable Editor become global variable types accessible in all V+ programs. Refer to the *eV+ Language User's Guide (Cat. No. I604)* or *V+ User's Manual (Cat. No. I671)* for more information about Auto and Local variables.

Real

A real V+ Variable type is a floating-point data type used to represent a real number.

String

A string V+ Variable type is a text-based data type used to represent characters.
Valid string characters include 128 8-bit ASCII characters.

Precision Point

A precision point variable type allows users to define a location by specifying a value for each robot joint. These joint values are absolute 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 robot arms with joint 4. They are also helpful where joint orientation is critical to the application or to move an individual joint.

Location

A location variable type specifies the position and orientation of the robot tool tip in three-dimensional space. Robot locations can be defined using Cartesian coordinates (transformations).
A transformation is a set of six components that identifies a location in Cartesian space and the orientation of the robot tool flange (X, Y, Z, Yaw, Pitch, Roll). A transformation can also represent the location of an arbitrary local reference frame (also know as a frame of reference). Refer to *Defining a Location* on page 6-12 for more information.

The coordinate offsets are from the origin of the World coordinate system which is located at the base of the robot by default.



Additional Information

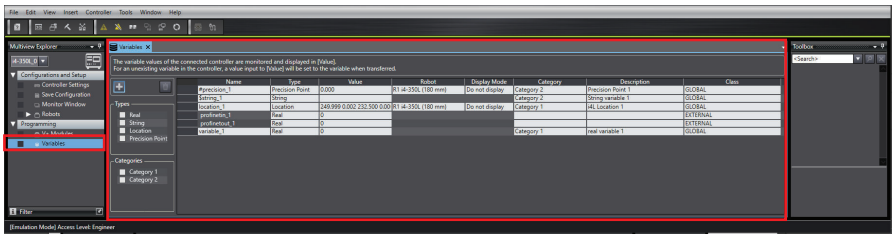
Refer to the following sections for more information.
- *1-5-1 Coordinate Systems* on page 1-15
- *5-3 3D Visualizer* on page 5-24
- *7-5-6 Location* on page 7-51

● Defining a Location

V+ Variable location values can be manually entered or acquired from the current robot position. Refer to *5-8 V+ Jog Control* on page 5-35 for more information.

6-4 V+ Variable Editor

The V+ Variable Editor is used to create and edit different types of V+ Variables. Refer to *eV+ Language User's Guide (Cat. No. I604)* or *V+ User's Manual (Cat. No. I671)* for more information. All variable types share the following properties but some variables have additional properties.
Access the V+ Variable Editor by double-clicking **Variables** in the Multiview Explorer (or right-click and select **Edit**).



6-4-1 V+ Variable Names

V+ Variables must have a unique name. Use the name to reference V+ Variables in V+ programs.
Use the following rules when naming the different V+ Variable types.

- Real and Location variables must begin with a letter and can be followed by any sequence of letters, numbers, periods, and underscores.
- String variables must begin with the $ symbol and can be followed by any sequence of letters, numbers, periods, and underscores.
- Precision Point variables must begin with the # symbol and can be followed by any sequence of letters, numbers, periods, and underscores.
- There is a 15 character limit for variable names.
- Variables are not case sensitive and always default to lower-case letters.
- Because ACE automatically creates default system variable names, avoid creating variable names that begin with two or three letters followed by a period to prevent coincidental variable name duplications. For example, sv.error, tsk.idx, and tp.pos1 are variable names that should be avoided. This restriction applies when creating variables in the V+ Editor and within V+ programs.

6-4-2 V+ Variable Properties

Variables contain properties that define the behavior and use within the ACE software. Use the V + Variable property types described below as new variables are created and edited. Refer to *eV+ Language User's Guide (Cat. No. I604)* or *V+ User's Manual (Cat. No. I671)* for additional information about V+ variable properties.

Name

The name property is used as a reference to a V+ Variable in a V+ program.

Type

The type property defines what type of data is stored in the variable.

Value

The value property is used to set the variable's default value before any program or other function alters it.

Robot

A robot in the ACE project can be assigned to a location or precision point variable. This property does not apply to other variable types. When assigning a robot current position to the location or precision point variable, a robot must be selected. The robot must be assigned for display purposes and for inclusion in location lists provided by the V+ Jog Control.

Display Mode

The following Display Mode options affect how a location or precision point variable appears in the 3D Visualizer. This property does not apply to other variable types.

- Do not display: the point will not appear in the 3D Visualizer.
- Display as point: the point will appear as a circle ().
- Display as frame: the point will appear as a coordinate symbol ().

Category

A category can be defined for each variable for organizational purposes. V+ Variables can be saved by category as well. Refer to *7-3 Save Configuration* on page 7-35 for more information.

Description

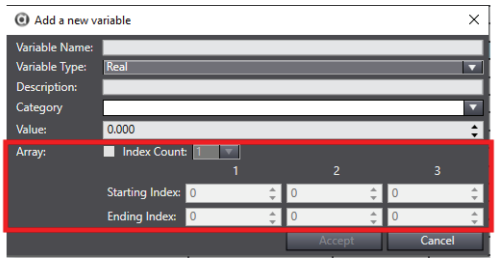
A description can be added to each variable for annotation and clarification purposes.

Class

A variable can be classified as either External or Global. Refer to the *V+ User's Manual (Cat. No. I671) for more information about variable classifications.*

Arrays

A variable can be created as a 1D, 2D, or 3D array. Use the array selection options in the *Add a new variable* Dialog Box to establish the dimensions of the array. Refer to *6-4-3 Creating a New V+ Variable* on page 6-15 for more information.



Additional Information

Limit a location array variable size to 120 or less. Exceeding this size may cause T20 Teach Pendant communication interruptions.

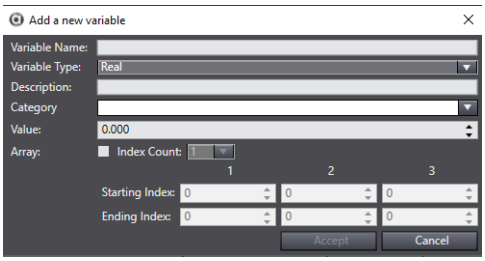
6-4-3 Creating a New V+ Variable

Use the following procedure to create a new V+ Variable with the V+ Variable Editor.

Additional Information

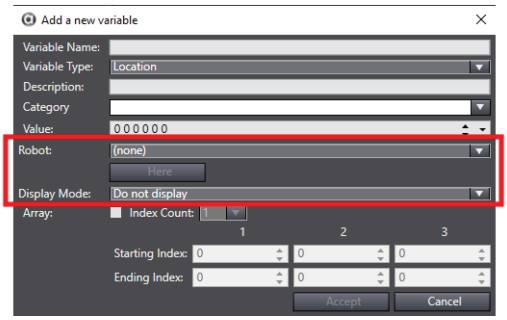
- V+ Variables can also be created from a V+ program. Refer to the *eV+ Language User's Guide (Cat. No. I604)* or *V+ User's Manual (Cat. No. I671)* for more information.
- New Precision Point and Location variable types can also be created from the V+ Jog Control. Refer to *5-8 V+ Jog Control* on page 5-35 for more information.
- Variables classified as External cannot be created in the V+ Variable Editor. These variables are automatically generated during the PROFINET configuration process. Refer to *Configure PROFINET Settings* on page 7-26 for more information.

1 Click the **Add** Button ( ). This will open the *Add a new variable* Dialog Box.



2 Input a unique variable name.

3 Select a variable type from the drop-down menu.
Selecting the variables types of location and precision point will expose the additional field for robot assignment, the *Here* button (for acquiring position values from a robot), and the *Display mode* selection.



4 Provide an optional description.

5 Enter a new category name or select an existing category for the new variable (optional).

6 Enter a value for the new variable.

7 If this variable is an array, select the array option, choose the Index Count, and enter the array dimensions.

8 After all fields are set, click the **Accept** Button and the new V+ Variable is created and visible in the V+ Variable Editor.

6-4-4 Editing an Existing V+ Variable

To edit an existing V+ Variable, click the field to edit in the V+ Variable Editor window and change the value accordingly.

To delete an existing V+ Variable, either select the variable row and click the **Delete** Button () or right-click the variable row and select **Delete**. A *Confirmation* Dialog Box will be shown.



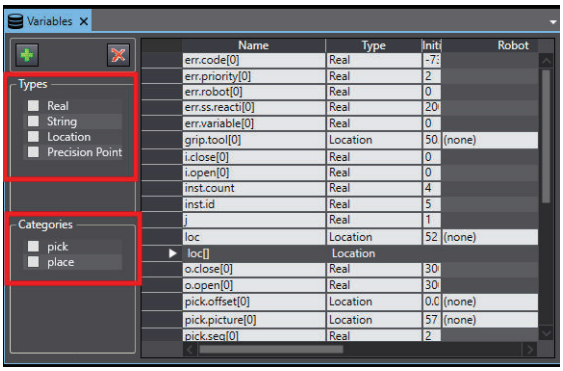
Additional Information

- Multiple variables can be selected by using the Shift or Control buttons and selecting multiple rows.
- Name and Type are not editable for variables that have an External classification property. Use the *Configure PROFINET Settings* area to edit variables with this classification type. Refer to *Configure PROFINET Settings* on page 7-26 for more information.

6-4-5 Filtering by Types and Categories

Existing variables can be filtered by type or category. Category selections will only appear if they have been pre-assigned to existing variables.

Make selections to filter variables by type and category.



6-4-6 Other V+ Variable Editor Functions

Several other functions are available in the V+ Variable Editor. These functions can be accessed by right-clicking an existing variable in the editor window as described below.

Cut, Copy, Paste, Delete

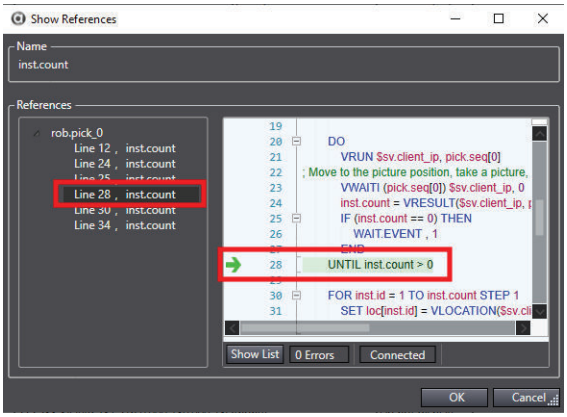
V+ Variable objects can be cut, copied, pasted, and deleted with these options. Pasting will prepend Copy_1_of_ to the name of the new V+ Variable object.

Additional Information

Variables that have an External classification property cannot be deleted. Use the *Configure PROFINET Settings* area to remove variables with this classification type. Refer to *Configure PROFINET Settings* on page 7-26 for more information.

Show References

Select **Show References** to display a list of program references where the variable is used. If a variable is used in programs, the program name and line numbers are provided to locate exactly where the variable is used. Clicking a line number will display the program reference as shown below.



Add to Watch

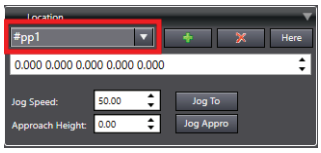
Selecting **Add to Watch** will place the C# Variable object in the V+ Watch window. Refer to *5-11 V+ Watch* on page 5-52 for more information.

Record Here

Selecting **Record Here** will acquire the robot's current position values and place them in the initial value field. This option is only available if a robot has been assigned in the variable. This option is only available for precision point and location variable types.

Select in Virtual Pendant

Choosing **Select in Virtual Pendant** will open the V+ Jog Control with the variable pre-selected in the Location area. This is a convenient method for jogging and teaching a robot position for a variable. This is only available if robot has been assigned in the variable. Refer to *5-8 V+ Jog Control* on page 5-35 for more information. This option is only available for precision point and location variable types.



Focus in 3D Visualization

Selecting **Focus in 3D Visualization** will open the 3D Visualizer and center the view at the coordinates. This option is only available for precision point and location variable types with an assigned robot.

6-5 Application Manager Programming

This section describes the basics of C# programs in the ACE software. These concepts apply to the other C# interfaces as well. For additional ACE API documentation and simple C# program examples, refer to the ACE API Reference Manual (Help Menu).

Precautions for Correct Use

If you are inexperienced with the object-oriented C# programming language, it is recommended that you explore Microsoft's documentation and tutorials to become familiar before attempting to create or edit these types of programs.

The Application Manager device provides C# interfaces for the following functions.

- C# custom program functions (see below)
- Custom Vision Tool (refer to *Custom Vision Tool* on page 8-323 for more information)
- Allocation Scripts (refer to *8-7-6 Allocation Script Object* on page 8-160 for more information)
- Recipe Manager Scripts (refer to *8-5-9 Recipe Manager Script* on page 8-80 for more information)

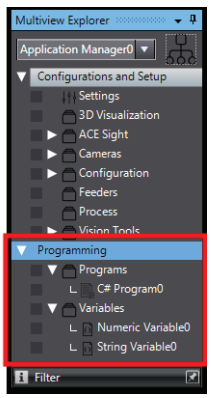
C# programs can be used for the following functions, for example.

- Managing product changes for a workcell or production line.
- Enabling or disabling processes of a Process Manager in a packaging application.
- Changing properties of vision tools or associating different vision models with a Locator Tool.
- Custom statistics tracking
- Manipulating the position and visibility of objects in the 3D Visualizer.

Using numeric and string variable objects is required to share variable values between multiple C# programs. C# programs and variables can be accessed in the Programming section of the Multi-view Explorer for a Application Manager device as shown below.

Additional Information

When an ACE project is saved, all C# programs and Variables objects are saved with the project.

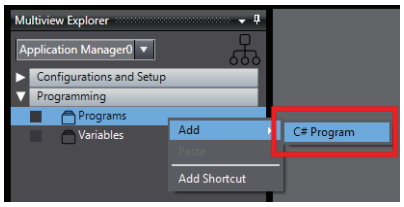


6-5-1 Creating a New C# Program

To create a new C# program, right-click **Programs** and select **Add** and then **C# Program**. A new C# program will be created.

Additional Information

New C# programs are created with minimum program structure in place as well as a basic trace message to indicate when a script has started.

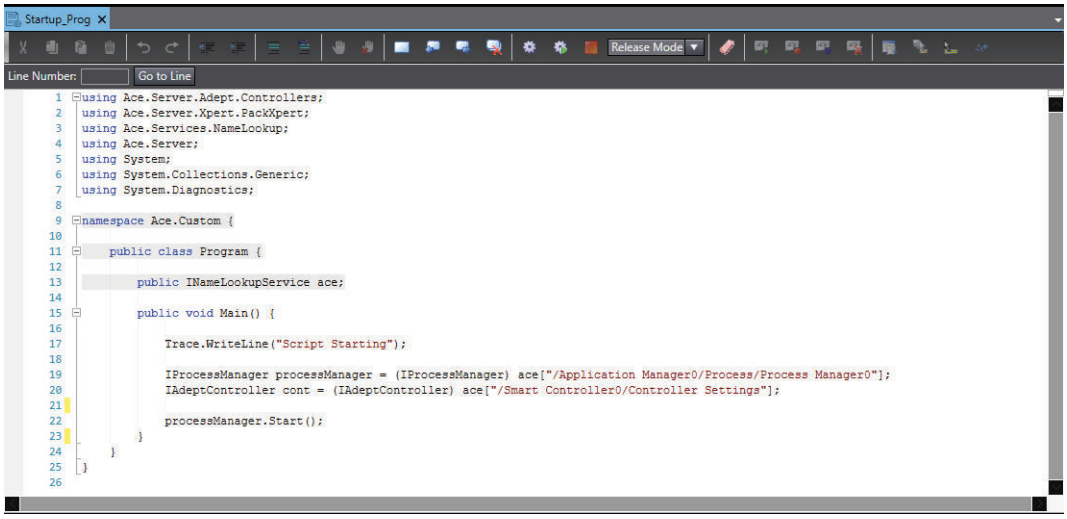


C# Program Names

Rename C# programs by right-clicking the program and selecting **Rename**. Program names can include letters, numbers, and special characters.

6-5-2 C# Program Editor

The C# program editor can be used to create and edit programs for performing various tasks and automation within the ACE project.
To access the C# Editor, double-click a program in the *Multiview Explorer*. The C# Editor will open in the *Edit* Pane.



C# Editor Details

The Editor Tool has the following functions.

● Drag and Drop Reference to Objects

Application Manager objects can be referenced in the C# program with a drag and drop action. For example, drag and drop a virtual camera object to place a reference to that object in the C# program.

● Access to Controller Memory Contents

Drag and Drop Controller Settings into a C# program to create a reference to a specific robot controller device. V+ variables, digital I/O, and more can be accessed from this reference.

● Copy/Paste

Use the copy and paste function to move ASCII text from other programs or other sources.

● Auto-complete

Type the significant letters of a statement and then press the Enter key, the editor then provides options to complete the keyword.

● Tool Tip Syntax

View the syntax and short description tool tip for a statement by hovering the cursor over that statement.

● Formatting and Syntax Checking

As each line of program is entered it is processed by ACE. This processing performs the formatting and checking, reports back the resulting format, and then the editor is updated to reflect this. If there is a problem with the entry, an error appears in the *Error* List tab below the *Edit* Pane. Refer to *Error List* on page 6-22 for more information.

● Drag and Drop

The editor supports drag and drop of ASCII text from another Windows program onto an open area of the editor. Use this feature to move lines of code within the editor by highlighting the text and then dragging it to the new position.

● Colorization

The code lines and text are colored to help visually distinguish between actual code, comments, and errors. Code lines can have the following colors.

- No Color: code line has no change.
- Yellow: code line has an unsaved edit.

Text can have the following colors.

- Green: text used as comments.
- Blue: definitions
- Red: Strings
- Black: non-keywords and operators as valid syntax

● Right-click Menu Functions

Right-clicking in the C# Editor opens a menu with several functions. In a addition to basic editing-commands (cut, copy, paste, delete, and select all), the following functions are also available.

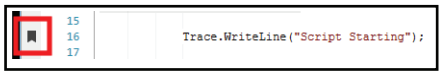
- Toggle All Outlining: collapse/expand all outlining regions of code.
- Toggle Outlining Expansion: collapse/expand selected outlined region of code.
- Collapse to Definitions: collapse areas for all members.
- Comment/Uncomment Lines: change a code line to/from commented text.

● C# Program Editor Comments

Add comments to a program for explanation and annotation. Commented text is not executed.

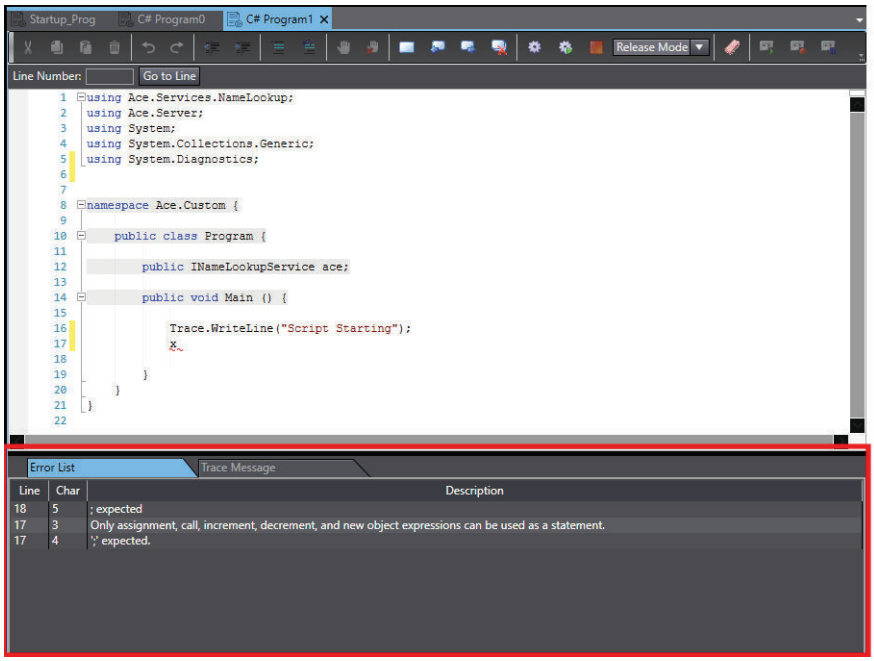
C# Program Editor Bookmarks

Add bookmarks to a program for quick access to specific program lines. Bookmarks do not affect program execution. A bookmarked program line will be indicated with the following symbol.



Error List

Program compile errors are shown in the Error List Tab below the C# program Edit Pane. When errors are present in the list, execution is not possible. Use the Error List Line, Char (character), and Description information to resolve program compile errors.

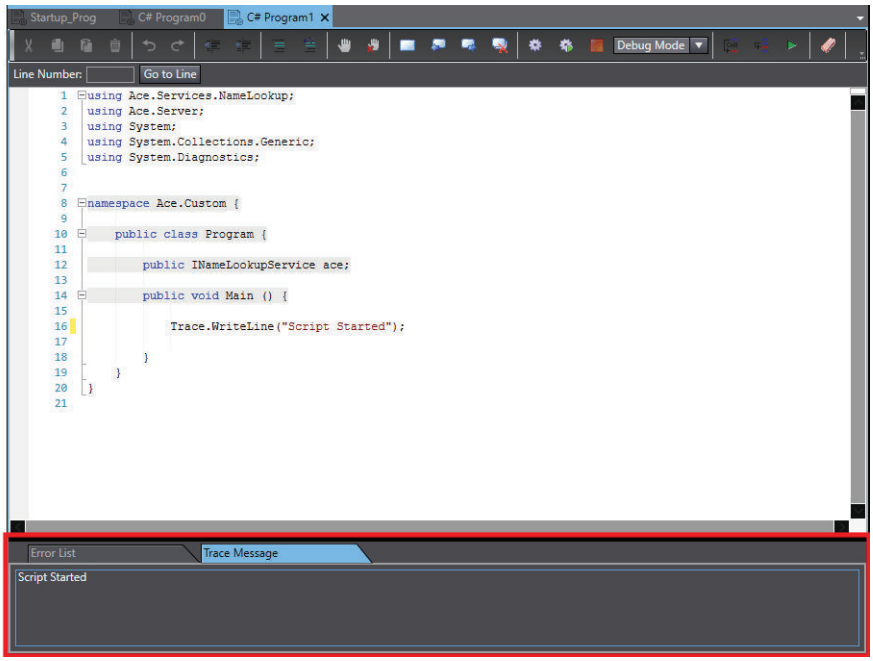


Trace Messages

Trace Messages are shown in the Trace Message Tab below the C# program Edit Pane. This displays messages created from any Trace.WriteLine() call in the program and can be useful when troubleshooting or debugging a program.

Additional Information

Erase all Trace Messages with the ( ) icon in the C# program editor toolbar. This will also clear Trace Messages from the Task Status Control Trace Messages area.

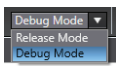


● Toolbar Items

The C# Editor has the following buttons on the toolbar.

C# Editor Toolbar Description

Icon	Name	Description
	Cut	Cut the selection and store on the clipboard.
	Copy	Copy the selection to the clipboard.
	Paste	Paste an item from the clipboard.
	Delete	Delete the selection.
	Undo	Undo the last action.
	Redo	Redo the last action.
	Outdent	Outdents the current line.
	Indent	Indents the current line.
	Comment section	Adds a semicolon to the beginning of a selected line to make it a program comment.
	Uncomment section	Removes a semicolon from the beginning of a selected line to make it an executable statement.

Icon	Name	Description
	Toggle breakpoint at current line	Add a breakpoint to the currently selected program line.
	Clear all breakpoints	Remove all breakpoints from the active program.
	Toggle bookmark	Add a bookmark to the selected line.
	Clear bookmark	Remove all bookmarks from the program.
	Previous bookmark	Move the cursor to the previous bookmark in the program.
	Next bookmark	Move the cursor to the next bookmark in the program.
	Compile	Compile the program.
	Compile and Run	Compiles and runs the program.
	Stop	Stops the program. Only available when the program is running.
	Debug/Release Mode	Selects the editor mode. Release Mode is the normal execution mode. Debug Mode is used to debug the program.
	Step Over	Single step operation that skips a program call. Only available when breakpoints are present in Debug Mode. A green arrow (➔) indicates the current program line where the stepping function occurs.
	Step Into	Single step operation that will enter a program call and single-step through the program. Only available when breakpoints are present in Debug Mode. A green arrow (➔) indicates the current program line where the stepping function occurs.
	Go	Runs the program. Only available when breakpoints are present in Debug Mode.
	Erase Trace	Clears messages from the Trace Message Tab and the Task Status Control Trace Messages Area.
	Run Recorded Macro	Runs a recorded Macro. This button is only available if a Macro has been recorded. Closing the C# Editor will erase any recorded Macros.
	Record Macro	Begin recording a Macro. Recording a Macro captures a series of keystrokes for replay with the **Run Recorded Macro** button. Click the **Record Macro** button while recording to end and save the Macro.
	Pause Recording	Pauses a Macro recording. This button is only available while a Macro is recording.

Icon	Name	Description
	Cancel Recording	Cancels a Macro Recording. This button is only available while a Macro is recording.
	Display an object member list	Based on the cursor location, this displays a list of the available object members.
	Display parameter info	While the cursor is on a command, this will display that command's parameter info.
	Display quick info	Displays the tooltip info for the cursor location (the same as the tooltip when the cursor hovers).
	Display Word Comple-tion	Based on the cursor location, this displays a list of the available object members.

6-6 C# Variable Objects

C# Variables objects are values used for storing numeric and string data types. After adding C# Variables objects, they can be edited with the C# Variable Editor. Use the information below to understand how to create and use C# Variables objects with the ACE software.

Additional Information

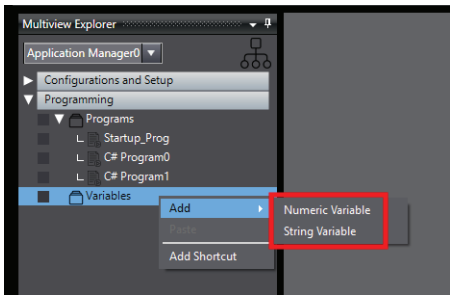
V+ Variables objects can also be created and edited, but these are used with V+ programs only. Refer to *6-2 Robot Controller Programming* on page 6-3 for more information.

6-6-1 C# Variable Object Types

Numeric and string C# Variable objects can be used to share data between C# programs that run separately. Numeric types hold numerical values and string types hold text-based values used to represent characters. Valid string characters include 128 8-bit ASCII characters.

6-6-2 Creating a New C# Variable Object

Right-click **Variables** in the *Multiview Explorer* and select **Add** and then **Numeric Variable** or **String Variable**. The new variable will be added to the *Variables list*.



6-6-3 Other C# Variable Object Function Keywords

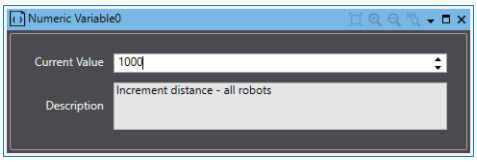
Several functions are available for existing C# Variables objects. Right-click an existing C# Variable object to display a menu with function selections described below.

Cut, Copy, Paste, Delete

C# Variable objects can be cut, copied, pasted, and deleted with these options. Pasting will prepend Copy_1_of_ to the name of the new C# Variable object.

6-6-4 Editing C# Variable Objects

After a C# Variable object has been created, double-click the variable in the *Multiview Explorer* to open the *C# Variable Editor* in the *Edit* Pane. Assign a current value and provide a description in the editor.



Add a watch to the *C# Edit* Pane, using the steps in *Set Value in C# Watch Window* on page 6-27. Refer to *6-5-2 C# Program Editor* on page 6-20 for more information.

Set Value in C# Watch Window

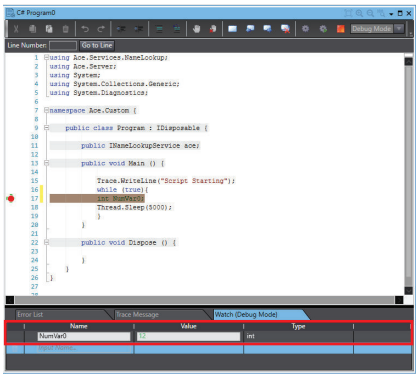
C# Variable object values can be set using the Watch function. Use the following procedure to add a Watch function and then set a value.

Additional Information

- The insertion of a watch item is at a break point.
- Debug mode must be enabled to use the Watch function.
- The C# program must be running.

1 Toggle a break point at the line with the C# variable to set.

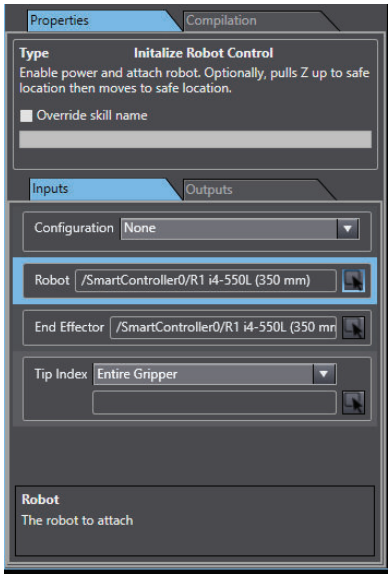
2 Right-click the editor at that line of code with the C# variable to set, and then select **Add Watch** from the drop-down menu.
A Watch line opens in the panel under the C# editor.



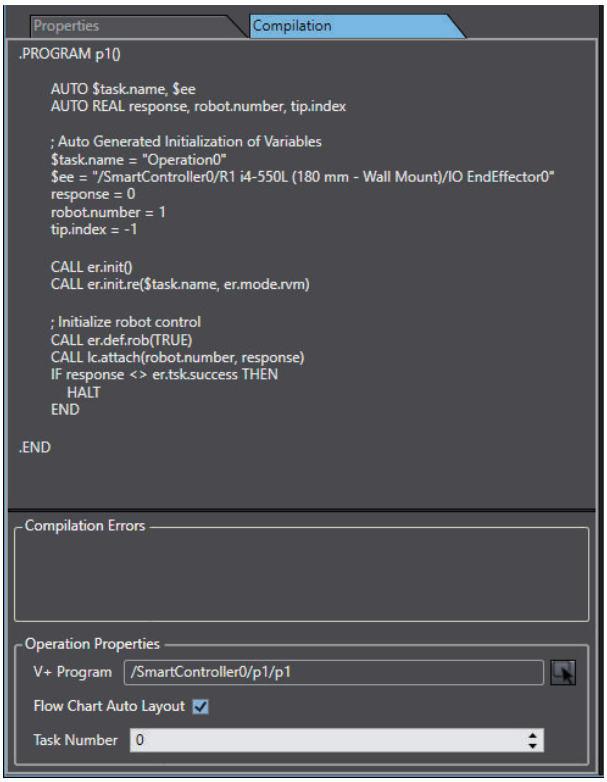
3 Input the new value and the press enter to set the new value.

6-7 Operation Programming

Operation programming is a simple method to achieve complex robot functions for a variety of common applications.
An Operation utilizes a flowchart arrangement of various Skills with different functions. Skills that are available in the Toolbox can be added and arranged to perform specific application operations. Add process flow controls, logic, and specialized functions to any Operation by adding specific Skills from the Skills Library. V+ and C# programs can also be included into the Operation by using the V+ Code and Run Script Skills.
After a Skill is added to the Operation flowchart, it must be configured by setting inputs and outputs in the properties area that are specific to that Skill.



Some Skills require references to other objects in the properties area such as Robot Locations, Encoders, or Vision Tools. It may be necessary to preconfigure other objects in the Multiview Explorer before some Skills can be used.
After a Skill is configured, V+ code is automatically generated and viewable in the Compilation area.



The generated V+ code can be viewed and edited within the designated robot controller program. There are two types of Operations; Operations and Process Operations. Operations Process Operations are unique because they are specifically designed to be compatible with Process Managers and contain Process Manager-specific Skills.
An Application Manager must be present in the system before Operation programming can occur. Operations can be created manually or automatically as described in the following sections.

6-7-1 Automatically Create an Operation

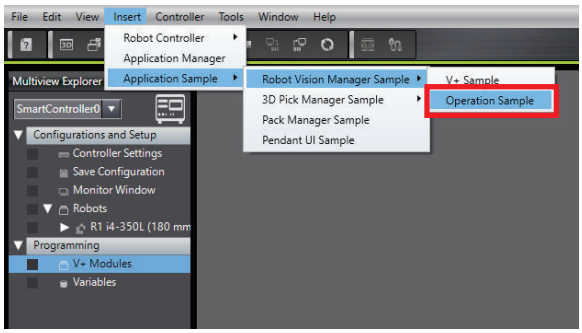
An Operation can be created automatically using the Application Sample wizards. Use these wizards to automatically create a Robot Vision Manager sample Operation or a 3D Pick Manager sample Operation.
A robot must be present in the project before creating a Robot Vision Manager sample Operation. The wizard will automatically add the following items to the project, plus additional items required for the application.

- Application Manager
- Operation

Additional Information

A 3D Pick Manager license must be enabled to use the 3D Pick Manager sample Operation.

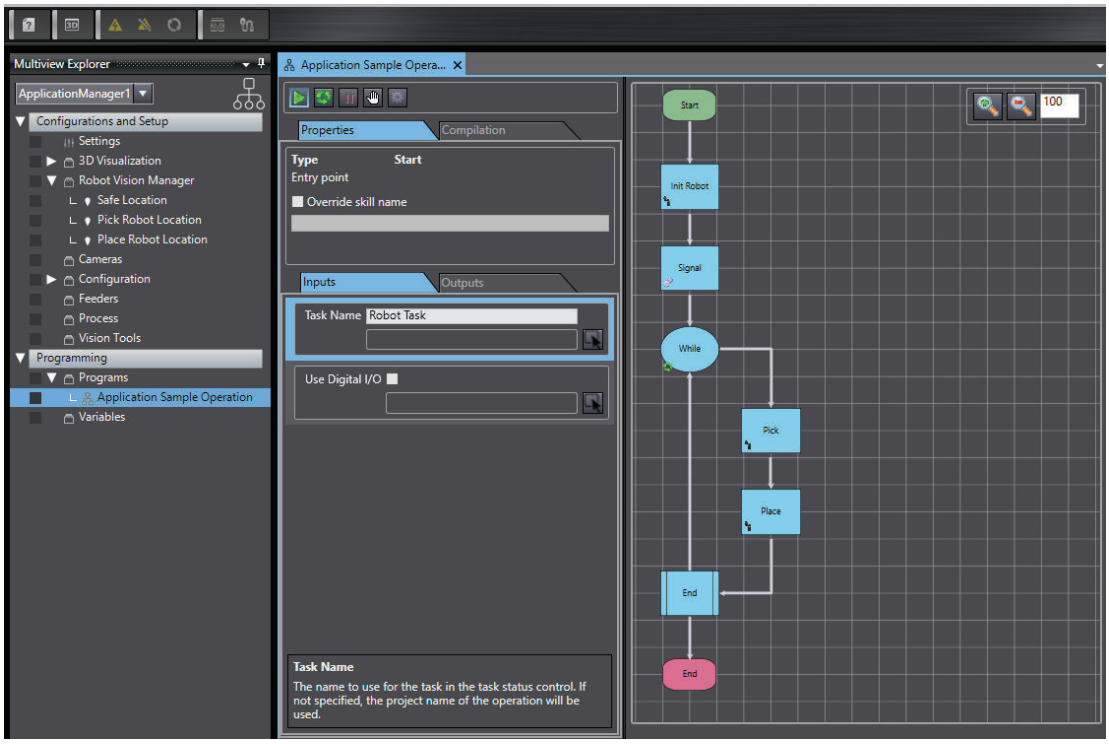
1 Select an **Application Sample** from the *Insert* Menu (Robot Vision Manager Sample or 3D Pick Manager Sample).



This will open an Application Sample wizard.

2 Proceed through the Application Sample wizard to make appropriate selections.

3 Click the **Finish** Button to complete this procedure.
After the Finish Button is clicked, the new Operation will be accessible in the ACE project.

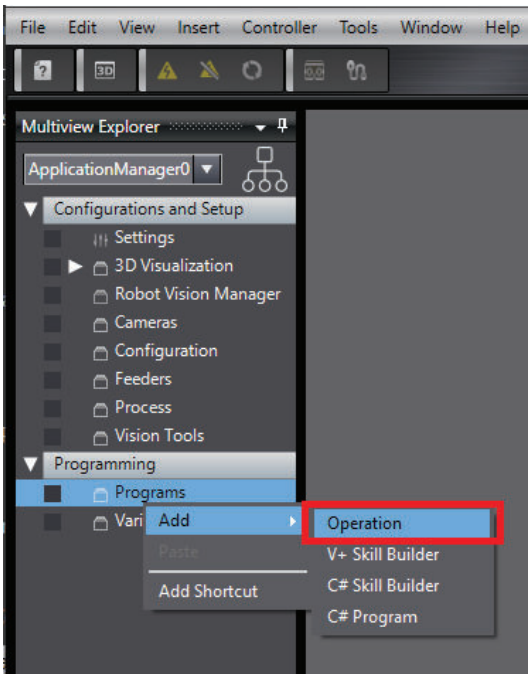


6-7-2 Manually Create an Operation

Use the following procedure to create an Operation manually.
The following items must be present in the project before manually creating an Operation.

- Robot controller
- Application Manager

1 In the Multiview Explorer of the Application Manager, navigate to the *Programming* section, right click **Programs**, then select **Add** and choose **Operation**.



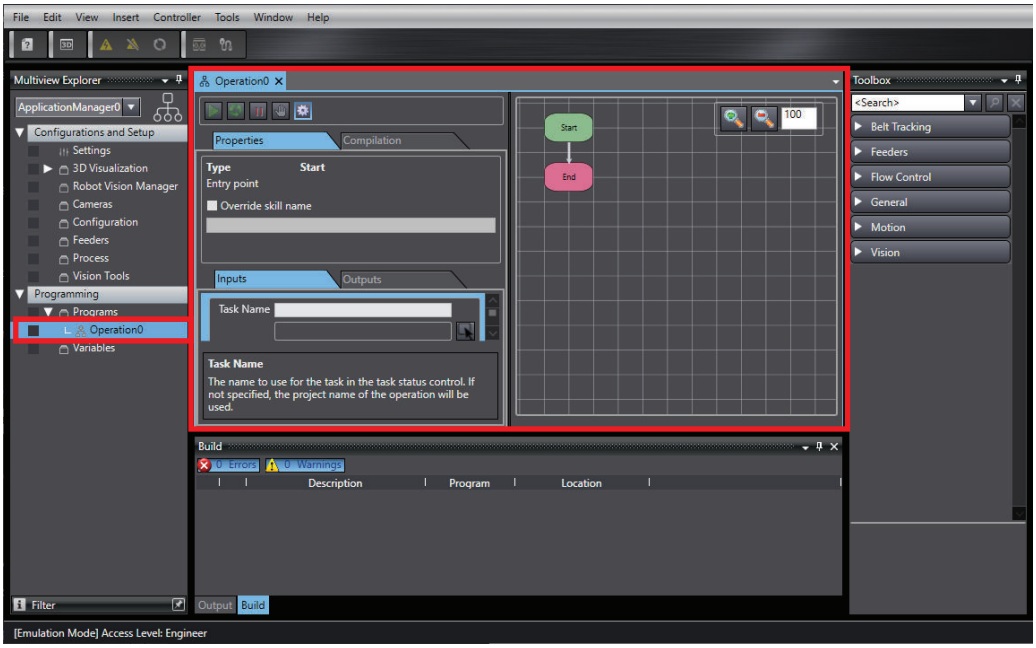
This will open the Create an Operation Wizard.

2 Select the robot controller object.

3 Select a Module or Program to be associated with the Operation.
Click the **Add New Program** Button to create a new program associated with the Operation. The new program name must be entered using V+ program naming requirements. Refer to *V+ User's Manual (Cat. No. I671)* for more information about V+ program naming requirements.

4 Select a task number that the Operation will run under.

5 Click the **Finish** Button to complete this procedure.
A new Operation will be opened in the Programs section of the Application Manager in the Multiview Explorer. This will create a new Operation with only a Start and End Skill.

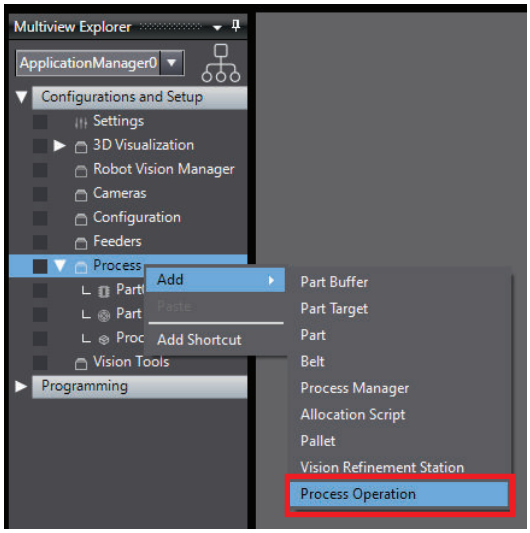


6-7-3 Manually Create a Process Operation

Use the following procedure to create a Process Operation manually.
The following items must be present in the project before manually creating a Process Operation.

- Robot Controller
- Application Manager
- Process Part object
- Process Part Target object
- Process Manager

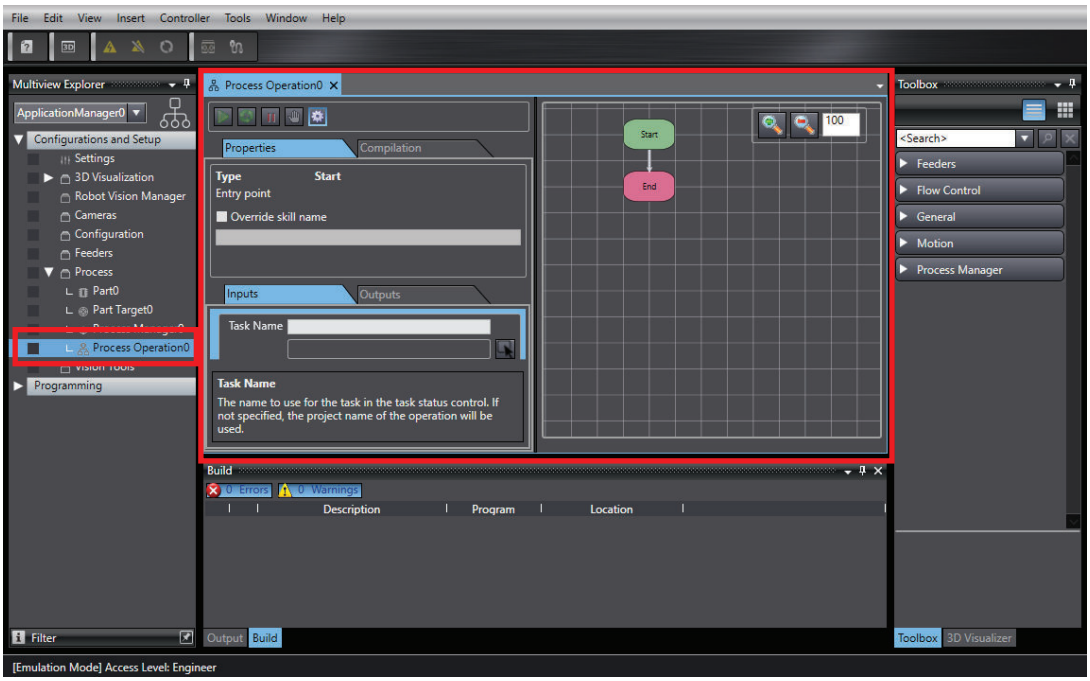
1 In the Multiview Explorer of the Application Manager, navigate to the *Configuration and Setup* section, right click **Process**, then select **Add** and choose **Process Operation**.



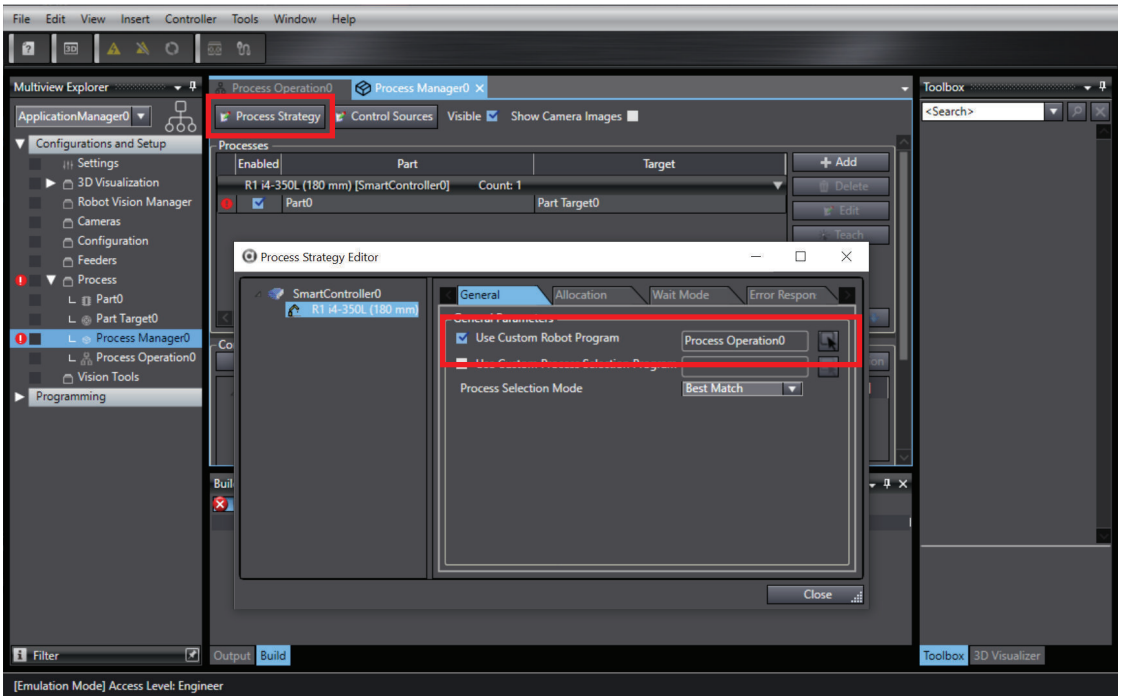
This will open the Create an Operation Wizard.

2 Select the Module or Program.
If a program is not present in the project, click the **Add New Program** Button. The new program name must be entered using V+ program naming requirements, refer to *V+ User's Manual (Cat. No. I671)* for more information. This will create a new Process Operation with only a Start and End Skill.

3 Click the **Finish** Button to complete the Create an Operation wizard and open the new Process Operation.
A new Process Operation will be present in the Process section in the Multiview Explorer.

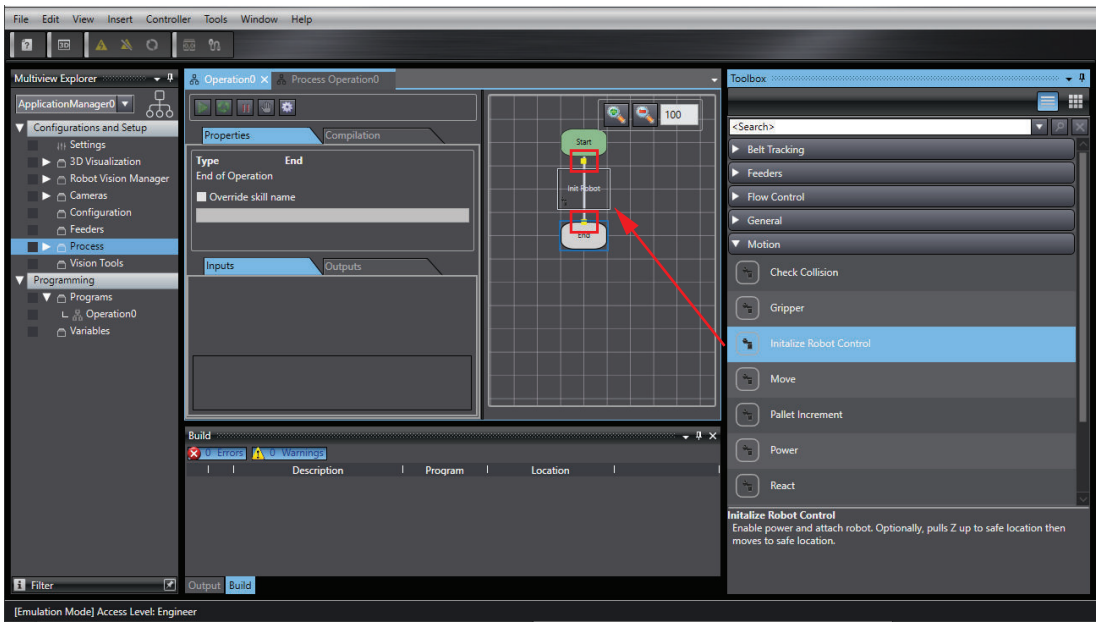


4 Access the Process Manager and then click the **Process Strategy** Button to set the previously created Process Operation as the *Use Custom Robot Program* selection to complete this procedure.



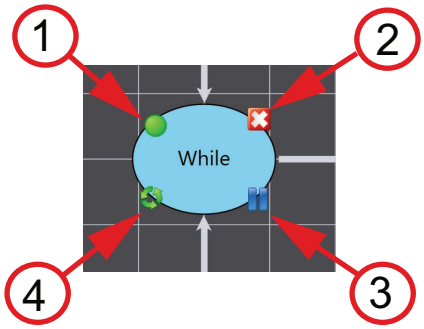
6-7-4 Add Skills to an Operation

Skills can be added to an existing Operation by simple drag-and-drop actions. Select a Skill from the Toolbox and place it in the Operation flow. Yellow endpoints appear to indicate acceptable locations to place the Skill.



6-7-5 Skill Icons

The Skill icons visually identify the Skill's type and status after they are placed into an Operation.



Item	Description
1	The indicator shows when the Operation is paused or operating in Toggle mode. The indicator colors are green for a True and black for a False setting in a Flow Control Skill.
2	A red box with a white X indicates the Skill requires configuration within its Properties panel. An Operation will not run if any Skill has this indicator.
3	The blue double bar indicator shows when the Operation is in Pause mode or Toggle mode.
4	This indicator corresponds to the Skill's category within the Toolbox. Refer to *6-7-7 Operation Skills* on page 6-36 for more category information.

Some skills from the Flow Control category of the Toolbox apply conditions to the process logic. These conditions, such as No or Yes, show at the output for the Skill.

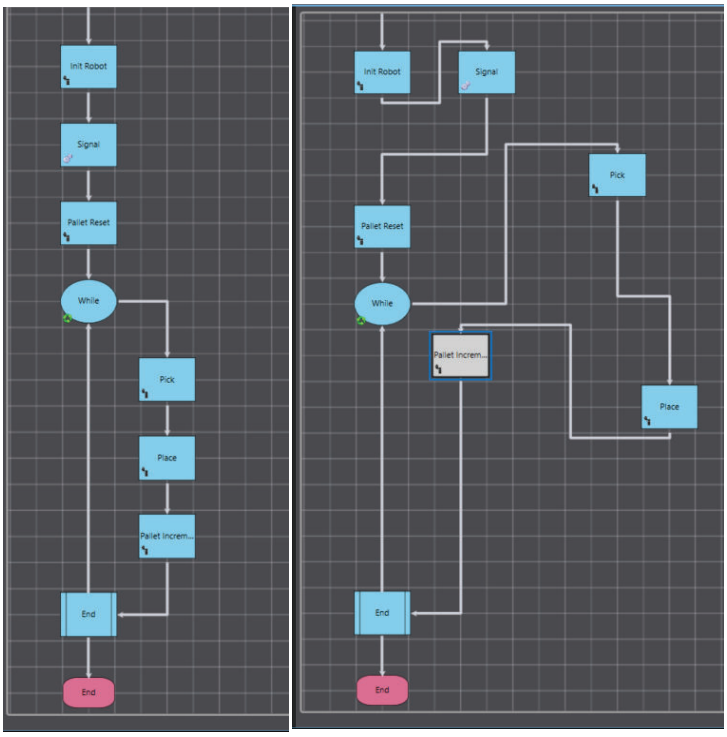


6-7-6 Edit an Operation

Operation editing is done using the Operation edit pane and Skills. Use the following methods to edit an Operation.

Move Skills

Left-click a Skill icon and drag it to another location on the Operation pane. The Skill keeps its connections to the skills before and after it in the Operation. The placement on the Operation pane changes without changes in the logic or flow of the Operation.



The Operation can be returned to a linear flow pattern by right-clicking in the grid, opening a menu, and selecting **Flowchart Auto Layout**.

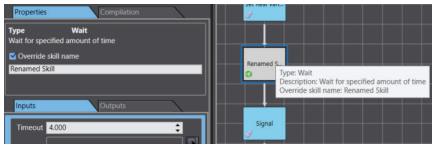
Copy and Paste Skills

Skills on an Operation pane are reusable and they can be copied and pasted. The following methods are used for the copy, paste and delete actions.
Skills can be cut and pasted using either of two methods. The first method is to select the skill, right-click and select **Cut**. Then place the cursor over the new location's flow line, right-click and select **Paste**. For the second method: Place the cursor on the skill, left-click and hold. Then hold the **Control** key and drag the skill to the new location and release the mouse button and **Control** key.

Remove and Rename Skills

Delete a skill from an Operation by left-clicking the skill and pressing the **Delete** key. An alternative is to right-click the skill icon and select **Delete** from the pop-up menu options.

An Operation is easier to understand when the skills are named to correspond to actions or specific equipment that is operating. Rename skills using the *Properties* panel of that skill. Select the skill icon to rename and left-click it. In the Properties panel select the **Override skill name** check box. Enter the name in the line under the check box to complete the renaming.



6-7-7 Operation Skills

Operation Skills are described below.

Skill Group	Skill Name	Description	Additional Information
Belt Tracking	Belt Initialization	Start the belt running.	*8-3-5 Belt Calibration* on page 8-23
	Track a Belt Latch or Camera Instance	Tracks an instance associated with a belt latch or belt camera location into range of the robot belt window.	• *8-3-11 Robot Location* on page 8-39 • *8-3-9 Communication Tool* on page 8-34
Custom	Custom Skills can be created from V+ programs or C# Programs.		*6-7-8 Building Custom Skills* on page 6-38
Feeders	AnyFeeder Execute	Executes an AnyFeeder command.	*8-6-1 AnyFeeder Object* on page 8-82
	AnyFeeder Process	Executes a vision sequence and series of AnyFeeder sequences.	• *8-3-7 Vision Sequence* on page 8-27 • *8-6-1 AnyFeeder Object* on page 8-82
	iPF Process	Executes a vision sequence and series of iPF sequences.	• *8-3-7 Vision Sequence* on page 8-27 • *8-6-3 iPF Object* on page 8-90
	iPF Execute	Executes an iPF command.	*8-6-3 iPF Object* on page 8-90
Flow Control	Call	Call a V+ program using the CALL Keyword.	V+ Keyword Reference Manual (Cat. No. I672)
	Case	Execute the V+ CASE Keyword.	
	Do Until	Create a loop using DO, UNTIL, and TRUE Keywords.	
	For	Create a loop using the FOR and END Keywords.	
	If	Performs a conditional check and branch based on the result using the IF, TRUE, THEN, and END Keywords.	
	Value	Execute the VALUE Keyword with a CASE Skill.	
	Wait	Wait for specified amount of time using the WAIT Keyword.	
	Wait Until	Wait until a condition is detected using the WAIT Keyword.	
	While	Create a loop using the WHILE, TRUE, DO, and END Keywords.	

Skill Group	Skill Name	Description	Additional Information
General	Log	Output a text string to the Monitor Window or write to a file on the controller.	
	Run Script	Run a C# script	*6-5 Application Manager Programming* on page 6-19
	Set Real Variable	Sets the value of a real V+ variable.	*6-3 V+ Variables* on page 6-11
	Set String Variable	Sets the value of a string V+ variable.	
	Signal	Controls the status of digital signals using the SIGNAL Keyword.	V+ Keyword Reference Manual (Cat. No. I672)
	Task Control	Control the execution status of a task using EXECUTE, ABORT, PROCEED, or CALL Keywords.	
	V+ Code	Insert a snippet of V+ Code into the generated program	*6-2 Robot Controller Programming* on page 6-3
Motion	Check Collision	Checks if the robot will collide with another object when positioned at the specified Robot Location.	*5-3-4 Graphical Collision Detection* on page 5-29
	Gripper	Perform an operation on the gripper.	*7-5-2 I/O EndEffectors* on page 7-43
	Initialize Robot Control	Enable power and attaches robot. Optionally, moves Z-axis to safe location then moves to safe location.	• *8-3-11 Robot Location* on page 8-39 • *8-3-12 Robot Precision Point* on page 8-41 • *7-5-2 I/O EndEffectors* on page 7-43
	Move	Move a robot to a location or relative to the current location.	• *8-3-11 Robot Location* on page 8-39 • *8-3-13 Robot Transfer Path* on page 8-42 • *7-5-2 I/O EndEffectors* on page 7-43 • *1-6-2 Approach and Depart* on page 1-27 • Refer to the ALIGN Keyword details in the V+ Keyword Reference Manual (Cat. No. I672) for more information.
	Pallet Increment	Increment pallet slot index for the specified Robot Location.	The Robot Location must refer to a Robot Frame with a pallet configuration.
	Power	Enable or disable robot High Power.	*5-9-2 Robot Power Control* on page 5-40
	React	Enable or disable REACTE and REACTI Keyword functions.	V+ Keyword Reference Manual (Cat. No. I672)
	Stop Motion	Stop the robot motion.	Refer to the BRAKE keyword when Stop Immediately is selected. Refer to the BREAK keyword when Stop Immediately is not selected in the V+ Keyword Reference Manual (Cat. No. I672).
	Wait Until Robot	Wait until the robot state matches a criteria	Refer to the WAIT and STATE keywords in the V+ Keyword Reference Manual (Cat. No. I672).

Skill Group	Skill Name	Description	Additional Information
Process Manager*1	Move to Wait	Move the robot to a waiting position.	V+ Module Reference Manual (Cat. No. I668)
	Part Status	Get the status of the part queue.	
	Part Target Status	Get the status of the target queue.	
	Pick Part	Pick a part.	
	Place Target	Place a target.	
	Refine	Refine an instance in the gripper.	
Vision	Load Vision Location Results	Find results using a vision sequence and associate them with a vision relative location.	*8-3-7 Vision Sequence* on page 8-27
	Read Vision Result	Reads a vision result from a vision sequence.	*8-3-7 Vision Sequence* on page 8-27
	Refine Part Gripper	Refine the part in the robot gripper.	• *8-3-7 Vision Sequence* on page 8-27 • *8-7-8 Vision Refinement Station Object* on page 8-167
	Run Vision Sequence	Run a vision sequence once or in continuous mode.	*8-3-7 Vision Sequence* on page 8-27
	Vision Results Increment	Returns the number of vision results associated with a vision relative location.	After a Run Vision Sequence, use Read Vision Result to return the number of objects. Use Vision Result Increment to increment to the next vision result for the next motion.

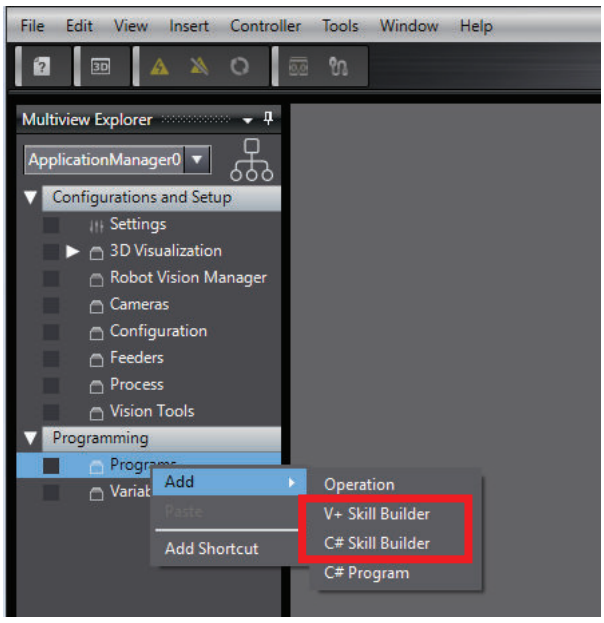
*1. Available when a Process Operation is present.

6-7-8 Building Custom Skills

Custom Skills can be created using the **Skill Builder** selections in the *Programming* area of the Multiview Explorer.

A custom Skill can be created from an existing V+ program or a C# program.

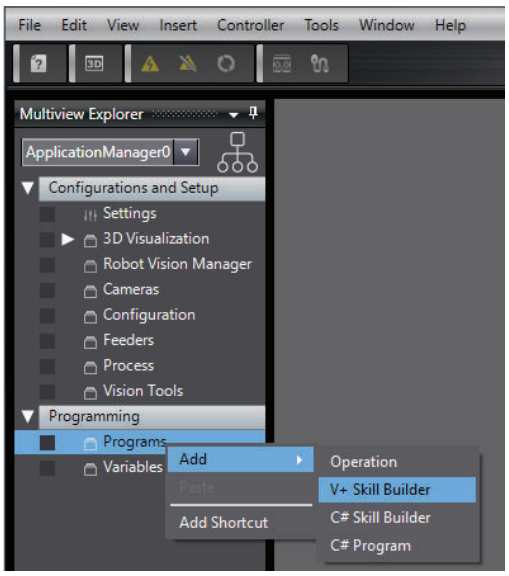
Once a custom Skill is created, it will be available in the Toolbox and Skill Library for use in Operations.



Building a Custom V+ Skill

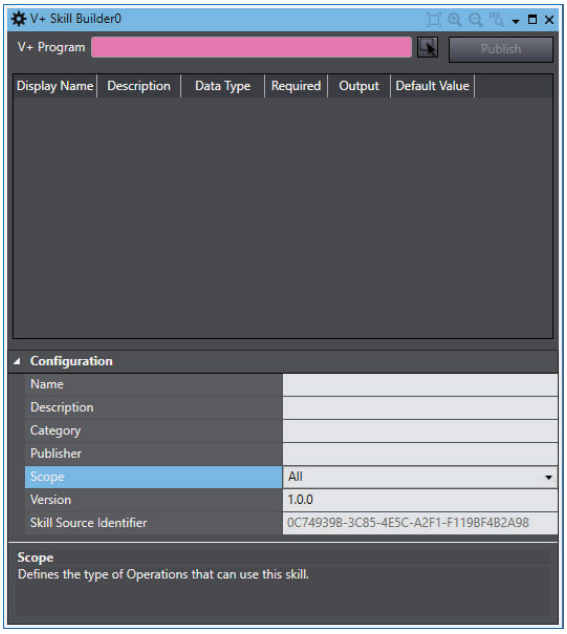
Use the following procedure to build a custom V+ Skill.
An existing V+ program will be used to create a custom V+ Skill. Make sure a V+ program exists for this purpose before beginning this procedure.
Refer to *Managing Custom Skills* on page 6-41 for information about importing, exporting, and removing custom Skills.

1 Right click the *Programs* object in the Multiview Explorer and select **V+ Skill Builder**.



The new object that was added can be manipulated by right-clicking it and making a selection from the menu.

2 Double click the new V+ Skill Builder object (or right click and select **Edit**) to access the V+ Skill Builder editor.



3 Select a pre-existing V+ program that the custom Skill will be based on.

4 Add a unique name for the new V+ Skill in the *Configuration* area of the editor.

5 Add optional Description, Category, and Publisher details in the *Configuration* area of the editor.
Custom Skills can be grouped in the Skill Library for organizational purposes using the *Category* field.

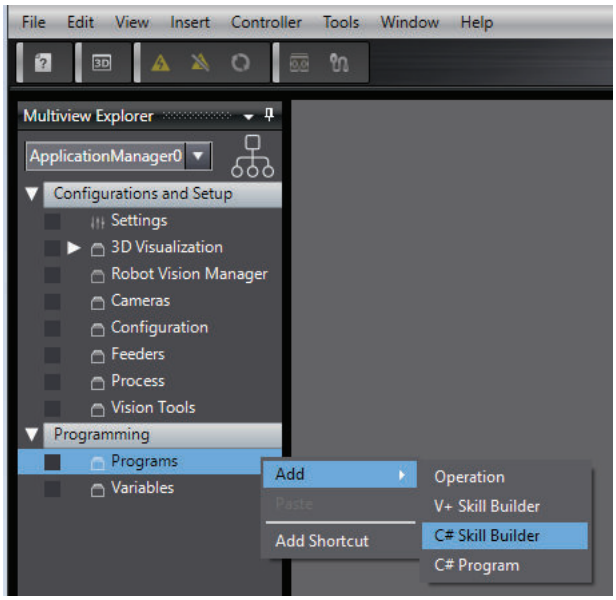
6 Adjust the Scope and Version if necessary.
The Scope field can dictate the use of the Skill to Robot Vision Manager Operations, Process Manager Operations, or all Operations.

7 Click the Publish Button to generate the new V+ Skill and register it in the Skill Library and Toolbox for use in Operations.
After the confirmation dialog box appears, click **OK** to acknowledge and complete this procedure.

Building a Custom C# Skill

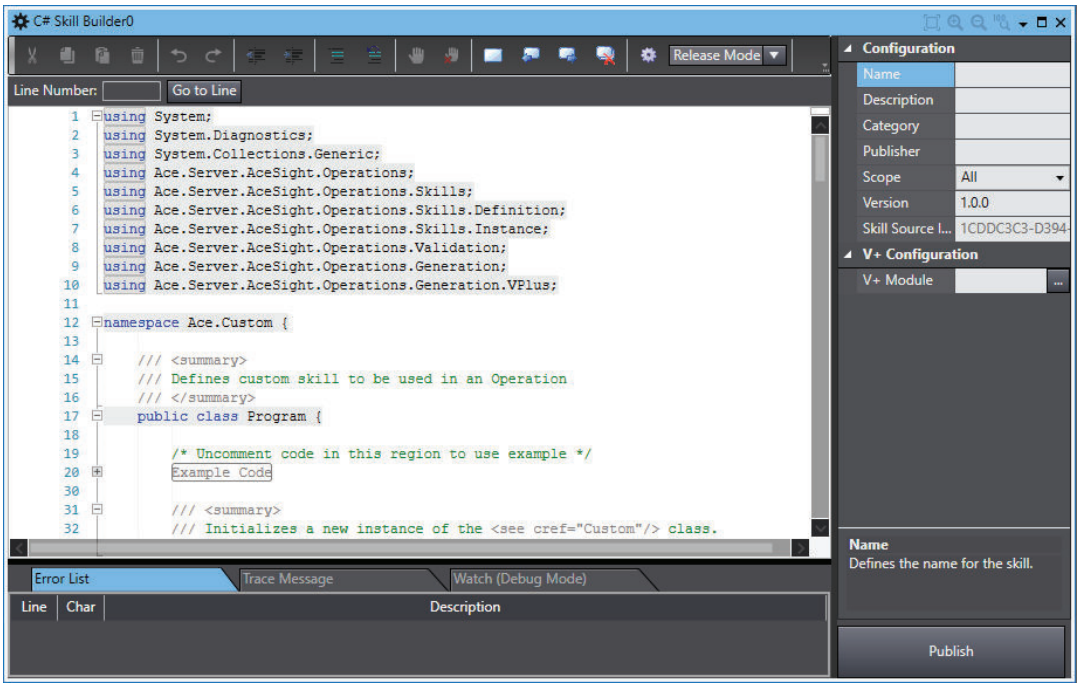
Use the following procedure to build a custom C# Skill.
Refer to *Managing Custom Skills* on page 6-41 for information about importing, exporting, and removing custom Skills.

1 Right click the *Programs* object in the Multiview Explorer and select **C# Skill Builder**.



The new object that was added can be manipulated by right-clicking it and making a selection from the menu.

2 Double click the new C# Skill Builder object (or right click and select **Edit**) to access the C# Skill Builder editor.



3 Use the editor to create a new C# program that the custom Skill will be based on.
A V+ Module can be used instead of a C# script. Select a reference to a V+ Module to enable this functionality.

4 Add a unique name for the new C# Skill in the *Configuration* area of the editor.

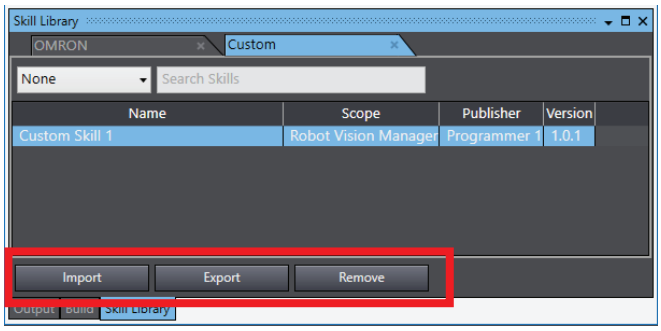
5 Add optional Description, Category, and Publisher details in the *Configuration* area of the editor.
Custom Skills can be grouped in the Skill Library for organizational purposes using the *Category* field.

6 Adjust the Scope and Version if necessary.
The Scope field can dictate the use of the Skill to Robot Vision Manager Operations, Process Manager Operations, or all Operations.

7 Click the Publish Button to generate the new C# Skill and register it in the Skill Library and Toolbox for use in Operations.
After the confirmation dialog box appears, click **OK** to acknowledge and complete this procedure.

Managing Custom Skills

Acess the Skill Library to manage custom Skills. Refer to *View Menu* on page 5-5 for more information.
Custom Skills can be imported, exported, and deleted using the buttons in the Skill Library.


Skill Library
OMRON
Custom
None
Search Skills
Name
Scope
Publisher
Version
Custom Skill 1
Robot Vision Manager
Programmer 1
1.0.1
Import
Export
Remove

7

Robot Controller Configuration and Setup

This section describes the configuration, objects, and programs used to create and install a robot controller.

7-1 Overview 7-2

7-2 Controller Settings 7-5

7-2-1 Configuration 7-5
7-2-2 Parameters 7-6
7-2-3 Control 7-7
7-2-4 Upgrade 7-9
7-2-5 Configure 7-14
7-2-6 Backup/Restore 7-31
7-2-7 Encoders 7-33

7-3 Save Configuration 7-35

7-4 Monitor Window 7-37

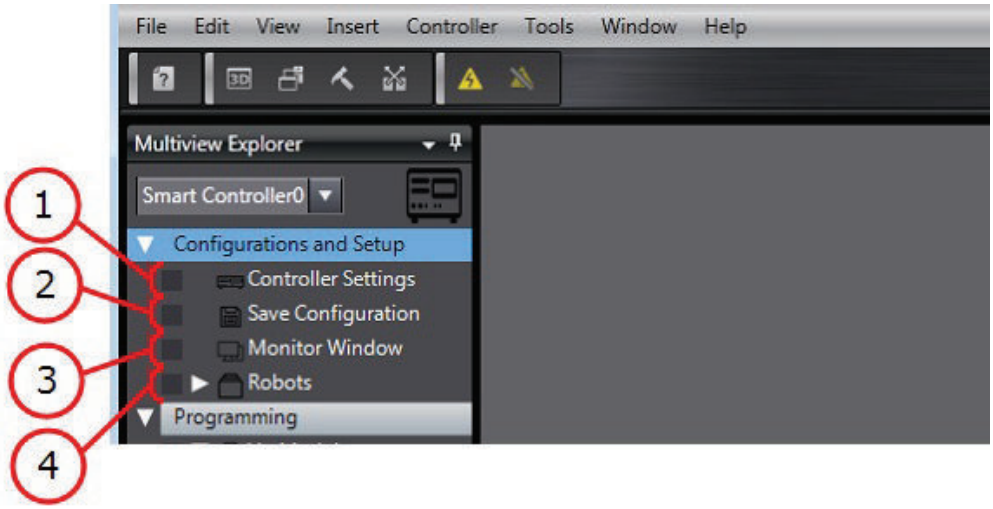
7-4-1 Using the Monitor Window 7-37
7-4-2 Monitor Command Entry 7-38
7-4-3 Monitor Command Keyword Parameters 7-38
7-4-4 Canceling a Monitor Command 7-39
7-4-5 Executing a Monitor Command 7-39

7-5 Robot Objects 7-40

7-5-1 Configure Robots 7-42
7-5-2 I/O EndEffectors 7-43
7-5-3 3D Visualization 7-46
7-5-4 Configuration 7-47
7-5-5 End-Effector 7-50
7-5-6 Location 7-51
7-5-7 Object 7-51
7-5-8 Configure 7-51
7-5-9 Control 7-56

7-1 Overview

The following items are available for the configuration and setup of a robot controller.
Use the Multiview Explorer to access these items as shown below.



Additional Information

These items do not apply when viewing an Application Manager device in the ACE project.

Item	Name	Contents	Reference
1	Controller Settings	Configuration	Refer to *7-2-1 Configuration* on page 7-5 for more information.
		Parameters	Refer to *7-2-2 Parameters* on page 7-6 for more information.
		Control • Set Time • Reboot V+ • Servo Reset • Zero Memory • Save Startup Specifications • Save Memory to a File • View Error Log	Refer to *7-2-3 Control* on page 7-7 for more information.
		Upgrade • Upgrading V+ • Upgrading FireWire Firmware	Refer to *7-2-4 Upgrade* on page 7-9 for more information.
		Configure • Configuring Licenses • Configuring Robot • Configuring Device Nodes • Configure I/O • Configure Belt Encoder Latches • Configure Robot Position Latches • Configure System Settings	Refer to *7-2-5 Configure* on page 7-14 for more information.
		Backup/Restore • Backup V+ • Restore V+ • Compare V+ to Backup	Refer to *7-2-6 Backup/Restore* on page 7-31 for more information.
		Encoders	Refer to *7-2-7 Encoders* on page 7-33 for more information.
2	Save Configuration		Refer to *7-3 Save Configuration* on page 7-35 for more information.
3	Monitor Window		Refer to *7-4 Monitor Window* on page 7-37 for more information.

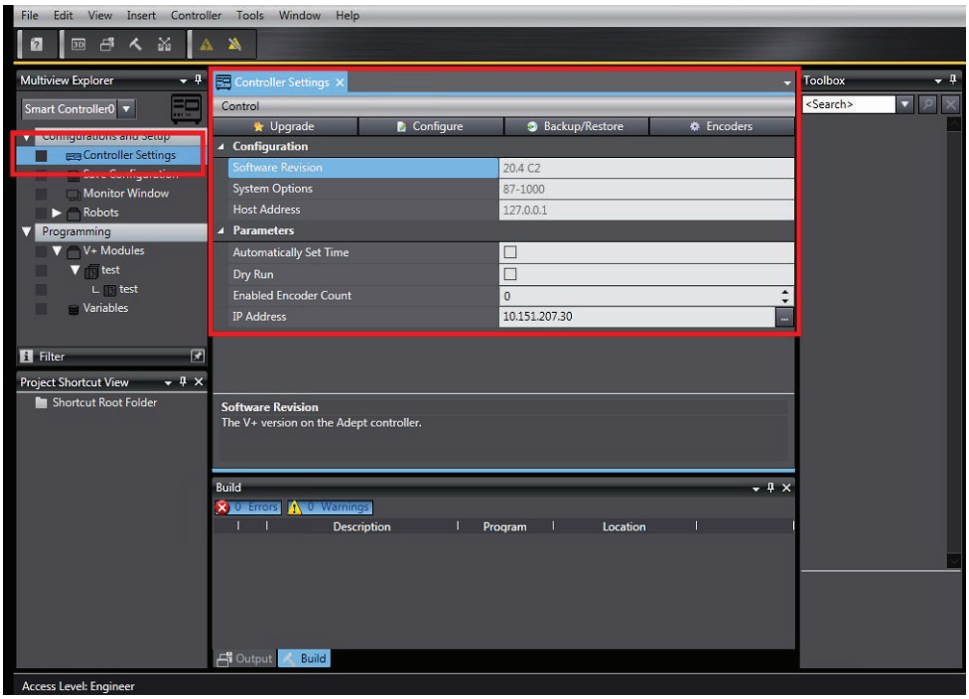
Item	Name	Contents	Reference
4	Robot Objects	3D Visualization	Refer to *7-5-3 3D Visualization* on page 7-46 for more information.
		Configuration	Refer to *7-5-4 Configuration* on page 7-47 for more information.
		I/O EndEffector	Refer to *7-5-2 I/O EndEffectors* on page 7-43 for more information.
		Location	Refer to *7-5-6 Location* on page 7-51 for more information.
		Object (Menu Item) • Expert Access	Refer to *7-5-7 Object* on page 7-51 for more information.
		Configure (Menu Item) • Save Startup Specifications • Load Spec File • Save Spec File • Axes, Options and Kinematics • Obstacles • S-Curve Profiles • Safety Settings	Refer to *7-5-8 Configure* on page 7-51 for more information.
		Control (Menu Item) • Hardware Diagnostics • Data Collection • Motor Tuning	Refer to *7-5-9 Control* on page 7-56 for more information.

7-2 Controller Settings

The Controller Settings item in the Multiview Explorer provides tools for installing and configuring the system. Use the information below to understand the functions of the Controller Settings area.
When Emulation Mode is enabled, some controller settings are not available. Refer to *Emulation Mode Differences* on page 1-9 for more information.

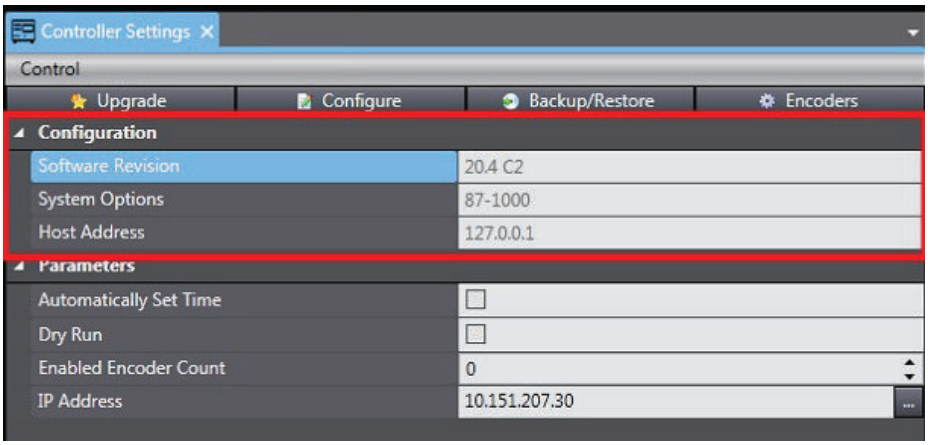
Additional Information

Many Controller Settings items are not available while offline.



7-2-1 Configuration

The Configuration area is found in the main view of the Controller Settings editor area. It contains the following read-only items for informational purposes.



Software Revision

The Software Revision field displays the V+ software revision number installed on the robot controller. This is displayed only while online with the controller.

System Options

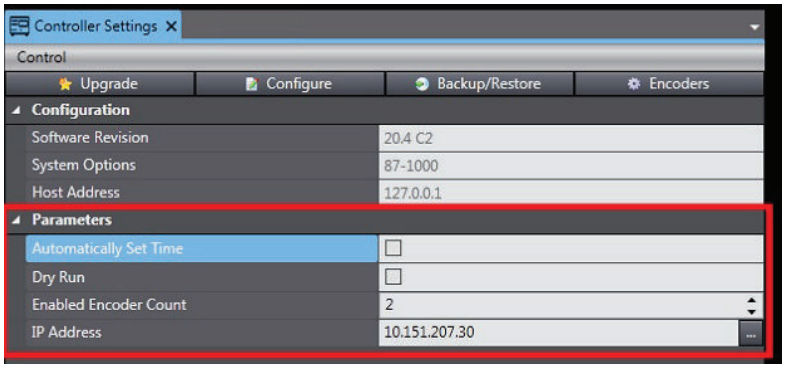
The System Options field displays the configuration of the V+ system in a two-word, hexadecimal format. This is displayed only while online with the controller.

Host Address

The Host Address field displays the IP address of the connected PC (host). This is displayed only while online with the controller.

7-2-2 Parameters

The Parameters area is found in the main view of the Controller Settings editor area. It contains the following items that are described below.



Automatically Set Time

Select *Automatically Set Time* to automatically synchronize the controller to the PC time.

Dry Run

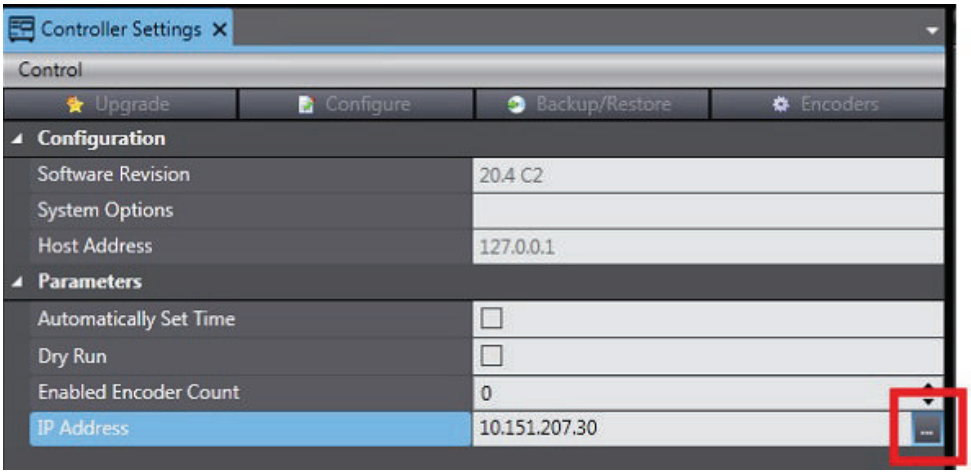
Select *Dry Run* to test a project without robot motion. This will cause robots to ignore motion commands. This is selectable only while online with the controller.

Enabled Encoder Count

Enabled Encoder Count displays the number of external encoders used by the system. The number displayed is related to the number of Belt Encoder Channels configured (refer to *Configure Belt Encoder Latches* on page 7-21 for more information).

IP Address

The IP Address field displays the current IP address of the controller. When offline, it is possible to change or select a controller's IP address by clicking the **More** Button as shown below. Clicking this button will display the *Select a Controller Address* Dialog Box.

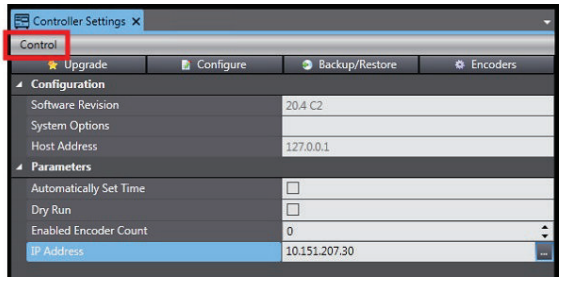


Additional Information

- If the desired controller is not available or the changed IP address will not allow an online connection, refer to *4-1-2 Detect and Configure Controller* on page 4-4 for more information.
- The controller IP address is also accessible in the configuration area. Refer to *Configure System Settings* on page 7-23 for more information.

7-2-3 Control

The Control Menu displays controller-specific setting items described below. These are selectable only while online with the controller.



Additional Information

When using ACE Controller Settings with a stand alone robot, the *Systems Options* line in the Configuration view will not be shown.

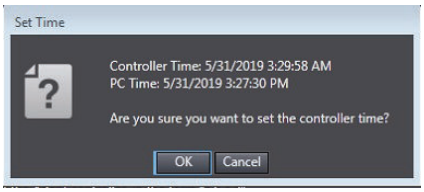
Set Time

In order to use error log time stamps for troubleshooting, the correct time needs to be set in the controller(s).

Selecting **Set Time** will manually synchronize the controller time to match the time detected on the connected PC. The following confirmation dialog box will be displayed.

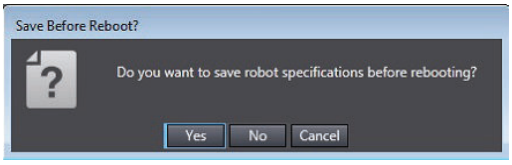
Additional Information

The Set Time function will not set the time for a robot node. Use the device node configuration to set robot node times. Refer to *Configure Device Nodes* on page 7-15 for more information.



Reboot V+

Selecting **Reboot V+** will reboot the controller. The following confirmation dialog box will be displayed.

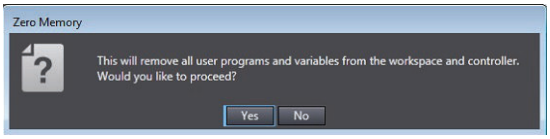


Servo Reset

Selecting **Servo Reset** will reset communication with the robot servo nodes.

Zero Memory

Selecting **Zero Memory** will clear all user programs and variables from the workspace and the controller. The following confirmation dialog box will be displayed.



Save Startup Specifications

Selecting **Save Startup Specifications** will save all robot and motor specifications to the V+ boot disk.

Additional Information

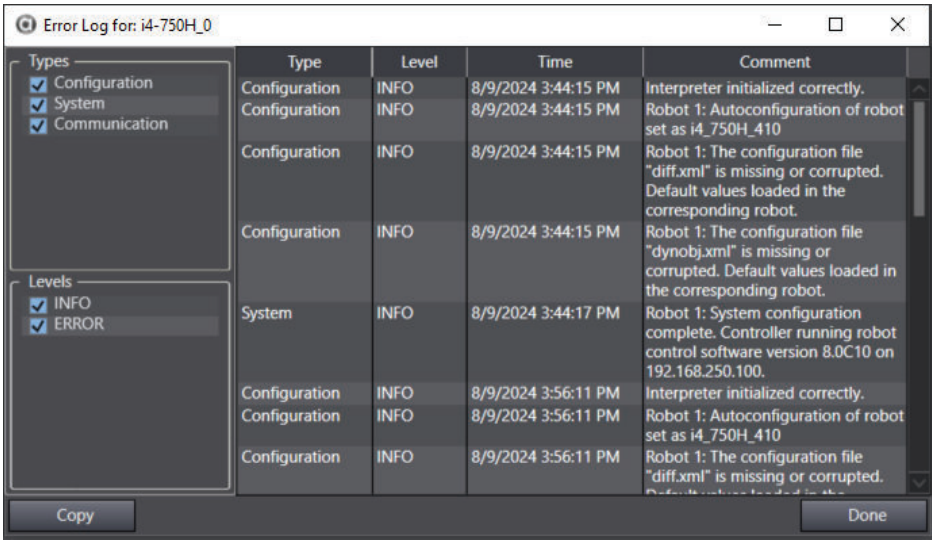
This is the same function that is present in the *Robot Object - Configure* menu. Refer to *7-2-3 Control* on page 7-7 for more information.

Save Memory to a File

Selecting **Save Memory to a File** will save all V+ programs to a file in the PC. A *Save As* Window will be displayed.

View Error Log

Selecting **View Error Log** will display a history of robot controller events. Refer to *9-3 Error Log* on page 9-5 for more information.



Additional Information

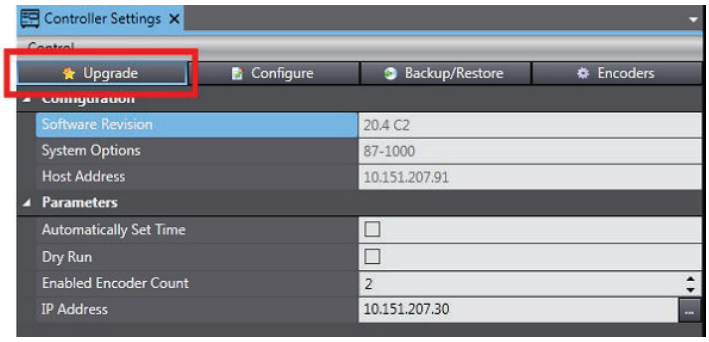
The Error Log is also accessible by right-clicking **Controller Settings** in the Multiview Explorer.

7-2-4 Upgrade

Use the Upgrade function to upgrade the V+ firmware on the robot controller or the FireWire node firmware.

To acess the Upgrade function left click Controller Settings and select Edit. This will open the Controller Settings window

Click the Upgrade Tab to display the *Upgrade Options* Dialog Box.



Additional Information

- The Upgrade function is not available in Emulation Mode.
- Upgrade functions are also available by right-clicking **Controller Settings** in the Multiview Explorer.
- The option to upgrade the FireWire node firmware is only available for the SCEX, eIAB, and eMB robot controllers. Refer to *FireWire Node Upgrade Procedure* on page 7-13 for more information.

V+ Upgrade Procedure

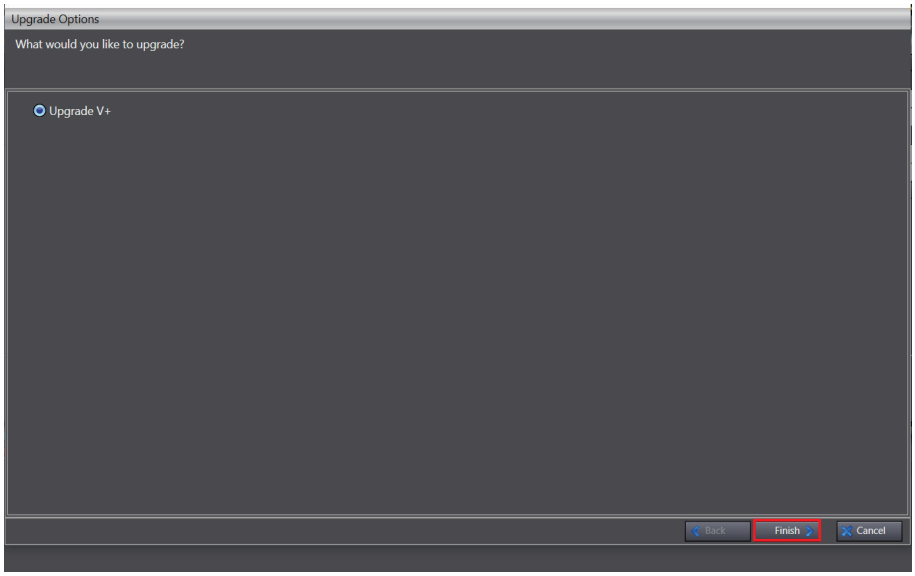
The *Upgrade V+* function is used to upgrade the V+ firmware on the robot controller.
A V+ distribution includes all compatible firmware files for robot nodes and the T20 Teach Pendant.

Precautions for Correct Use

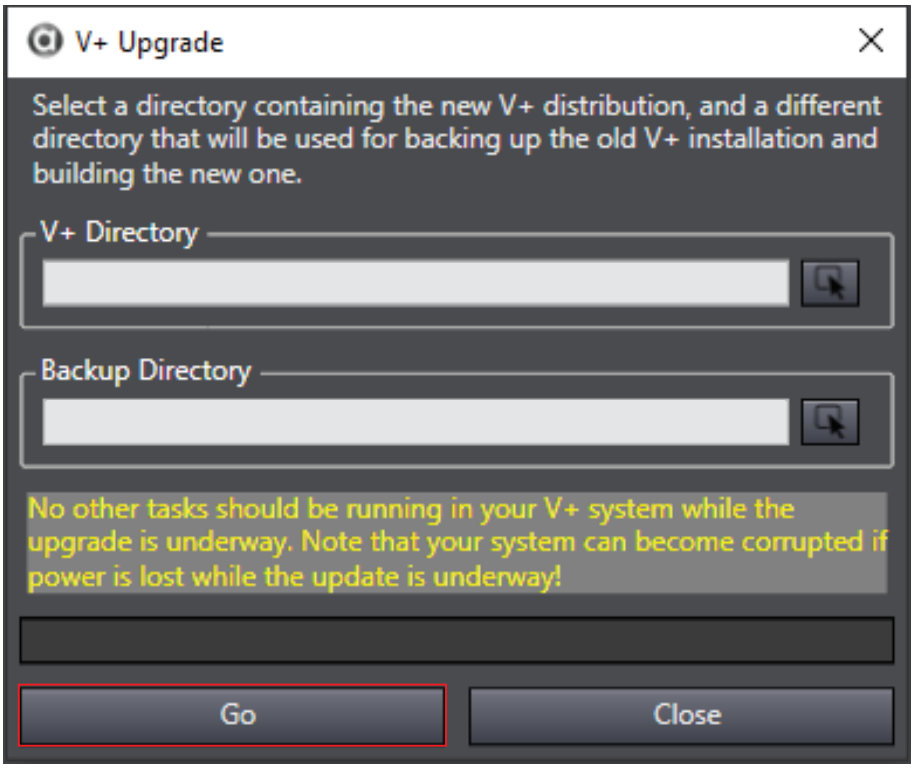
- No other tasks should be running while a V+ upgrade is underway.
- The system can become corrupted if power is lost while a V+ upgrade is underway.

Use the following procedure to upgrade V+ firmware

1 Select *Upgrade V+* in the *Upgrade Options* Dialog Box and then click the *Finish* Button.



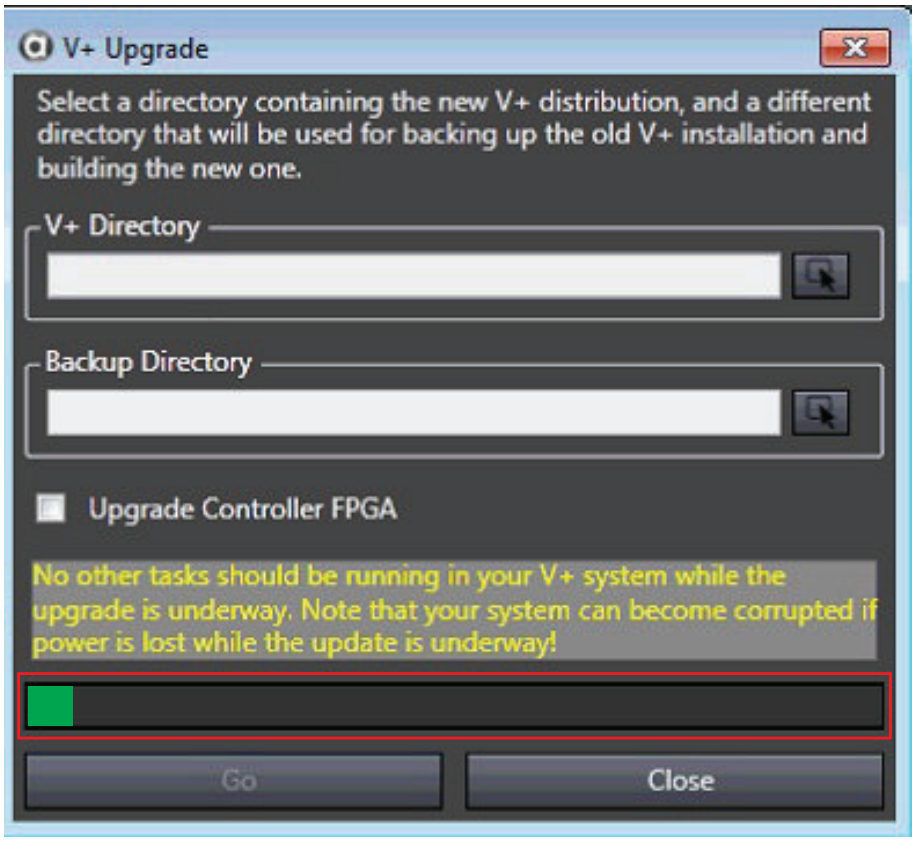
2 The *V+ Upgrade* window will appear. Select a file path for both the *V+ Directory* and the *Backup Directory*. After the *V+ Directory* and the *Backup Directory* fields are selected, click the *Go* Button to proceed with the upgrade process.



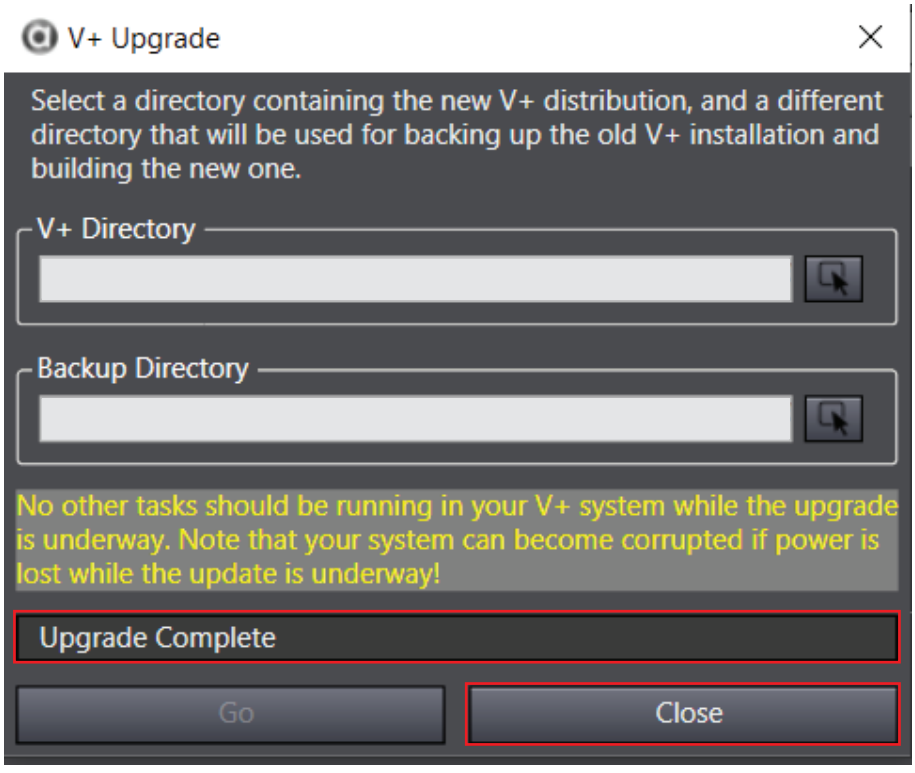
Additional Information

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.

Ensure no other tasks are running on the system while the upgrade is underway. The process can take several minutes to complete. A progress bar is displayed while the system is being updated.

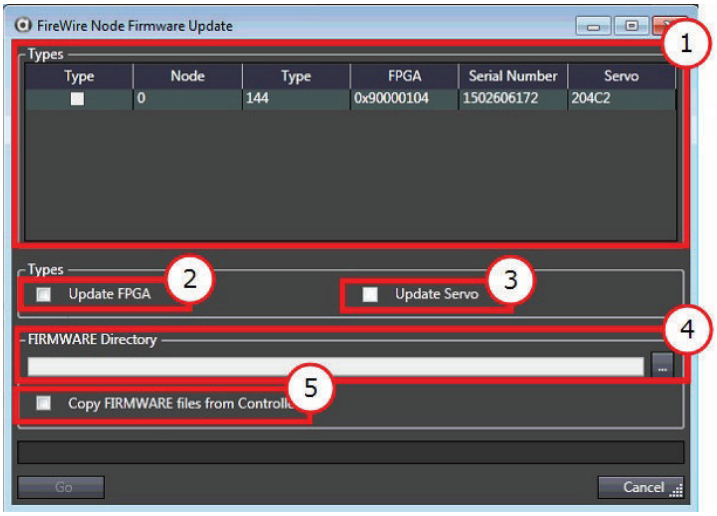


3 Once the upgrade is finished the *V+ Upgrade* window will display the message *Upgrade Complete*. Click the *Close* Button to close the *V+ Upgrade* window. The firmware upgrade procedure is now complete.



FireWire Node Upgrade Procedure

The Upgrade FireWire Firmware function is used to upgrade a FireWire node's firmware. Selecting **Upgrade FireWire Firmware** from the *Upgrade Options* Dialog Box and then clicking the **Finish** Button will display the following window.



Item	Description	Details
1	FireWire Node Identification and Selection Area	Select nodes to be updated.
2	Update FPGA	Updates the firmware for the FireWire node.
3	Update Servo	Enable to update the servo binary image stored in the selected node's flash memory. Not performing this relies on successful dynamic download from controller to robot during boot up (see note below). Performing the servo update eliminates any potential risk of dynamic download failure during boot up.
4	Firmware Directory	Select the PC directory where the new firmware files are located. Provide a file path to the \SYSTEM folder of the distribution file directory.
5	Copy FIRMWARE files from Controller	Use firmware files in controller for FireWire node updates.

Use the following procedure to upgrade a FireWire node.

Additional Information

- The firmware upgrade process will require several minutes for each node in the distributed network.
- During boot, a SmartController will attempt a dynamic download to robot nodes to give them a compatible servo version (stored in RAM), if the compatible version is not already present in RAM. Updating servo firmware with this method removes the risk of issues arising if dynamic download fails during boot up.

1. Select *Upgrade FireWire Firmware* in the *Upgrade Options* Dialog Box and then click the *Finish* Button.

2. Select all detected robot amplifier nodes. This process is used to check and update servo code firmware on robot nodes. If a newer version is not provided, an upgrade will not occur. Alternatively, select only the node to upgrade or downgrade with a specific file from the PC.

3. Select *Update FPGA* and *Update Servo* options.

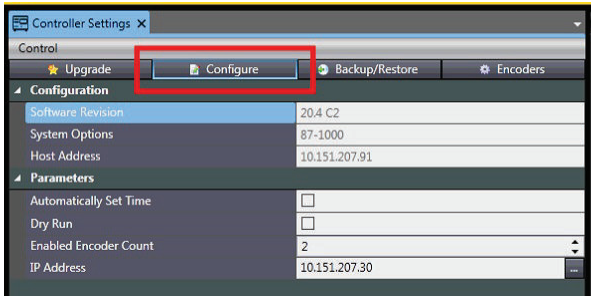
4. Select the source of the firmware update files from either the PC or the controller. If the update files are located on the PC, provide the firmware directory where the new firmware distribution is located. Provide a file path to the \SYSTEM folder of the distribution file directory.Do not select the option for *Copy FIRMWARE* files from controller in this case. If the update files are located on the controller, select the option for *Copy FIRMWARE files from controller*. The *FIRMWARE* directory field is not accessible or necessary in this case.

5. After all fields and selections are made, click the *Go* Button to proceed with the update process. The process can take several minutes to complete. A progress bar is displayed during the upgrade procedure.

6. After updating the Servo / FPGA, a prompt opens, requesting a power cycle for the 24 VDC power to the updated nodes. The update process is complete when the power is cycled.

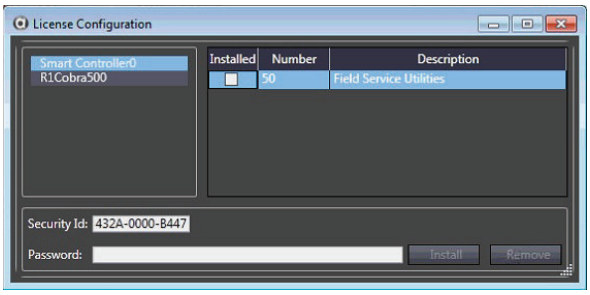
7-2-5 Configure

Clicking the **Configure** Button displays the *Configure Options* Dialog Box. Several configuration items are selectable as described below. Configuration items that appear will be specific to the robot controller type that is connected. The **Configure** Button is available only while online with the controller.



Configure Licenses

Select **Configure Licenses** to view, install, or remove V+ licenses. Each license is uniquely paired to a corresponding Security ID and every device has a unique Security ID number associated with the memory card.
To install or remove a license, first select the device from the left side list, enter the license key in the Password field, and then click either the **Install** or **Remove** Buttons.



Configure Device Nodes

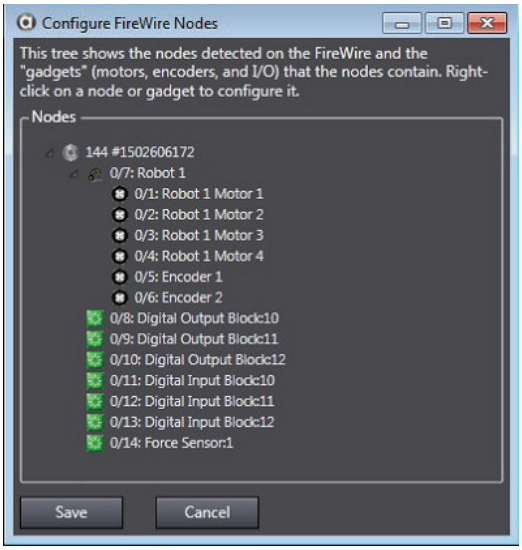
Select **Configure Device Nodes** to create a valid device node network configuration to enable communication with devices such as robots, belt encoders, digital I/O, and force sensors. Selecting **Configure Device Nodes** and clicking the **Finish** Button will open the dialog box shown below.

Precautions for Correct Use

If a device node network configuration is modified and saved, the physical connections will be identified and restored accordingly on the next boot.

Additional Information

Always review the initial device node configuration to ensure it is compatible and appropriate for the application. When connecting to a controller, ACE software will check for device node configuration conflicts. If a conflict is present, error messages will be displayed. Refer to *9-4 Device Node Event Log* on page 9-7 for more information.



Right-click an item in the list to display information and settings for each device node. The following information and settings are available.These may vary depending on the device type.

- View and clear device node logs for controllers
- Set a robot node clock
- View a robot node's information
- Unconfigure or designate a robot's device node number
- Designate encoder device node numbers
- Designate force sensor device node numbers

- Configure I/O Block address ranges

Configure I/O for DeviceNet

Select **Configure I/O** to map I/O to numerical signal numbers for use in programming. The Configure I/O function provides DeviceNet scanning, configuration, and mapping.

Additional Information

Device nodes must be configured before attempting to map I/O to numerical signal numbers.

DeviceNet Configuration

The following steps are used for configuring DeviceNet I/O with ACE 4. The required equipment for DeviceNet I/O configuration is:

- SmartController EX
- Slave DeviceNet Device

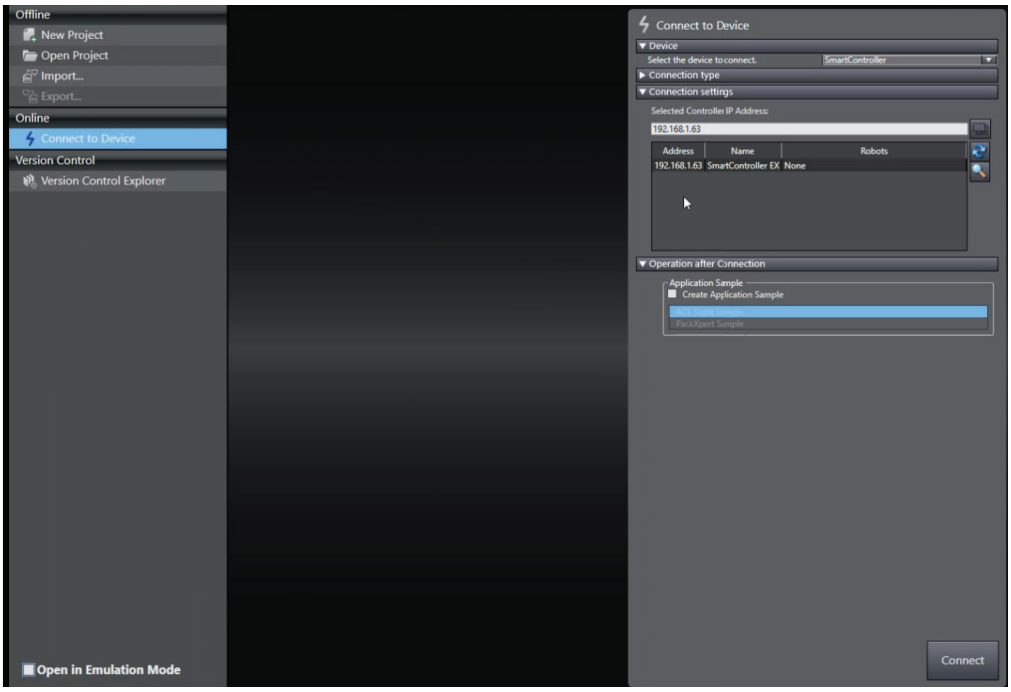
Confirm that the SmartController EX and slave DeviceNet devices are connected according to the DeviceNet topology. For more information, see the ODVA website and download the DeviceNet Planning and Installation Manual.

Once all the components are installed and connected, use ACE 4 and connect to the SmartController EX.

Additional Information

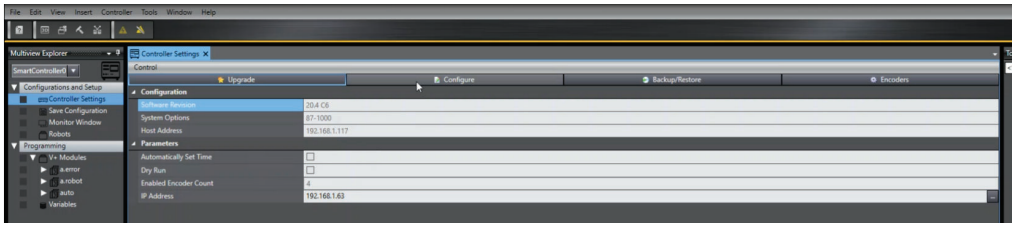
The configuration cannot be done in Emulation Mode.

1 Open Multiview Explorer and click **Connect to Device** in the left-side panel. On the right-side panel, click the controller address to connect to, as shown in the following figure.



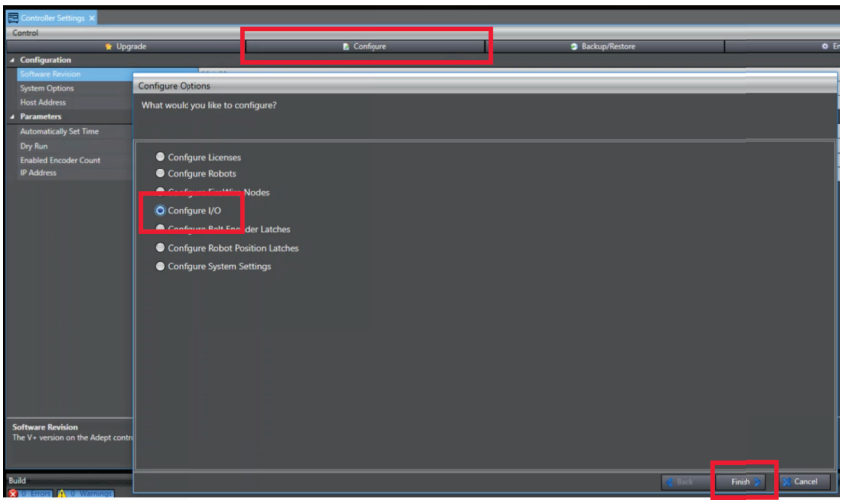
2 Click the **Connect** Button. Multiview Explorer shows the connection to the device has been established.

3 Expand the *Configurations and Setup* drop down menu, and select **Controller Settings** to display Controller Settings window.



4 Click the **Configure** Button on the Controller Settings window.

5 When the panel opens, select the **Configure I/O** option and then click **Finish** Button. This starts the Configure I/O sequence



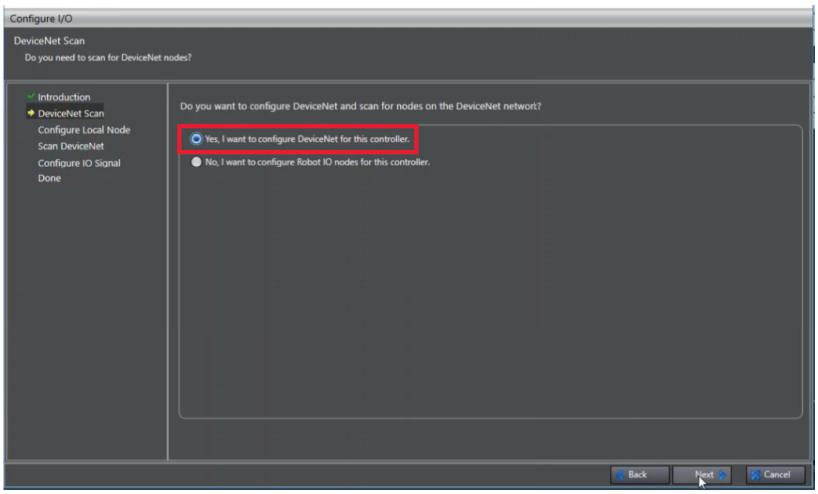
6 When the *Introduction* pop-up appears, click **Next** Button.

7 At the DeviceNet Scan step, select **Yes, I want to configure DeviceNet for this controller** then click the **Next** Button

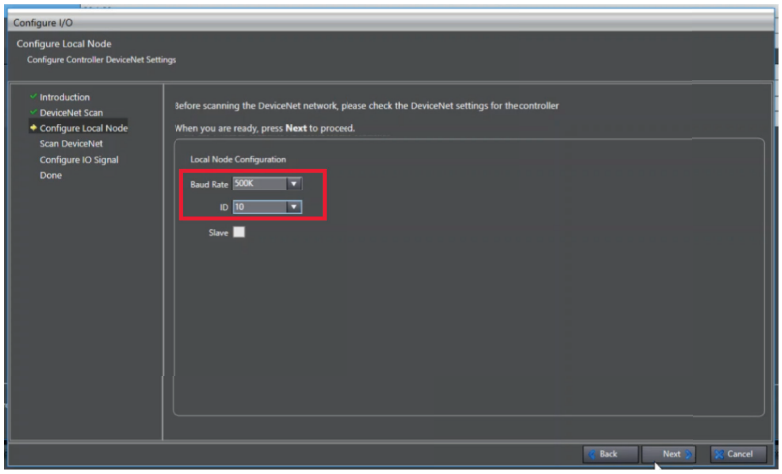


Additional Information

The second option in the following figure, **No, I want to configure Robot IO nodes for this controller**, is used for configuring SDIO and IOBlox signals ranges, if the default configuration is not sufficient.

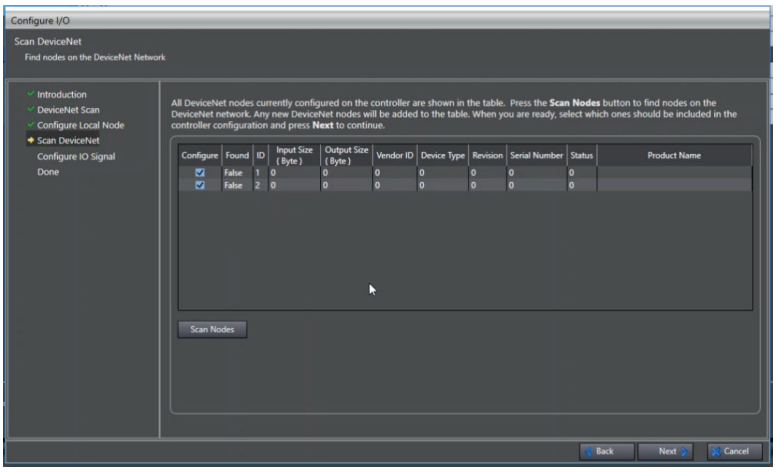


8 The panel opens the *Configure Local Node* step for selecting the network Baud Rate and the SmartController EX node ID. In the example, the Baud Rate is 500K and the SmartController EX ID is 10. Set the configuration for the node, and click the **Next** Button.

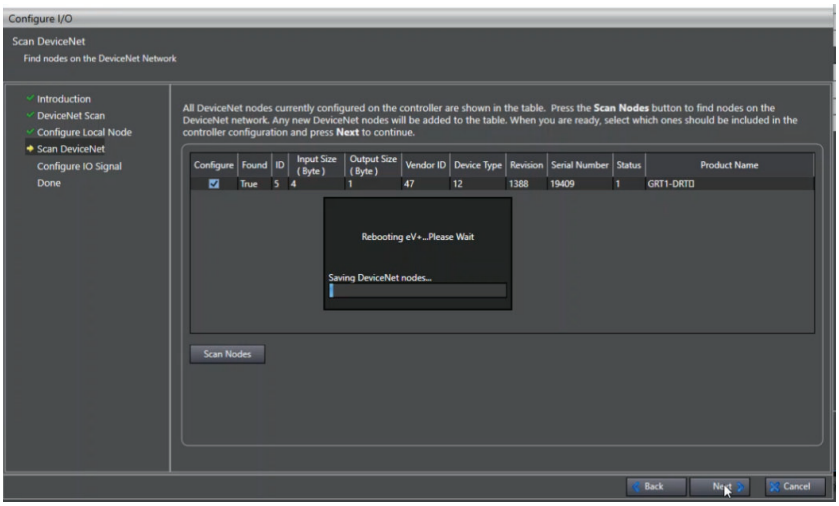


9 If the SmartController EX had an existing network configuration, the selections will be displayed. If it is the first time the configuration has been entered, a pop-up message requesting a reboot will appear. Click the **Yes** Button to continue.

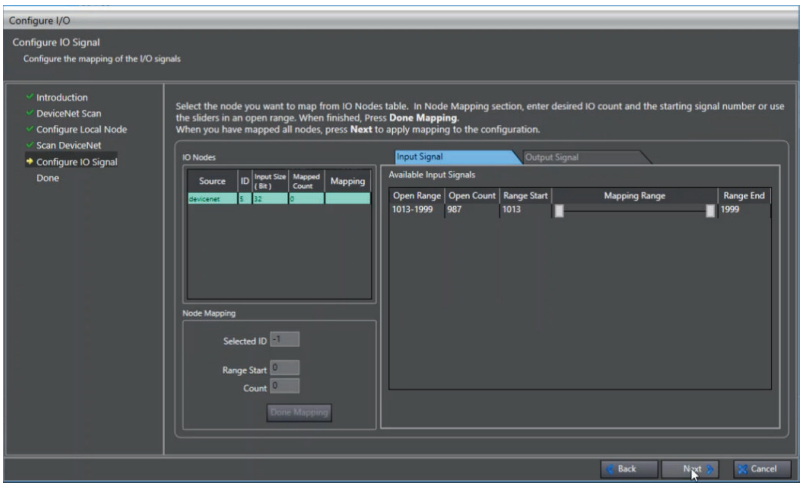
10 Once the reboot is completed, a default list with node IDs will appear. Click the **Scan Nodes** Button to start scanning the network. Scanning the nodes may require a few minutes. When the scan completes a panel opens showing the discovered nodes.



11 To add a device in the robot I/O configuration, check the **Configure** box for the device and then click the **Next** Button. The SmartController EX will reboot.

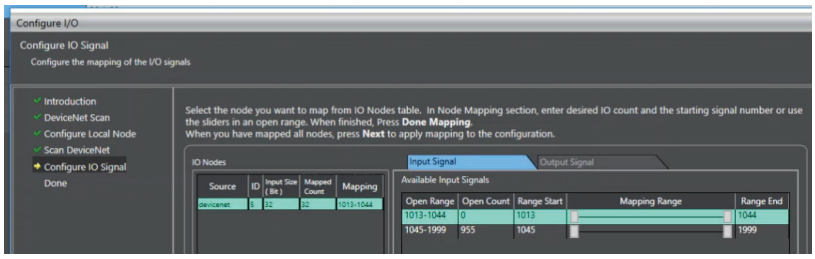


12 When the reboot completes the *Configure IO Signal* windows opens and shows all the selected Devices and IDs. Configure the inputs and outputs to the robot I/O.



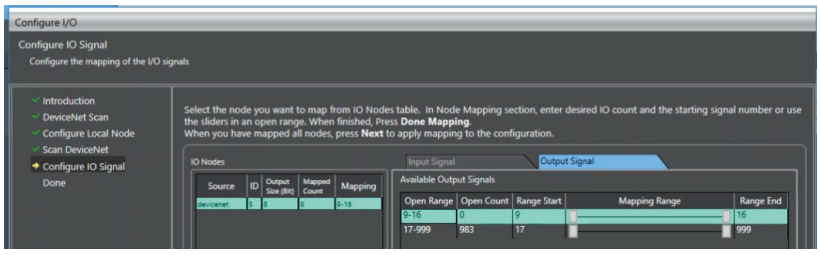
13 Configure the inputs, by locating the device in the I/O Nodes windows and clicking on the line. The Selected ID then appears in the *Node Mapping*.

14 Click the **Open Range** line to select the Range Start. If the Range Start is not what is desired, enter a different signal number. Next, enter the Count of bits that will be used. Usually, it is equal to the Input Size (Bit). When all values are entered, click on the **Done Mapping** Button. A new line with the same color as the device will appear on the *Available Input Signals* list.

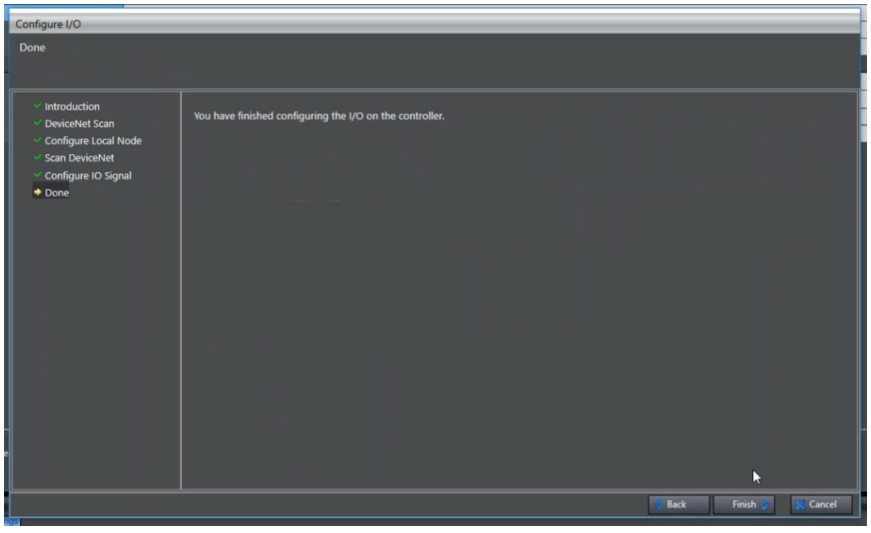


15 Click the **Output Signal** tab. Configure the outputs, by locating the device in the I/O Nodes windows and clicking on the line. The Selected ID then appears in the *Node Mapping*.

16 Click the **Open Range** line to select the Range Start. If the Range Start is not what is desired, enter a different signal number. Next, enter the Count of bits that will be used. Usually, it is equal to the Input Size (Bit). When all values are entered, click on the **Done Mapping** Button. A new line with the same color as the device will appear on the *Available Output Signals* list.



17 Once all Input and Output are configured, click the **Next** Button to reboot the Controller and apply the updates to the I/O configuration. When the reboot finishes, click the **Finish** Button to close the *Configure IO* window.



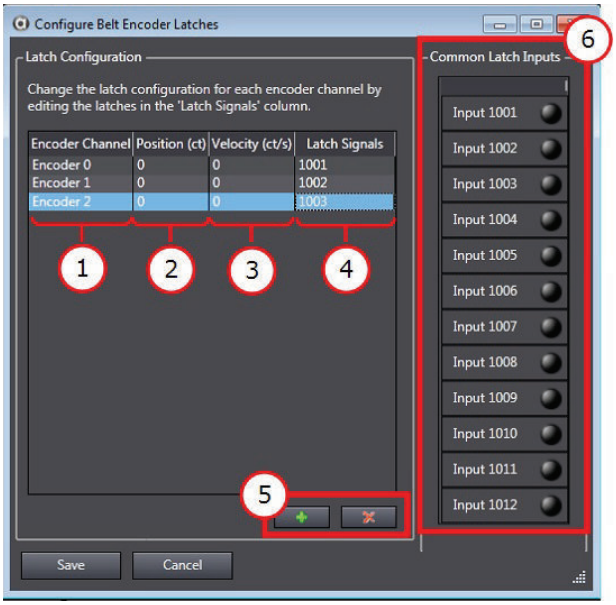
Configure Belt Encoder Latches

Select **Configure Belt Encoder Latches** to view and change latch signals for each encoder channel of the controller. A Belt Encoder Latch refers to the capture of a conveyor belt encoder position value when an input signal (latch) changes state. Once configured, the system will monitor and record the latch signal number and corresponding encoder position (one encoder value) in a first-in-first-out buffer for later access and use in a program. This area can also be used to check current position and velocity for each encoder channel.

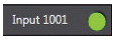
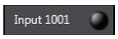
Additional Information

- Belt encoder latches are not required for conveyor belt tracking but are recommended for positional accuracy. Pack Manager requires use of latch signals for all belt tracking applications for positional accuracy, as well as cross-controller latch handling.
- Encoders should be uniquely numbered in the device node configuration before configuring encoder channels/latches.

Selecting **Configure Belt Encoder Latches** and clicking the **Finish** Button will display the dialog box shown below.

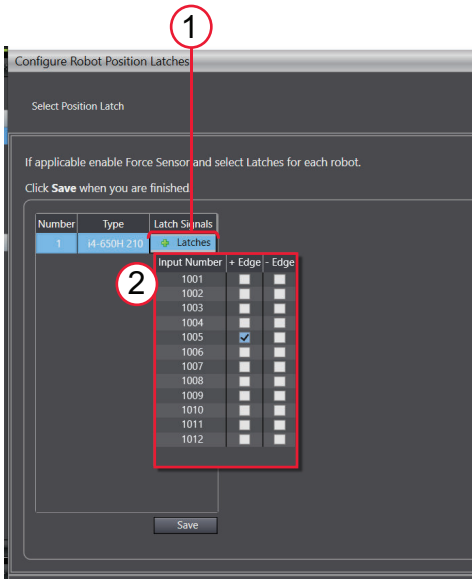


Item	Name	Description
1	Encoder Channel	Shows the belt encoder channels for the system.
2	Position (ct)	Shows the current position in counts of the corresponding belt encoder.
3	Velocity (ct/s)	Shows the current velocity in counts per second for the corresponding belt encoder.

Item	Name	Description
4	Latch Signals	Shows the latch signal assignments for the corresponding belt encoder. One or more signals can be configured as described below. Use the minus sign to designate a falling edge condition: • 1001: When signal 1001 changes from OFF to ON, the encoder channel position data will be latched. • -1001: When signal 1001 changes from ON to OFF, the encoder channel position data will be latched. Use a space between signals to configure multiple signals: • 1001 1004: When either signal 1001 or 1004 changes from OFF to ON, the encoder channel position data will be latched. • -1001 1003: When signal 1001 changes from ON to OFF or signal 1003 changes from OFF to ON, the encoder channel position data will be latched.
5	Add/Remove Encoder Channels	Add or remove an encoder channel from the configuration. If an encoder channel is not configured here, that encoder input will not be accessible in V+ programs or be visible in other Encoder windows in the project. Similarly, if extra encoder channels are present here, they will not provide input. A correct configuration will allocate one encoder channel for each belt encoder on the device node network.
6	Common Latch Inputs	Displays the state of available latch input signals. • ON:  • OFF:  Input signals 1001-1008 are the position latch signals for e-Series controllers on the XIO terminal block. Input signals 1001-1012 are the position latch signals for the SmartControllerEX on the XDIO terminal block. These signals are also used for Robot Position Latches (refer to *7-5-8 Configure* on page 7-51 for more information).

Configure Robot Position Latches

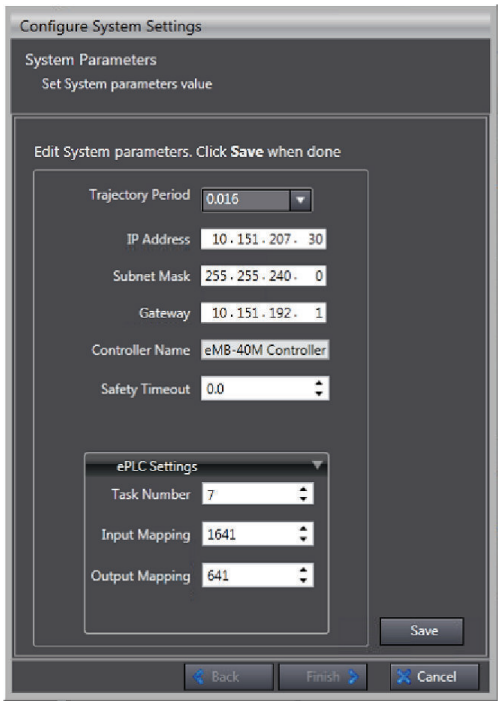
Select **Configure Robot Position Latches** and click **Finish**, to view and change latch signals for each encoder channel of the controller. A Robot Position Latch refers to the capture of a robot's position when an input signal (latch) changes state. A robot's position is captured as a precision point, which is a data structure containing one value for each joint of the robot. Once configured, the system will monitor and record the latch signal number and corresponding robot joint positions in a first-in-first-out buffer for later access and in a program. Robot position latches are most frequently used for applications that require vision-guided position refinement without stopping robot motion.



Item	Name	Description
1	*Add Latches* Button	Click the button to display a list of available input signals to trigger the Robot Position Latch.
2	Latch Type	Select rising edge + *Edge* or falling edge - *Edge* as the latch type for each *Input Number*.

Configure System Settings

Selecting **Configure System Settings** and clicking the **Finish** Button will display the dialog box shown below.



The following items are accessible in this area.

● Trajectory Period

The trajectory period defines the interval between set points calculated during robot motion trajectory generation. The default setting of 16 ms is adequate for most applications because the servo loop receives set points and controls motor position based on micro-interpolation at 8 khz (125 µs). In some situations it is helpful to decrease trajectory period, which results in more frequent set point generation to decrease path following error. However, reducing this value can have a noticeable impact on quantity of calculations and processor usage, especially for embedded controllers.

Additional Information

Unnecessary decrease of trajectory period may consume processing time needed for user programs running in the controller. Refer to *5-9-10 Profiler* on page 5-48 for more information

● IP Address

View and change the IP address of the connected controller.

● Subnet Mask

View and change the subnet mask of the connected controller.

● Gateway

View and change the Gateway of the connected controller.

● Controller Name

View and change the connected controller's name.

● Safety Timeout

View and change the Safety Timeout setting for the connected controller. This controls the behavior of the robot high power request sent by a connected PC, executing program, Front Panel High

Power Enable button, or optional pendant. If a safety timeout is enabled, the robot power button on the front panel will blink for a specified amount of seconds after a high power request is made. If the robot power button is not pressed within the specified time, a safety timeout occurs and robot power is not applied.

Precautions for Safe Use

Ensure adequate safety measures are taken if the safety timeout is disabled.

The default setting for the Safety Timeout is 10 seconds. Use the following settings to adjust the Safety Timeout.

- 0 seconds: disables the high power request secondary action and robot power is applied immediately.
- 1 to 60 seconds: enables the high power request secondary action and robot power is applied if the Robot Power button is pressed on the front panel within the specified amount of time.

Precautions for Safe Use

Robot maintenance and troubleshooting procedures require the safety timeout to be enabled. The robot's flashing high power indicator is used to verify that the lamp is not burned out and this should be considered before disabling the safety timeout feature.

● Auto Start

Enable or disable an Auto Start program. If Auto Start is enabled, as the controller completes boot process it will load and execute D:\AUTO.V2. An AUTO.V2 program has no user-defined error handling and should be kept simple to load modules or variable files and execute a program that handles system startup. Refer to *7-3 Save Configuration* on page 7-35 for more information.

● Task Number (ePLC Settings)

View and change the Task Number used for an ePLC configuration. This defines the first task that ePLC programs will start executing on when Auto Start is enabled (eplc_autostart). ePLC task numbers must be between 7 and 15 and will occupy at least 5 tasks when Auto Start is enabled.

Additional Information

ePLC settings typically do not need to be changed after Auto Start is enabled.

● Input Mapping (ePLC Settings)

View and change the Input Mapping used for an ePLC configuration. This defines the first input signal in the range of signals mapped by ePLC programs.

● Output Mapping (ePLC Settings)

View and change the Output Mapping for an ePLC configuration. This defines the first output signal in the range of signals mapped by ePLC programs.

● Auto Start (ePLC Settings)

Select the Auto Start behavior when using the ePLC function. When Auto Start is enabled, D:\ADEPT\UTIL\ePLC3.v2 will be loaded and executed on the designated task number and initialized with the Input and Output Mapping signal numbers.

Configure PROFINET Settings

Use the **Configure PROFINET Settings** area to add, edit, or delete variables associated with PROFINET data exchange.

Set IO Type to I for input data from the PROFINET Controller. Set IO Type to O for output data from the robot controller PROFINET device.

Selecting **Configure PROFINET Settings** and clicking the **Finish** Button will display the dialog box shown below. This area can also be accessed by double-clicking the PROFINET label or status indicator color in the *Task Status Control* area. Refer to *5-9 Task Status Control* on page 5-40 for more information.

The following PROFINET configuration and information is available in this area.

- Enable or disable PROFINET fieldbus communication.
- PROFINET module V+ variable data exchange configuration.
- Robot behavior settings in the event of PROFINET communication errors.
- Import and Export configuration information.



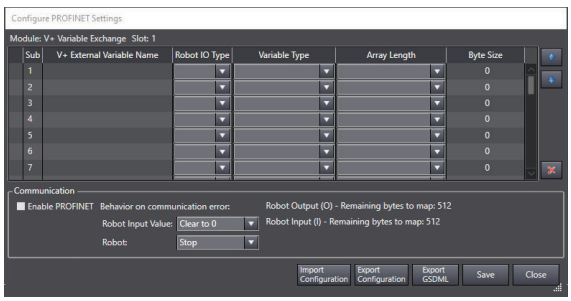
Additional Information

To activate the PROFINET fieldbus, you must configure at least one variable in addition to enabling the checkbox in the Configure PROFINET Settings Window. If no variables are defined, PROFINET will not be enabled.

Make the following considerations when configuring the PROFINET variable data exchange.

- The information in this area must directly correspond with the PROFINET Controller configuration.
- There are no V+ External Variables that are defined by default.
- Brackets [] must follow the variable name when declaring an array variable type. Array variables must also have an array length set.
- A $ character must precede the variable name when declaring a string variable type. String variables must also have an array length set. V+ variable naming convention must be followed. Refer to *6-4-1 V+ Variable Names* on page 6-13 and *V+ User's Manual (Cat. No. I671)* for more information about naming variables.
- Saving a new PROFINET configuration requires a controller reboot to apply the changes.
- After you save the PROFINET configuration and reboot the robot, the variables are automatically created in the robot controller. To confirm the presence of the new variables, execute the **Pull V + Memory** function to view them in the Variable Editor area. Refer to *6-2-6 Pull V+ Memory* on page 6-5 and *6-4 V+ Variable Editor* on page 6-13 for more information.
- PROFINET fieldbus configuration and set up are possible in emulation mode; however, active PROFINET communication is not supported in emulation mode.

Use the information below to understand other functions in the *Configure PROFINET Settings* area.



Item		Description
Communication Area	Enable PROFINET	Enables or disables PROFINET communications for the currently selected robot controller. Enabling the fieldbus automatically configures and activates the network ports in the robot for PROFINET communication. In the default robot state, both fieldbuses are disabled and all ports are enabled for Ethernet communication.
	Behavior on communication error	Determines the behavior when a communication error occurs. • Robot Input Value: Clear to 0 (default) All Input variables have a 0 value. • Robot Input Value: hold last value All Input variables retain their values when the error occurred. • Robot: Stop (default) The robot stops motion. • Robot: Stop and Disable Power The robot stops and then disables High Power. • Robot: Ignore No change to robot behavior.
	Remaining bytes to map	Displays the available bytes remaining while considering the existing configuration that was entered.
Import Configuration Button		Import an existing configuration file from another ACE project.
Export Configuration Button		Export the configuration file for use with another ACE project.
Export GSDML Button		Export GSDML information to the local PC for use with a PROFINET configuration tool.
Save Button		Save the existing PROFINET configuration. This will require a robot controller reboot to apply the new configuration. After the configuration is saved and the controller is rebooted, all new variables associated with PROFINET data exchange will be available for use with V+ programs.
Close Button		Close the **Configure PROFINET Settings** area.

Precautions for Safe Use

Fieldbus communications are not secure. Additional measures must be taken if compliance with security standards is required.

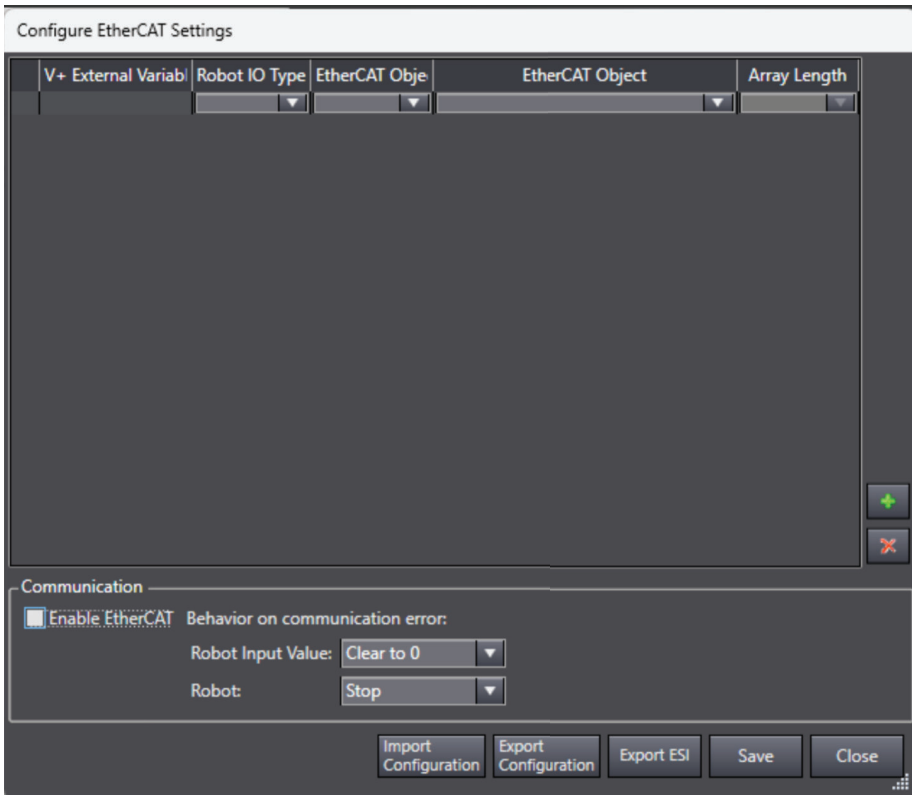
Additional Information

Refer to the *Industrial Robot Fieldbus Configuration User's Guide (Cat. No. M130)* for more information about PROFINET configuration.

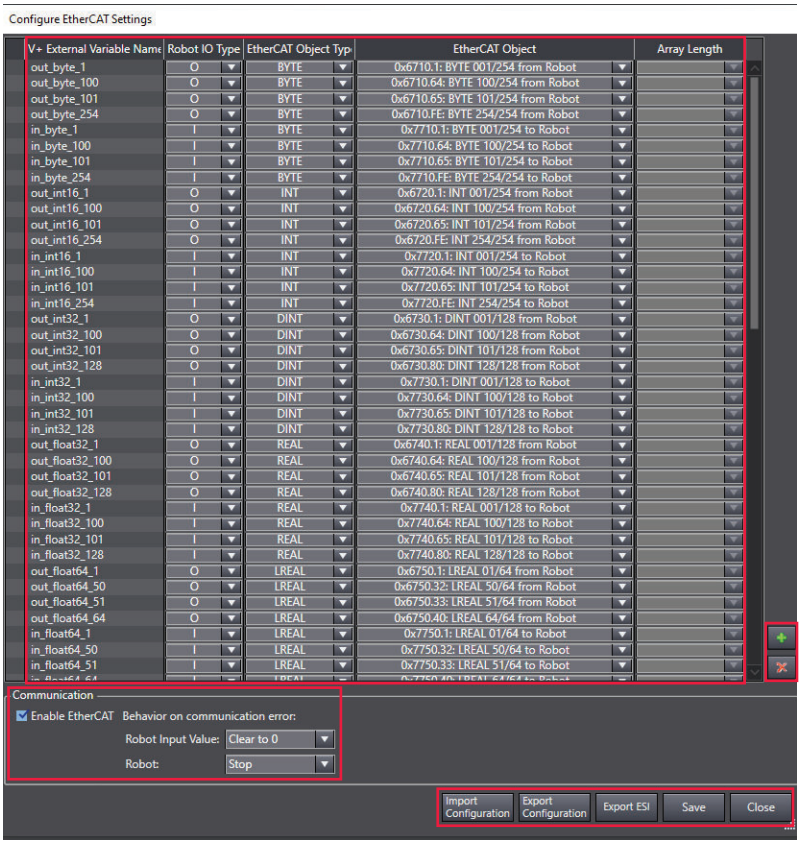
Configure EtherCAT Settings

Use the **Configure EtherCAT Settings** area to enable or disable the EtherCAT fieldbus, and add, edit, or delete variables associated with EtherCAT communication.

Access the **Configure EtherCAT Settings** area by double-clicking the **EtherCAT** label or status indicator color in the *Task Status Control* interface. Refer to *5-9-9 Fieldbus Status* on page 5-46 for more information. This area can also be accessed in the Multiview Explorer by selecting **Configurations and Setup** > **Controller Settings** > **Configure** > **Configure EtherCAT Settings** and clicking the **Finish** Button to display the dialog box below.



The following EtherCAT configuration for the EtherCAT SubDevice (robot) is available in this area.



- Enable or disable EtherCAT fieldbus communication.
- EtherCAT External V+ variable data exchange configuration. ACE allows you to map V+ External variables by their name to EtherCAT objects. Use the  and  buttons to add and delete variables.
- Robot behavior settings in the event of EtherCAT communication errors.
- Import and Export configuration information.

⚠ CAUTION

Always physically disconnect the robot from the EtherCAT network before disabling its EtherCAT communication.

Make the following considerations when configuring the EtherCAT variable data exchange.

- The information in this area is available for data exchange with the EtherCAT MainDevice.
- There are no V+ External Variables that are defined by default.
- Variable names must begin with an alphabetic character. V+ variable naming convention must be followed. Refer to *6-4-1 V+ Variable Names* on page 6-13 and *V+ User's Manual (Cat. No. I671)* for more information about naming variables.
- Variable names can consist of alphanumeric characters (0-9, A-Z, a-z), as well as period and underscore.
- The maximum length of a variable name is limited to 15 bytes.
- Brackets [] must follow the variable name when declaring an array variable type.
- A $ character must precede the variable name when declaring a string variable type.

- Enter the V+ External Variable Name, then set the IO Type, EtherCAT Object Type, EtherCAT Object, and Array Length in the table. Refer to the *Industrial Robot Fieldbus Configuration User's Guide (Cat. No. M130)* for more information.
- If you define an incorrect variable name, an error icon appears in the left column. Hover over the icon to display the cause of the error. Until the error is rectified, the configuration cannot be saved.
- Saving a new EtherCAT configuration requires a robot controller reboot to apply the changes.
- After you save the EtherCAT configuration and reboot the robot, the variables are automatically created in the robot controller. To confirm the presence of the new variables, execute the **Pull V + Memory** function to view them in the Variable Editor area. Refer to *6-2-6 Pull V+ Memory* on page 6-5 and *6-4 V+ Variable Editor* on page 6-13 for more information.
- EtherCAT fieldbus configuration and set up are possible in emulation mode; however, active EtherCAT communication is not supported in emulation mode.

Use the information below to understand other functions in the *Configure EtherCAT Settings* area.

Item		Description
Communication Area	Enable EtherCAT	Enables or disables EtherCAT communications for the currently selected robot controller. Enabling the fieldbus automatically configures and activates the network ports in the robot for EtherCAT communication. In the default robot state, both fieldbuses are disabled and all ports are enabled for Ethernet communication.
	Behavior on communication error	Determines robot behavior when a communication error occurs. • Robot Input Value: Clear to 0 (default) All Input variables have a 0 value. • Robot Input Value: hold last value All Input variables retain their values when the error occurred. • Robot: Stop (default) The robot stops motion. • Robot: Stop and Disable Power The robot stops and then disables High Power. • Robot: Ignore No change to robot behavior.
Import Configuration Button		Import an existing configuration file from another ACE project.
Export Configuration Button		Export the configuration file for use with another ACE project.
Export ESI Button		Export the EtherCAT SubDevice ESI file containing the SubDevice profile, from the robot controller to your local PC. This file contains the SubDevice's profile supported by the version of firmware with which it is associated. The device profile contains the layout of all allowed EtherCAT objects (data type and array length). To enable EtherCAT communication between the SubDevice and the MainDevice, import this saved ESI file into the EtherCAT MainDevice using Sysmac Studio or your EtherCAT MainDevice application.

Item	Description
Save Button	Save the existing EtherCAT configuration. Reboot the robot controller to apply the new configuration. After the configuration is saved and the controller is re-booted, all new variables associated with EtherCAT data exchange will be available for use with V+ programs.
Close Button	Close the *Configure EtherCAT Settings* area.

Precautions for Safe Use

Fieldbus communications are not secure. Additional measures must be taken if compliance with security standards is required.

Precautions for Correct Use

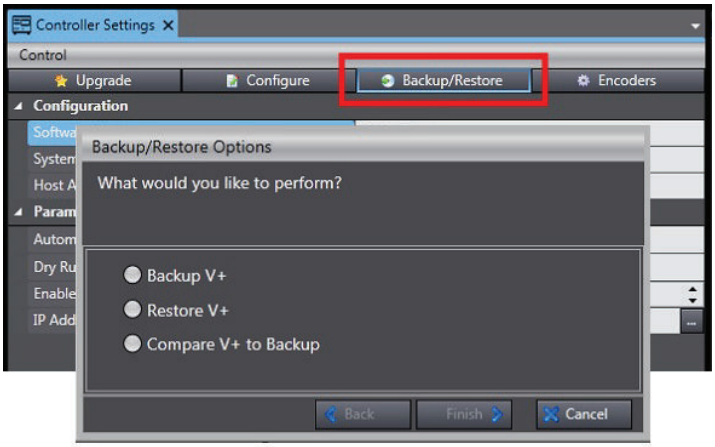
Disconnecting EtherCAT communications while the robot is transitioning between control states may cause an invalid fieldbus state. Cycle power to the robot to recover from this condition.

Additional Information

Refer to the *Industrial Robot Fieldbus Configuration User's Guide (Cat. No. M130)* for more information about EtherCAT configuration.

7-2-6 Backup/Restore

Use the Backup/Restore function to backup, restore, or compare the V+ operating system files and directories. Clicking the **Backup/Restore** Button displays the Backup/Restore Options Dialog Box. This is selectable only while online with the controller.

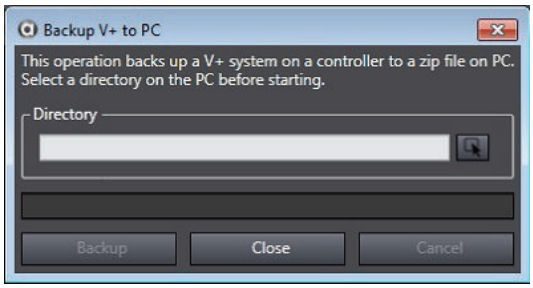


Backup V+

Use the Backup V+ function to back up the V+ operating system files and directories to the connected PC. Select *Backup V+* and then click the **Finish** Button to display the following window.
Choose a PC directory for the V+ operating system files and directories to be stored and then click the **Backup** Button to proceed.

Additional Information

The backup process takes several minutes. If there are files present in the selected PC directory, a prompt opens to remove them before the backup process can begin.

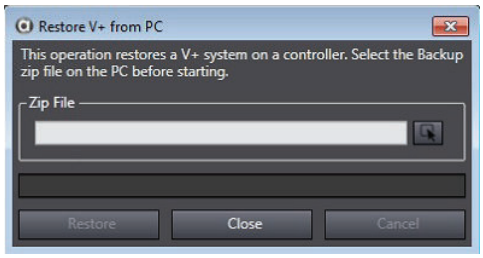


Restore V+

Use the Restore V+ function to restore the V+ operating system files and directories from a directory on the connected PC. Select **Restore V+** and then click the **Finish** Button to display the following window.
Choose a PC directory where the desired V+ system zip file is located and then click the **Restore** Button to proceed.

Additional Information

The restore process takes several minutes.

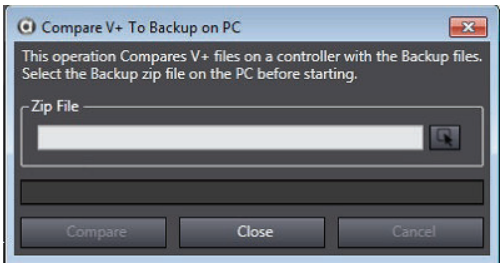


Compare V+ to Backup

Use the Compare V+ to Backup function to compare V+ operating system files and directories stored on the connected PC with those in the connected controller. Select **Compare V+ to Backup** and then click the **Finish** Button to display the following window.
Select a PC directory where the V+ system zip file for comparison is located and then click the **Compare** Button to proceed.

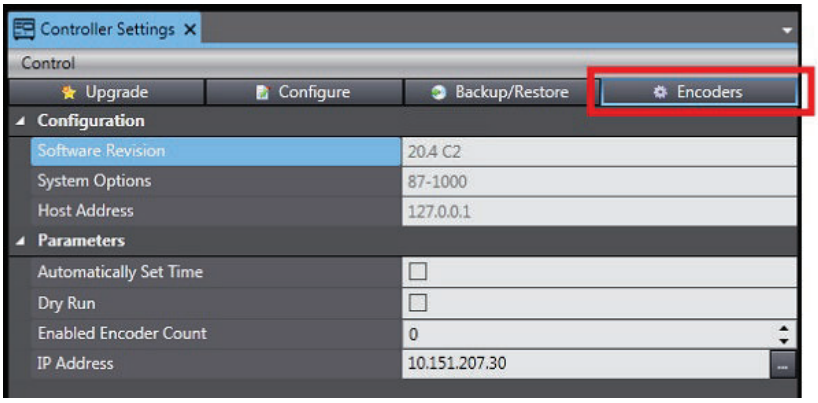
Additional Information

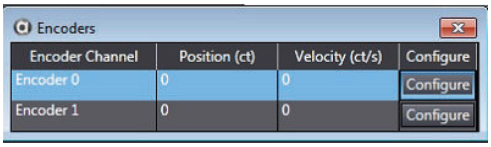
The compare process takes several minutes.



7-2-7 Encoders

Use the Encoders function to configure and check operating status of devices connected to the controller's encoder channels. Clicking the **Encoders** Button displays the *Encoders* Dialog Box.





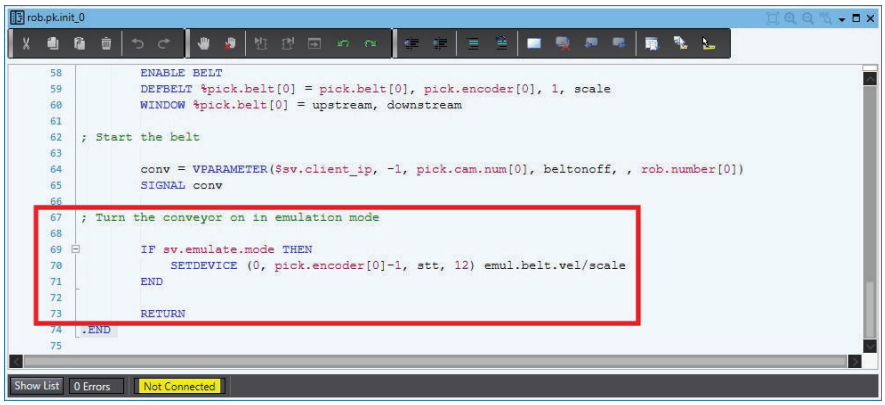
The *Encoders* Dialog Box displays all encoder channels that have been configured. The Configure Button is reserved for future use.

If an encoder channel that is present does not appear, check the Device Node Configuration and Belt Encoder Latch configuration.

To view and configure all encoder channels, ensure that all encoder channels are added in the Configure Belt Encoder Latches Dialog Box even if latches are unnecessary. Refer to *Configure Belt Encoder Latches* on page 7-21 for more information.

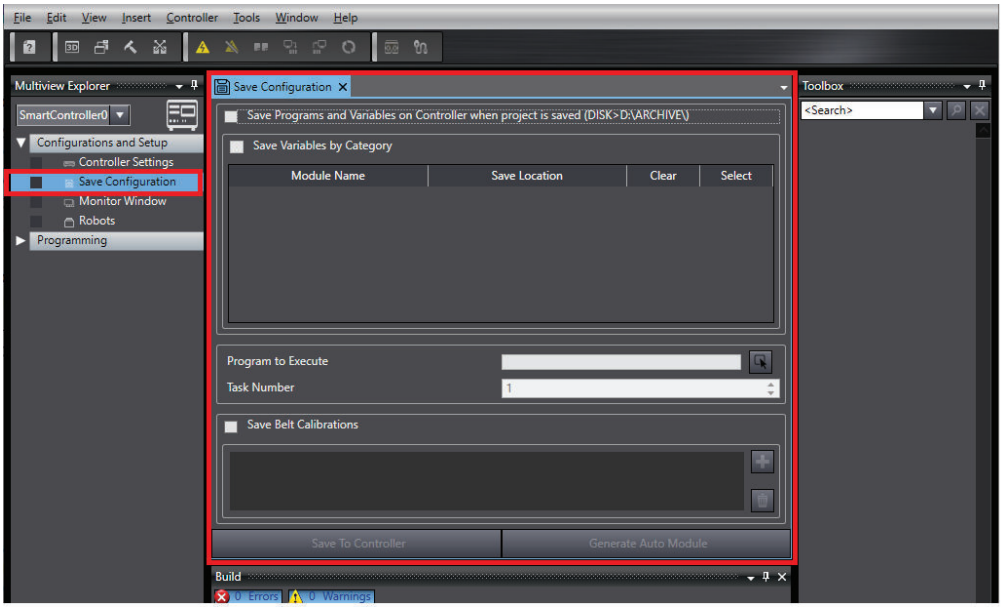
Additional Information

- It is possible to enter position and velocity values for an encoder while in Emulation Mode. However, typical interaction with emulated belt encoder travel is handled in a V+ program or a Process Belt object when working with a Process Manager.
- Robot Vision Manager Sample wizards that include belt tracking demonstrate an example of turning on a simulated belt encoder in a V+ program. Refer to the rob.pick.init V+ program created with a Robot Vision Manager Sample using belt tracking, as shown below.


rob.pk.init_0
ENABLE BELT
DEFBELT %pick.belt[0] = pick.belt[0], pick.encoder[0], 1, scale
WINDOW %pick.belt[0] = upstream, downstream
; Start the belt
conv = VPARAMETER($sv.client_ip, -1, pick.cam.num[0], beltonoff, , rob.number[0])
SIGNAL conv
; Turn the conveyor on in emulation mode
IF sv.emulate.mode THEN
SETDEVICE (0, pick.encoder[0]-1, stt, 12) emul.belt.vel/scale
END
RETURN
.END
Show List
0 Errors
Not Connected

7-3 Save Configuration

The Save Configuration selection in the Multiview Explorer provides options for saving programs, variables, and belt calibrations to the controller in the DISK>D:\ARCHIVE\ location.



This area also provides the capability to generate an AUTO.V2 program that can load modules, variable file contents, and execute a specific program on a specific task. The *Program to Execute* and *Task Number* Fields are used when generating the contents of the AUTO.V2 program after clicking the **Generate Auto Module** Button. The generated AUTO.V2 will also include program instructions for loading the saved modules. In order to execute this program at bootup, enable Auto Start (refer to *Auto Start* on page 7-25 for more information).

If *Save Programs and Variables on Controller when project is saved* is enabled, the following items will automatically save to the controller every time the project is saved.

- V+ programs
- Variables
- Belt Calibrations

If *Save Variables by Category* is enabled, individual files will be saved based on the category name. Variables without a category designation will be saved to a file named OTHERS.VAR. Refer to *Category* on page 6-14 for more information about variable categories.

All variables will be saved in a single file named GLOBALS.VAR if *Save Variables by Category* is not selected. Modules will be saved in a file called {module name}.pg. Use the Select Button ( ) to choose an alternate storage location for the Variable and Module files on the controller. The AUTO.V2 program will be stored at D:\AUTO.V2 on the controller when clicking the **Save to Controller** Button. Use the **Save to Controller** Button to manually save all items to the controller.

The **Save to Controller** and **Generate Auto Module** Buttons function only when an online connection is present.

If Robot Vision Manager Belt Calibrations are present, select *Save Belt Calibrations* and then use the **Add** ( ) or **Remove** ( ) Buttons to choose calibrations that are to be saved to the controller. Refer to *8-3-5 Belt Calibration* on page 8-23 for more information.

Belt Calibrations will be saved to DISK>D:\ARCHIVE\BELT.DAT when the **Save To Controller** Button is clicked.



Additional Information

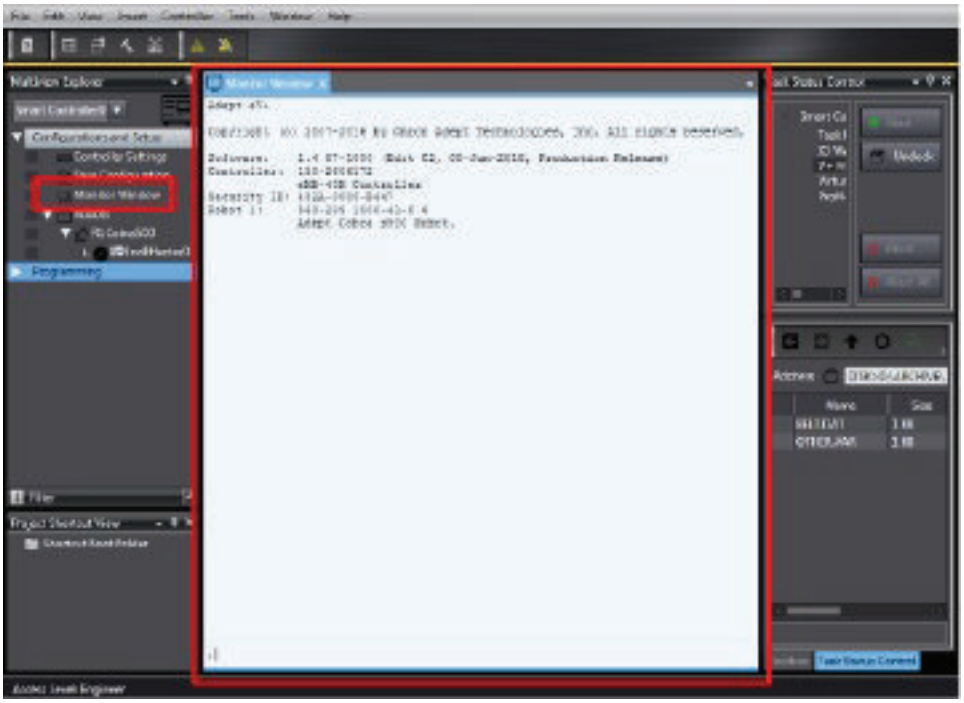
- The transfer may be interrupted if power is lost at the controller while saving data.
- Many *Save Configuration* items are not available while offline.
- The selection for *Save Programs and Variables on Controller* must be enabled to save belt calibrations to the controller
- When Emulation Mode is enabled, saving data to the virtual controller may result in a loss of data.

7-4 Monitor Window

The Monitor Window is used to input V+ Monitor Commands and receive responses from a controller. When the Monitor Window is open it can also output error information when an error occurs. User program output from the TYPE keyword can also be displayed within this window.

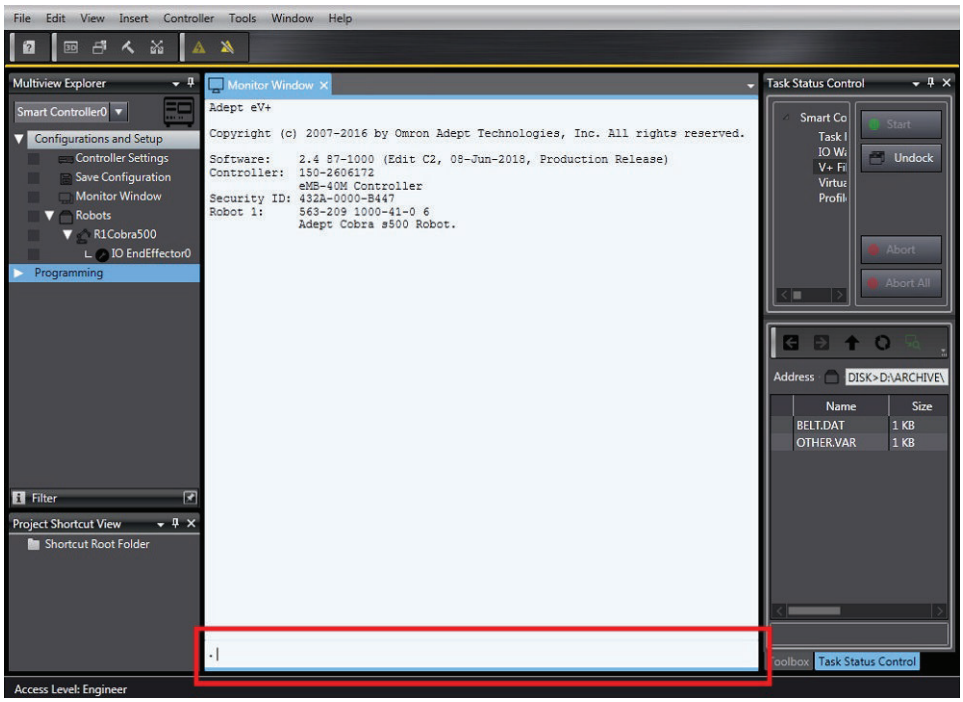
Additional Information

- The Monitor Window operates in read-only mode while offline.
- For a complete list of V+ Monitor Command Keywords, refer to the *eV+ Language Reference Guide (Cat. No. I605)* or the *V+ Keyword Reference Manual (Cat. No. I672)*.



7-4-1 Using the Monitor Window

When the input prompt cursor is blinking, the system is ready to accept Monitor Commands. Find the input prompt cursor at the bottom of the Monitor Window.
Up and down arrow keys can recall recent Monitor Commands for convenient re-execution.
Moving the scroll bar away from the bottom will pause auto-scroll. Return the scroll bar to the bottom to resume auto-scroll.
Closing the Monitor Window will clear the recent Monitor Command history.



7-4-2 Monitor Command Entry

Monitor Command do not have to be typed completely. Type only enough characters to uniquely identify the Monitor Command Keyword. For example, the only Monitor Command Keyword that begins with STAT is STATUS. Therefore, typing STAT is sufficient to execute the STATUS Monitor Command. Typing STA will result in the Ambiguous Name error because it could refer to either the STATUS or STACK command.

7-4-3 Monitor Command Keyword Parameters

Most Monitor Command Keywords require additional information that tells the system exactly how to execute the command. This additional information is specified as parameters on the command line. Specify unique information for each Monitor Command Keyword parameter.

Monitor Command Keyword parameters must be entered in the order they are listed and they must be separated (delimited) in exactly the fashion shown. A comma is normally used to separate parameters. Occasionally, an equal sign is used. A space must be typed between a command and its list of parameters.

Spaces before and after parameter separators are optional. Monitor Command Keyword parameters can be optional or required. If a parameter is required, a value must be entered on the command line or the Monitor Command will not execute correctly. If a parameter is optional, its value can be omitted and the system will substitute a default value. For example, the command STATUS has one optional parameter. If the command STATUS is entered, status information for all the used system tasks will be displayed. If the command STATUS 1 is entered, status information will be displayed only for system task number 1.

If one or more parameters follow an omitted parameter, the parameter separator(s) must be typed. If all the parameters following an omitted parameter are optional, and those parameters are omitted, the separators do not need to be typed.

7-4-4 Canceling a Monitor Command

To cancel an executing Monitor Command, press CTRL+C. This will NOT halt a program that was started with the EXECUTE command (the ABORT command halts an executing program)

7-4-5 Executing a Monitor Command

To execute a Monitor Command, type the command at the input prompt cursor. The results of the command are displayed in the same window.

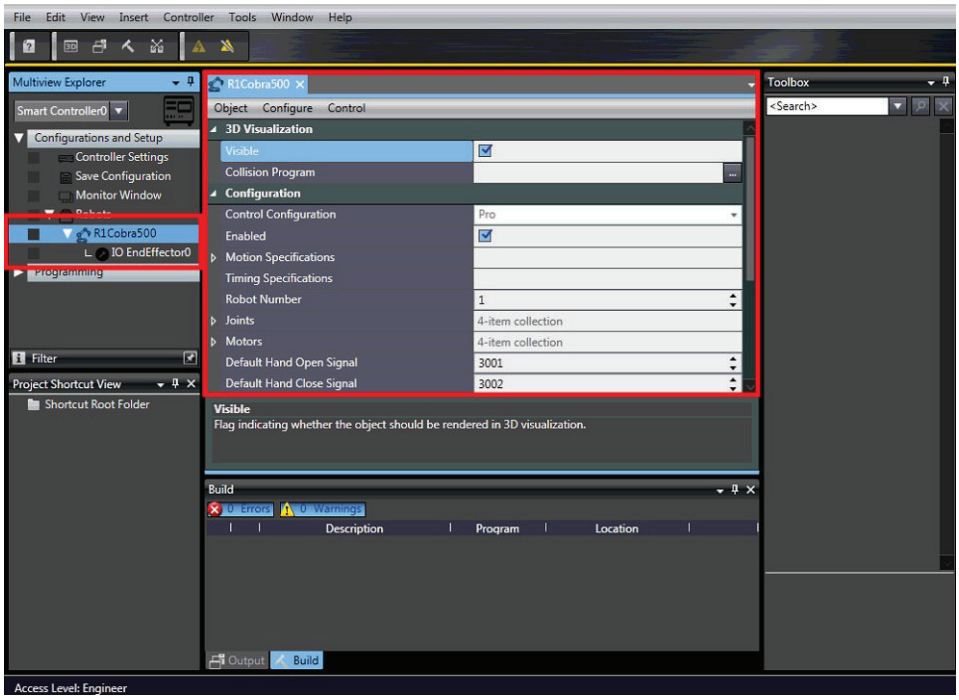
ACE tasks cannot be aborted from the Monitor Window in an ACE project. To abort an ACE task, use the Monitor Window that is provided at the *Connect to Device* Dialog Box. Refer to *4-1 Going Online with a Robot Controller* on page 4-2 for more information.

7-5 Robot Objects

Robot Objects represent robots that are installed on a controller. After a robot is installed, it appears in the Multiview Explorer to provide access to robot configuration and other related settings.

Additional Information

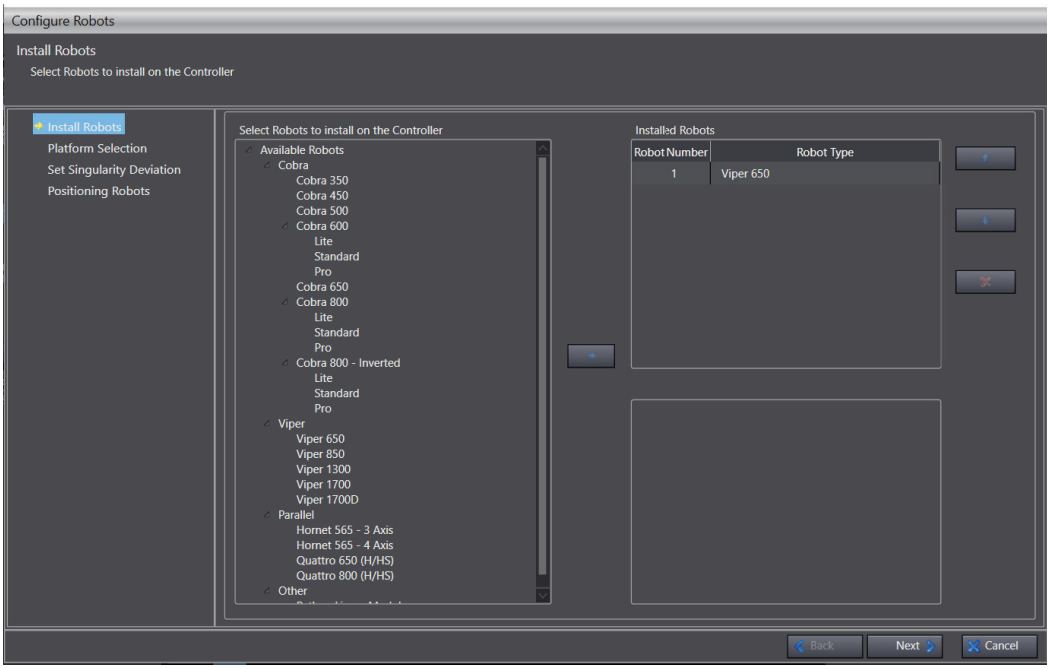
- Many Robot Object configuration and control settings are not available while offline. To access all available items, open the Robot Object while online. Refer to the following section for more details.
- When Emulation Mode is enabled, certain Robot object settings are not available.



Robot Objects must be added to a new ACE project while online with a controller. Right-click **Robots** in the Multiview Explorer to display the *Configure Robots* option, shown below . Use this option to add a new robot to an ACE project, or to manually add robots to a project running in Emulation Mode. When connecting to a physical controller, robots the connected robots will be present in the project after the connection is established.

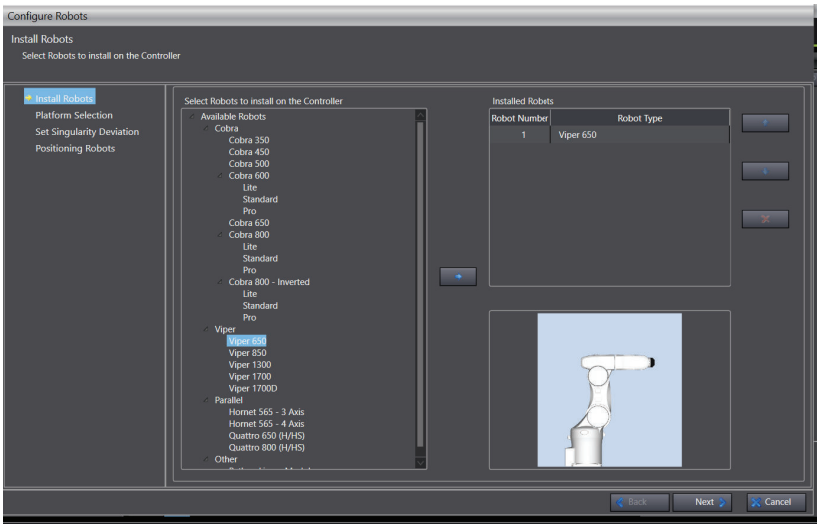
Additional Information

The *Configure Robots* Dialog Box can also be accessed through the Controller Settings area. Refer to *7-2-5 Configure* on page 7-14 for more information.

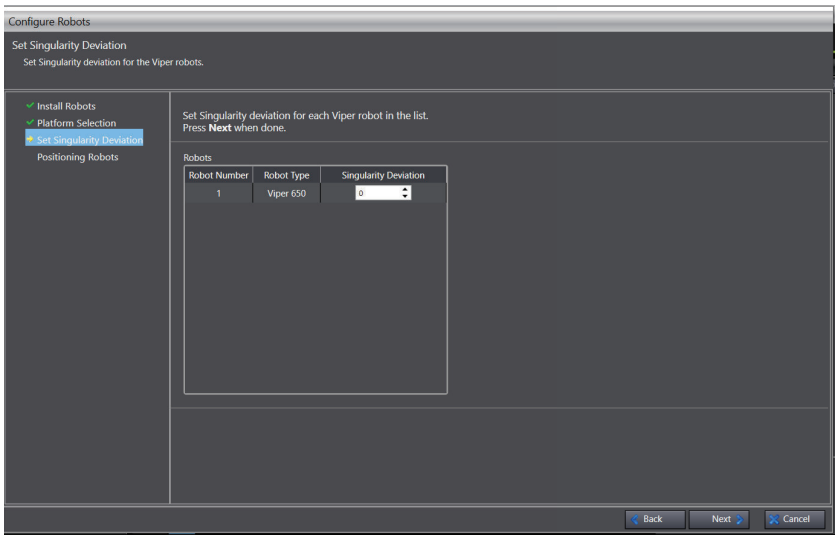


Depending upon the configuration and robot workspace, a configuration singularity may develop. A configuration singularity can be defined as a location in the robot workspace where two or more joints no longer independently control the position and orientation of the tool. As a robot executes a straight-line motion that moves close to a configuration singularity, the robot joint speeds necessary to achieve that motion become excessive. This speed control and motion can be controlled through the use of the Singularity Deviation.

When the robot is selected, the *Singularity Deviation* can be set to prevent the condition from occurring thereby preventing excessive joint motor speeds. The addition and testing of the *Singularity Deviation* offset should be done in JOG mode.



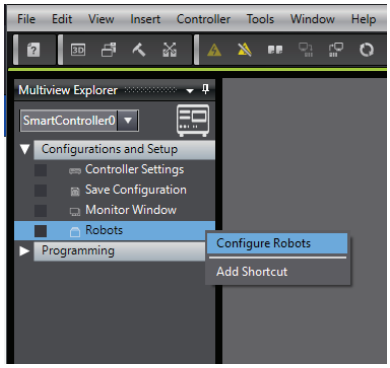
After the robot is selected, click **Set Singularity Deviation** to enable that option.



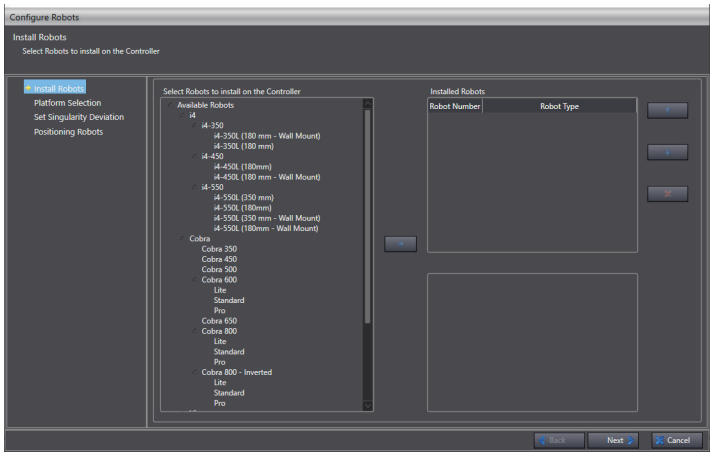
Set the value for the Singularity Deviation between 0, for no deviation to 100. The deviation is not linear and applies to the relative joint positions and motor speeds as the robot moves through the workspace.

7-5-1 Configure Robots

Access the *Configure Robots* menu item by right-clicking **Robots** and click **Configure Robots**.



Use the *Configure Robots* Dialog Box to select robots to install on the controller. In the *Configure Robots* Dialog Box, select specific robots to manually install on the controller or select **Auto Configuration**. After selecting robots to install on the controller, platform selection (if applicable) and robot positioning are required to complete the Configure Robots process.





Additional Information

Refer to *5-3 3D Visualizer* on page 5-24 for more information about robot positioning in a 3D workspace.

Auto Configuration Considerations

The default robot configuration uses Auto Configuration for identifying the physically connected robot type(s). The identification process occurs when V+ accesses a connected robot's RSC (Robot Signature Card) data during boot up.
Auto Configuration is convenient when a controller is used with a varying number of connected robots (such as demonstrations or other applications needing interchangeable robots).
The V+ log may contain extra errors or warnings if there are fewer robots connected than present in the controller configuration.
If more robots are physically connected than listed in the controller configuration, they will exist on the device node network but need configuration before ACE and V+ programs can fully support them.



Additional Information

Auto Configuration is not available when ACE software is running in Emulation Mode.

7-5-2 I/O EndEffectors

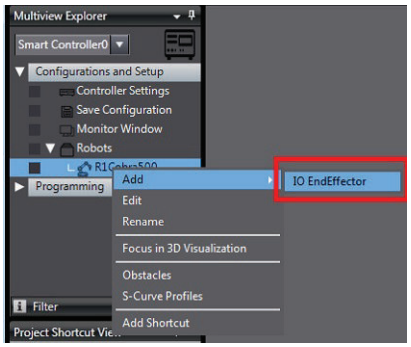
I/O EndEffectors, also called tools, end effectors, and grippers can be used in picking, placing, dispensing, or other functions common to robotic applications. End effectors are often driven by digital outputs to grip, extend, retract, or dispense. Inputs are commonly used on end effectors to detect parts presence and extend / retract status.
Most robots have a single end effector, but some may have multiple end effectors to pick and place multiple objects at the same time. ACE supports these variations with the I/O EndEffector settings described below.
By default, one end effector object is automatically added with a Robot Object (I/O EndEffector0). 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 I/O EndEffector objects may be added as needed for a specific application, such as calibration pointers used to perform calibration for systems with multi-tip process grippers.

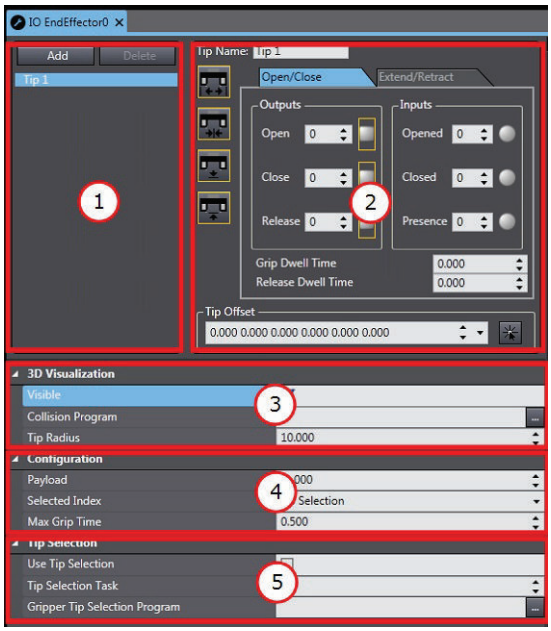


Additional Information

I/O EndEffector objects represent grippers that are wired to and controlled by the robot controller device. I/O EndEffectors are defined in the robot controller device because they are associated with specific robots, but full functionality of the I/O EndEffector is utilized by a Process Manager configured in an Application Manager device.

To add additional I/O EndEffector objects, right-click a **Robot Object** in the Multiview Explorer, select **Add** and then click **I/O EndEffector**.





Item	Name	Description
1	Add/Delete Buttons	Used to add a new gripper tip or to delete an existing gripper tip.
	Tip List	Displays the list of defined gripper tips. 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 center line of the gripper will be used as the reference tool in the calibration procedure. If multiple tips are defined, it can be useful to create a second I/O EndEffector object that stores a single tip offset for use when performing calibrations because the calibration process allows a choice for which I/O EndEffector object to use, regardless if the physical end effector needs to be changed. Refer to *8-7-5 Process Manager Object* on page 8-117 for more information.

Item	Name	Description
2	Outputs / Inputs	Defines the Open / Close or Extend / Retract activation signals and Opened / Closed or Extended / Retracted status signals for the selected tip. Define multiple signals by entering the signal numbers separated by a space (for example: 97 98). If the output signals are not valid signals, they are ignored. When multiple signals are defined, the following icon colors apply: ON OFF multiple signals (not all signals are ON or OFF) When Emulation Mode is enabled, the input signals are ignored, but soft signals can be substituted for testing purposes.
	Presence	Defines an input signal from a part-presence sensor that 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. Input signals are ignored in Emulation Mode, but soft signals will be monitored.
	Grip Dwell TimeRelease 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.
	Open tip / Close Tip Buttons	Use the **Open Tip** ( ) and **Close Tip** ( ) Buttons to send a open tip or close tip signal to the selected tip.
	Extend Tip / Retract Tip Buttons	Use the **Extend Tip** ( ) and **Retract Tip** ( ) Buttons to send an extend tip or retract trip 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 in the 3D Visualizer. To change the offset, click the **Teach Tool Tip Wizard** Button ( ), which starts the Tool Offset Wizard.
3	Visible	Enables the display of the I/O EndEffector tip positions in the 3D Visualizer. Refer to *8-2 3D Visualization* on page 8-10 for more information.
	Collision Program	Select a program that is invoked when the 3D Visualizer detects a collision. Refer to *5-3-4 Graphical Collision Detection* on page 5-29 for more information.
	Tip Radius	The radius of the tip when drawing the I/O EndEffector in the 3D Visualizer. Refer to *8-2 3D Visualization* on page 8-10 for more information.

Item	Name	Description
4	Payload	The weight of the gripper plus the weight of the heaviest part the robot will carry (in kg). Setting the correct payload will lower tracking error and settling time, making the robot motion more precise and faster at the end of the motion. A negative value restores the default payload for this robot.
	Selected Index	The currently selected /IO EndEffector tip. Choose *Centerline* when multiple tips are present.
	Max Grip Time	Maximum time allowed for the grip to be verified and part presence to be satisfied, in seconds.
5	Use Tip Selection	When enabled, the tip selection program will be called when a tip is selected.
	Tip Selection Task	The task used when executing the tip selection program.
	Gripper Tip Selection Program	The V+ program that is called when a tip is selected. 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.

I/O EndEffector Settings

ACE software provides an interface for setting various gripper-related parameters, such as end effector tips, minimum grip time, and maximum grip time. To open the I/O EndEffector settings area, double-click the **I/O EndEffector** Object in the Multiview Explorer.

Additional Information

I/O EndEffector properties can be accessed from V+ programs using VPARAMETER and VLOCATION keywords. Refer to the *V+ Keyword Reference Manual (Cat. No. I672)* for more details or create a Robot Vision Manager application sample that includes vision for V+ program code examples. Using Robot Vision Manager keywords requires an online connection between the ACE software and a controller.

Precautions for Correct Use

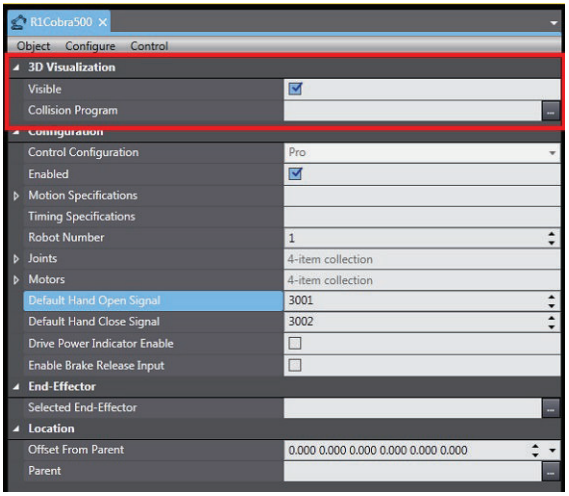
I/O EndEffector settings are not saved on the controller with the Save Configuration function.

7-5-3 3D Visualization

The 3D Visualization setting area is found in the main view of the Robot Object editor area. It is used to adjust 3D visualization settings for the Robot Object. It contains the following items.

Additional Information

Refer to *8-2 3D Visualization* on page 8-10 for more information about 3D Visualizer.



Visible

The *Visible* Check Box indicates whether the Robot Object should be rendered in 3D visualization.

Collision Program

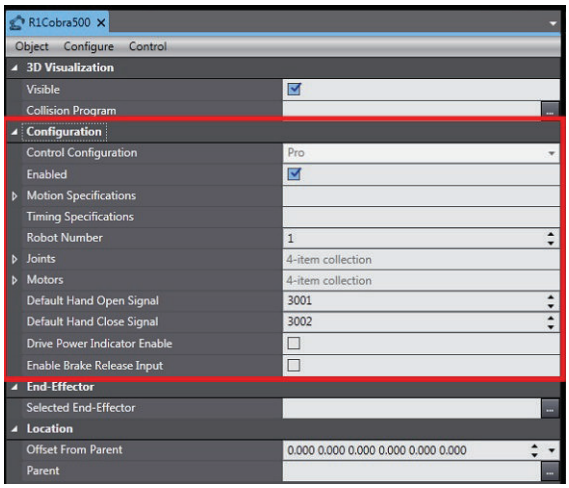
The Collision Program field allows selection of a C# program that is invoked when the 3D Visualizer detects a collision between two objects. Use the *Selection* Button (...) to select a program.

Additional Information

Refer to *6-5 Application Manager Programming* on page 6-19 for more information about C# programs. Refer to *5-3-4 Graphical Collision Detection* on page 5-29 for more information about collision detection with the 3D Visualizer.

7-5-4 Configuration

The Configuration setting area is found in the main view of the *Robot Object* Editor Area. It is used to adjust robot configuration settings for the Robot Object. It contains the following items.
Configuration setting changes take affect immediately, but will not be retained after a power cycle or reboot unless Save *Startup Specifications* is selected from the *Configure* Menu. Refer to *Save Startup Specifications* on page 7-51 for more information.



Control Configuration

The *Control Configuration* Item displays the license status of the Robot Object.

Enabled

The *Enabled* Selection enables or disables control of this robot. This is typically used during debugging and troubleshooting.

Motion Specifications

The *Motion Specifications* Item can be expanded to display several settings for robot speed and acceleration as described below.

Item	Description
Cartesian Rotation Acceleration	Specifies the Cartesian rotation acceleration at ACCEL 100 (deg/sec^2).
Cartesian Rotation Speed	Specifies the Cartesian rotation speed at SPEED 100 (deg/sec).
Cartesian Translation Acceleration	Specifies the Cartesian translation acceleration at ACCEL 100 (mm/sec^2).
Cartesian Translation Speed	Specifies the Cartesian translation speed at SPEED 100 (mm/sec).
Max Percent Acceleration	Specifies the Maximum allowable percent acceleration from an ACCEL command.
Max Percent Deceleration	Specifies the Maximum allowable percent deceleration from an ACCEL command.
Max Percent Speed	Specifies the Maximum allowable speed from a SPEED command.
SCALE.ACCEL Upper Limit	Specifies the Program speed above which accelerations are saturated when SCALE.ACCEL is enabled.

Timing Specifications

Timing Specifications are only available with Expert Access.

Robot Number

The *Robot Number* Field specifies the robot's designated number for the device node configuration. Use the *Configure Option*s - *Configure Device Nodes* to change this value. Refer to *7-5-8 Configure* on page 7-51 for more information.

● Robot Number Considerations

If only one robot is present in the system, it must be configured as Robot Number 1.
If multiple robots are present in the system, they must have unique Robot Numbers.

Joints

The *Joints* Item can be expanded to display settings for range of motion limit, full speed limit, and full acceleration limit of each robot joint.

Motors

The Motors item can be expanded to display settings for motor gain and nulling tolerances for each servo motor in the robot.

Item	Description
Motor Gains	Only available with Expert Access enabled. Contact your local OMRON representative for more information.
Fine Nulling Tolerance	Specifies the tolerance for the number of servo motor encoder feedback counts to consider a move complete when a move is made after specifying FINE tolerance. Refer to the *eV+ Language Reference Guide (Cat. No. I605)* or the *V+ Keyword Reference Manual (Cat. No. I672)* for more information.
Coarse Nulling Tolerance	Specifies the tolerance for the number of servo motor encoder feedback counts to consider a move complete when a move is made after specifying COARSE tolerance. Refer to the *eV+ Language Reference Guide (Cat. No. I605)* or the *V+ Keyword Reference Manual (Cat. No. I672)* for more information.

Default Hand Open Signal

The Default Hand Open Signal field specifies the output for the V+ keywords OPEN, OPENI, CLOSE, and CLOSEI. Refer to the *eV+ Language Reference Guide (Cat. No. I605)* or the *V+ Keyword Reference Manual (Cat. No. I672)* for more information.

Default Hand Close Signal

The Default Hand Close Signal field specifies the output for the V+ keywords OPEN, OPENI, CLOSE, and CLOSEI. Refer to the *eV+ Language Reference Guide (Cat. No. I605)* or the *V+ Keyword Reference Manual (Cat. No. I672)* for more information.

Drive Power Indicator Enable

The *Drive Power Indicator Enable* Selection is used to enable or disable the signal to indicate robot power. When enabled, an external indicator can be used to signal when robot power is ON. Refer to the appropriate robot user's manual for more information.

Precautions for Safe Use

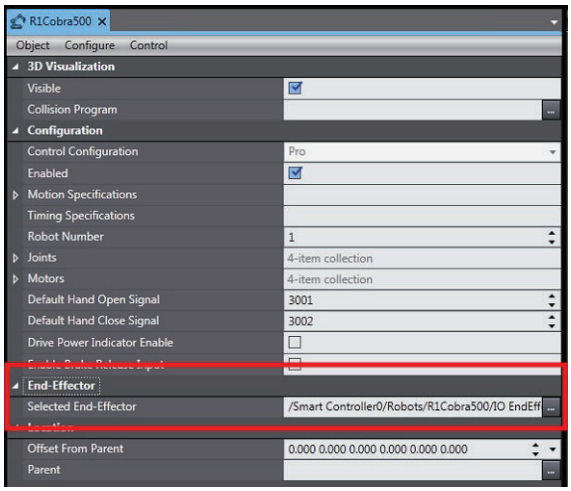
The user is responsible to connect and install the appropriate visual indicator(s) as per the requirements of UL 1740. This function is required to be enabled to meet UL 1740 requirements.

Enable Brake Release Input

The *Enable Brake Release Input* Selection is used to turn ON or OFF the brake release input signal. Selecting this item will allow an external signal to be used for robot brake control.

7-5-5 End-Effector

The *End-Effector* Setting is used to select an I/O EndEffector for the Robot Object. This allows a Process Manager to reference the number of tips available for that robot when defining multi-pick process, and to control the gripper signals for a specific robot when the Process Manager is active.



Offset From Parent

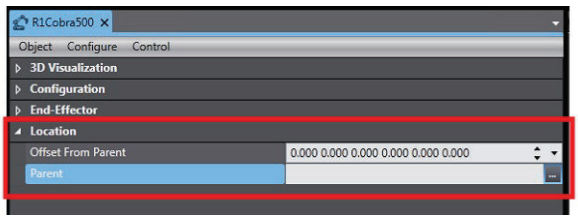
The *Offset From Parent* Field specifies a coordinate offset for the Robot Object relative to a parent item for 3D Visualization. This allows relative positioning of objects in workspace coordinates. A robot is typically a parent to other objects. The values are specified as X, Y, Z, yaw, pitch, and roll.

Parent

The *Parent* Selection specifies the object this robot will be relative to (using the *Offset From Parent* Parameter). Refer to *8-2-1 Adding Shapes* on page 8-10 for more information.

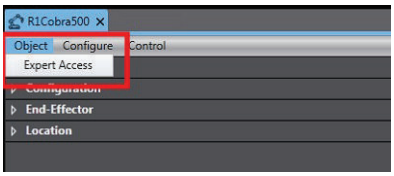
7-5-6 Location

The *Location Setting* Area is used to set the workspace coordinates of the Robot Object in 3D Visualization. It contains the following items.



7-5-7 Object

The *Object* Menu displays the Expert Access options described below.

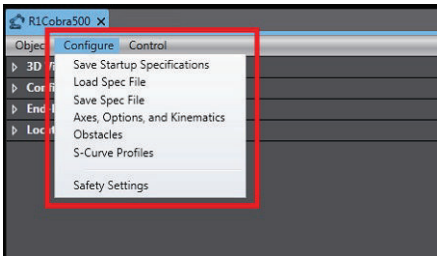


Expert Access

Expert Access grants access to all available parameters and settings. Contact your local OMRON representative for more information.

7-5-8 Configure

The *Configure* Menu displays the configuration items described below.

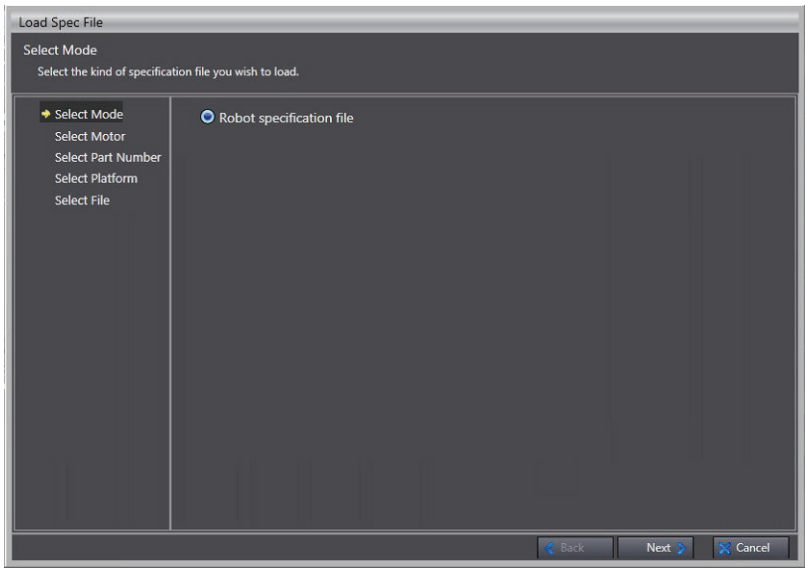


Save Startup Specifications

Selecting *Save Startup Specifications* will save all robot and motor specifications to the V+ boot disk. This is the same function that is present in the *Controller Settings - Control* Menu. Refer to *7-2-3 Control* on page 7-7 for more information

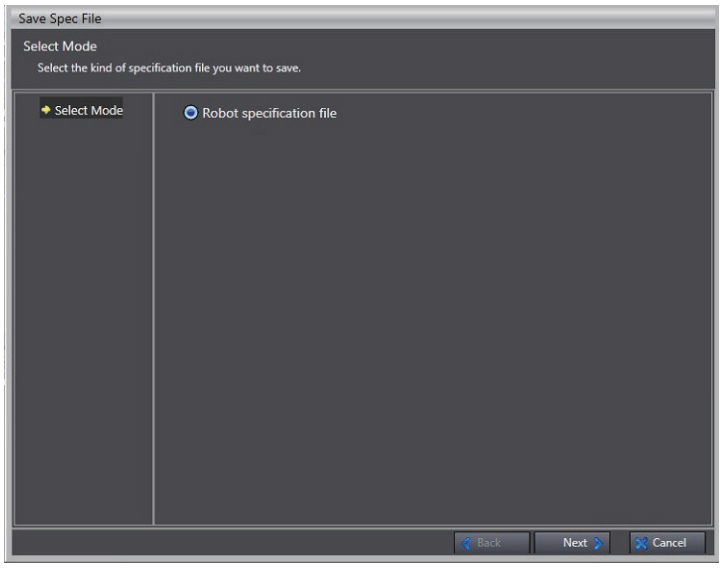
Load Spec File

A Spec file can be used to restore robot and motor specifications from a saved file. Selecting **Load Spec File** will open the *Load Spec File* Dialog Box. Choose a location where the saved Spec File is stored and then click the **Next** Button to proceed.



Save Spec File

A Spec file can be saved to store robot and motor specifications. Select **Save Spec File** to open the Save Spec File Dialog Box. Choose a location on the PC to save the Spec File and then click the **Next** Button to proceed.

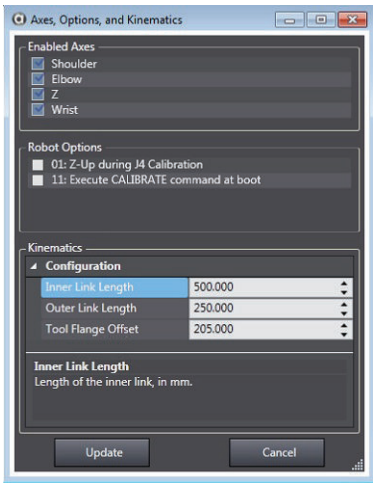


Axes, Options and Kinematics

Some robots have a variable number of joints and option bits that control the presence of special features and kinematic parameters used in position calculations. The *Axes, Options, and Kinematic Parameters* Dialog Box enables the editing of these parameters.

Precautions for Correct Use

Improper editing of robot joints, option bits, and kinematic parameters can cause the robot to malfunction or become inoperable. Editing must be performed by qualified personnel only.



● Enabled Axes

The *Enabled Axes* Area is used to enable / disable the joints (axes) of the robot. If the robot does not have joints that can be enabled / disabled, the *Enabled Axes* Check Boxes will be disabled.

● Robot Options

Use the *Robot Options* Area is used to select the robot option bits. Refer to the robot kinematic module documentation for the robot option bits that apply to the specific robot. See the table below for some common option bits.

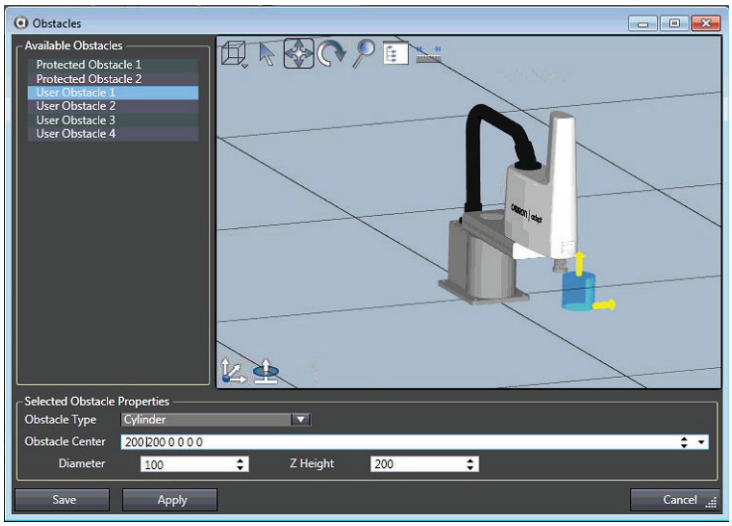
Item	Description
Free mode power OFF	Robot power is turned OFF rather than disabling the individual amplifier.
Execute CALIBRATE command at boot	Calibrate the robot after the V+ operating system boots. This is set by default on all Viper and Cobra s350 robots. This only works if the robot can calibrate with power OFF. It does not work on Cobra robots because they must move joint 4 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 Cobra robots, J4 must rotate slightly during calibration. This causes J3 to retract before moving J4.
J6 multi-turn	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, use software motor limits to restrict motor motion.

● Kinematics

The Kinematics area is used to display the kinematic parameters for a specific robot.

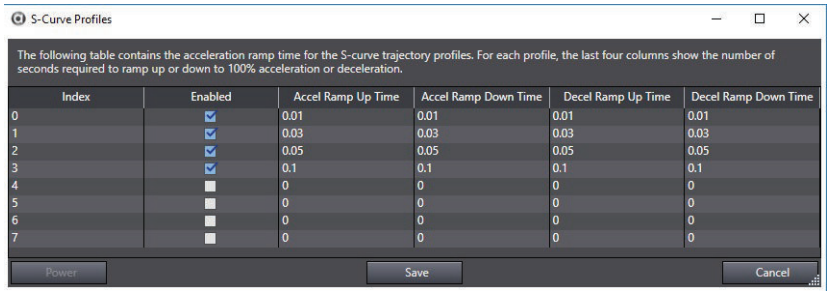
Obstacles

Select **Obstacles** to add / edit the location, type, and size of workcell obstacles. This will open the *Edit Obstacles* Dialog Box. It contains the following items.
Obstacles are predefined and evaluated in the controller when commanding robot motion and should not be confused with Collision Detection in the 3D Visualizer. Refer to *5-3-4 Graphical Collision Detection* on page 5-29 for more information.



S-Curve Profiles

The *S-Curve Profile Configuration* Dialog Box is used to configure the four time values required to create a new s-curve profile for use in robot motion (also called a trapezoidal acceleration curve). Select **S-Curve Profiles** to display the *S-Curve Profiles* Dialog Box.



● S-Curve Profile Considerations

An S-Curve Profile 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 can be controlled for a smoother motion.
For many mechanisms, controlling the acceleration is significant because high 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 time values that can be specified.

- Ramp up to maximum acceleration

- Ramp down from maximum acceleration
- Ramp up to maximum deceleration
- Ramp down to zero acceleration.

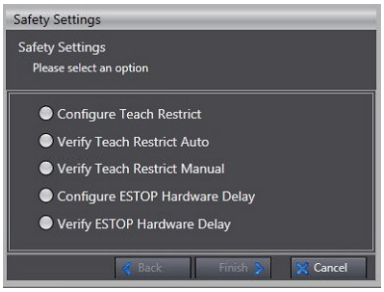
Each of these four acceleration values can be individually specified and a set of the four values defines a specific acceleration profile for use in programming robot motion routines.

Safety Settings

Safety Settings are used to restore the E-Stop hardware delay and the teach-mode restricted speed to the factory settings. Select **Safety Settings** to display the dialog box shown below. It contains the following items.

Additional Information

Safety Settings are not available in Emulation mode.



● Configure Teach Restrict

Selecting **Configure Teach Restrict** and clicking the **Next** Button will step through the procedure for setting predetermined speed limits for each robot motor. This configuration process is tuning circuitry in the amplifier units based on characteristics of specific motors. This process should be performed any time an amplifier is replaced in an existing system.

The objective of the Teach Restrict feature is to comply with safety regulations which require the speed to be limited while the robot is in manual mode. This is hardware-based safety functionality to prevent rapid robot motion in manual mode even in the unexpected event of software error attempting to drive a robot faster than allowed. While jogging the robot in manual mode, if any joint exceeds its configured speed limit the system will disable high power.

● Verify Teach Restrict Auto

Selecting **Verify Teach Restrict Auto** and clicking the **Next** Button will step through a procedure to verify that Teach Restrict is operating properly with a series of automatic commanded motions.

● Verify Tech Restrict Manual

Selecting **Verify Teach Restrict Manual** and clicking the **Next** Button will step through a procedure to verify Teach Restrict is operating properly with a series of jogging operations performed by the user with a T20 pendant. This may also be useful for troubleshooting or testing individual joints when Teach Restrict commissioning process or automatic verification fails.

- **Configure ESTOP Hardware Delay**

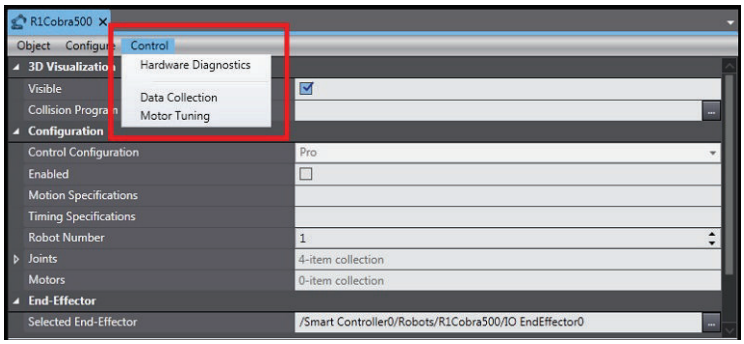
  Selecting **Configure ESTOP Hardware Delay** and clicking the **Next** Button will step through the procedure for configuring the delay on the ESTOP timer circuit. The objective of the ESTOP hardware delay feature is to comply with safety regulations which require the robot to have the capability of disabling high power without software intervention in an emergency stop scenario.

- **Verify ESTOP Hardware Delay**

  Selecting **Verify ESTOP Hardware Delay** and clicking the **Next** Button will step through the procedure to verify that robot high power is disabled without software intervention when an ESTOP is triggered.

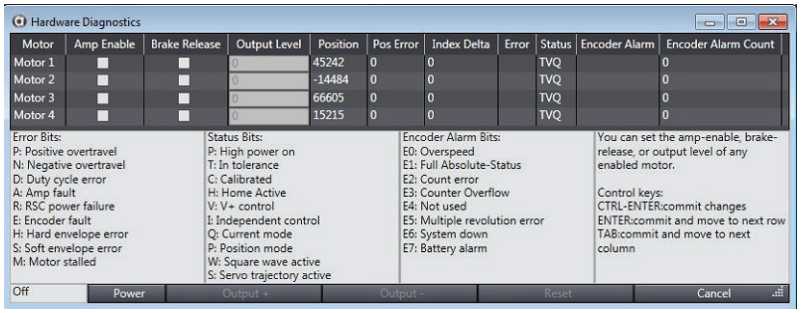
7-5-9 Control

The *Control* Menu displays Hardware Diagnostics, Data Collection and Motor Tuning items described below. These items are not available in Emulation Mode or while offline.



Hardware Diagnostics

Hardware Diagnostics are used to check robot motor status. For example, when a robot's segmented display shows encoder error E2, this means encoder error on Motor 2. Hardware Diagnostics can be used to determine what Encoder Alarm Bit on Motor 2 is triggering the encoder error.
Selecting **Hardware Diagnostics** will display the *Hardware Diagnostics* Dialog Box.



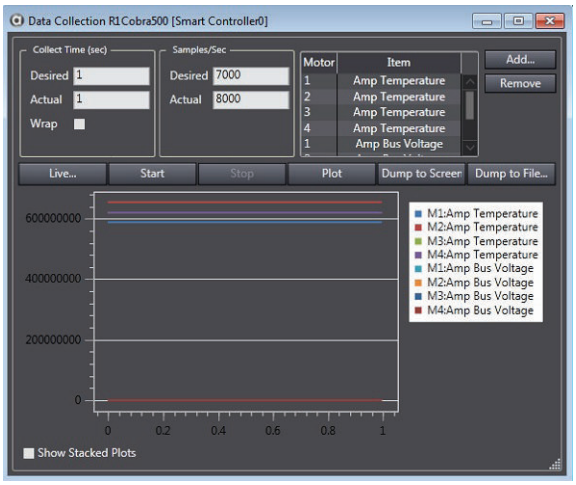
Item	Description
Amp Enable	Enables / disables the amplifier for the selected motor.
Brake Release	Enables / disables the brake release for the selected motor.

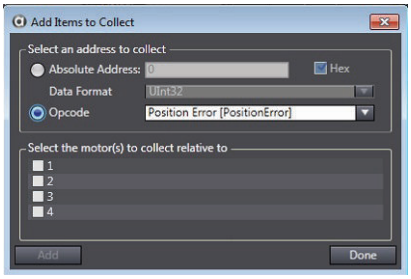
Item	Description
Output Level	Specifies a commanded torque, which is used to test the operation of the selected motor. The range is from -32767 to 32767, or the range specified by the Max Output Level parameter for that motor in the Robot Editor (restricted Expert Access parameter).
Position	Displays 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 following error codes for the selected motor. • P - Positive overtravel • N - Negative overtravel • D - Duty cycle error • A - Amp fault • R - RSC (Robot Signature Card) power failure • E - Encoder fault • H - Hard envelope error • S - Soft envelope error • M - Motor stalled
Status	Displays the following status codes for the selected motor. • P - High power on • T - In tolerance • C - Calibrated • H - Home sensor active • 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 state).
Output + Output -	DAC output controls. Click the **Output +** Button to increase the DAC output to the selected motor Click the **Output -** Button to decrease the DAC output to the selected motor.

Data Collection

Data Collection can be used to view, store, and plot various robot system data while online with a controller. A maximum of 8 data items can be examined at up to an 8 khz sampling rate, up to the memory limit of the controller's data buffer.

Selecting **Data Collection** will display the following window. Data Collection is not available in Emulation Mode.



Item	Description
Collect Time (sec)	Specifies the data collection time in seconds. The default value is 1.
Samples/Sec	Specifies the data collection rate in samples per second. The default value is 1000.
Add... Remove	Add or remove motor items to examine with these buttons. Clicking the **Add** Button will display the *Add Items to Collect* Dialog Box.  Enter an Absolute Address or select **Opcode** and choose an available data item from the drop-down list. Then select the motor(s) to collect data from.
Live	Displays a window that shows the real-time data being collected.
Start	Click to start the data collection. The data collection will continue until either the Stop button 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 *Data Collection* Dialog Box.
Dump to Screen	Displays the collected data in a *Data Dump* Window in text-file format.
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.

● Data Collection Sample

Data collection can be customized and collected from within a C# program. The script is comprised of several major elements:

- C# class definition
- INameLookupService field
- Main method entry point

The script contains lines that cannot be modified and are considered protected. These are designated by the light gray background. The namespace, class name, IAceServer field name, and Main method lines are protected and cannot be changed.

When the C# Program is executed, the text in the script is dynamically compiled in memory and the Main method is called. It is up to the user to define the logic of the script.

The following sample shows how to use data collected in this way.

```
using Ace.Services.NameLookup;
using Ace.Server;
using System;
using System.Collections.Generic;
using System.Diagnostics;

namespace Ace.Custom {

    public class Program {

        public INameLookupService ace;

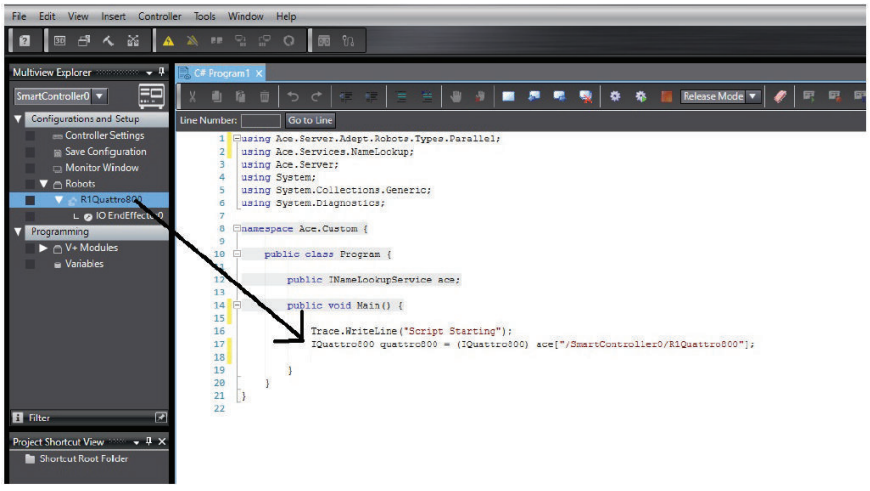
        public void Main () {

            Trace.WriteLine("Script Starting");

        }
    }
}
```

■ Referencing Other Objects and Tools

It is useful to reference other objects in the workspace when running a script. Reference other objects by dragging and dropping objects from the Workspace Explorer into the body of the script as shown below.



■ Data Collection Overview

In the following sample, the script will collect the encoder position for all 4 motors of a Cobra robot while the robot is moving, to do this, the following steps will be required:

- Acquire a handle to a robot.
- Identify the data to be collected.
- Collect the data.
- Extract the data.

1 Acquire a handle to a robot.

Drag and drop a robot from the workspace into the script. Since the script will be moving the robot, the script will ensure that robot power is enabled and the robot is calibrated.

```
//Drag n Drop Reference to a robot
IQuattro800 robot = (IQuattro800) ace["/SmartController0/R1Quattro800"];

if (!robot.Power){
    robot.Power = true;
    robot.Calibrate();
}
```

2 Identify the data to be collected.

The script must create DataCollectionConfiguration collection and add DataCollectionItem objects associated with each data item that should be collected. In this case, a DataCollectionItem must be created for each motor of the robot, and then each associated with the ServoData for collection.

The DataCollectionConfiguration also must be told how many samples are to be collected each second and how many seconds to run the test. The last step is to associate the DataCollectionConfiguration with the robot.

```
//Reference to robot data configuration
DataCollectionConfiguration config = robot.DataCollectionConfiguration;

//Clear all items that may be present in config from previous collection
config.Clear();

//Set data collection parametersa
config.TimeToCollect = 1; // Units in Seconds
config.SamplesPerSecond = 3;
int numberOfSamples = Convert.ToInt32(config.TimeToCollect * config.SamplesPerSecond);
int maxItemCount = robot.MaximumDataCollectionItemCount;

ServoData[] dataTypesToCollect = new ServoData[] {ServoData.CommandedPosition,
                                                  ServoData.PositionError};

//Add data items to data collection configuration
foreach (ServoData datatype in dataTypesToCollect)
    foreach (IMotor motor in robot.Motors)
        config.Add(new DataCollectionItem(motor, datatype));

//Assign configuration to the robot
robot.DataCollectionConfiguration = config;
```

3 Collect the data.

Once the data collection is enabled, the script should wait for the data collection to complete.

```
//Start Collection in robot controller
robot.DataCollectionEnabled = true;

//Option 1: wait for collection to complete
while(robot.DataCollectionEnabled)
    System.Threading.Thread.Sleep(200);

//Option 2: End collection prematurely (after 500ms)
// System.Threading.Thread.Sleep(1500);
// robot.DataCollectionEnabled = false;
```

4 Extract the data.

After data collection completes, the data resides on the Controller and must be extracted. This extraction is done using the AdeptRobot::ReadDataCollection method as noted below.

```
//Read Data from buffer in robot controller
List<double[]> rawData = new List<double[]>();

for (int i=0; i<config.Count; i++)
    rawData.Add(robot.ReadDataCollection(config[i],false,0,0));

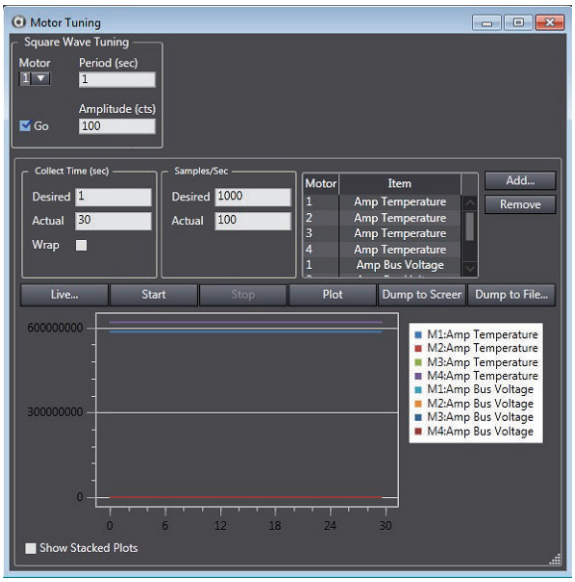
//Output Data to Trace Message window, or write to file
string header =  "Sample";
foreach (DataCollectionItem item in config)
    header += "\t| " + item.ToString();
Trace.WriteLine(header);

for (int i=0; i<rawData[0].Length; i++){
    string entry = i.ToString().PadLeft(3,' ');
    foreach (double[] item in rawData)
        entry += " | " + String.Format("{0:F3}",item[i]).PadLeft(30, ' ');
    Trace.WriteLine(entry);
}
```

Motor Tuning

Motor Tuning is used to send a square wave positioning command to the specified motor and observe the response for servo motor tuning purposes. Observing the response has the same functionality as Data Collection (refer to *Data Collection* on page 7-57 for more information).

Selecting **Motor Tuning** will display the following window. Motor Tuning is not available in Emulation Mode.



Item	Description
Motor	Specifies the motor that will receive the square wave positioning command.
Period (sec)	Specifies the length of the square wave in seconds.
Amplitude (cts)	Specifies the amplitude of the square wave in servo counts.
Go	Turns ON/OFF the square wave positioning command to the specified motor.

8

Application Manager Configuration and Setup

This section describes the functions, configuration, and setup of Application Manager items and objects.

8-1 Remote Application Manager 8-3
- 8-1-1 Remote Application Manager Set Up 8-3
- 8-1-2 ACE Server Instance 8-6
- 8-1-3 ACE Client Instance 8-7
- 8-1-4 Synchronization 8-8
- 8-1-5 Remote User Management 8-9

8-2 3D Visualization 8-10
- 8-2-1 Adding Shapes 8-10
- 8-2-2 Adding and Configuring CAD Data 8-11
- 8-2-3 Update 3D Shapes 8-14
- 8-2-4 Connection Points 8-14

8-3 Robot Vision Manager 8-18
- 8-3-1 Robot Vision Manager Object Configuration 8-19
- 8-3-2 Saving and Loading Calibration Data 8-19
- 8-3-3 Belt Latch Calibration 8-19
- 8-3-4 Camera Calibration 8-21
- 8-3-5 Belt Calibration 8-23
- 8-3-6 Gripper Offset Table 8-26
- 8-3-7 Vision Sequence 8-27
- 8-3-8 Overlap Tool 8-33
- 8-3-9 Communication Tool 8-34
- 8-3-10 Robot Frame 8-38
- 8-3-11 Robot Location 8-39
- 8-3-12 Robot Precision Point 8-41
- 8-3-13 Robot Transfer Path 8-42

8-4 Camera Objects 8-46
- 8-4-1 Virtual Cameras 8-46
- 8-4-2 Emulation Cameras 8-54
- 8-4-3 Physical Cameras 8-56
- 8-4-4 Custom Devices 8-57
- 8-4-5 Visualizer Capture Device 8-58

8-5 Configuration 8-61
- 8-5-1 Controller Connection Startup 8-61
- 8-5-2 Data Mapper 8-63
- 8-5-3 Note Object 8-64
- 8-5-4 OPC Container 8-65
- 8-5-5 Program System Startup 8-68

8-5-6 Recipe Overview 8-69
8-5-7 Recipe Manager 8-70
8-5-8 Recipe Editor 8-74
8-5-9 Recipe Manager Script 8-80

8-6 Feeders 8-82
8-6-1 AnyFeeder Object 8-82
8-6-2 I/O Feeder Object 8-88
8-6-3 iPF Object 8-90

8-7 Process Objects 8-97
8-7-1 Part Buffer Object 8-98
8-7-2 Part Target Object 8-101
8-7-3 Part Object 8-104
8-7-4 Belt Object 8-108
8-7-5 Process Manager Object 8-117
8-7-6 Allocation Script Object 8-160
8-7-7 Pallet Object 8-161
8-7-8 Vision Refinement Station Object 8-167
8-7-9 Process Manager Customization 8-170
8-7-10 Process Operation Object 8-189

8-8 Vision Tools 8-190
8-8-1 Adding Vision Tools 8-192
8-8-2 Vision Tool Editor 8-193
8-8-3 Region of Interest 8-194
8-8-4 Color Spaces 8-197
8-8-5 Finder Tools 8-199
8-8-6 Inspection Tools 8-247
8-8-7 Reader Tools 8-280
8-8-8 Calculation Tools 8-295
8-8-9 Image Process Tools 8-304
8-8-10 Custom Tools 8-323

8-1 Remote Application Manager

ACE 4 introduces the ability to install and run ACE as an Application Manager. This allows transfer of Application Manager device contents from one IPC to another. This in turn allows flexibility for users to:

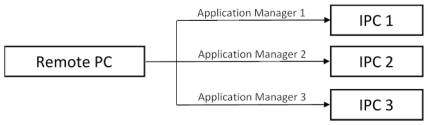
- Maintain one project file for multiple devices on a network
- Manage multiple Application Manager ACE instances from one interface
- Allows for projects to be modified offline and transferred to network connected PCs
- Monitor and modify project data over network connection to an IPC

Using a remote Application Manager requires two types of ACE instances. The first is called a server instance and is primarily used for application control during runtime functionality. The second is called a client instance and is used to create projects and transfer them to the server instances. A client instance is virtually identical to a standard ACE instance and is created just by opening ACE. It is called a client instance solely due to its connection to one or more server instances.

The client instance is connected to the server instances by using Ethernet connections between different computers. Typically, the server instances will be opened on IPCs connected to necessary cameras and controllers while the client instance will be on a separate computer, called the Remote PC in this section. When correctly configured correctly, client instance can be transferred to the server instances to provide system interaction. When the Application Manager is then configured and operating as desired, the Remote PC can go Offline and disconnect from the system without stopping production.

An example layout is shown below where the project on the client instance, in this example, has three Application Managers that are synchronized with a different server instance.

All remote Application Manager functionality requires the Application Manager device AM101-1000 version 2.0.

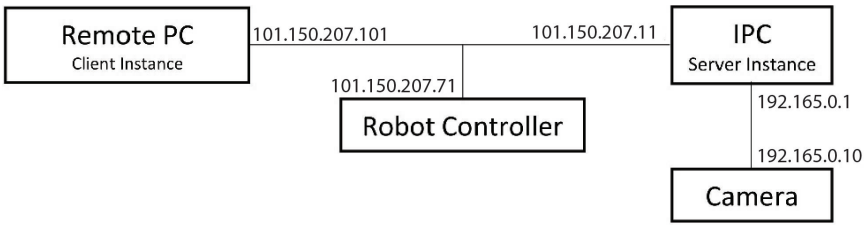


8-1-1 Remote Application Manager Set Up

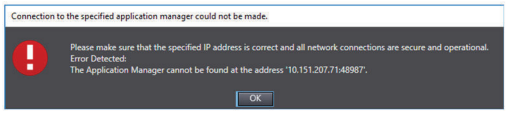
Establishing a connection between an instance of ACE and an Application Manager server instance requires the following:

- One computer (called the Remote PC in this section)
- At least one IPC application controller with necessary cameras
- One network switch and Ethernet cables
- At least one robot controller
- A recent version of ACE 4 installed on all computers

Any cameras need to be connected directly to their respective IPC, but the robot controllers should be arranged so both the Remote PC and the IPC can go Online with them. Refer to the following image for an example of the recommended layout. The sample IP addresses are shown to illustrate sample networks.



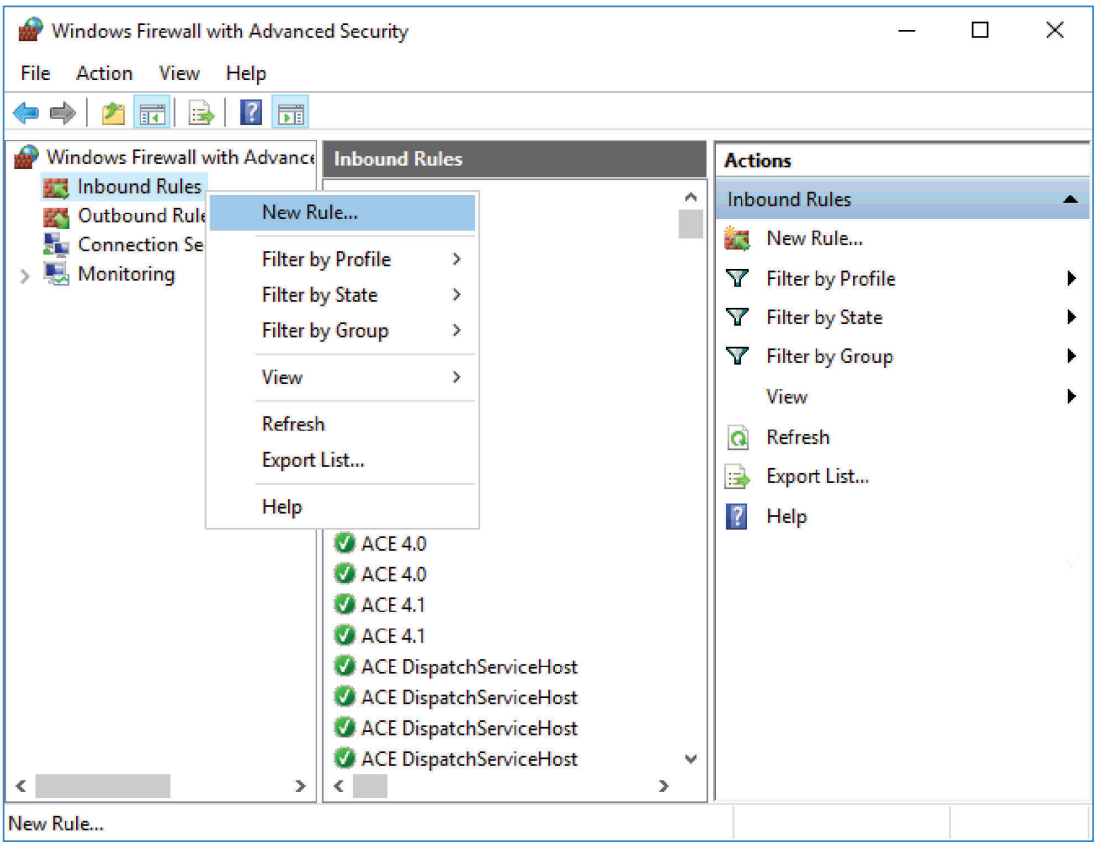
To transfer a project from the remote PC to the IPC, an exception must first be made for the port in the Windows firewall. Continuing without the exception will cause an error message, shown below:



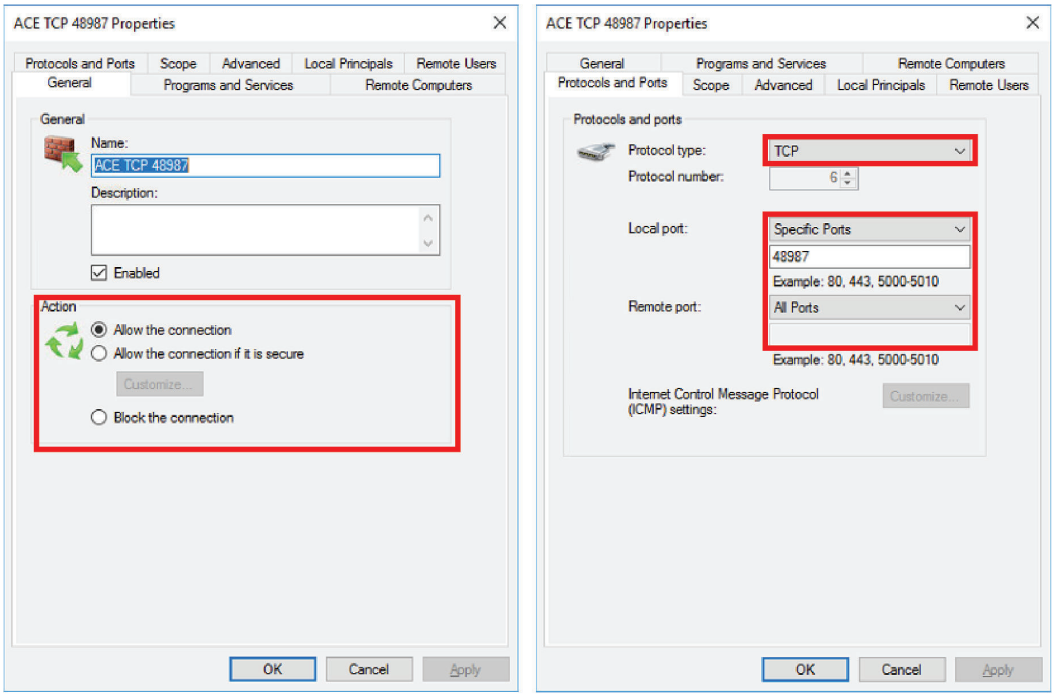
Remote Application Manager Set Up Steps

A firewall exception for all TCP and UDP ports should be automatically created during installation, but if additional security is needed, an exception for the default port can be created with the following steps:

1. Open Windows Firewall with Advanced Security. The name of this program may vary depending on the software version and computer model.
2. Right-click on **Inbound Rules** in the navigator on the left and then click on **New Rule**….
3. Select the bubble next to *Port* in the Rule Type step. Click the **Next** Button.
4. Confirm the bubbles next to TCP and Specific Local Ports are selected in the Protocol and Ports step. Type the number 48987 in the *Specific local ports* field. Then click the **Next** Button.
5. Verify the **Allow the connection** is selected in the Action step and click the **Next** Button.
6. Confirm that all checkboxes are selected in the Profile step and click the **Next** Button.
7. Assign a name and description as necessary and click **Finish**.
   Click on **Inbound Rules** to see the new rule listed among the others.

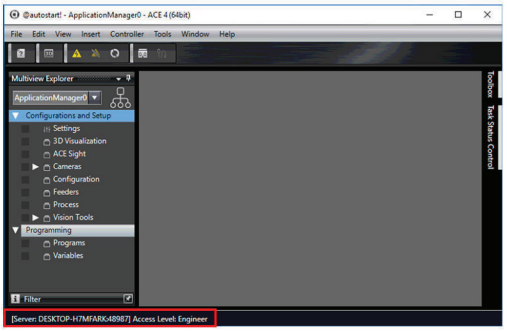


Double-clicking on the rule will show its properties. Refer to the following image for an example of what this should look like. Double-check the areas outlined in red and ensure they match the image.



8-1-2 ACE Server Instance

An ACE server instance is used as a recipient for projects from a remote computer. While it has similar functionality to a standard instance of ACE, when synchronized from a client instance, the active project will become the one from that client instance. A server instance is identified with a tag in the bottom-left corner, located in the same place as the Emulation Mode label. This tag also shows the name of the computer it is operating on as well as the port number. An example of this is outlined in red in the following figure.



Creating a Server

During the installation of ACE, selecting the Application Manager option allows ACE to open as a server, and installs a desktop short cut for this purpose. Opening an existing installation of ACE as a server requires the use of a keyword. ACE uses utilizes the following keywords in Command Prompt open ACE for different purposes:

- startclient is used to start a client session, which is the same as opening a standard instance of ACE.
- startserver is used to open a server instance.
- start is used to open a server instance with specified defaults
- help displays information about the command and offers options.
- version displays information about the currently installed version of ACE.

● Creating a Server Steps

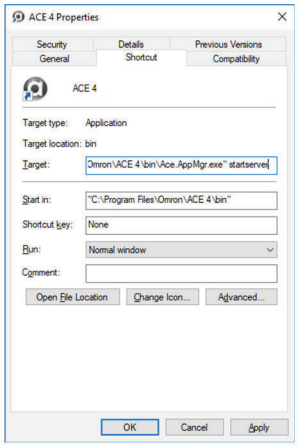
In general, the only one needed for the architecture illustrated above is startserver. A client session can be created by opening ACE normally or using the startclient keyword. However, startserver is required to create a server instance using Windows Command Prompt, as shown in the steps below:

1 Open the Windows Command Prompt.

2 Change to the directory containing the ACE 4 bin folder. The command should be similar to cd C:\Program Files\Omron\ACE 4\bin. Depending on the computer storage, the folder may be in a different location.

3 Type the command Ace.AppMgr.exe startserver.
Note that Steps 2 and 3 can be done in the same command by including the directory navigation in the command.

A server instance can also be opened by creating a short cut specifically for it. This is done by opening the properties of an ACE short cut and adding the startserver keyword after the quotation marks in the Target field.



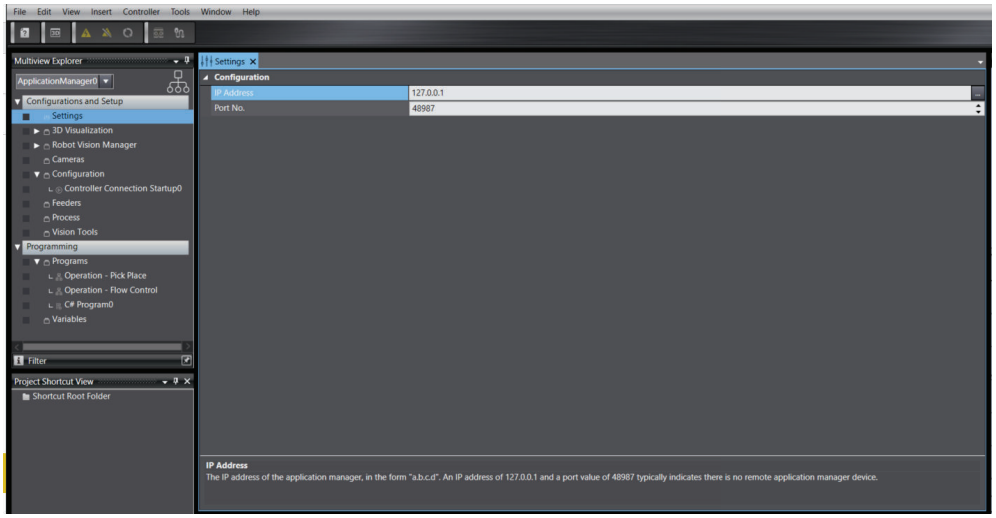
Additional Information

Opening a server instance of ACE will automatically open the project named @autostart! if it exists. Otherwise, it will open the ACE Start Page and will look like a regular instance, but it will still function as a server. See *8-1-4 Synchronization* on page 8-8 for more information.

If the server instance must is a different port number than the default, add --tcpport=[port number] after the keyword startserver. The target port number should be indicated.

8-1-3 ACE Client Instance

An ACE client instance is the standard instance of ACE that is used to connect to a server. This is done by opening the Settings of an Application Manager from the Multiview Explorer and entering the IP address of the server computer into the IP Address field. The Port No. should remain as the default 48987 to match the firewall exception. The connection is shown in the following figure.
If a different port number is required due to the needs of the application, a new Inbound Rule must be created for that number. See *8-1-1 Remote Application Manager Set Up* on page 8-3 for this process.

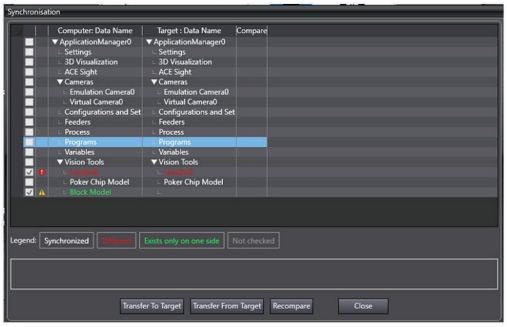


If a server instance exists on a computer with the defined IP address, the client instance can connect to it. This is done using the Online ( ) icon in the toolbar. If the connection cannot be established, an error message will be displayed.

Otherwise, the client will go online with the server instance. To disconnect, click the Offline ( ) icon.

8-1-4 Synchronization

The **Synchronize** Button is used to transfer Application Managers between clients and servers. Synchronization can only occur when the client instance is online with the server instance.



The main portion of the window shows the objects in the client and the server Application Managers in comparison to one another. The *Computer: Data Name* column shows the items in the client, and the *Target: Data Name* column shows those in the server. The text in each row is color-coded depending on the results of the comparison:

- White: The items are synchronized between the computer and the target. No differences are detected.
- Red: There are detected differences between the computer and the target. They must be synchronized for the project to properly function. This is also indicated by a red icon in the left column.
- Green: The item only exists on one side. It must be transferred to the other. This is also indicated by a yellow icon in the left column.
- Gray: The item has not been checked. This will only occur if there is an error outside of the Synchronization window.

The area at the bottom of the Synchronization window displays messages from the synchronization process.

The four buttons at the bottom of this window are described below.

- **Transfer To Target** Button synchronizes all checked items from the client to the server.
- **Transfer From Target** Button synchronizes all checked items from the server to the client.
- **Recompare** Button analyzes the computer and target data synchronization status. This also occurs when the Synchronization window is opened.
- The **Close** Button closes the *Synchronization* window.

In the example shown in the figure above, clicking the **Transfer To Target** Button will have the following effect:

- The Locator0 Vision Tool will be synchronized and updated on the server.
- The Block Model Vision Tool will be created on the server.

In the example shown in the figure above, clicking the **Transfer From Target** Button will have the following effect:

- The Locator0 Vision Tool will be synchronized and updated on the on the client.
- The Block Model Vision Tool will be removed from the client.

Successfully transferring an Application Manager from a client instance to a server instance will create an ACE project on the server with the name @autostart!. Transferring a different Application Manager to the server instance will replace an existing @autostart! project with the new data. Refer to *3-1-4 Clear Application Manager Memory* on page 3-6 for information about interference from server objects during the synchronization process.

8-1-5 Remote User Management

By default, client instances of ACE have Engineer access and, by default, server instances have Operator access. The Engineer level has the access to make process changes, and because all changes should be made in the client instance and then transferred to the server instance. In some cases, however, it may be necessary for changes to be made directly to the server. This activity can be managed and limited by creating users and passwords for the higher access levels, such as Engineer.
If changes are made directly on the server instance, it is recommended to immediately transfer the changes to a client instance. The changes will be erased if a new project is transferred to the server instance, so it is best to save it on another system as soon as possible.
The server instance can further be protected by adding additional users with various access levels and setting passwords for them. These can be created on the server instance directly or they can be created on a client instance and then transferred to the server instance using Synchronization.
Refer to User Management on page 1 for more information.

8-2 3D Visualization

In addition to robots, several other objects that are configured in the Application Manager are represented in the 3D Visualizer. The location, orientation and dimensions of these items are defined during the initial configuration of each item and can be adjusted to precisely simulate the application. The following items will appear in the 3D Visualizer after they are configured.

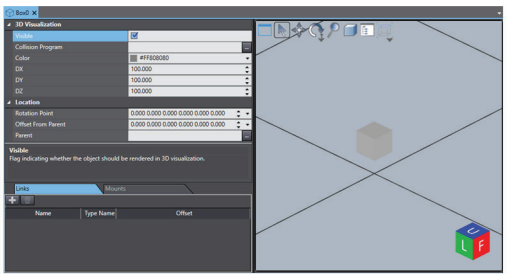
- Process Belts (Belt Calibrations, Windows, and Allocation Limits)
- Latch sensor calibrations
- Camera calibrations and field of views
- Part and Part Target instances
- Boxes, Cylinders, and imported CAD files

8-2-1 Adding Shapes

Boxes and cylinders can be added to the 3D Visualizer to represent objects in the robot workspace. Use the following procedure to add cylinders and boxes to the 3D Visualizer.

1 Right-click **3D Visualization** in the Multiview Explorer and select **Add** and then choose **Box** or **Cylinder**. A new object will be added to the Multiview Explorer under 3D Visualization and it will appear in the 3D Visualizer Window.

2 Access the properties for this object by double-clicking it or right-click and select **Edit**. This opens the properties editor in the Edit Pane.



Object	Description
3D Visualization	
Visible	Display or hide the object from the 3D Visualizer.
Collision Program	Select a program that is invoked when the 3D Visualizer detects a collision. Refer to *5-3-4 Graphical Collision Detection* on page 5-29 for more information.
Color	Choose a color for the shape in the 3D Visualizer.
DX/DY/DZ	Only available for Box. Defines the dimensions of the shape in the respective direction. DX and DY are both centered about the origin while DZ is measured from the box base.
Radius/Height	Only available for Cylinder. Defines the respective dimension of the shape. Radius is from the center axis and Height is measured from the cylinder base.

Object	Description
Location	
Offset From Parent	Set an offset distance from a parent object (X Y Z Yaw, Pitch, Roll).
Parent	Select an optional reference to a parent object where the base of the shape will be positioned. If a parent object is selected, when the parent object location moves, this object will be moved accordingly. Using robot objects as Parent will result in the object moving relative to the robot tool flange. This is particularly useful for moving end effectors with the robot tool flange, and visualizing objects held by the gripper. A C# program can be used to manipulate visibility of Visualizer objects (for example, to visualize parts on an end effector while the gripper output signals are ON).

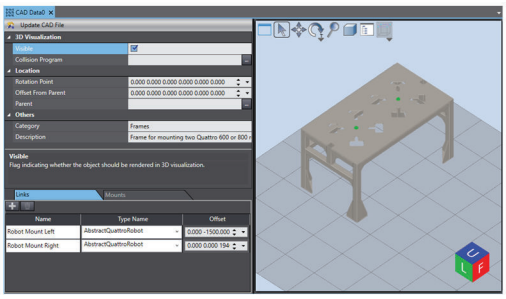
3 Adjust the shape's properties according to the application.

8-2-2 Adding and Configuring CAD Data

ACE version 4.4 can open a project created in an earlier ACE version, modify it, save it, then re-open in that earlier version if no change is made to the device type from the earlier version to 4.4. Once a device is explicitly upgraded in version 4.4, the device then it is no longer compatible with the earlier ACE version.

A CAD Data object is a three-dimensional shape created with computer-aided design (CAD) software. ACE supports the following CAD file types: .stp, .step, .igs, .iges, and .stl. CAD Data objects can be added to the 3D Visualizer. When a CAD Data object is created, an import wizard is opened to guide user through the process of importing the file.

To create a CAD Data object, right-click the 3D Visualization object in the Multiview Explorer and select **Add** and then click **CAD Data Object.** An import CAD file wizard will appear with two methods for adding CAD Data as described in the following sections. Once created, the CAD Data Object can be opened and edited like any other object in the Multiview Explorer.

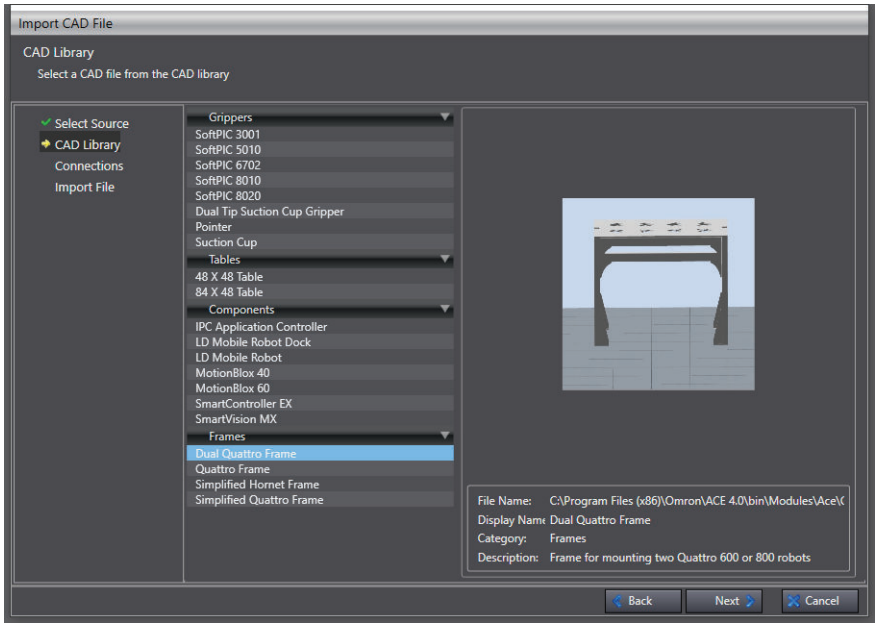


Standard CAD Library Object

Select **CAD Library**, as shown in the following figure, and then click the **Next** Button to access the CAD Library section of the wizard. The displayed objects appear in one of the following categories:

- Grippers - vacuum, pointers, and other gripper types.
- Tables - typically used for mounting SCARA and articulated robots.

- Components - controllers, mobile robots, and accessories.
- Frames - typically used for mounting parallel robots.



Selecting one of the CAD objects will show a preview of it in the window on the right as well as the details down below, including its Name, Category, and Description. An example of this is shown with the Dual Quattro Frame, as shown in the above figure.

Once the correct part has been adjusted, click **Next** to continue to the Connections step. This step is optional, but it can be used to define how objects are connected in the 3D Visualizer and will automatically set Parent and Offset From Parent values based on the links and mounts associated with that library object. See*8-2-3 Update 3D Shapes* on page 8-14 for more information.

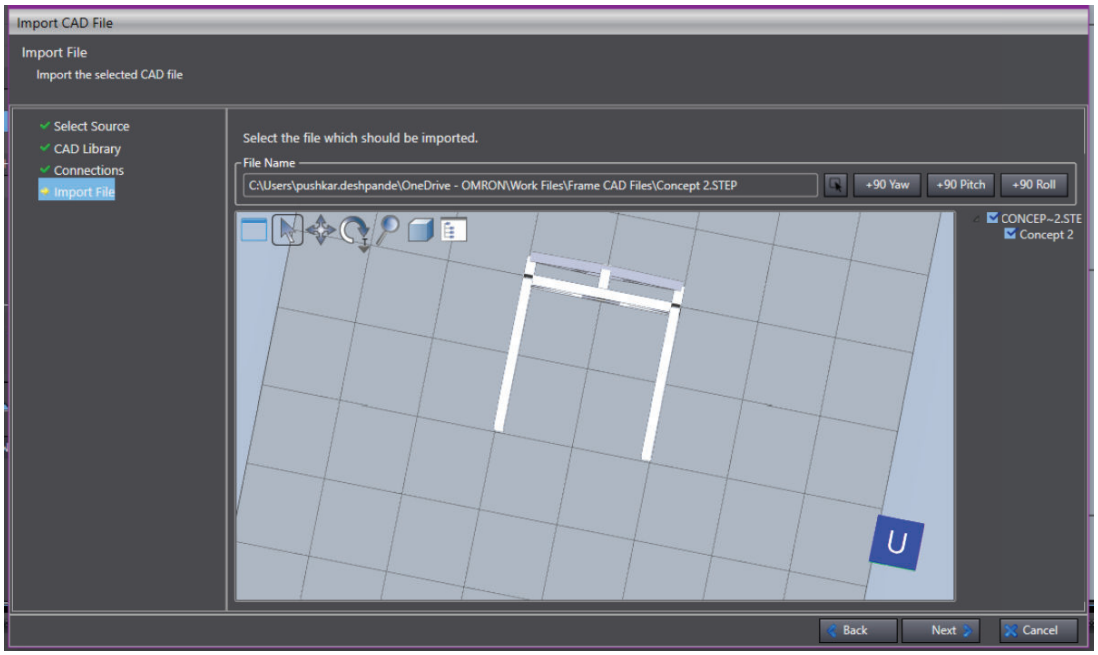
Clicking **Finish** in this step of the wizard will create the CAD Data object. The initial placement of the object depends on the connections and category:

- If no connections are added, it will always be created at the origin.
- If the CAD Data is in the Gripper category and a robot is selected for the Robot Flange connection, it will be created with the specified robot as its parent and positioned at the end effector. When the end effector moves, the CAD Data will move with it.
- If a connection is created to mount a robot to the CAD Data, such as mounting a SCARA robot onto a table, the CAD Data will be created at the origin and the specified robot will be moved to the correct mounting position with the CAD Data as its parent. Note that this cannot be undone and if the previous robot position is required, the parent CAD Data will need to be moved to accommodate this.

Once created, the CAD Data object can be repositioned by accessing its coordinates in the editor or using the Edit Workspace Position icon to drag it around the window. For more information, See *8-2 3D Visualization* on page 8-10.

● Imported CAD File

Select **Open my own CAD file** in the first step of the Import CAD File wizard to access the Import File step. Then click the selection icon next to the *File Name* field. A browser window will open to select the CAD file. The supported CAD file format is STEP. Selecting a file and clicking open imports the file into the wizard.



Once imported, the main window of this step shows the CAD Data file as it will be saved in the project. Some modifications can be made to it here before it is fully integrated. First, the +90 Yaw and +90 Pitch rotate the object to control its standard orientation in the 3D Visualizer. This is particularly useful if the file orientation is interpreted differently than ideal. For example, custom frames, as in the above figure, should be positioned so their feet are flat against the base plane. If they are shown in a different orientation in this window, the buttons can rotate them into the correct position. The navigator on the right side shows all of the components in the CAD file. The checkboxes define which of the parts are imported. By default, all parts will be selected. Deselecting these will omit them from the resulting CAD Data object in the project.
When the necessary adjustments are made, click **Next** to close the wizard and import the file.

■ Configuration of CAD Objects

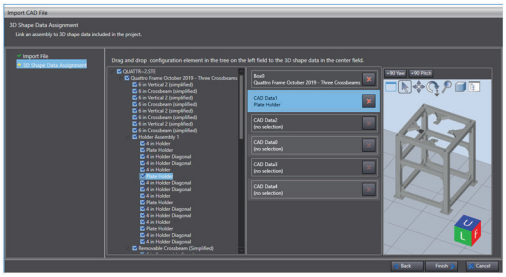
Object	Description
3D Visualization	
Visible	Display or hide the object from the 3D Visualizer.
Collision Program	Select a program that is invoked when the 3D Visualizer detects a collision. Refer to *5-3-4 Graphical Collision Detection* on page 5-29 for more information.
Color	Choose a color for the shape in the 3D Visualizer.
Location	
Rotation Point	Set the offset of the object's center of rotation from the origin.
Offset From Parent	Set an offset distance from a parent object (X Y Z Yaw, Pitch, Roll).
Parent	Select an optional reference to a parent object where the base of the shape will be positioned. If a parent object is selected, when the parent object location moves, this object will be moved accordingly. Using robot objects as Parent will result in the object moving relative to the robot tool flange. This is particularly useful for moving end effectors with the robot tool flange, and visualizing objects held by the gripper. A C# program can be used to manipulate visibility of Visualizer objects (for example, to visualize parts on an end effector while the gripper output signals are ON).
Others	

Object	Description
Category	Defines the category of the CAD Data file in the CAD Library. By default, this field is blank for imported files.
Description	Shows a brief description of the item and its purpose By default, this field is blank in imported files.

8-2-3 Update 3D Shapes

Boxes and Cylinders used to represent objects in the robot workspace may eventually need to be replaced by custom CAD Data. In addition, CAD Data objects may be revised outside of ACE, requiring the object in the workspace to be updated. Both of these tasks can be accomplished using Update All 3D Shapes.

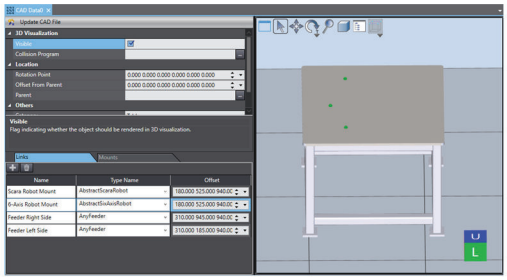
To update 3D Shapes, right-click in 3D Visualization in the Multiview Explorer and select **Update All 3D Shapes**. This opens a wizard similar to importing a custom CAD Data object, refer to *8-2-2 Adding and Configuring CAD Data* on page 8-11. The first updating process step is to select a CAD file saved on the local drive. Once selected, the wizard will allow replacement of the existing 3D Visualization objects as shown in the previous figure. The left side shows the hierarchy of the selected CAD file, including all parts and sub-assemblies. The right side of this is a list shows all the existing 3D Visualization objects. The far right shows the imported file in the 3D Visualizer.



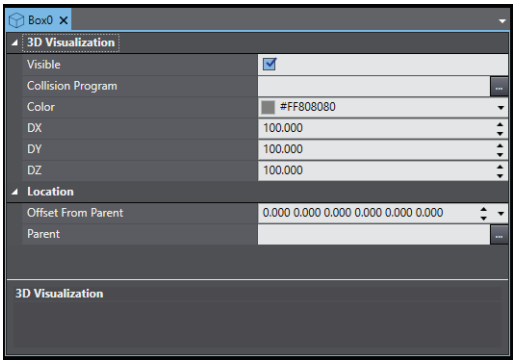
The 3D Visualization objects can be replaced by the imported file or any part or sub-assembly contained within it. To do this, first set the specifications of the imported file by unchecking any parts and sub-assemblies that should not be imported. Also, set the orientation using the **+90 Yaw** and **+90 Pitch** Buttons above the visualizer window. Then click the appropriate entry in the hierarchy and drag it to the respective visualization object. The text beneath the object name will change from (no selection) to the name of the CAD Data. Once the necessary selections are made, click **Finish** to close the wizard and replace the objects. The previous image shows an example of this process. Click **Finish**, and Box0 will be replaced by the Quattro Frame October 2019 – Three Crossbeams assembly and CAD Data1 will be replaced by the Plate Holder part. None of the other objects will be updated.

8-2-4 Connection Points

Each object in the 3D Visualizer can have associated connection points. These allow the user to easily create connections between objects.



There are two types of connection points: Links, where the associated object is the parent item, and Mounts, where the connected item is the parent. For example, if a connection is made where a robot is attached to a table, the connection point for the table will be a link and the connection point for the robot will be a mount. In this way, the table becomes the parent of the robot. Moving the robot will not affect the table, but moving the table will also move the robot.
The links and mounts for an object are accessed in the lower part of 3D Visualization object editors, as shown above. Clicking the tabs above the displayed items will toggle between the two connection types. Connection points have the following properties:



Object	Definition
Name	User-defined name of the connection. This has no functional effect and is primarily used so the user can easily show the purpose of the connection.
Type Name	Defines the type of connection for which the point is designed. Links and mounts will only be able to connect if they have the same Type Name.
Offset	Set the offset of the connection point from the object origin.

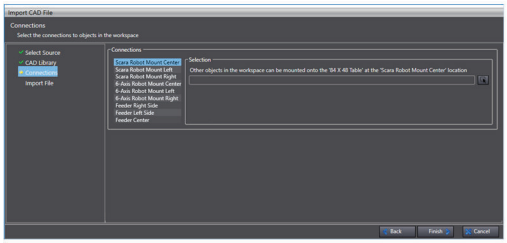
The connections will be displayed in the editor 3D Visualizer window as green dots.
To create a new connection point, open the correct tab in the editor and click the plus button. A new entry will appear in the editor. It is recommended that the name is set to something distinct to differentiate it from others. Click the **Type Name** Drop-down menu and set the type from the following options:

- AbstractHornetRobot: Defines a connection to any Hornet robot. This is primarily used for robot-to-frame connections.
- AbstractQuattroRobot: Defines a connection to any Quattro robot. This is primarily used for robot-to-frame connections.
- AbstractScaraRobot: Defines a connection to any Scara robot. This is primarily used for robot-to-frame or robot-to-table connections.
- AbstractSixAxisRobot: Defines a connection to any six-axis robot. This is primarily used for robot-to-frame or robot-to-table connections.
- AdeptRobot: Defines a connection to a robot end effector.

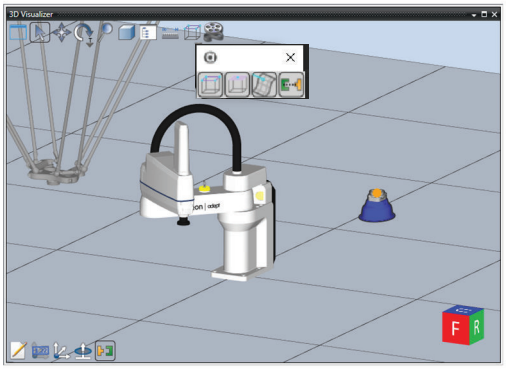
- AnyFeeder: Defines a connection to an AnyFeeder object. This is primarily used for AnyFeeder-to-table connections.
- Box: Defines a connection to any Box object.
- CadData: Defines a connection to any CAD Data object.
- Cylinder: Defines a connection to any Cylinder robot.
- IODrivenEndEffector: Defines a connection to any robot end effector. This is primarily used for grippers.

Connecting Links and Mounts

Mounts can be connected to links in two ways. The first is when a CAD Data object is created from the library. In this case, the last step of the wizard is the Connections step, where connections are selected for the object.



The rows on the left side of the Connections pane show the links and mounts associated with the CAD Library part. Clicking the selection icon on the right enables the choice of an existing 3D Visualizer object to link to the new object. Only objects with connection points that match the highlighted one will appear in the menu. For example, the option selected in the previous image connects to SCARA Robot Mount Center, so only robots of that type will be available to select.
The second method to make a connection is to open the 3D Visualizer and use the Snap feature in the 3D window.



Icon	Name	Definition
	Snap to Edge	Snaps the origin of the selected object or a selected mount point to either an endpoint or the midpoint of the edge of another object.
	Snap to Face	Snaps the origin of the selected object or a selected mount point to the centroid of a face of another object.

Icon	Name	Definition
	Snap to Surface	Snaps the origin of the selected object or a selected mount point to the centroid of a surface of another object.
	Snap to Link	Snaps a mount to a link of the same Type Name. Only links of the selected mount will be visible in the Visualizer.

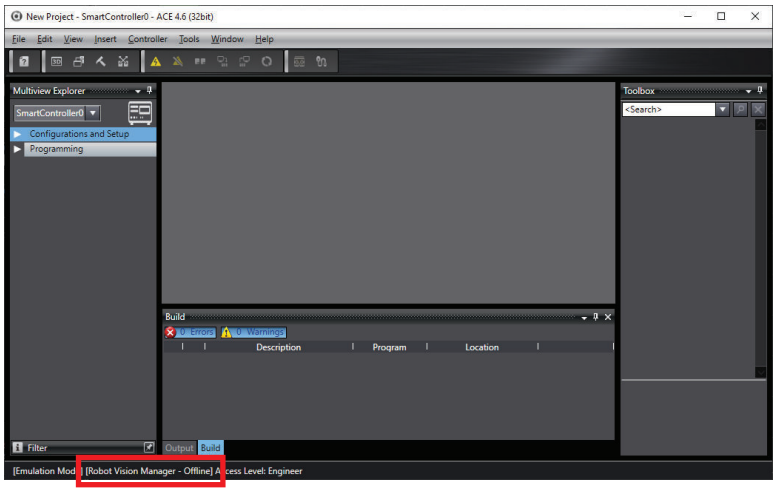
Selecting a Box, Cylinder, or CAD Data object in the Visualizer will allow selection of either Snap To Edge, Snap To Face and Snap to Surface. However, Snap To Link can only be activated by first selecting an existing mount. To do this, select an object with a mount and click the Show/Hide Mount Points icon at the bottom of the window (). Then select a mount point. The link points are shown by hovering the cursor over a linked object, at which point one can be selected. This will snap the object with the mount to the link.
Snapping cannot change the relationship of one of the objects to be the parent of the other. If the two objects need to be moved as a group, the parents need to be set manually using the editors.

8-3 Robot Vision Manager

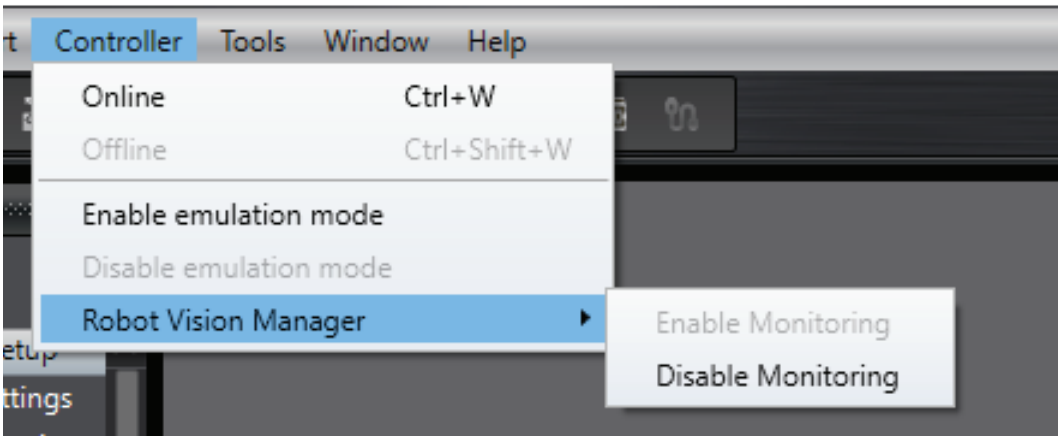
Robot Vision Manager enables the integration of vision and V+ programs. Refer to the *Robot Vision Manager User's Manual (Cat. No. I667)* for more information.
The ACE software provides a collection of Robot Vision Manager objects for the purposes of creating and storing calibrations as well as communicating vision results to V+ programs. These objects provide the following functions.
Robot Vision Manager operation requires that a controller connection exists between the Application Manager and the NJ device.

- Belt Latch Calibration: calibrating a robot to a latch signal. Refer to *Belt Latch Calibration* on page 8-140 for more information.
- Belt Calibration: calibrating a robot to a conveyor belt. Refer to *8-3-5 Belt Calibration* on page 8-23 for more information.
- Camera Calibration: calibrating a robot to a camera. Refer to *8-3-4 Camera Calibration* on page 8-21 for more information.
- Gripper Offset Table: defining the offset on a part to be picked from the actual pick point to the part origin. Refer to *8-3-6 Gripper Offset Table* on page 8-26 for more information.
- Vision Sequence: displays the order and dependency of vision tools while providing program access to the results. Refer to *8-3-7 Vision Sequence* on page 8-27 for more information.
- Overlap Tool: define a method to prevent double processing of belt-relative vision results located in more than one image acquired by a camera. Refer to *8-3-8 Overlap Tool* on page 8-33 for more information.
- Communication Tool: this tool is added to a Vision Sequence and communicates belt-relative vision results to a controller queue for processing by a robot. Refer to *8-3-9 Communication Tool* on page 8-34 for more information.
- Saving calibration data to a file and loading calibration data from a file. Refer to *8-3-2 Saving and Loading Calibration Data* on page 8-19 for more information.

Only one instance of ACE software can run with Robot Vision Manager functions. If an additional instance of the ACE software is started on the PC, *[Robot Vision Manager Offline]* will be displayed in status bar. In this state, any Robot Vision Manager Keywords may return an error.

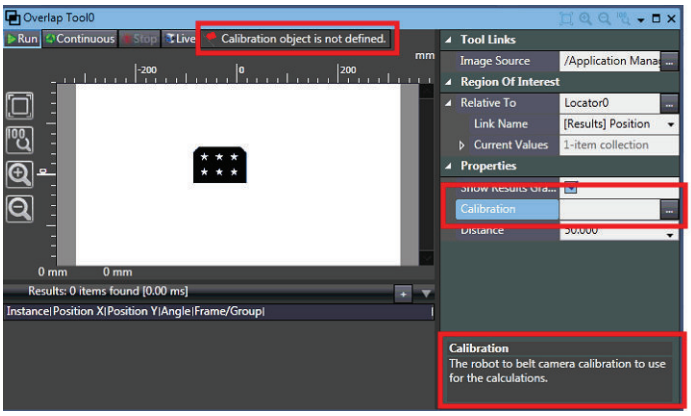


Robot Vision Manager can be enabled or disabled for an application, to select which V+ application receives the Robot Vision Manager communication. Click the **Controller** and then select **Robot Vision Manager**. Once selected, choose **Enable** or **Disable** as shown below.



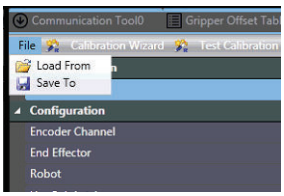
8-3-1 Robot Vision Manager Object Configuration

Many Robot Vision Manager objects are dependent on other ACE software objects. When configuring a new Robot Vision Manager object, the editor will provide information about other objects that may need to be configured or defined. When configuring a new Robot Vision Manager object, the Edit Pane will indicate missing dependencies as shown below (as an example).



8-3-2 Saving and Loading Calibration Data

All Robot Vision Manager calibrations (Belt Latch, Belt, and Camera) can be saved to a file or loaded from a file. Use the File Menu item in the object's editor to save or load a configuration file.



8-3-3 Belt Latch Calibration

Belt 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 on a conveyor belt. This synchronizes the rest of the system with the position of the object.

Requirements

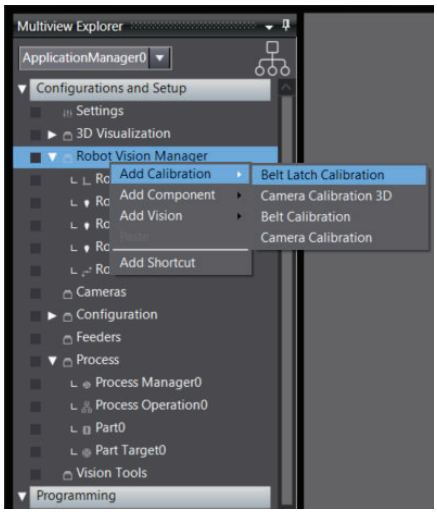
- The PC running the ACE software must be connected to the controller for the robot.
- A belt calibration must be completed.
- The robot, controller, and belt encoder must be properly connected and functioning.
- The belt encoder position latch signal must be configured in robot controller. Refer to *Configure Belt Encoder Latches* on page 7-21 for more information.

A latch signal number is not required while in Emulation Mode.

Add Belt Latch Calibration Object Procedure

Follow the procedure below to add a Belt Latch Calibration object.

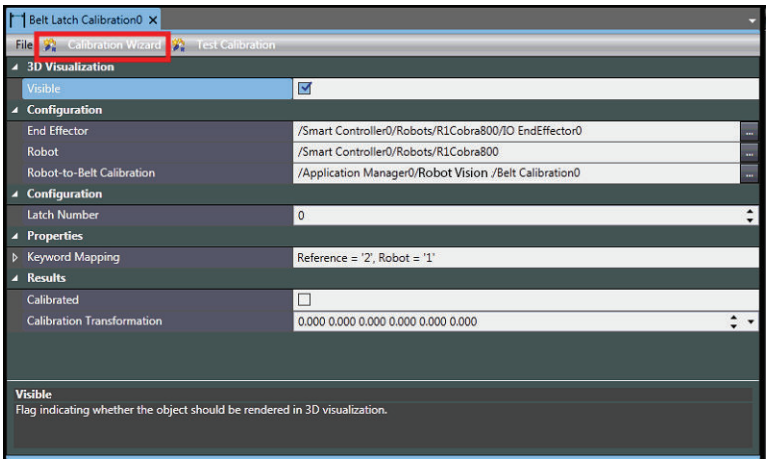
1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Calibration**, and then click **Belt Latch Calibration**. The Robot Vision Manager Robot-to-Belt Latch Calibration Wizard will open.



2 Follow the Calibration Wizard steps to select the robot, end effector, and belt calibration. Clicking the **Finish** Button will create the Belt Latch Calibration object in the Multiview Explorer.

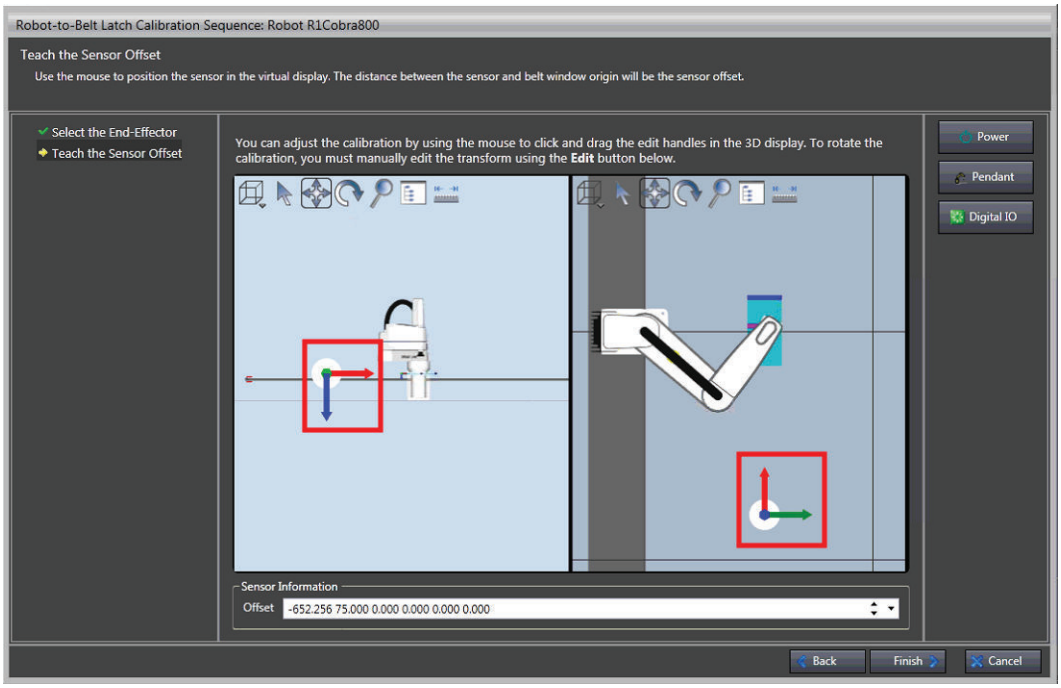
3 Open the new Belt Latch Calibration editor by right-clicking the object and selecting **Edit** or double-clicking the object. The Belt Latch Calibration editor will open in the Edit Pane.

4 Open the Calibration Wizard by clicking the **Calibration Wizard** Button. The Robot-to-Belt Latch Calibration Sequence will open.



5 Make a selection for the end effector and set the latch sensor offset position. After completing these steps, click the **Finish** Button. This will close the Robot-to-Belt Latch Calibration Sequence.

The latch sensor is depicted in the wizard's virtual display as shown below.



6 Review the Belt Latch Calibration object properties in the Belt Latch Calibration editor to confirm the configuration. Use the Robot-to-Belt Latch Calibration Test Sequence to confirm the configuration is correct by clicking the **Test Calibration** Button (Test Calibration). If the configuration is correct, the calibration procedure is complete.

8-3-4 Camera Calibration

Camera Calibration calibrates a robot to a camera. This calibration is necessary when using a vision system with a robot.

Configuring this object will establish a relationship between the following objects (where applicable).

- Camera
- Belt

- Robot
- Robot end effector (robot tool)

Requirements

- The robot, controller, belt (if used), and camera must be correctly connected and functioning.
- The Virtual Camera calibration (mm/pixel) must be complete.
- The PC running the ACE software must be connected to the controller for the robot (and belt).
- The Belt Calibration Wizard must have completed successfully if a conveyor belt is used.

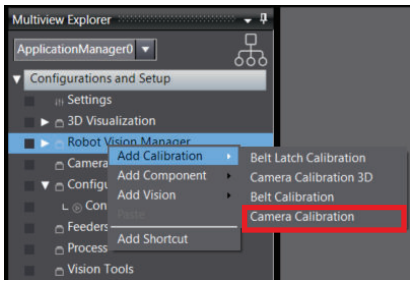
Additional Information

Cameras must be connected and configured before attempting to calibrate them. Refer to *Automation Control Environment (ACE) Version 4 Camera Configuration User's Manual (Cat. No. I679)* for more information.

Add Camera Calibration Object Procedure

Follow the procedure below to add a Camera Calibration object.

1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Calibration** and then select either **Camera Calibration** or **Camera Calibration 3D** as applicable. The Robot Vision Manager Camera Calibration Wizard will open.



2 Follow the Calibration Wizard steps to select the robot, end effector, camera, scenario, camera link, and belt calibration. Clicking the **Finish** Button will create the Camera Calibration object in the Multiview Explorer.

3 Open the new Camera Calibration editor by right-clicking the object and selecting **Edit** or double-clicking the object. The Camera Calibration editor will open in the Edit Pane.

4 Open the Calibration Wizard by clicking the **Calibration Wizard** Button. The Calibration Sequence will open.

Additional Information

The Calibration Sequence will vary depending on the scenario choice selections made during the Calibration Wizard.

5 Make selections for all steps of the Calibration Sequence. After completing these steps, click the **Finish** Button. This will close the Calibration Sequence.

6 Review the Camera Calibration object properties in the Camera Calibration editor to confirm the configuration. Also, use the **Test Calibration** Button to confirm the configuration is correct. If the configuration is correct, the calibration procedure is complete.

8-3-5 Belt Calibration

Belt Calibration calibrates a robot to a conveyor belt. Configuring this object will establish a relationship between the belt, its encoder, and the robot. This calibration is necessary when the robot will handle parts that are moving on a conveyor belt. Refer to *Robot-to-Belt Calibration* on page 1-23 for more information.

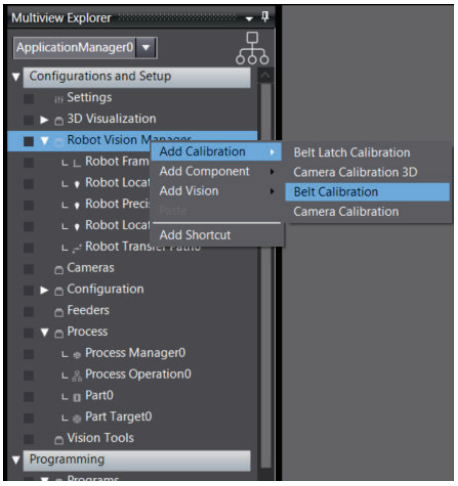
Requirements

- The robot, controller, and belt must be correctly connected and functioning.
- The PC running the ACE software must be connected to the controller for the robot and belt.
- The robot and gripper must be defined in the ACE software

Add Belt Calibration Object Procedure

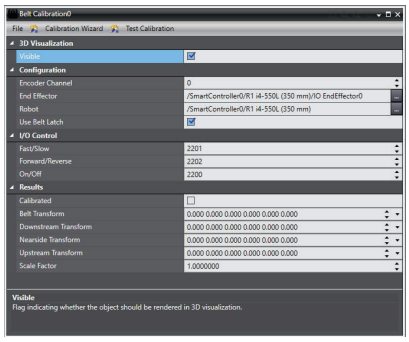
Follow the procedure below to add a Belt Calibration object.

1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Calibration**, and then click **Belt Calibration**. The Robot Vision Manager Robot-to-Belt Calibration Wizard will open.

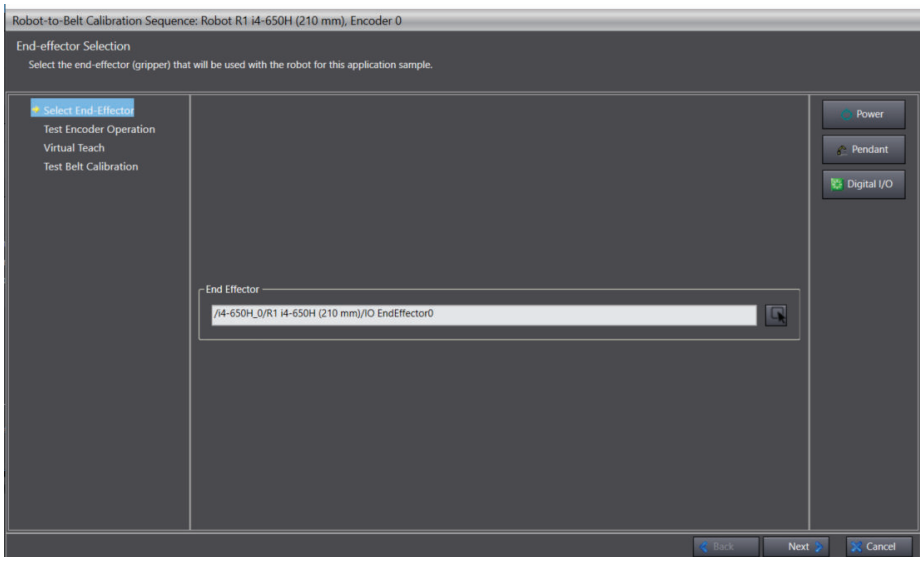


2 Follow the Calibration Wizard steps to select the robot, end effector, and belt calibration. Clicking the **Finish** Button will create the Belt Calibration object in the Multiview Explorer.

3 Open the new Belt Calibration editor by right-clicking the object and selecting **Edit** or double-clicking the object. The Belt Calibration editor opens.

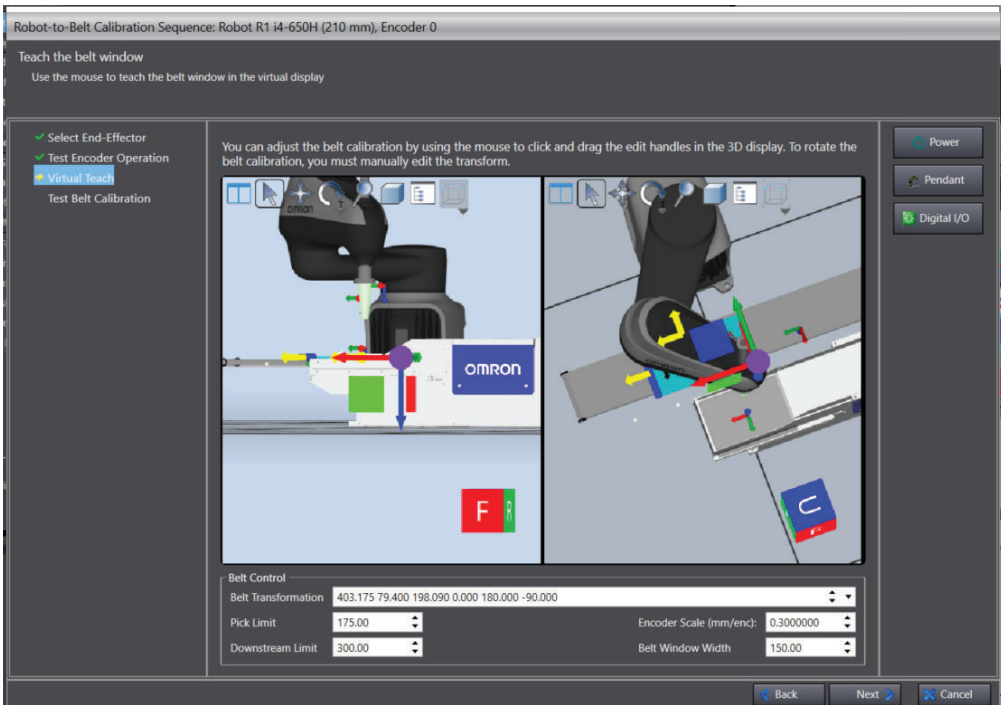


4 Open the Calibration Wizard by clicking the **Calibration Wizard** Tab. The Robot-to-Belt Latch Calibration Sequence opens.



5 Complete the selections and click the **Next** Button.

6 Use the 3D Visualizer display in the Virtual Teach and Test Belt Calibration steps to orient the robot and placement.



7 After completing these steps, click the **Finish** Button. This will close the Robot-to-Belt Calibration Sequence.

Belt Calibration Results

The belt calibration results are used to define the belt area for the robot to access. When using these values in V+ programs, they may need to be adjusted based on the application and will vary based on factors like robot travel times, part flow rates, belt speed, and other timing conditions.
The values of the items below are set during the belt calibration process.

Additional Information

A Robot Vision Manager Application Sample demonstrates how these results can be used in V + programs.

● Belt Transform

The Belt Transform is the frame that is generated from the downstream and upstream transform values to define the orientation of the belt window.

● Downstream Transform

The Downstream Transform is used to define the downstream belt window limit in a V+ program. This sets the downstream threshold where the robot is allowed to access an object. If the robot is tracking a part when it reaches this point, a Belt Window Violation will occur.

● Nearside Transform

The Nearside Transform is the third point taught in the calibration and is used to define the width of the belt window.

- **Upstream Transform**

  The Upstream Transform is used to define the upstream belt window limit in a V+ program.

- **(Encoder) Scale Factor**

  The belt encoder Scale Factor sets the amount of millimeters per encoder count for the belt's encoder.

8-3-6 Gripper Offset Table

The Gripper Offset Table 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.



Additional Information

The Gripper Offset Table can be useful when a robot must pick a part in different poses / orientations located by different models. It may be necessary to create a pick point in a different orientation than the part has been detected.

A gripper can have two types of offsets:

1. A tip offset from the center of the tool flange to the gripper tip(s).
   This offset is established when the robot end effector is defined. This offset will be applied to that gripper whenever it is used, including calculating values for the Gripper Offset Table. Refer to *7-5 Robot Objects* on page 7-40 for more information.
2. The offset(s) from the actual pick point to the part origin which indicates where the robot must pick the part in relation to the origin of the part. This is defined in the Gripper Offset Table and is assigned to a specific robot.

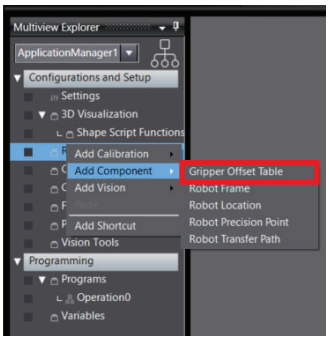
Requirements

- The robot and controller must be correctly connected and functioning.
- The PC running ACE software must be connected to the controller.
- All associated objects such as belts, cameras, and vision tools must be defined and configured if used.

Add Gripper Offset Table Object Procedure

Follow the procedure below to add a Gripper Offset Table object.

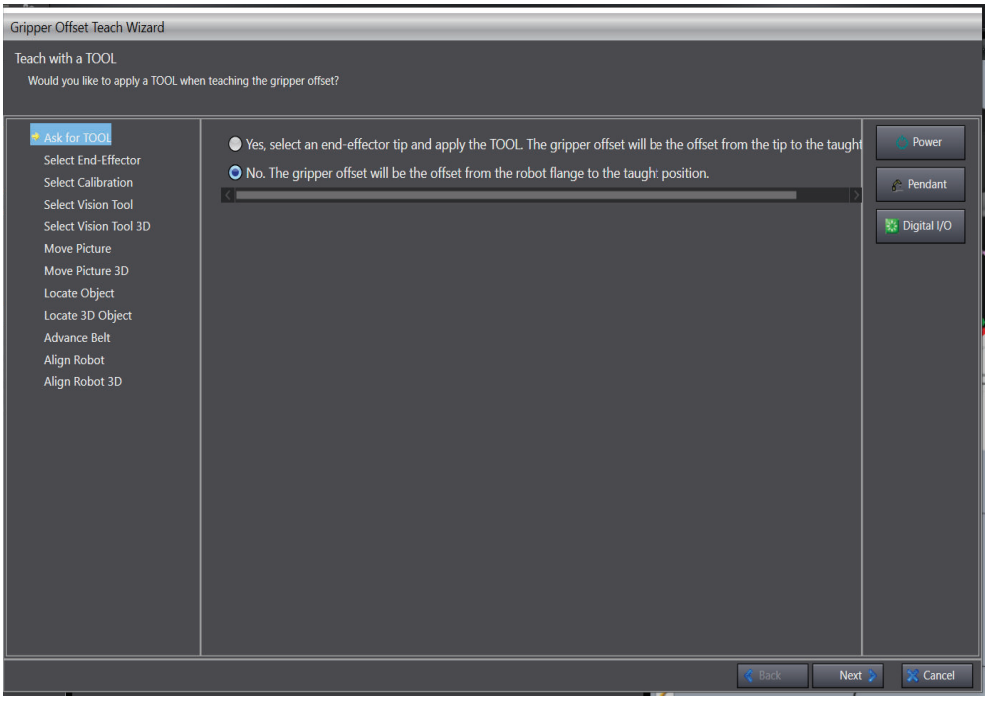
1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Component**, and then click **Gripper Offset Table**. The Gripper Offset Table Creation window will open.



2 Follow the Gripper Offset Table Wizard steps to select the robot. Clicking the **Finish** Button will create the Gripper Offset Table object in the Multiview Explorer.

3 Open the new Gripper Offset Table editor by right-clicking the object and selecting **Edit** or double-clicking the object. The Gripper Offset Table opens.

4 Click the **Teach** Button to open the Editor and change the table.



5 Click **Next** to advance through the Editor, and **Finish** when complete.

8-3-7 Vision Sequence

A Vision Sequence displays the order and dependency of vision tools that will be executed while giving V+ programs a means for retrieving results from vision tools. The Vision 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.
The sequence cannot be modified from the Sequence Display Window. It 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 the Relative To parameter is specified for each of the tools to be included in the sequence.

When a Vision Sequence object is added to the project, the Vision Tool parameter determines the Top-Level tool, and all the tools specified as the Relative To parameter in the chain under that will automatically show up as members of the sequence, in the order they are set.
A robot-to-camera calibration is specified in a sequence. The calibration is applied to any result accessed by a VLOCATION transformation function.
A Vision 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 (this will be the last tool executed). This enables access to a Vision Sequence with the sequence_id parameter from a V+ program command.
V+ programs 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. The Default Calibration is applied to all results, even if they are not the top-level tool.

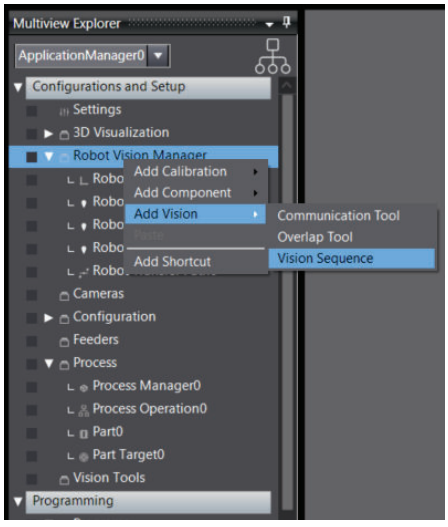
Requirements

- The robot and controller must be correctly connected and functioning.
- At least one Vision Tool must be configured to define in the sequence.
- A camera must be defined in the ACE software.
- A Camera Calibration must be completed if a VLOCATION command is used in a V+ program.

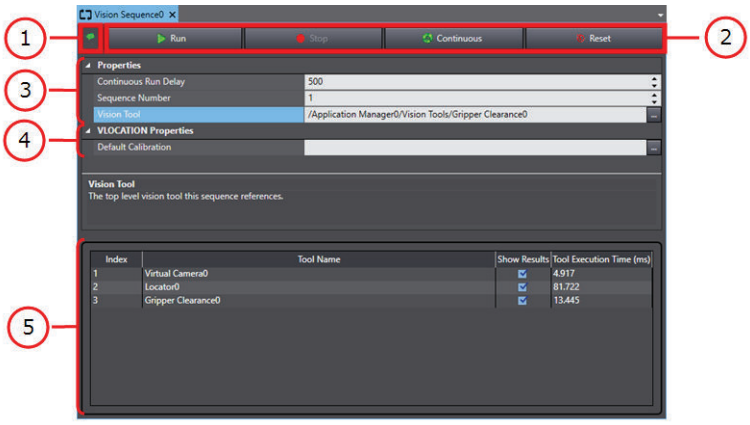
Add Vision Sequence Object Procedure

Follow the procedure below to add a Vision Sequence object.

1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add**, and then click **Vision Sequence**. A Vision Sequence object will be added to the Multiview Explorer.



2 Open the new Vision Sequence editor by right-clicking the object and selecting **Edit** or double-clicking the object. The Vision Sequence editor will open in the Edit Pane.
Items in the Vision Sequence editor are described below.



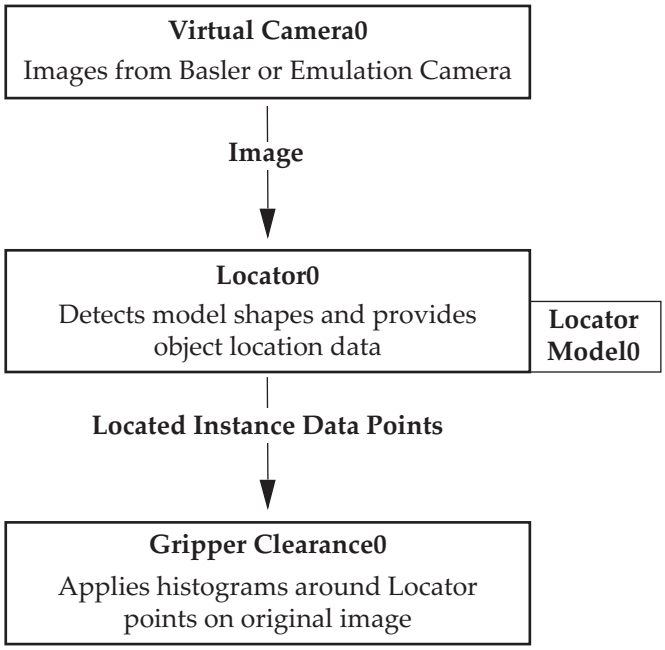
Item	Description
1	Configuration Status • A red flag ( ) indicates incomplete configuration. • A red flag ( ) indicates incomplete configuration. The configuration status will update after the **Run** Button is clicked. Hover over the red flag for information about the incomplete configuration.
2	Sequence Control Buttons • **Run**: Runs the sequence once. • **Stop**: Stops the running sequence. This can only be selected if the sequence is in the middle of a run or the **Continuous** Button has been clicked. • **Continuous**: Runs the sequence repeatedly. • **Reset**: Reset the communication queue, which flushes all instances on the PC that have been identified but not yet sent to a V+ program and resets the Overlap tool tracking of previously-located instances.
3	Properties • Continuous Run Delay: Amount of time in milliseconds to delay between execution of the Vision Sequence in continuous mode. This is the interval at which the Vision Sequence will be executed for V+ conveyor tracking programs using Robot Vision Manager. • Sequence Number: The number associated with the sequence. This sequence number is used in V+ programs to reference this vision operation. • Vision Tool: Top-level vision tool this sequence references. All tools required to operate the tool selected here will be included in the sequence.
4	VLOCATION Properties Default Calibration: Default camera calibration used to apply to a VLOCATION transformation function from V+ programs. This must reference a Camera Calibration object shown in the Multiview Explorer.
5	Sequence Layout The Vision Sequence for the specified Vision Tool.

3 Make the necessary configuration settings in the Edit Pane to complete the procedure for adding a Vision Sequence object.

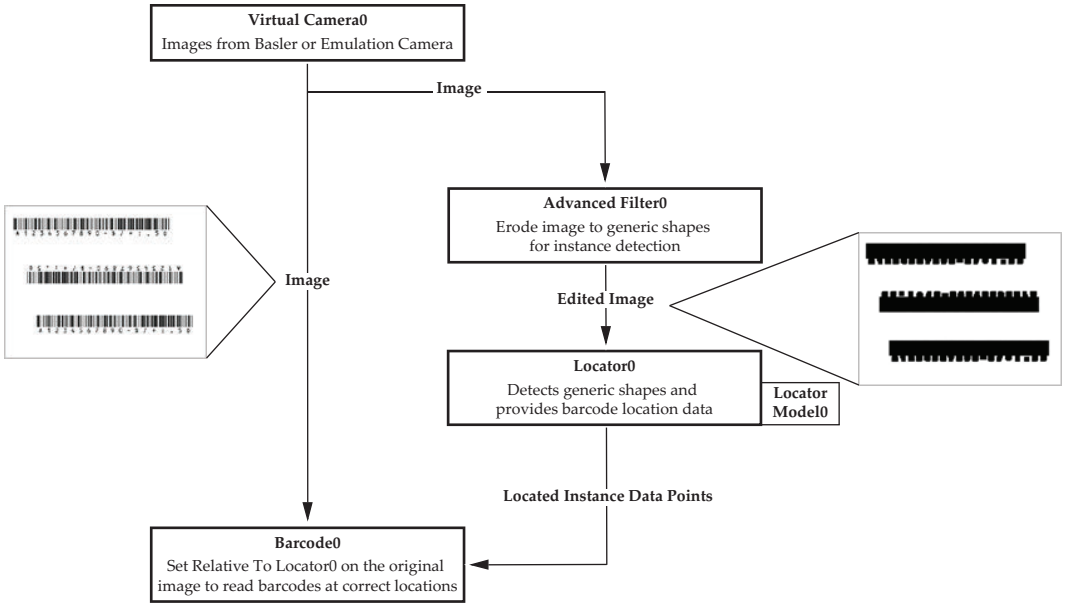
Vision Sequence Configuration

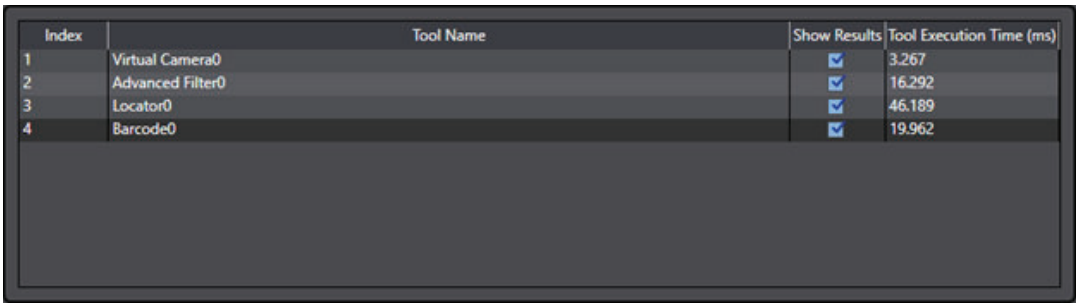
The Vision Sequence editor shows the list of tools that will be executed as part of the sequence, the order in which they will be executed, and the associated index number of each. The tools are executed in ascending Index value.

The main property of a Vision Sequence is the Vision Tool parameter that defines what tool marks the end of the sequence. Once it is selected, the Sequence Layout will be populated by the top-level tool's dependencies down to the initial image source. For example, the Vision Sequence is based on a Gripper Clearance tool that has Relative To set to a Locator tool. These are included by starting with the top-level tool and its Image Source and Relative To properties and working through the same properties of subsequent tools. This is laid out in the following figure that shows how the tools are associated and what data is passed between them.



Another example is shown in the following figure, where the image is modified for instance locating before the bar codes are actually read.





Index	Tool Name	Show Results	Tool Execution Time (ms)
1	Virtual Camera0	☑	3.267
2	Advanced Filter0	☑	16.292
3	Locator0	☑	46.189
4	Barcode0	☑	19.962

The sequence itself cannot be changed in the Vision Sequence editor. The Relative To properties in the tools themselves must be used to do this.
For some applications, it may be beneficial to have multiple sequences where one sequence is a subset of another.
Vision Sequences also provide a means for V+ programs to obtain information from vision tools using a sequence_id parameter. In addition, the robot-to-camera calibration in the Default Calibration property can be accessed using a VLOCATION transformation function. The returned values are provided in the following table.

Additional Information

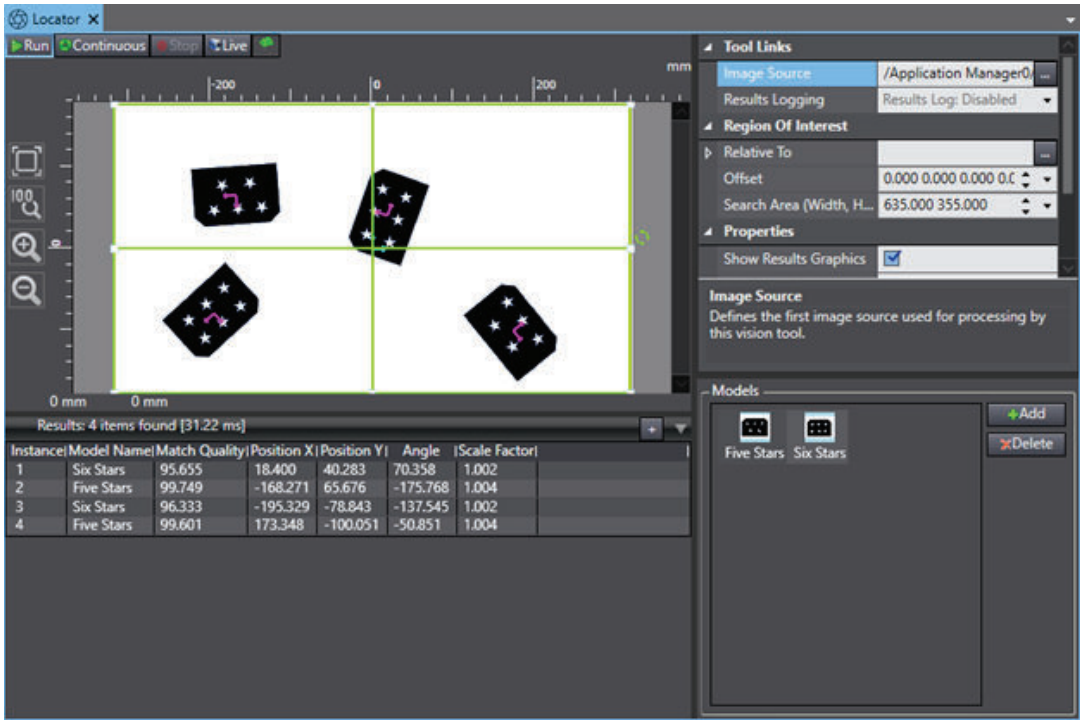
Refer to the *V+ Keyword Reference Manual (Cat. No. I672)* for more information about the correct use of the VLOCATION Keyword.

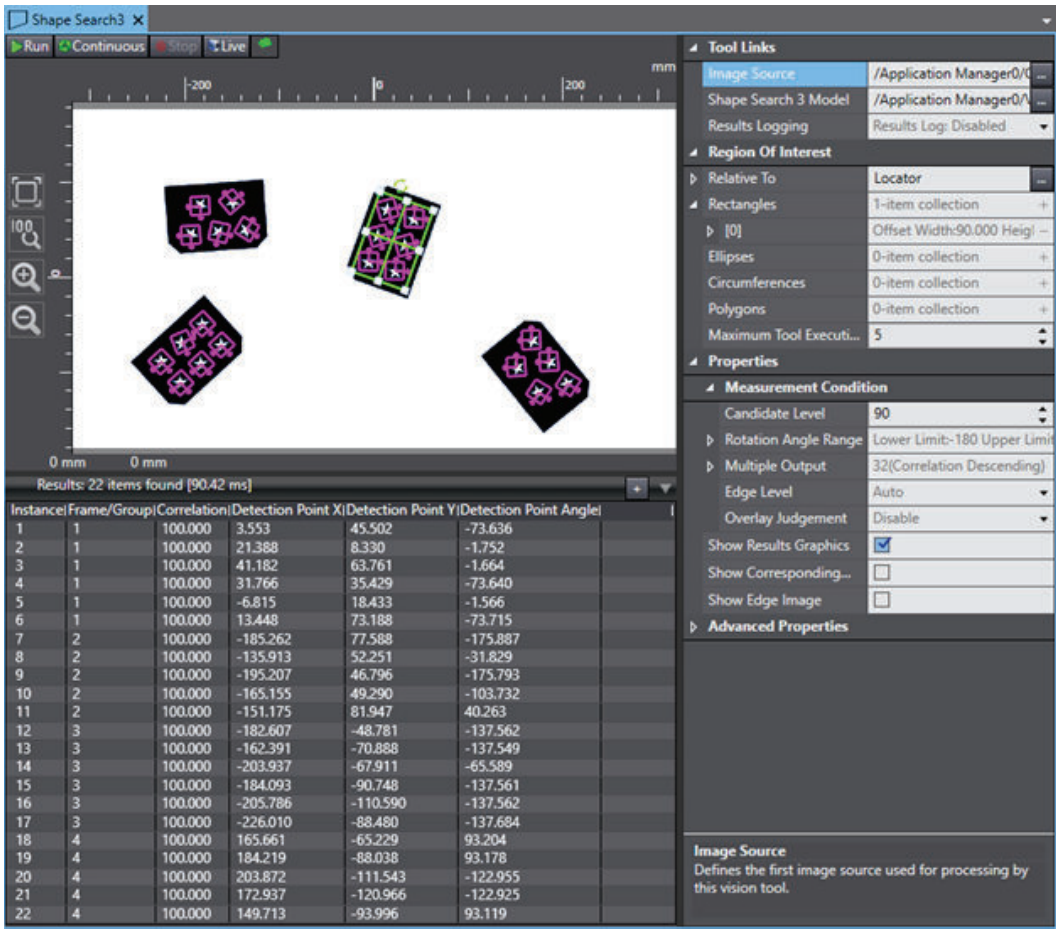
Value	Description
Index	Tool number within the sequence. The tools are run in ascending order.
Tool Name	Name of the tool to run.
Show Results	Defines if the tool results will be shown in the Vision Window. This is based on the Show Results Graphics property of the respective tool. This option is not selectable for tools that do not have the Show Results Graphics property.
Tool Execution Time (ms)	Time (in milliseconds) taken for the tool to run in the most recent execution of the sequence.

Frame and Group Behavior

The Frame / Group result defines the number of the frames referenced in the calculation of the tool. If the tool is not set relative to any other tool, the results for this column will all be returned as 1. However, when the tool is set relative to a tool with more than one returned instance, the Frame / Group value reflects the result instance of the previous tool.
For example, the two figures below show a sequence that is designed to find the stars within the defined shapes. The Locator tool in the first figure below disambiguates between the shapes with five stars and those with six. Then, the Shape Search 3 tool shown in the second figure below locates the stars within the shape.
The Results section shows that the Frame / Group results directly correlate with the Instance result from the Locator in the first figure below. Instances 1 through 6 in the second figure below are in

Frame / Group 1 since they are located with respect to Instance 1 in the first figure below. Instances 7 through 11 are in Frame / Group 2 for the same reason. The other regions correlate in the same way.





The Frame / Group can also be used as an argument for the VLOCATION transformation function in V + program code to limit the returned location to a particular instance in another tool. The syntax of the VLOCATION transformation is provided below.

```
VLOCATION ($ip, sequence_id, tool_id, instance_id, result_id, index_id, frame_id)
```

If there is an argument for frame_id, then instance_id will be evaluated based on that frame. Otherwise, it will be evaluated with reference to all instances. For example, the following code returns the location (21.388, 8.330, 0, 0, 0, -1.752) based on Instance 2 in the results of the second figure above.

```
VLOCATION ($ip, 1, 3, 2, 1311, 1)
```

However, when the frame_id is added as shown below, the line returns (-135.913, 52.251, 0, 0, 0, -31.829) since that is Instance 2 of Frame / Group 2.

```
VLOCATION ($ip, 1, 3, 2, 1311, 1, 2)
```



Additional Information

Refer to the *V+ Keyword Reference Manual (Cat. No. I672)* for more information about the correct use of the VLOCATION Keyword.

8-3-8 Overlap Tool

The Overlap Tool ensures that parts moving on a conveyor belt are processed only once if located in multiple images. A part found by the Locator Vision Tool (or other input tools) may be present in multiple images acquired by the camera and this tool ensures that the robot is not instructed to pick up or process the same part more than once. The input required by the Overlap Tool can be provided by any tool that returns a transform instance. This tool is typically used in conveyor tracking applications.



Additional Information

It is often desirable to have images taken at an interval that is one-half of the field of view of belt travel. This makes it possible to get two images of the same part. In this case, the Overlap Tool can be used to filter out duplicates.

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.
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. Refer to *8-3-7 Vision Sequence* on page 8-27 for more information.

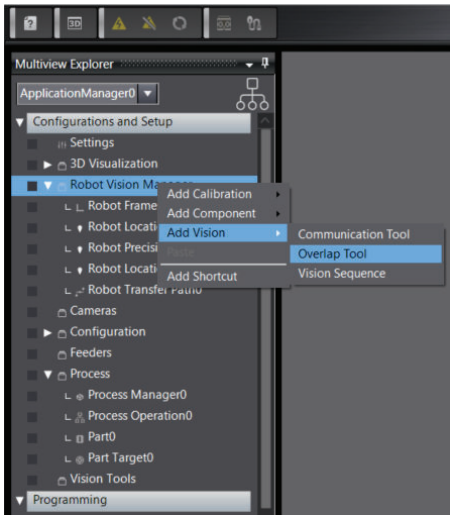
Requirements

- The camera, robot, and conveyor belt are calibrated, connected, and correctly functioning.
- The PC running the ACE software must be connected to the controller for the robot and belt.
- The tool is receiving latched values from the input tool. The belt latch must be wired to the controller and properly configured.
- A vision input tool is defined and configured.
- The conveyor belt and the controller have been correctly assigned to a camera.

Add Overlap Tool Object Procedure

Follow the procedure below to add an Overlap Tool object.

1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Vision**, and then click **Overlap Tool**. An Overlap Tool object will be added to the Multiview Explorer.



2 Open the new Overlap Tool editor by right-clicking the object and selecting **Edit** or double-clicking the object. The Overlap Tool editor will open in the Edit Pane.

3 Make all necessary settings in the Overlap Tool editor. When all settings are completed, the Overlap Tool object configuration procedure is complete.

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.

- 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.

Additional Information

5 mm is a recommended starting value for the Distance parameter.

Rotation is ignored by the Overlap Tool. Only the difference in X and Y is considered.

8-3-9 Communication Tool

The Communication Tool is a tool for conveyor tracking applications. The purpose of the Communication Tool is to transfer belt-relative vision results into a controller queue for processing by a robot.
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 input to the Communication Tool can also be provided by other tools that output instances, such as an Inspection or a Locator Tool. The Communication Tool processes the input instances by applying region-of-interest parameters.

The Communication Tool acts as a filter in the following way.

- 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, such as another communication tool.

Order of the Communication Tool in a Vision Sequence

In a simple vision-guided, belt-tracking 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.

Using Multiple Communication Tools

In many applications, it may be useful to use two or more Communication Tools. Examples when multiple Communication Tools are necessary are provided below.
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.

- Use two Communication Tools for managing either side of a conveyor belt. Each Communication Tool sends instances to a robot that picks parts on one side of the belt only.
- Use 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.
- Use multiple Communication Tools to send instances to multiple robots positioned near a single conveyor belt with a single camera.

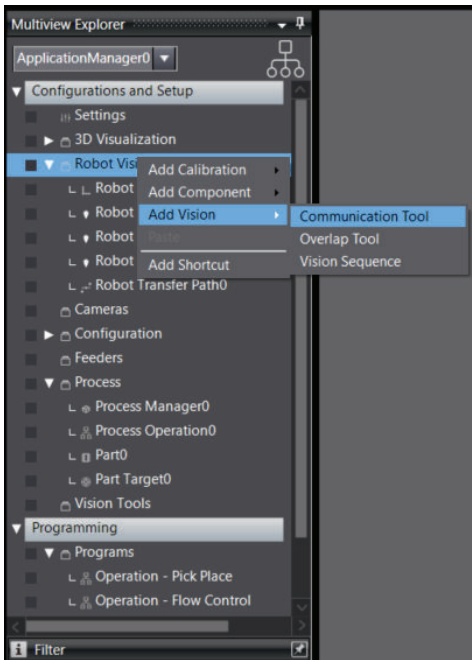
Requirements

- The camera, robot, and conveyor belt are calibrated, connected, and correctly functioning.
- The conveyor belt and the controller have been correctly assigned to a camera.

Add Communication Tool Object Procedure

Follow the procedure below to add a Communication Tool object.

1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Vision**, and then click **Communication Tool**. A Communication Tool will be added to the Multiview Explorer.



2 Open the new Communication Tool editor by right-clicking the object and selecting **Edit** or double-clicking the object. The Communication Tool editor will open in the Edit Pane.

3 Make all necessary settings in the Communication Tool editor. When all settings are completed, the Communication Tool object configuration procedure is complete.

Communication Tool Parameter Details

The Communication Tool has unique parameters that are described below.

Search Area

Search Area is the size of the region of interest is defined the width and height of 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, an 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 region of interest.

Robot

The Robot parameter selects the robot that will handle or pick the instances output by the Communication tool.

Ensure the selected robot and the robot of the selected camera calibration are the same. If not, the transformation may be invalid.

Queue Parameters

The Communication Tool sends instances that pass its criteria to its queue, which is configured with the following parameters.

Queue Index

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. Choose a value from 1 to 100.

In a V+ program, this queue index must be used to access the instances sent to the controller by the communication tool. For example, in the Robot Vision Manager application sample for belt camera pick to a static place, the rob.pick program will use pick.queue variable to store the queue index used when obtaining instances. This occurs with the following V+ program call.

```
CALL getinstance(pick.queue, peek, inst.loc, model.idx, encoder.idx, vision.x, visi
on.y, vision.rot)
```

Queue Size

Queue Size specifies the number of instances that can be written to the queue. The ideal queue size varies greatly and may require some testing to optimize this value for a specific application and environment. Choose a value from 1 to 100.

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. The Queue Update options are described below.

- After Every Instance: The After Every Instance setting sends 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+ 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 to the queue at once. 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.

Both Queue Update settings have the same function when only one instance is found.

Use Soft Signal

Select this option to enable the Soft Signal.

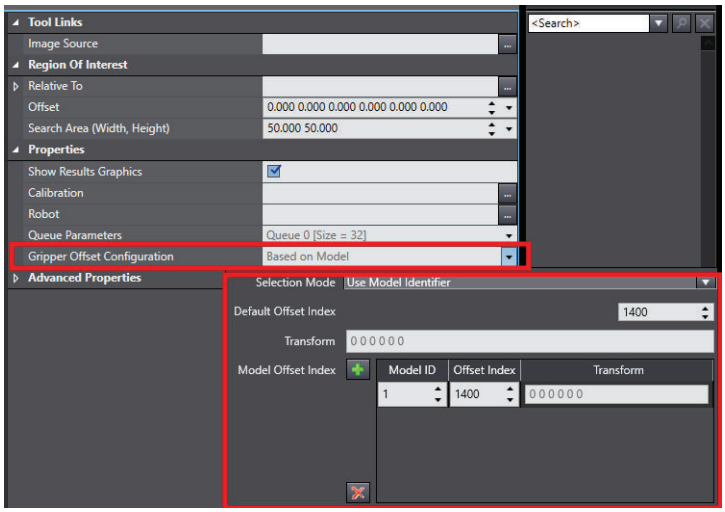
Soft Signal

This sets the value of the Soft Signal to use when Use Soft Signal is enabled. The signal can be used by a V+ program 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 Offset Configuration

This specifies the method and details needed for determining the offset index of the gripper. Refer to *8-3-6 Gripper Offset Table* on page 8-26 for more information.

Use the item descriptions below to make the appropriate configuration settings.



Selection Mode

Make one of the following selections as described below.

Disabled: No Gripper Offset is applied.

Use Default: Use the value set in the Default Offset Index field.

Use Model Identifier (Model ID): Use the values set in the Model Offset Index area.

Default Offset Index

This specifies the index in the Gripper Offset Table to apply as the gripper offset. This setting can be from 1400 to 1499.

Model Offset Index

Use this area to create associations between Model IDs from a Locator Model and Gripper Offset Table index numbers. This refers to the Custom Model Identifier property of the Locator Model.

Use the **Add** Button () and **Remove** Button () to create necessary associations.

Tool Relative Coordinates

Selecting this option indicates that locations should be returned relative to the robot tip position when the picture was taken. This is only used if the selected camera calibration was performed with the calibration object attached to the robot tool (an upward looking camera for example).

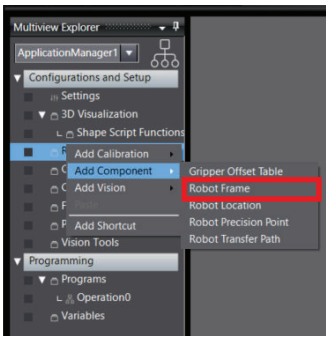
8-3-10 Robot Frame

Robot frames (also known as reference frames) are useful because they allow users to teach locations relative to the frame. If the location of the frame changes in the workcell, simply update the frame information to reflect its new location. It is then possible to use any locations created relative to that frame without further modifications.

Add Robot Frame

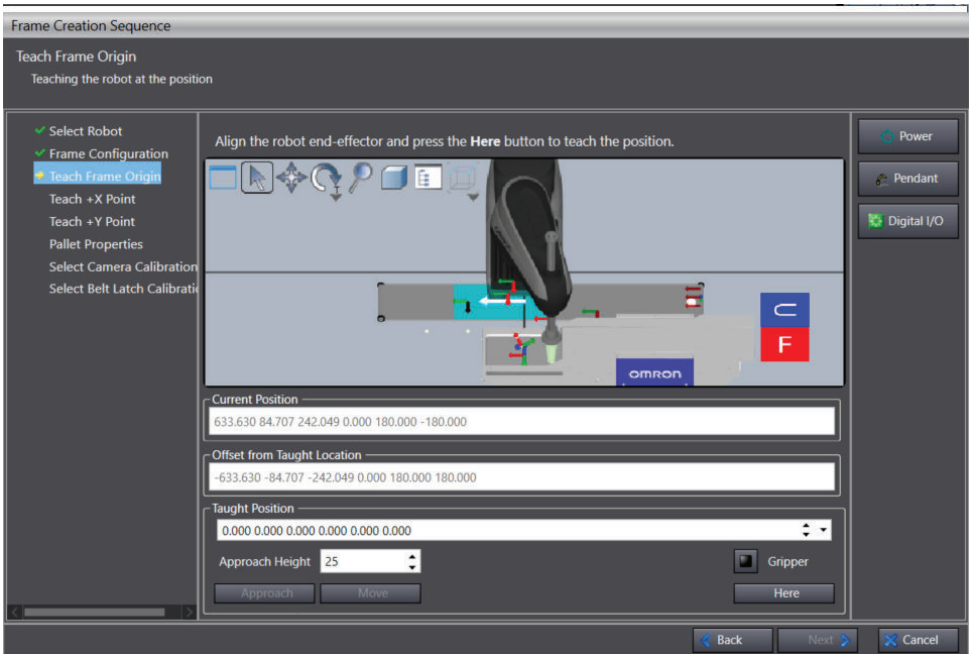
Follow the procedure below to add a Robot Frame object.

1. Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Component**, and then click **Robot Frame**. The *Frame Creation Sequence* Window opens.



2 Follow the Frame Creation Sequence Wizard steps. clicking **Next**, at the end of each step.

3 Teach the points by placing the cursor at the tool tip, left-clicking and dragging the tip to the location. Then release the left button and click **Here**. Click **Next**, to move to the next step.



4 After the last step, click the **Finish** Button.

5 Edit the Robot Frame by selecting it in the Multiview Explorer view, right-clicking and selecting **Edit**.

6 Click the **Teach** Button to open the *Frame Creation Sequence* to make the edits.

8-3-11 Robot Location

The Robot Location is a Cartesian location relative to the world coordinate system, another location, or a frame of reference

Requirements

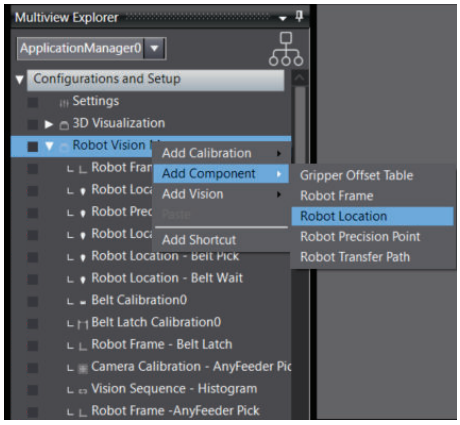
- The robot and controller must be correctly connected and functioning.

- The PC running ACE software must be connected to the controller.
- All associated objects such as belts, cameras, and vision tools must be defined and configured if used.

Add Robot Location

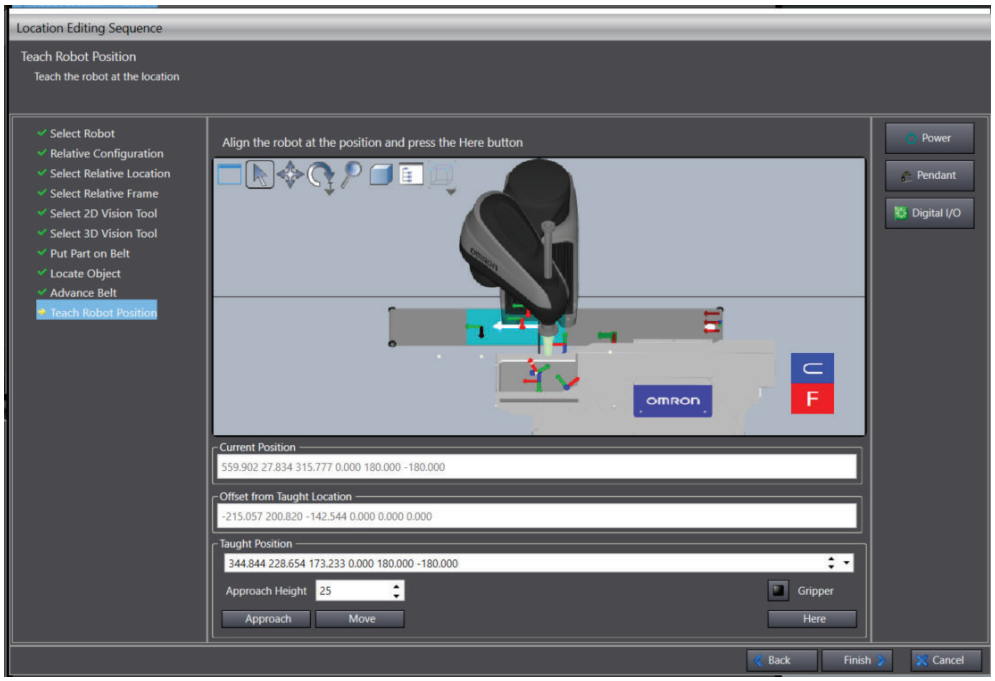
Follow the procedure below to add a Robot Location object.

1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Component**, and then click **Robot Location**. The *Location Creation Sequence* Window opens.



2 Follow the Location Creation Sequence Wizard steps. clicking **Next**, at the end of each step.

3 Teach the location by placing the cursor at the tool tip, left-clicking and dragging the tip to the location. Then release the left button and click **Here**. Click **Next**, to move to the next step.



4 After the last step, click the **Finish** Button.

5 Edit the Robot Location by selecting it in the Multiview Explorer view, right-clicking and selecting **Edit**.

6 Click the **Teach** Button to open the *Frame Creation Sequence* to make the edits.

8-3-12 Robot Precision Point

A Robot Precision Point describes a location in space by the joint angles of the robot. Because it is described via joint angles, it will always be the same, independent of the tool offset.

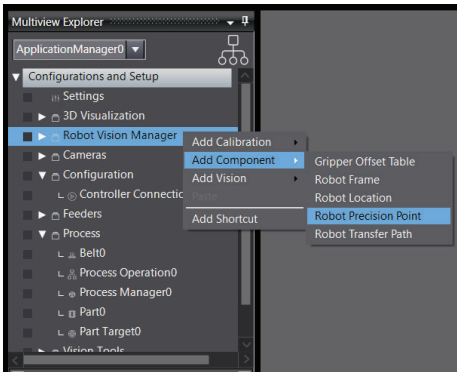
Requirements

- The robot and controller must be correctly connected and functioning.
- The PC running ACE software must be connected to the controller.
- All associated objects such as belts, cameras, and vision tools must be defined and configured if used.

Add Robot Precision Point

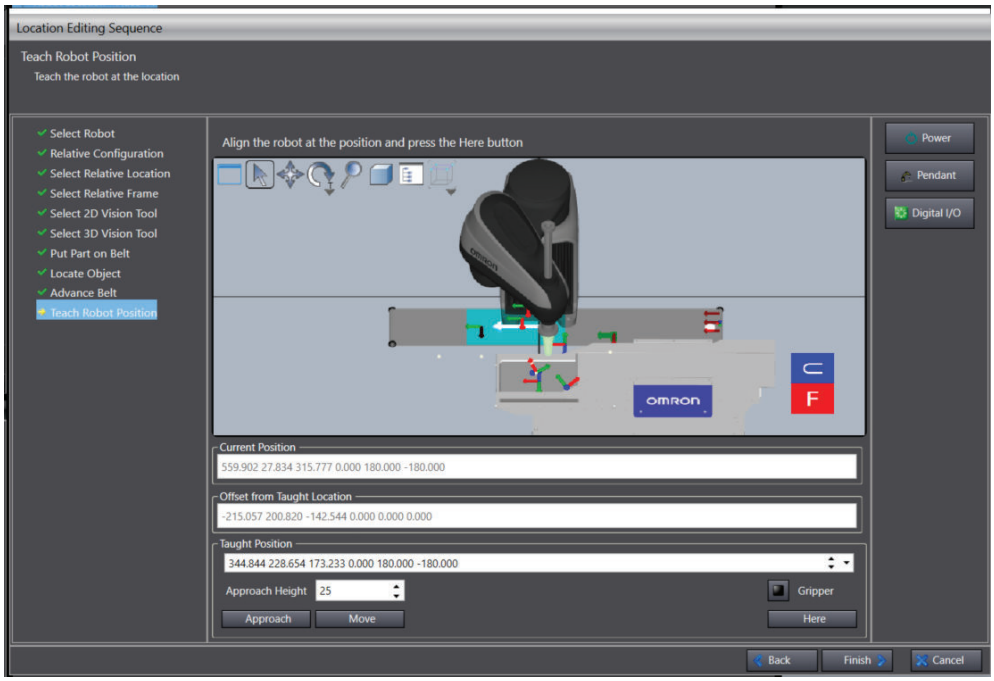
Follow the procedure below to add a Robot Precision Point object.

1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Component**, and then click **Robot Precision Point**. The *Teach Position* Window opens.



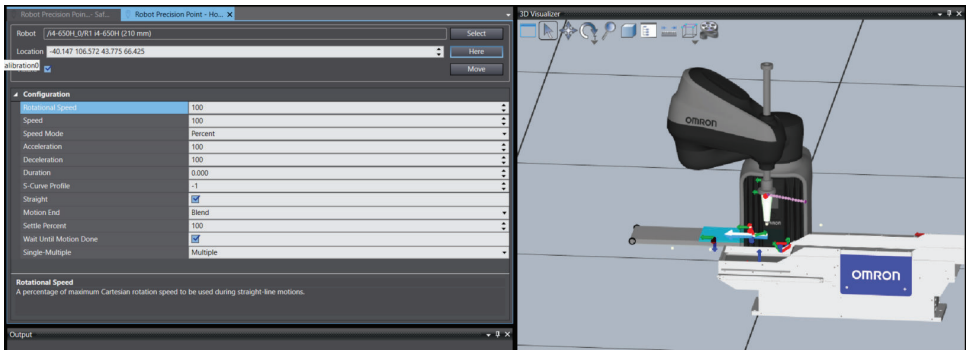
2 Follow the Teach Position Wizard steps. clicking **Next**, at the end of each step.

3 Teach the position by placing the cursor at the tool tip, left-clicking and dragging the tip to the location. Then release the left button and click **Here**. Click **Next**, to move to the next step.



4 After the last step, click the **Finish** Button.

5 Edit the Robot Location by selecting it in the Multiview Explorer view, right-clicking and selecting **Edit**.



6 The position can be changed by dragging the robot tool tip to a new location or manually entering new location values. Click the **Here** Button to store the new position.

7 Additional parameters can be changed in the editor. Once completed, close the editor.

8-3-13 Robot Transfer Path

A transfer path is a series of locations, when combined form a path, that the robot follows from one point to another while avoiding obstacles. A transfer path describes a path through empty space, free of any obstacles. The robot will auto-calculate the point in the path closest to the robots current location and the point closest to the target location, which are called the Entry and Exit Points. The robot will go first to the Entry point, then follow the path to the Exit point where it will exit the path and go to the target destination.

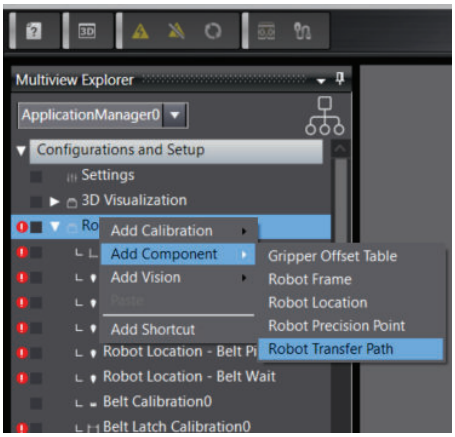
Requirements

- The robot and controller must be correctly connected and functioning.
- The PC running ACE software must be connected to the controller.
- All associated objects such as belts, cameras, and vision tools must be defined and configured if used.

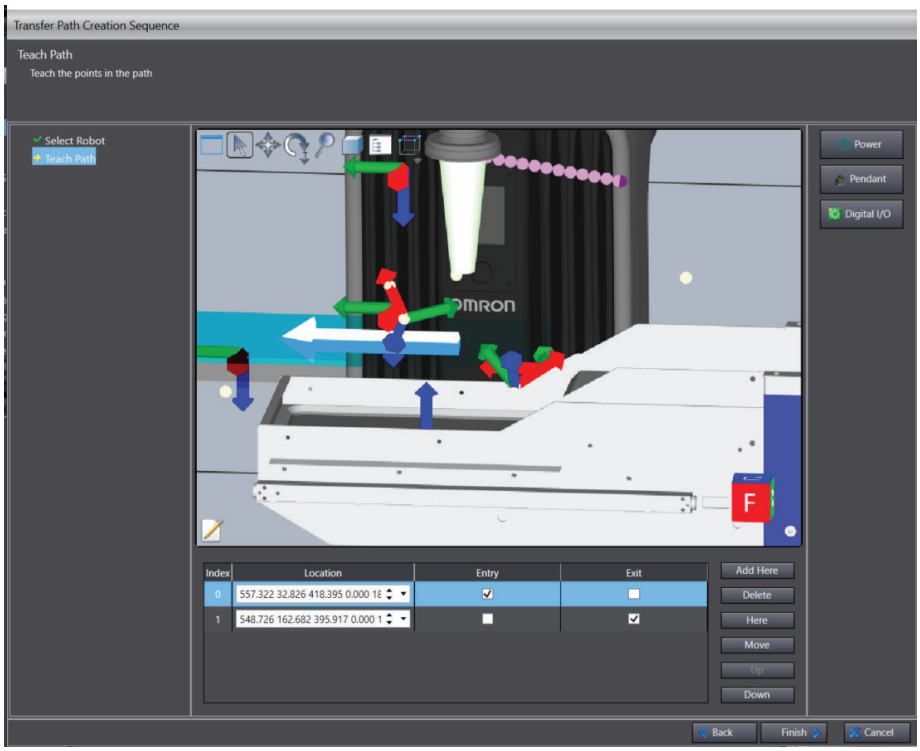
Add Robot Transfer Path

Follow the procedure below to add a Robot Transfer Path object.

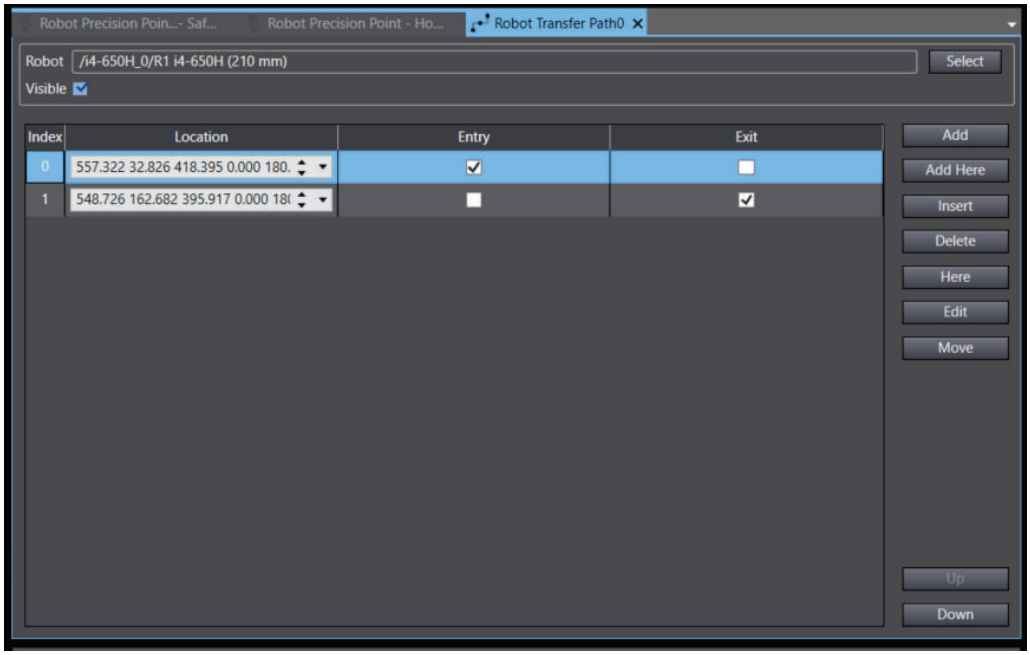
1 Right-click **Robot Vision Manager** in the Multiview Explorer, select **Add Component**, and then click **Robot Transfer Path**.



2 The Robot Transfer Path Creation window opens. Follow the Robot Transfer Path Wizard steps to create the path. Clicking the **Finish** Button will create the Robot Transfer Path object in the Multiview Explorer.



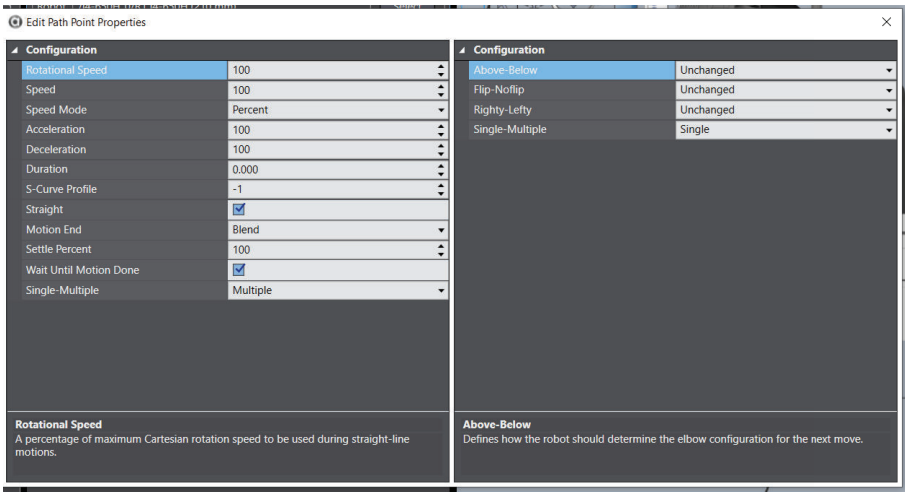
3 Edit the Robot Transfer Path by selecting it in the Multiview Explorer view, right-clicking and selecting **Edit**.



4 Use the **Add** Button to add a location to the path. Left-clicking and dragging the tool tip and then clicking the **Add Here** Button also adds another location in the path. Once added, check the box to identify it as an Entry or Exit point.

5 Insert a new location by clicking the **Insert** Button. Select a location by clicking in the Index column and then click **Delete**, to remove that location.

6 Click **Edit** to open the Path Point Properties. When finished making changes, close the editor.


Edit Path Point Properties
Configuration
Rotational Speed
100
Speed
100
Speed Mode
Percent
Acceleration
100
Deceleration
100
Duration
0.000
S-Curve Profile
-1
Straight
Motion End
Blend
Settle Percent
100
Wait Until Motion Done
Single-Multiple
Multiple
Rotational Speed
A percentage of maximum Cartesian rotation speed to be used during straight-line motions.
Configuration
Above-Below
Unchanged
Flip-Noflip
Unchanged
Righty-Lefty
Unchanged
Single-Multiple
Single
Above-Below
Defines how the robot should determine the elbow configuration for the next move.

8-4 Camera Objects

Camera objects can be added to the ACE project to provide a method for obtaining an image to be used for processing. A camera object can be either a physical camera, virtual camera, emulated camera, or a visualization capture device. Multiple camera objects can be added to an ACE project.
Use the information in this section to understand the different camera object types and their configurations in an ACE project.
To add a camera object to the Application Manager in the Multiview Explorer, right-click **Cameras**, select **Add** and then choose a camera type. A new camera object will be added to the Multiview Explorer.

Additional Information

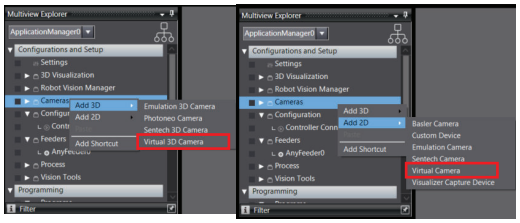
When adding an Emulation camera for a Basler Camera, Sentech Camera, Phoxi 3D camera or Custom Device, there is also the option to also add a Virtual Camera. A Virtual Camera is typically required for most applications.

8-4-1 Virtual Cameras

The Virtual Camera object provides an interface between a Vision Sequence and the object used to acquire an image. The Virtual Camera object is typically used as the Image Source reference for vision tools (except when Image Processing tools are used).
The Virtual Camera object editor provides access to pixel-to-millimeter calibration data, acquisition settings, image logging, and references for the camera object to use for acquiring an image. The Default Device setting for the Virtual Camera designates the object used to acquire an image. When configured properly, this can provide a seamless transition between a physical camera and an Emulation Camera without changing vision tool settings.

Adding a Virtual Camera

To add a Virtual Camera, right-click **Cameras**, select either **Add 3D** or **Add 2D**, and then click **Virtual Camera**. A new Virtual Camera will be added to the Multiview Explorer.

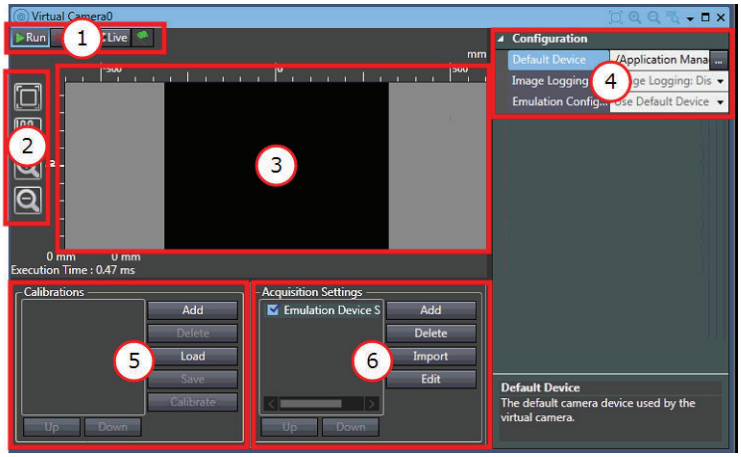


Additional Information

- The option to add a Virtual Camera is present when adding an Emulation Camera, Basler, Sentech, FH and Photoneo Camera or a Custom Device. This is typically how a Virtual Camera is added to the ACE project.

Virtual Camera Configuration

To access the Virtual Camera configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Virtual Camera editor in the Edit Pane.



Item	Description
1	Camera Control Buttons • The **Run** button will acquire the image once from the camera device specified. • The **Stop** button will stop continuous Live image acquisition. • The **Live** button will start continuous image acquisition from the camera device specified.
2	Camera Image View Adjustments Use these buttons to adjust the image view area. An alternative is to use the mouse scroll to zoom in and out.
3	Camera Image View This area shows the most recent image acquired by the Virtual Camera.
4	Virtual Camera Configuration Items This area is used to configure the Virtual Camera. Refer to *Virtual Camera Configuration Items* on page 8-47 for more information.
5	Virtual Camera Calibrations Area This area is used to calibrate the Virtual Camera. Refer to *Grid Calibration with Sample Dot Target Files* on page 8-50 for more information.
6	Virtual Camera Acquisition Settings This area is used to adjust image acquisition settings for the camera. Refer to *Acquisition Settings* on page 8-52 for more information.

Virtual Camera Configuration Items

The following sections provide detail on each camera configuration item.

● Default Device

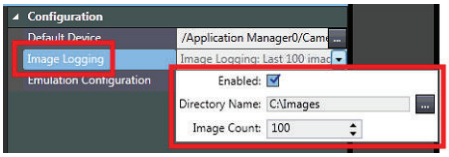
Select a default camera device used by the Virtual Camera (when not in Emulation Mode).

● Image Logging

When logging images from a physical, belt-relative camera for use in Emulation Mode, record the average belt velocity and picture interval in effect while logging images. These two pieces of information are necessary for the picture spacing on an emulated belt to be similar. If using a Process Manager, the belt velocity can be recorded from the System Monitor and the picture interval can be

recorded from Process Manager in the Control Sources area. If using a V+ program and Robot Vision Manager, the belt velocity can be recorded from the Controller Settings - Encoders area and the picture interval can be recorded from the Robot Vision Manager Sequence, Continuous Run Delay area.

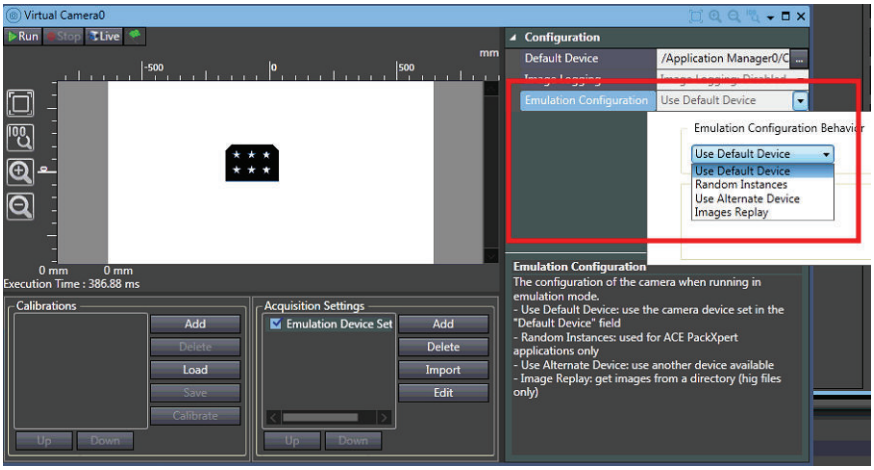
Without recording using this information, the replayed images may be overlapping and part flow may be different than expected. When using the logged images in emulation, be sure to apply these values in the appropriate fields for the flow of images to match the physical system.



Item	Description
Enabled	Enable or disable the image logging function with this selection.
Directory Name	Select a directory on the PC where images will be stored.
Image Count	Enter the number of images to store. Up to 1000 images can be stored.

● Emulation Configuration

When Emulation Mode is enabled, the Virtual Camera object editor uses an Emulation Configuration parameter that specifies one of the following modes for the image source.



Emulation Configuration Behavior	Description
Use Default Device	This setting will use the camera device specified in the Default Device field.

Emulation Configuration Behavior	Description
Random Instances	This setting allows generation of a random quantity of randomly oriented vision results. Specify the range of the possible instance count with minimum and maximum values, but there is no control over the random X, Y, and Roll values of the results generated. If this level of control is required, consider using a Custom Vision tool with user-defined logic. • When using this mode, vision tools will display an error *There is no image in the buffer*. • 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.
Use Alternate Device	This setting allows pictures to be obtained from another vision device that is configured in the project (typically an Emulation camera object). Select the alternate vision device with this setting if required.
Images Replay	This setting allows a set of images to be displayed from a specified directory (.hig files only).

Virtual Camera Calibration

Virtual camera calibration is required before vision tools are created to ensure that accurate measurements and locations are acquired. This is a spatial adjustment that corrects for perspective distortion, lens distortion, and defines the relationship between the camera pixels and real-world dimensions in millimeters. There are two methods available for virtual camera calibration as described below.

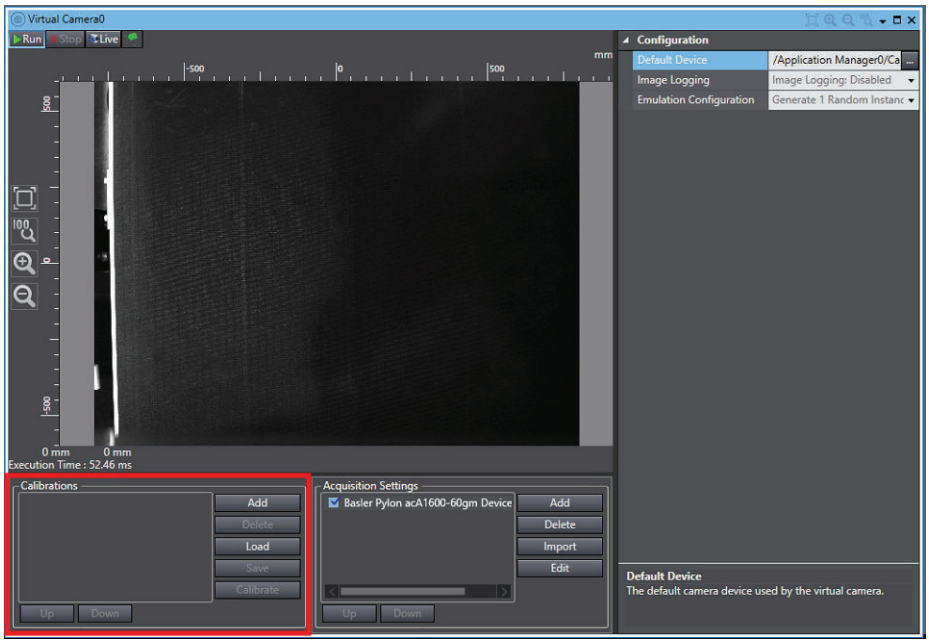


Additional Information

- The camera should be calibrated before creating any vision tools.
- 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. Refer to *8-3-4 Camera Calibration* on page 8-21 for more information.

Access the camera calibration in the Virtual Camera's editor that is associated with the camera device. An example is provided below.

- Click the **Add** Button to begin the calibration procedure. Click the **Delete** Button to remove a previous calibration.
- Click the **Load** Button to load a previously saved calibration. Click the **Save** Button to save an existing calibration to the PC.
- Click the **Calibrate** Button to adjust an selected calibration.



● Grid Calibration with Sample Dot Target Files

Calibrate the camera using a grid with known spacing. Sample dot target files are provided with the ACE software installation. Find these in the default installation directory with the following file names:

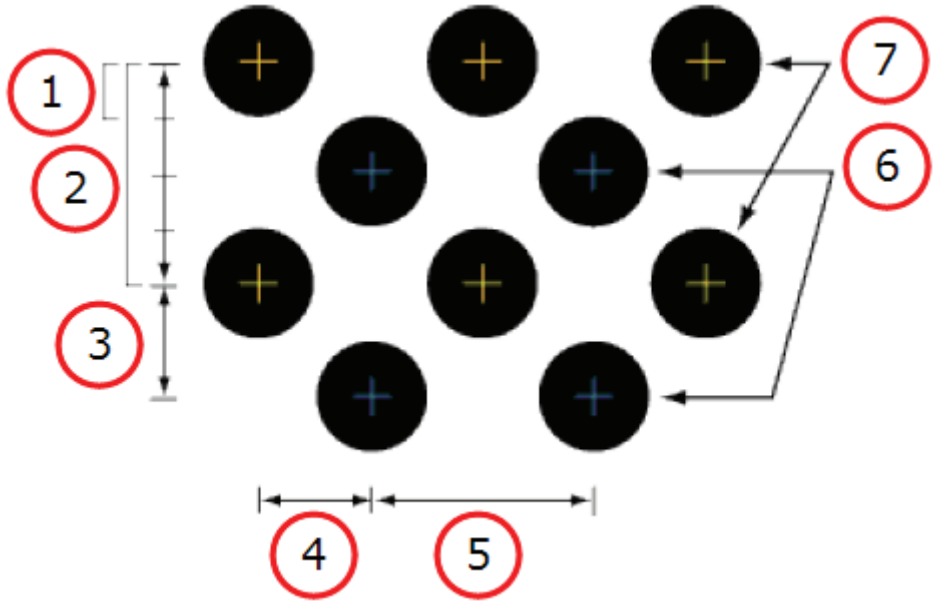
- DotPitchOthers_CalibrationTarget.pdf
- DotPitc10_CalibrationTarget.pdf

Additional Information

- 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. Commercially available targets can be acquired through third parties such as Edmund Optics or Applied Image Inc.

● Creating a Dot Target for Grid Calibration

Dot targets are commercially available, but targets can also be created using the following guidelines, provided below. The quality and precision of a grid of dots target have a direct impact on the overall precision of the application.



Item	Description
1	Dot Radius
2	Dot Pitch
3	1/2 Dot Pitch
4	1/2 Dot Pitch
5	Dot Pitch
6	Validation Dots (blue markers)
7	Calibration Dots (yellow markers)

- 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.
- 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.

Additional Information

- It is important to measure the pitch of the dots after printing a grid to confirm that the printer did not change the scale of the grid. If the dots are not exactly the defined pitch, the camera calibration will be inaccurate.
- The dot grid rows and columns must be aligned with the field of view. Ideally after executing the Calibrate function, there should be a uniform distribution of yellow and blue alternating dots. A region without blue dots indicates the calibration is not sufficient in that region to predict the location of the validation dots.

With the above concepts in mind, use the following steps to create specific dot targets. Measure the width or height of the FOV by using a ruler inside the camera image in live mode to obtain the length of the FOV. Using the camera manual, take the matching pixel resolution, Then calculate the resulting mm-to-pixel ratio. With this information, identify the optimal dot pitch and dot radius for the target. As an example, with a 400mm FOV in the horizontal direction and 1600 pixels in the horizontal direction, camera, the calculation for that direction would be:
Dot pitch = (Field of View (mm)/Pixel (count))*36
Dot radius = (Field of View (mm)/Pixel (count))*9
Dot pitch = (400 mm / 1600)*36 = 9 mm
Dot radius = (400 mm / 1600)*9 = 2.25 mm
A standard dot grid with a pitch of 10mm would still be within the scope for accuracy.

● Fixed Pixel Calibration Method

Fixed-pixel calibration specifies 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.

Acquisition Settings

Acquisition settings are used to view information about the camera and make other image adjustments for vision devices used by the Virtual Camera.
When configuring a Virtual Camera that uses an Emulation Camera, the settings in this area are limited to only gray scale conversion and image selection.
When using a Virtual Camera that uses a vision device such as a Basler camera, several adjustments can be made to the image such as shutter, gain, and exposure along with other camera related settings as described below.

Additional Information

The settings in this area will vary depending on the vision device associated with the Virtual Camera.

● 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 enables users to 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 the down-arrow is clicked (the default selection is recommended).
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, it is recommended that a Timeout value is always used. 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 and the actual processing time may be slightly greater.

● Video Format

Use the Video Format tab to 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.
Some of the minimum and maximum values, particularly for Gain, will differ depending on the camera being used.

● Exposure Adjustment Considerations

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 the sensor is acquiring still or moving object images. Adjust Exposure, Gain, and Black Level (in that order) to improve the quality of acquired images with the following considerations.

- If the object is not moving, choose a high exposure time setting (i.e., a long exposure interval).
- 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.



Additional Information

Acquisition parameters are validated before being sent to the camera. If an unsupported exposure time for a camera is entered, the time will be adjusted to be valid. Use the left and right arrows to set valid exposure times.

● Gain Adjustment Considerations

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. Increase the image contrast by increasing the camera's gain setting.
Unless the application requires extreme contrast, make sure that image details remain visible in the brightest portions of the image when increasing gain. Increasing gain will increase the image brightness and increase signal noise. Set the gain only as high as is necessary.

● Black Level Adjustment Considerations

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.

● White Balance Considerations

Balance Red, Balance Green, and Balance Blue are only available when using a color camera. 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 grayed out).

● Trigger

The Trigger tab is used to enable an external trigger for taking a picture and set parameters that pertain to that trigger.

Most applications will not use trigger mode and the image is taken when requested by the PC, but some applications need to reduce latency in communication. In this type of situation, a trigger signal would be wired directly to a camera input and trigger mode is enabled and configured in the Virtual Camera. A V+ program would execute a VRUN command to execute a Vision Sequence but instead of acquiring an image, it will create the image buffer and wait to receive the image from the camera when it is triggered. A camera exposure active signal could still be used for position latching if necessary.

● Other Acquisition Settings Functions

Use the **Add** and **Delete** buttons to add or remove acquisition settings.

The **Import** button opens a selection window to copy acquisition settings from another Virtual Camera in the ACE project.

The **Edit** button opens a window for changing acquisition settings for the selected device.

Use the **Up** and **Down** buttons to arrange multiple acquisition settings in a specific order.

8-4-2 Emulation Cameras

An Emulation Camera is a stored collection of images, which the ACE software can treat as if they were coming from a physical camera. Emulation Cameras are mostly used in Emulation Mode when a camera is not available or when viewing images from another application.

Emulation Cameras require imported images to be the same size (pixel X by Pixel Y) and format (color vs mono). Emulation Cameras should be used with limited quantities of images because they directly impact the size of projects. If a large quantity of images must be used, it is advisable to use Image Replay and provide a directory of logged images.

Since there are no .hig image file viewers, Emulation Cameras can be used to import logged .hig images and export those images as other common formats for viewing with other software (.jpg, .png, etc).

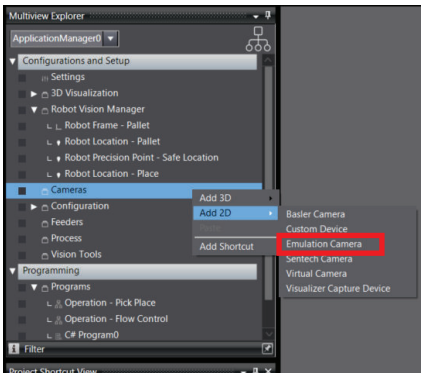


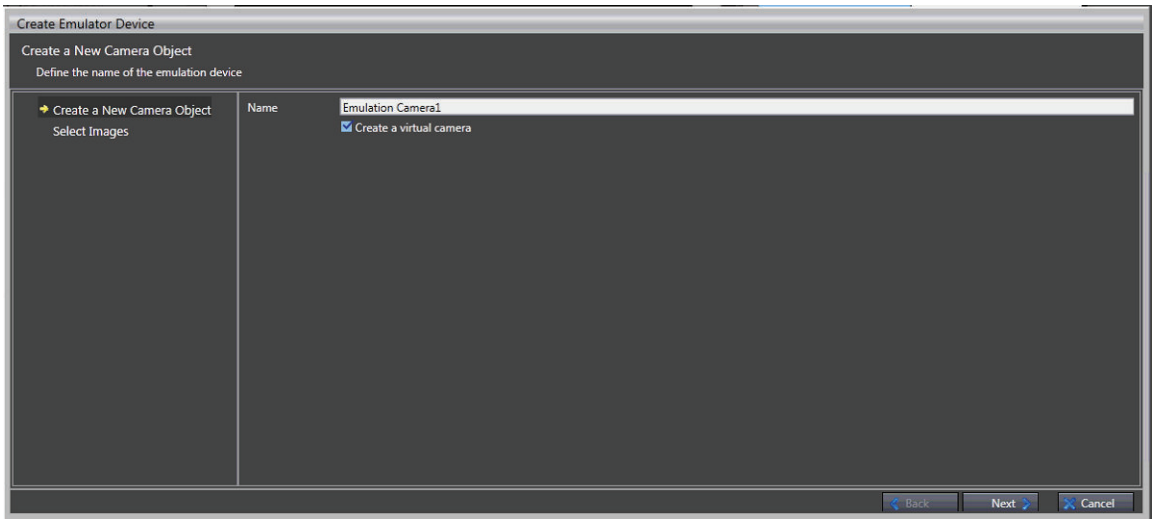
Additional Information

Emulation Camera acquisition settings in the Virtual Camera allow color-to-gray scale conversion for convenience.

Add an Emulation Camera

To add an Emulation Camera, right-click **Cameras**, select **Add 3D** or **Add 2D**, and then click **Emulation Camera**. The Create Emulator Device window will open.





Provide a name for the Emulation Camera and make a selection for creating a Virtual Camera associated with this device. Then, click the **Next** Button and click the **Add** Button to load images to be used with this device. Click the **Finish** Button when all images are added and then the Emulation Camera (and Virtual Camera) object will be added to the ACE project.

Emulation Camera Configuration

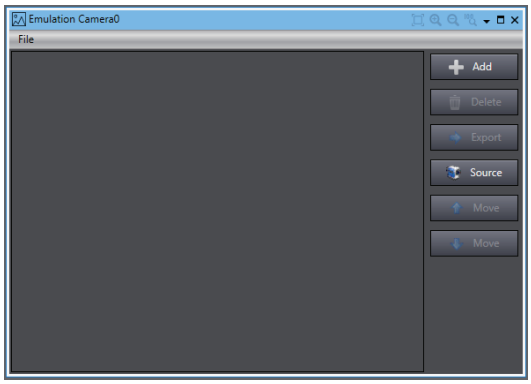
The Emulation Camera Configuration area allows you to manage all images used for this device. The order of the added images will be used in the associated Virtual Camera during subsequent image acquisition.

To access the Emulation Camera configuration, right-click the **Emulation Camera** object in the Multiview Explorer and then select **Edit** or double-click the object. This will open the Emulation Camera editor in the Edit Pane.

Use the information below to understand the Emulation Camera configuration items.

Additional Information

When adding images, image files must be of the same type (color vs. monochrome) and the same size.



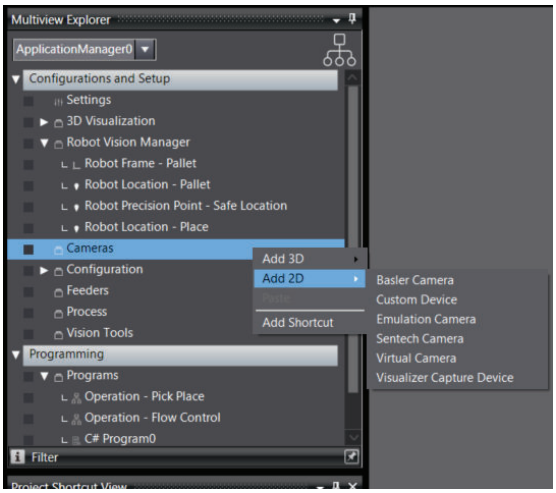
Item	Description
Add / **Delete** Buttons	Add or delete images
Export Button	Export images from the Emulation Camera to files on the PC
Source Button	Add images from another camera source within the ACE project
Move Buttons	Arrange the images in a specific order
File Menu - *Load from Database*	Load a collection of images from an emulation database file (.hdb)
File Menu - *Save All To Database*	Save a collection of images to an emulation database file (.hdb)
File Menu - *Sort Images*	Sort images in alphabetical order by file name

8-4-3 Physical Cameras

Physical cameras objects are associated with camera hardware devices that are physically connected to the system such as Basler, Sentech, or other types listed in the *Add 2D/3D* sub-menus. These physical camera devices are used for image acquisition.
The physical camera object stores information necessary for communicating with the camera hardware, such as the Device Friendly Name, Model, Vendor, and Device Full Name.

Adding a Physical Camera

To add a physical camera, right-click **Cameras** and then select **Add 3D** or **Add 2D**. Select the specific camera device and then complete the steps provided in the Create Device wizard that follows.



Additional Information

- Confirm the camera's configuration utility is closed before adding a physical camera. If the utility is opened, an *Unable to open camera* error will result.
- Physical cameras must be pre-configured before they are added in ACE. Refer to *Automation Control Environment (ACE) Version 4 Camera Configuration User's Manual (Cat. No. I679)* for more information.

Physical Camera Configuration

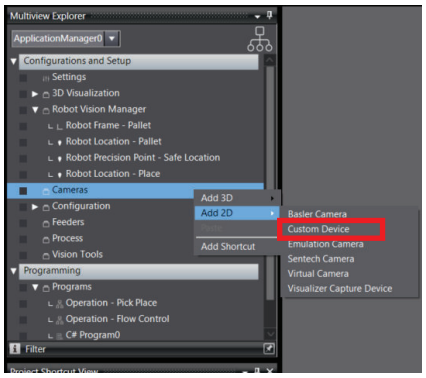
Physical cameras are configured using software that is supplementary to ACE. Refer to *Automation Control Environment (ACE) Version 4 Camera Configuration User's Manual (Cat. No. I679)* for more information.

8-4-4 Custom Devices

The Custom Device object is a C# program that executes at the level of a camera object and can be used to acquire an image from any camera device or external vision system. It can also be used to manipulate image data from one Virtual Camera's image buffer before input to a second Virtual Camera that is linked to a Vision Sequence.
A C# program template will be created when the Custom Device is added to the ACE Project.

Adding a Custom Device

To add a Custom device, right-click **Cameras**, select **Add**, and then click **Custom Device**. A new Custom Device will be added to the Multiview Explorer. Use this object to access the C# program associated with the device.



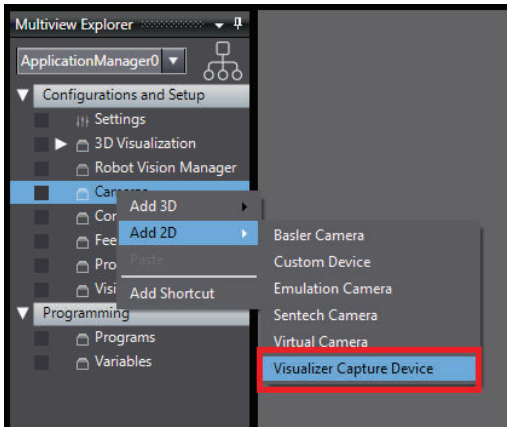
8-4-5 Visualizer Capture Device

The Visualizer Capture Device camera object is used to extract images from the 3D Visualizer for use with a virtual camera. When CAD models are imported into the 3D visualizer, the camera object can be used to assist in configuring vision system models and tools.

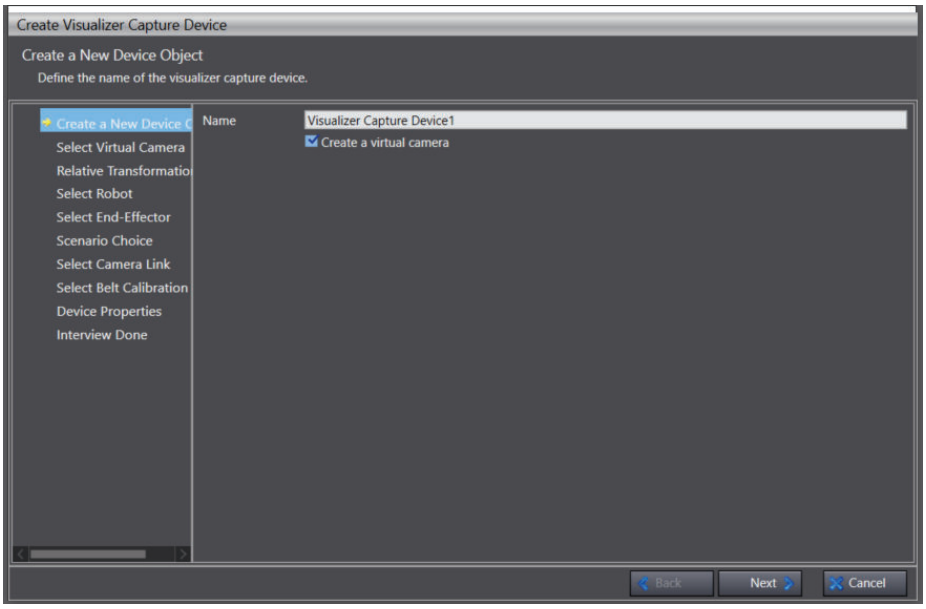
Create a Visualizer Capture

Prior to making a Visualization Capture, all objects in the project must be added and configured.

1 In Application Manager, right-click **Cameras**, select **Add 2D** and then select **Visualizer Capture Device**.

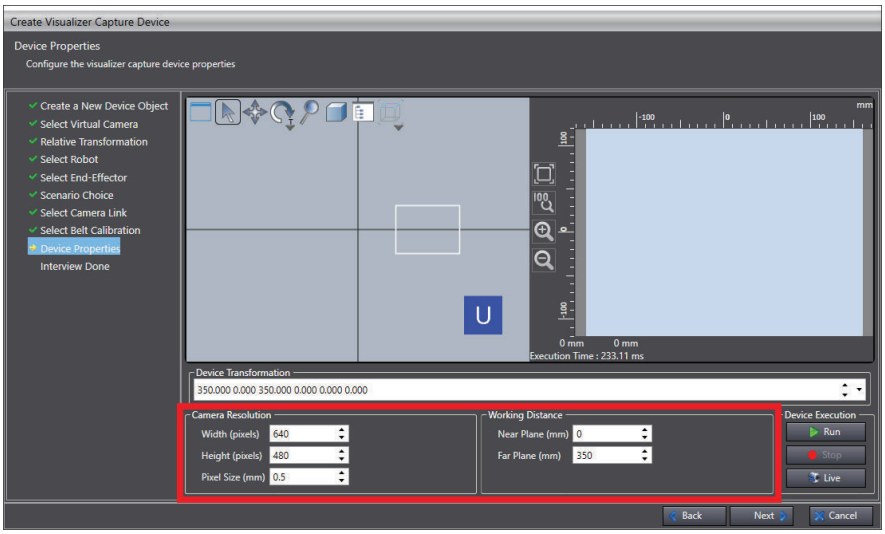


2 The *Create Visualizer Capture Device* window opens with prompts for configuration selections. Make selections and then click the **Next** Button after each selection.



3 At the *Scenario Choice* Window, make the selections that apply to the particular workspace. Then click the **Next** Button.

4 At the *Device Properties*, in the left-side panel, use the Camera Resolution area to set the camera's field-of-view. Use the *Working Distance* input areas to enter camera depth-of-field values to complete the configuration.

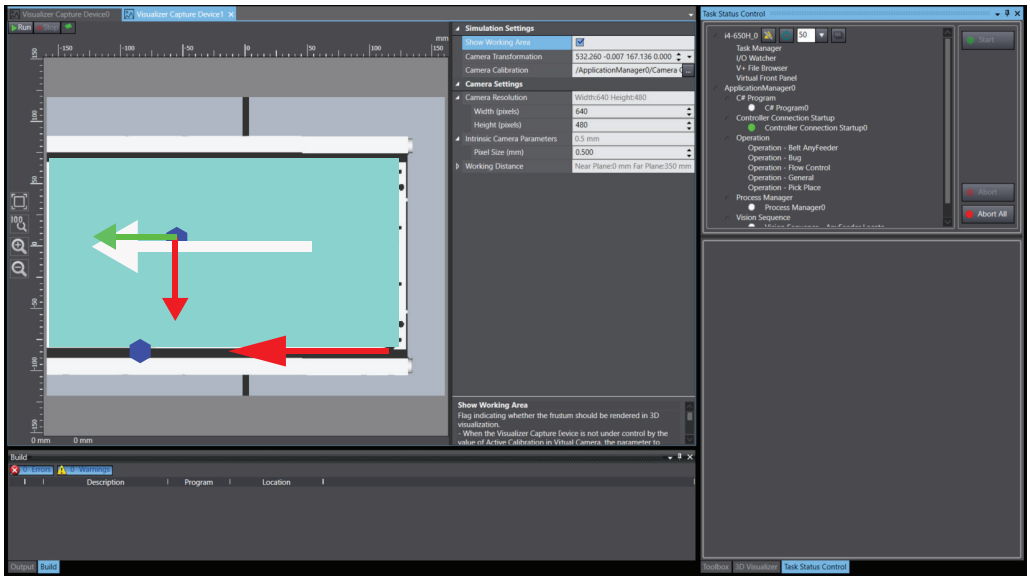


5 Click the **Run** Button to view the results of the Visualizer setup.

6 Select individual number segments in the Device Transformation line to further change refine the view. Click **Run** after each change to see the changes.

7 Make changes to the working distance values, based on the location viewed in the *Run* Window. After each change, click the **Run** Button to see the changes. When the refinement is complete, click **Next**.

8 Locate and double-click the **Visualizer Capture Device** in the Application Manager to open it.



9 Use the *Simulation Settings* to adjust the Visualizer as needed.

8-5 Configuration

Various configuration objects can be added to the ACE project for different control, setup, and other functions. Use the information in this section to understand the functions of the Configuration objects.

8-5-1 Controller Connection Startup

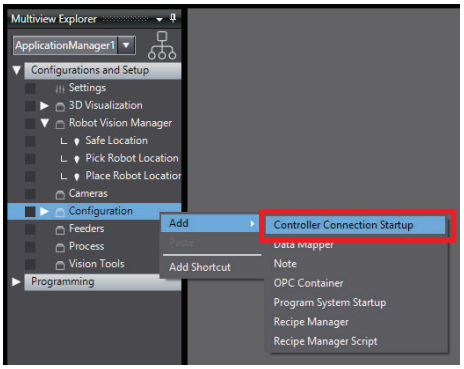
The Controller Connection Startup object is used to establish a connection, monitor, and maintain the connection to one or more controllers upon the startup of the ACE project. It can also be configured to start a V+ program on a specified task number upon startup.

When the Controller Connection Startup object is configured, opening the ACE project will automatically initiate a connection to a controller and program execution, once the connection is fully established.

Although the Controller Connection Startup function and the Auto Start function share some similarities, the startup file for the Controller Connection Startup function is stored on the PC whereas the Auto Start function files are stored on the controller. This allows the Auto Start function to provide a headless (no PC required) operation of the application. Refer to *Auto Start* on page 7-25 for more information.

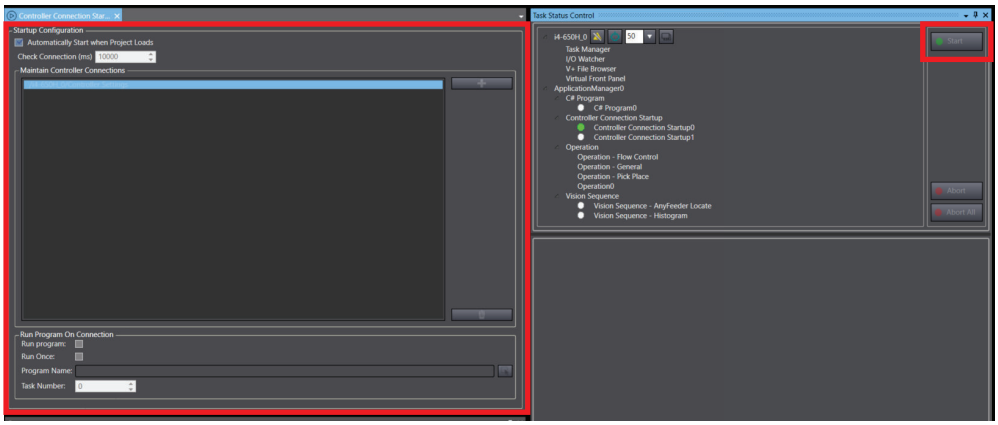
Adding a Controller Connection Startup Object

To add a Controller Connection Startup object, right-click **Configuration**, select **Add**, and then click **Controller Connection Startup**. A new Controller Connection Startup object will be added to the Multiview Explorer.



Controller Connection Startup Configuration

To access the Controller Connection Startup configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Controller Connection Startup editor in the Edit Pane.



The Controller Connection cannot be modified when it is running. Use the table below to understand the Controller Connection Startup configuration items.

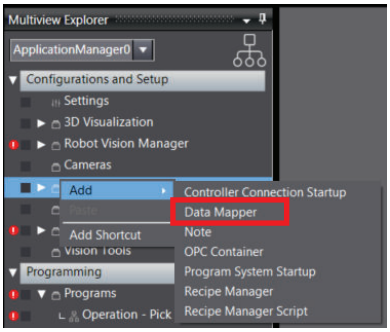
Group	Item	Description
Startup Configuration	Enable Startup Object	Enable or disable the function of the Controller Connection Startup object when the ACE project is opened.
	Add / **Delete** Buttons	Add or remove controllers to/from the Maintain Controller Connections list. A controller can be added only when it already exists in the ACE project. A new controller connection in the configuration.
	Check Connection (ms)	Specifies the interval in milliseconds for checking the connections to the controllers specified on the Maintain Controller Connections list.
	Maintain Controller Connections	Displays a list of controller connections being monitored.
Run Program on Connection	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 the program needs to run again, ACE must be shut down and restarted.
	Program Name	Specifies the program to run when the selected controller is connected.
	Task Number	Specifies the V+ task number for the V+ program. The specified V+ task number must be idle. Otherwise, the program will not run on connection.

8-5-2 Data Mapper

The Data Mapper provides a method to associate different data items within the ACE project. For example, a Process Manager object can be triggered to run when a digital input signal turns on. Any data items that are associated in the Data Mapper will be continuously checked while the ACE project is open.

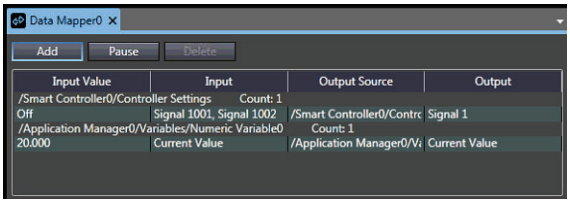
Adding a Data Mapper Object

To add a Data Mapper object, right-click **Configuration**, select **Add**, and then click **Data Mapper**. A new Data Mapper object will be added to the Multiview Explorer.



Data Mapper Configuration

To access the Data Mapper configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Data Mapper editor in the Edit Pane.



● Data Mapper Editor Buttons

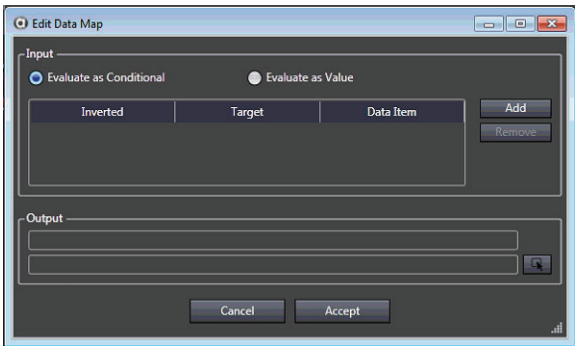
Use the Data Mapper Editor buttons to add, delete, and pause/run Data Mapper items.
Pausing a selected Data Mapper item will prevent it from executing. Click the **Run** Button to resume a paused Data Mapper item.

● Data Mapping

When the **Add** Button is clicked in the Data Mapper editor, an Edit Data Map Dialog Box will open. This is used to create and edit Data Mapping items. The Data Mapping configuration is described below.
To edit an existing Data Mapper item, double-click the item in the Data Mapper list to access the Edit Data Map Dialog Box.

Additional Information

The Data Mapping input and output items that are available depend on the objects present in the ACE project.



- **Evaluate as Conditional**

  When **Evaluate as Conditional** is selected, the Data Mapper will interpret all input conditions 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. The output condition is asserted when the input item transitions (rising edge/falling edge).

  Additionally, when Evaluate as Conditional is selected, the expected value of an input item can be inverted. This means that if the value is 0, the condition is considered to be ON.

- **Evaluate as Value**

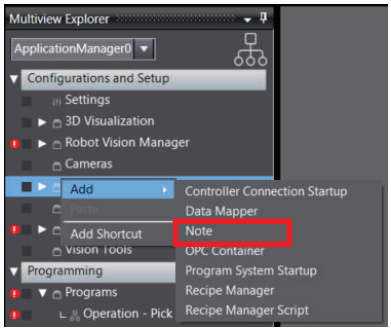
  When **Evaluate as Value** is selected, the value of all input conditions are added together and written to the output value.

8-5-3 Note Object

The Note object provides a means for creating documentation within an ACE project. Use this object to create detailed notes for future reference.

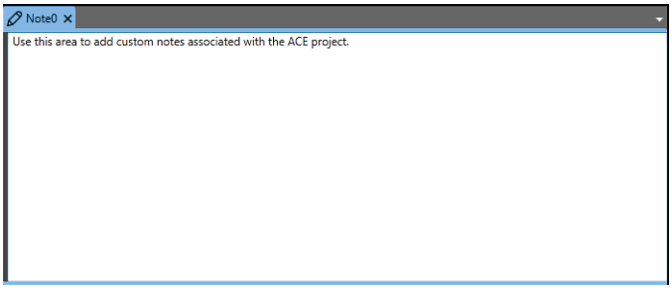
Adding a Note Object

To add a Note object, right-click **Configuration**, select **Add**, and then click **Note**. A new Note object will be added to the Multiview Explorer.



Note Editing

To access the Note object for editing, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Note editor in the Edit Pane.



8-5-4 OPC Container

The purpose of an OPC container is to provide a standardized infrastructure for the exchange of process control data that accommodates different data sources, connections, and operating systems.
OPC stands for Object Linking and Embedding (OLE) for Process Control. It uses Microsoft's Component Object Model (COM) and Distributed Component Object Model (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 which allows various applications to retrieve data in exactly the same format regardless of whether the data source is a PLC, DCS, gauge, analyzer, software application, or anything else with OPC support. The data can be available through different connections such as serial, Ethernet, or radio transmissions for example. Different operating systems such as Windows, UNIX, DOS, and VMS are also used by many process control applications.
The OPC protocol consists of many separate specifications. OPC Data Access (DA) provides access to real-time process data. Using OPC DA, an OPC server can be queried for the most recent values of anything that is being measured, such as flows, pressures, levels, temperatures, densities, and more.
OPC support in ACE software is limited to the DA specification.
For more information on OPC, please see the OPC Foundation website.
An OPC container can be configured for the following functions.

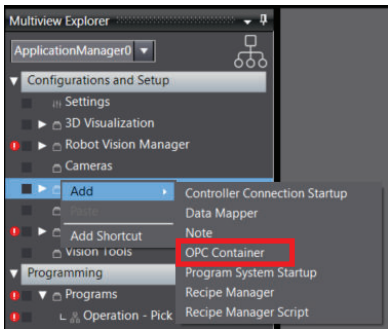
- Communicate values of V+ Global Variables and Process Manager statistics.
- Start or stop a process within an ACE application.

OPC Test Client

An OPC test client is included with the ACE software installation. Refer to *8-5-4 OPC Container* on page 8-65 for more information.

Adding an OPC Container

To add an OPC Container object, right-click **Configuration**, select **Add**, and then click **OPC Container**. A new OPC Container object will be added to the Multiview Explorer.



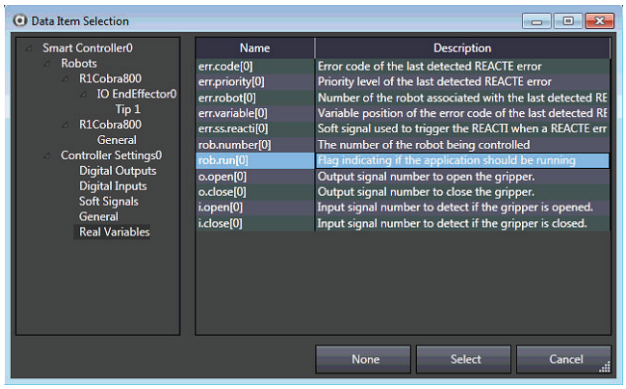
OPC Container Configuration Procedure

Use the following procedure to configure an OPC Container object.
Depending on the ACE project configuration, the items available in the Data Item Selection list will vary. The procedure below has one robot configuration that was generated using a Robot Vision Manager Application Sample with a fixed pick and a fixed place arrangement.

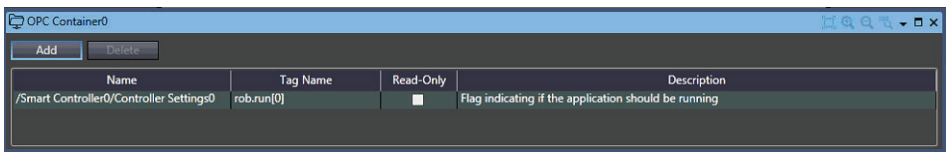
1 To access the OPC Container configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the OPC Container editor in the Edit Pane.



2 Click the **Add** Button to select a Data Item. The *Data Item Selection* Dialog Box will appear.



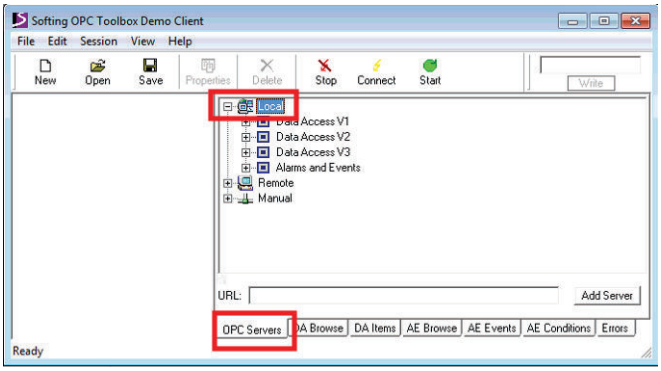
3 Select an item from the list and then click the **Select** Button. The item will be added to the OPC Container publish list in the Edit Pane.



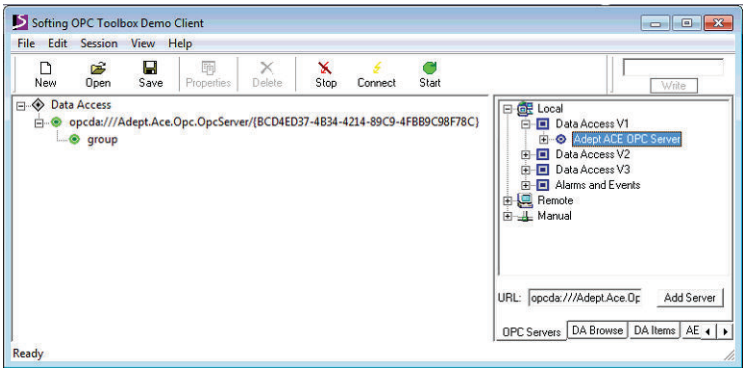
4 Make a selection for the **Read-Only** option. If checked, an external OPC client cannot write to the item. If the Read-Only option is not checked, the OPC client has access to read and write the item's value.

5 Once all items have been added to the OPC Container publish list, the OPC client can be configured as shown in the next steps. Run the SOClient.exe file found in the default ACE installation directory (C:\Program Files\OMRON\ACE 4\OPC Test Client).

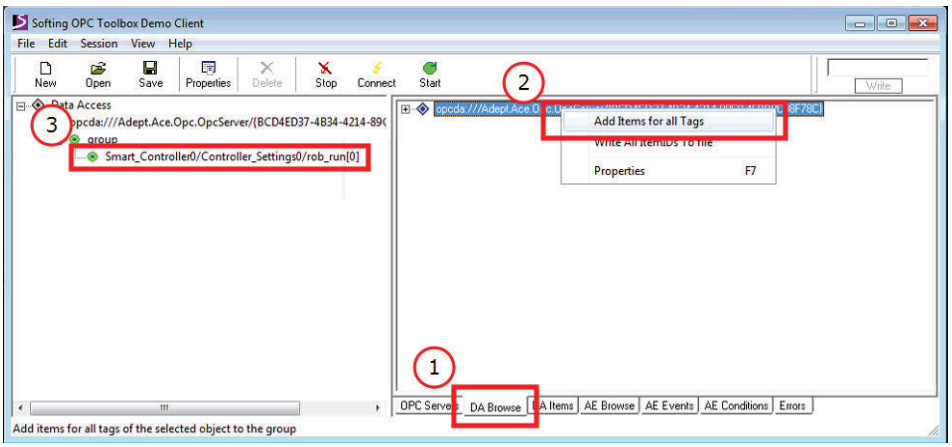
6 Select the **OPC Server** Tab and then expand the **Local** item to expose all Data Access items.



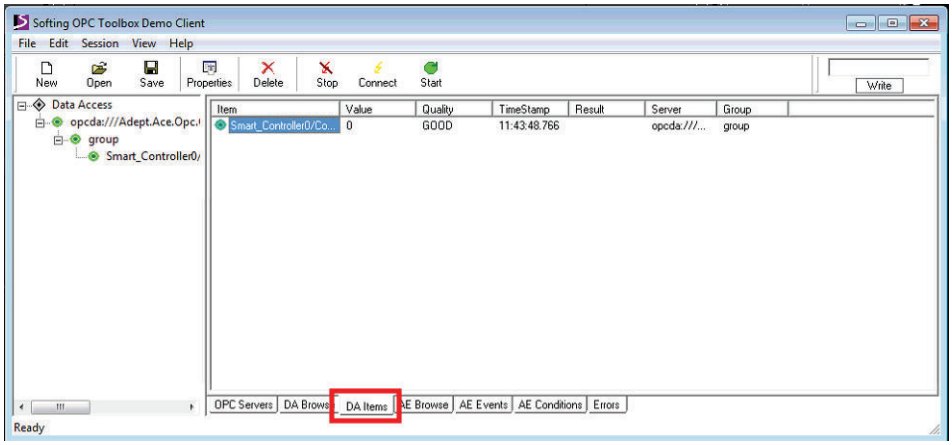
7 Expand a Data Access item to show the Adept OPC Server item. Add the Adept OPC Server to the left pane window by double-clicking it or, right-clicking the **Adept OPC Server** item and selecting **Add Server**.



8 Select the **DA Browse** Tab and then right-click the server on the right window and select **Add Items for all Tags**. This will add all associated tags to the server in the left window.

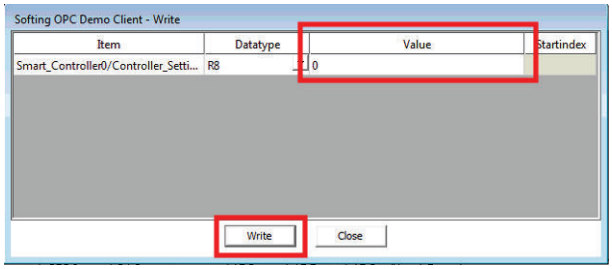


9 Select the **DA Items** tab to see the item's value and other related information.



The Quality column indicates that the communication with the OPC DA server worked normally (GOOD indicates successful communications).
The TimeStamp column indicates the last update time of the tag.

10 Tags that were configured with the Read-Only selection unchecked can be modified with the OPC client. To change the value from the OPC client, right-click the tag and select **Write**. The dialog box below will appear. Enter a new value and then click the **Write** Button to change the value in the ACE application.



11 Values that have been updated from the ACE application can be verified using the read function in the OPC client. Right-click the tag and select **Read** to update values in the OPC client. If the values are updating correctly, the configuration procedure is complete.

- **Add or Delete Items to Publish on OPC DA**

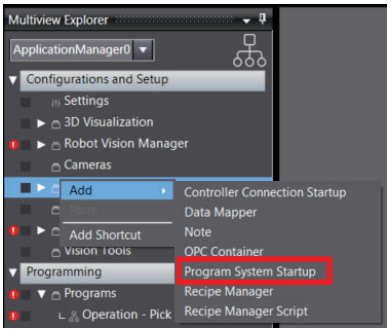
  Use the **Add** Button to add a new item to publish on OPC DA. Use the **Delete** Button to remove a selected item from the list.

8-5-5 Program System Startup

The Program System Startup object is used to specify a C# program to run when the ACE project opens.
Although the Program System Startup function and the Auto Start function share some similarities, the startup file for the Program System Startup function is stored on the PC whereas the Auto Start function files are stored on the controller. This allows the Auto Start function to provide a headless (no PC required) operation of the application. Refer to *Auto Start* on page 7-25 for more information.

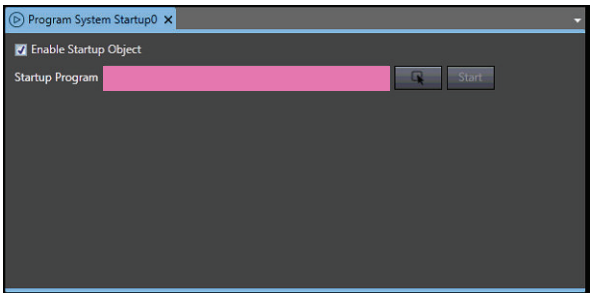
Adding a Program System Startup Object

To add a Program System Startup object, right-click **Configuration**, select **Add**, and then click **Program System Startup**. A new Program System Startup object will be added to the Multiview Explorer.



Program System Startup Configuration

To access the Program System Startup configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Program System Startup editor in the Edit Pane.



Use the **Selector** Button ( ) to make a startup C# program selection.

Uncheck the **Enable Startup Object** Option to disable the object.

Use the **Start** Button (Start) to manually test the Program System Startup object.

8-5-6 Recipe Overview

Manufacturing processes often require frequent product changeover resulting in the need for changes to variable values, vision models, vision tool parameters, pallet layouts, motion parameters, process definitions, motion offsets, and more. The ACE software provides a Recipe Manager that simplifies the complex process of saving and restoring large amounts of data to minimize downtime during frequent product changeover.

There are three steps for recipe management in the ACE software, as described below.

Recipe Definition

Recipe definition involves selecting which objects will be the Sources of recipe data. Sources are similar to ingredients of a traditional cooking recipe. A recipe will contain a copy of the data for each Source. Recipes can only store data of objects that are defined as Sources in the Recipe Manager edit pane. All other objects will have common parameters for all recipes. When a recipe is created, it will contain a copy of the data that is currently present in the Source objects. This can significantly reduce the number of objects that must be created and maintained.

For example, consider a situation where a camera is used to locate a product to be packaged. In this example, the system can process five different types of products, but only one product type at a time. Rather than creating five Locator Models and five Locators, create one Locator and one Locator Model, add each as a source, and create five recipes containing the Model data and Locator parameters optimized for each product type. If two types of product must be recognized by the same Locator, create two Locator Model objects and include both as sources.

Recipe Creation and Modification

After the recipe Sources have been defined in the Recipe Manager Edit Pane, recipes can be created in the Recipe Manager section of Task Status Control.

Typically, a system should be sufficiently optimized for one product before creating recipes. This minimizes the amount of parameters in Source objects that need to be individually edited in each recipe. Although this is recommended, it is not mandatory.

Task Status Control provides a Recipe Editor that can be used to edit the parameters of all Source types that are commonly modified by operators. When a recipe is selected, the entire ACE software interface becomes the editor for the active recipe. The Recipe Editor does not provide an editor window for all Sources. For example, if a Process Manager is a source for a recipe, it will not be visible in the Recipe Editor, however the Process Manager edit pane can be used to make modifications to all Process Manager parameters for the selected recipe.

It is important to recognize that if changes are made to objects that are Sources in a recipe while no recipe is selected, when a recipe is selected, those changes may be lost. Be sure to always use the Recipe Editor or select a recipe before making changes to recipe Source objects.

Refer to *8-5-8 Recipe Editor* on page 8-74 for more information regarding creating, modifying, and deleting recipes.

Recipe Selection

Recipe Selection is a single item selection process for applying the parameters stored in the recipe to the Source objects.

Once all recipes have been defined and optimized automate the recipe selection process so that it does not need to be performed from Task Status Control. This can be achieved using a V+ program and Robot Vision Manager, or a C# program.

When a Recipe is selected, the parameters saved in the Recipe are applied to the ACE project. 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 ACE project.

Use the information in the following sections to understand recipe management configuration.

8-5-7 Recipe Manager

The Recipe Manager is used to define all sources that will be used when creating individual recipes. Sources must be added to the Recipe Manager before creating a Recipe. A recipe is a dedicated object that is added to Application Manager.

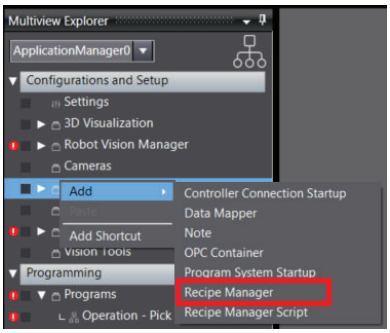
The following objects in an ACE project can be used as sources for the Recipe Manager.

Item	Description
V+ Variables	Specify V+ Variables to be included in a recipe. It is important to identify how the variable is displayed to the user and what access level a user will have.

Item	Description
Vision Tools	All vision tools can be accessed with the Recipe Manager, except the following tool types. • Calculation Tools • Image Process Tools • Custom Tools
Virtual Camera	Specify Virtual Camera object data to be included in a recipe.
AnyFeeder	Specify AnyFeeder object data to be included in a recipe.

Adding a Recipe Manager Object

To add a Recipe Manager object, right-click **Configuration**, select **Add**, and then click **Recipe Manager**. A new Recipe Manager object will be added to the Multiview Explorer.

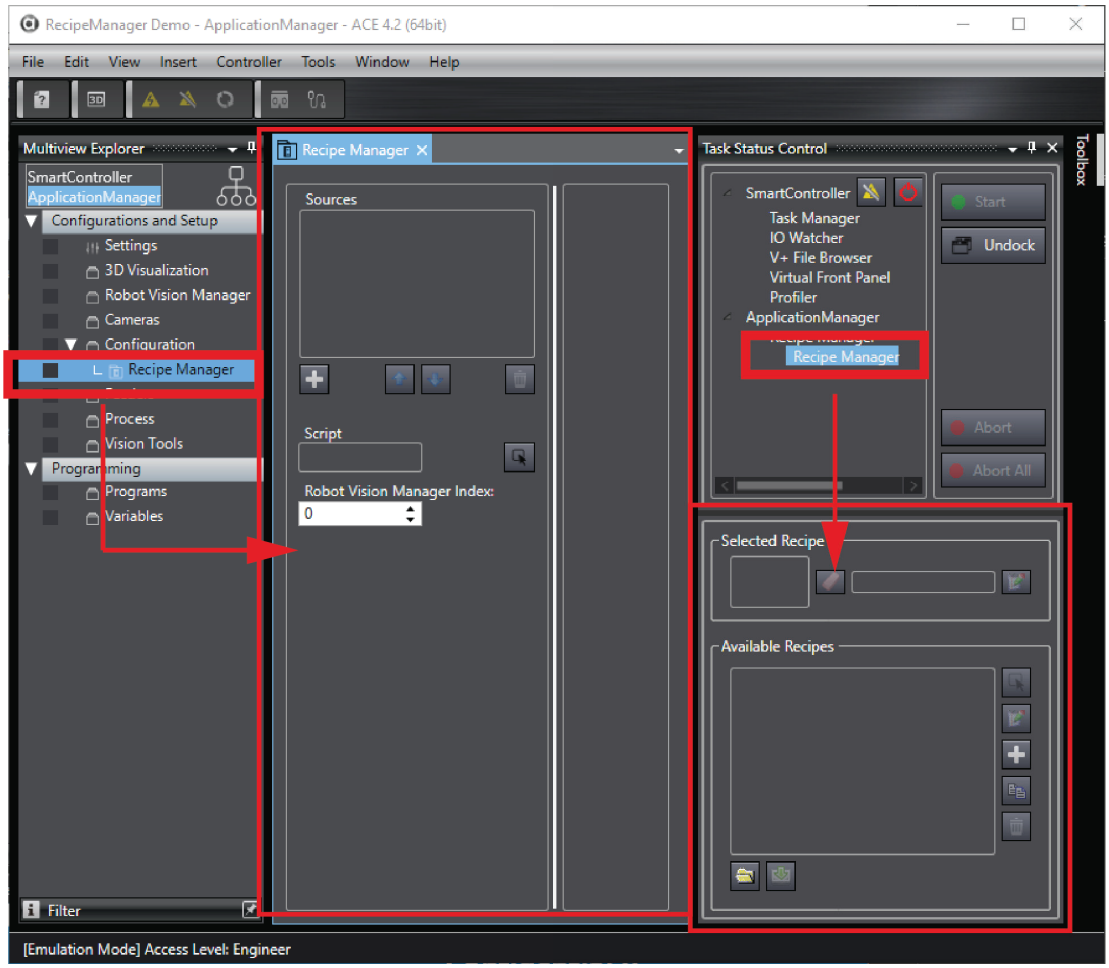


Additional Information

A Recipe Manager object can be dragged and dropped into a C# program. Refer to RecipeManager Topics in the ACE API Reference Manual (Help Menu) for more information.

Recipe Manager Configuration

To access the Recipe Manager configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Recipe Manager editor in the Edit Pane, shown in the center of the following figure.

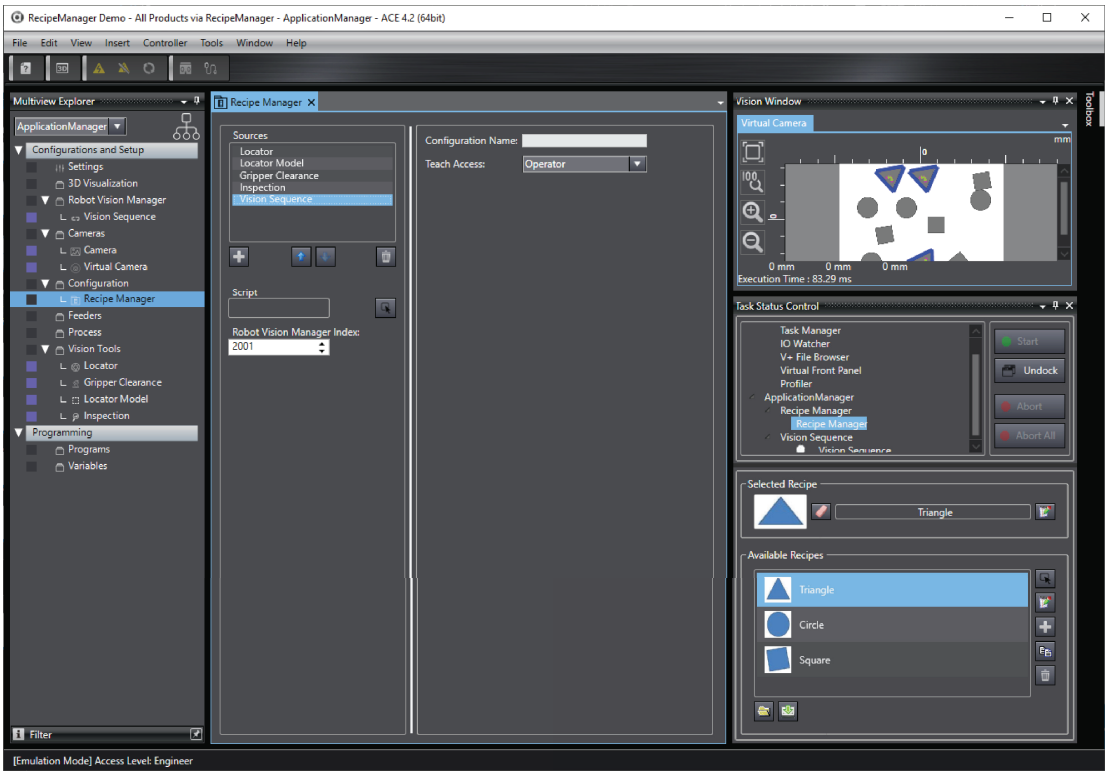


The Recipe Object can also be opened by selecting it in the Task Status Control, which is shown on the right side of the above figure.

● Sources

The Sources area displays a list of all active sources available for Recipe creation and editing. Use the **Add** button ( ) and **Remove** button ( ) to build a list of sources for use in Recipe creation.

The **Up** and **Down** buttons ( ) change the order of the items in the list and order of these items in the tabs of the Recipe Editor. Place frequently used items near the top of the list.



After a data source is added to the Sources list, select it to display the settings in the configuration window. Configuration window options will vary based on the Source type selected. All Source types include settings for the following items.

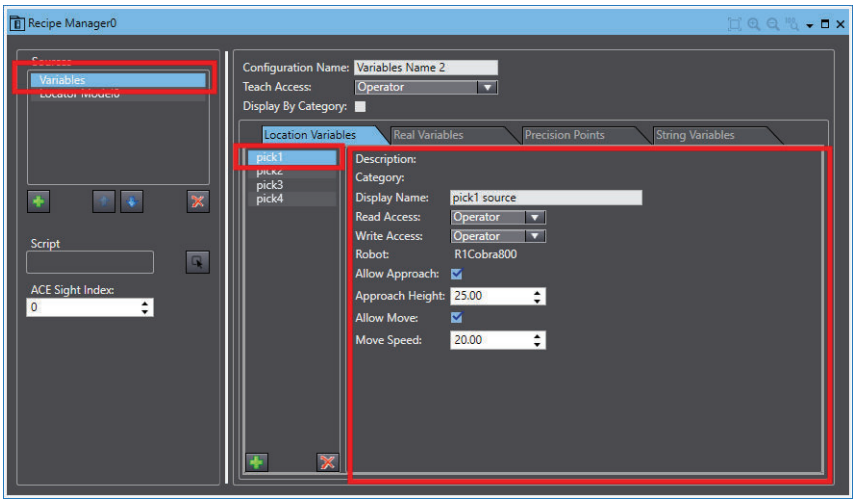
- Configuration Name - Provide a unique name for the Source. This is the name that is displayed in the Recipe Editor.
- Teach Access - This is used to restrict write access to Recipe Manager sources by setting user access levels. Refer to *5-13 User Management* on page 5-55 for more information.

● V+ Variable Sources

When configuring a V+ Variable Source in the Recipe Manager, individual variables must be added in the configuration area. Use the **Add** Button () and **Remove** Button () to make a list of V + Variables that will be used in individual Recipes.

Each type of variable contains different properties that affect how the variable is presented to the user in the Recipe Editor. For example, the variables can have defined unique display names and access levels. To access the V+ Variable Recipe attributes, make a Variable selection and then adjust its properties as shown below.

V+ Variable values must be edited in the recipe component directly with the Recipe Editor or a C# program. These values are used to initialize V+ Variables when the recipe is selected, but the V+ variable values may change while the recipe is active (without it being stored in the recipe).



● Vision Tool Sources

Finder, Inspection, and Reader vision tools can be added to a Recipe configuration. A copy of the vision tool will be saved with each Recipe.

When a Recipe is selected with a vision tool, it is linked with the vision tool object in the ACE project that they correspond to. When a vision tool included in the Recipe configuration is modified in the ACE project object, the selected recipe copy of the vision tool is automatically updated. Likewise, when the vision tool is modified in the Recipe Editor, the ACE project vision tool object is automatically updated. Because of this linking between the Recipe and ACE project object, a vision tool object is configured and will be saved with the active Recipe.

The Recipe Editor will vary depending on the vision tool object in use. Typically, the Recipe Editor is only a small subset of vision tool object's properties. Refer to *8-5-8 Recipe Editor* on page 8-74 for more information.

● Recipe Script Selection

Select a Recipe Script object created with the Recipe Manager Script editor. Refer to *8-5-9 Recipe Manager Script* on page 8-80 for more information.

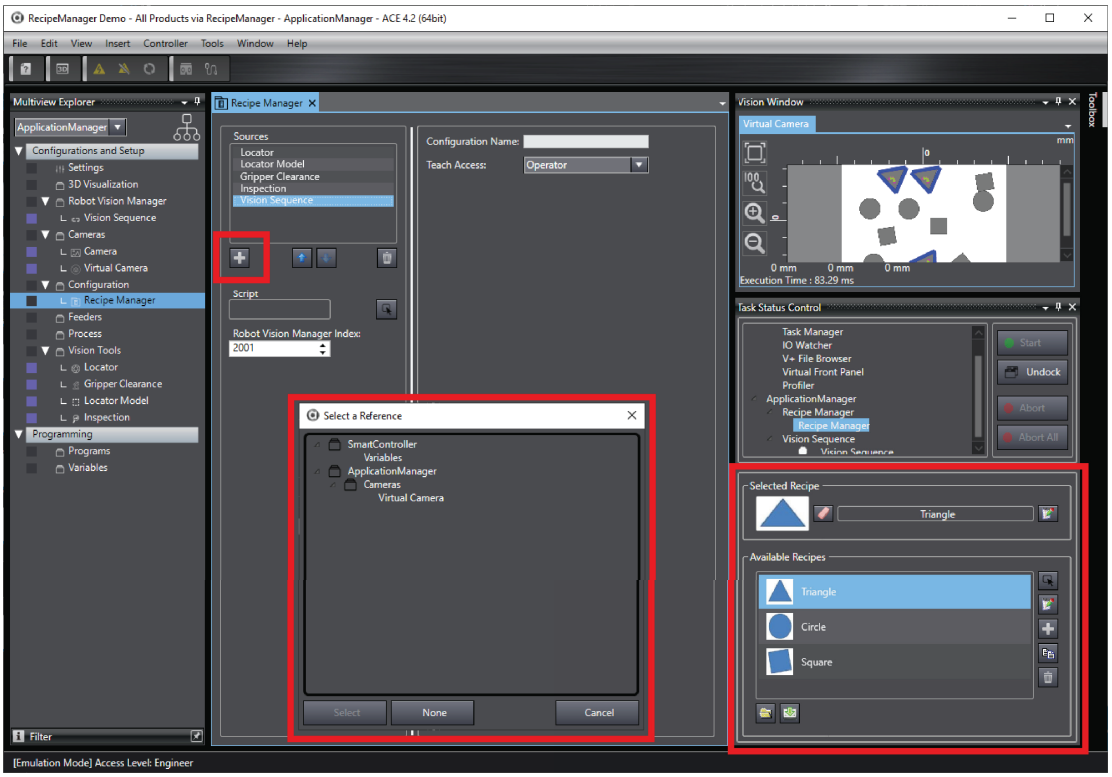
● Robot Vision Manager Index Setting

The Robot Vision Manager index setting defines the index used as the sequence_id when accessing the Recipe Manager object from a V+ program. Recipe numbers in a range of 0 to 16.777.217 are possible. Refer to Recipe Properties in the ACE API Reference Manual (Help Menu) for more information.

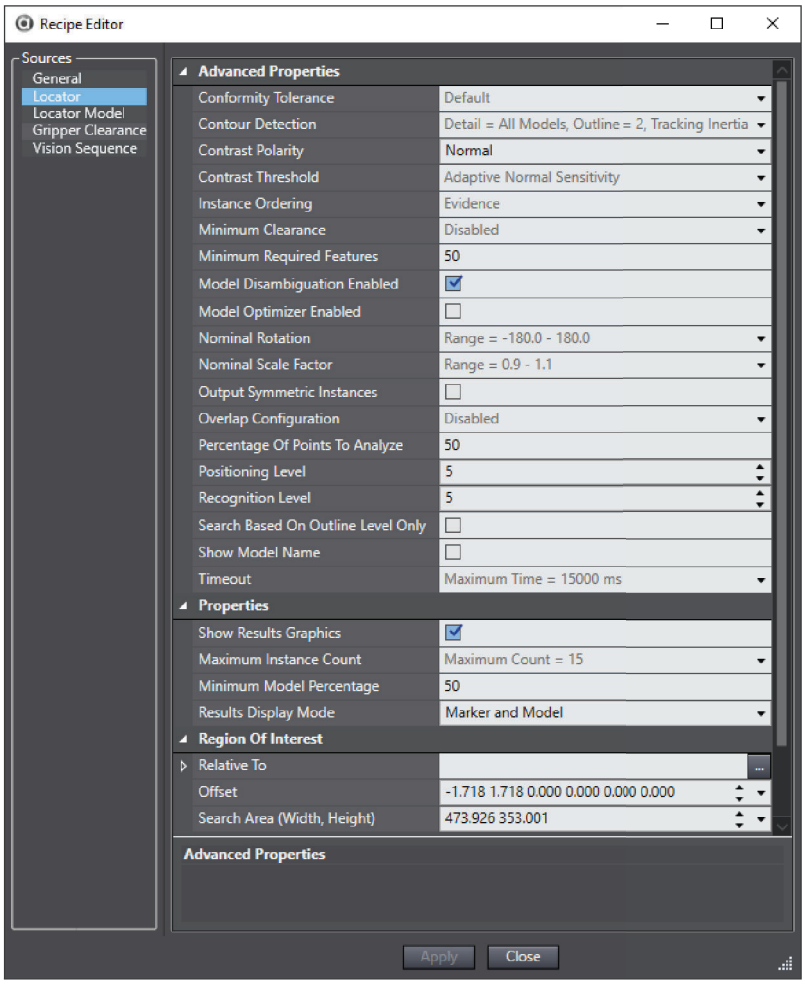
8-5-8 Recipe Editor

After the Recipe Manager object has been configured and all sources are defined, individual Recipes can be created with the Recipe Editor.

The Recipe Editor can be accessed from the Task Status Control area. Refer to *5-9 Task Status Control* on page 5-40 on page 1 for more information. The Recipe Editor is described below.



With the Multiview Explorer opened, select the Recipe Manager and, click the **Add** Button, bordered in red above. Those items shown in the colored boxes, under Recipe Manager, in the left side panel can then be added. The select an available option in the Available Recipe, which opens the center highlighted box. Clicking the **Edit** Button in the Selected Recipe box then allows editing of that specific recipe. The Recipe Editor is shown in the following figure.



The following table identifies the control buttons in the Vision Window used for Recipe editing. The available recipes can be exported and reloaded from a directory on the PC.

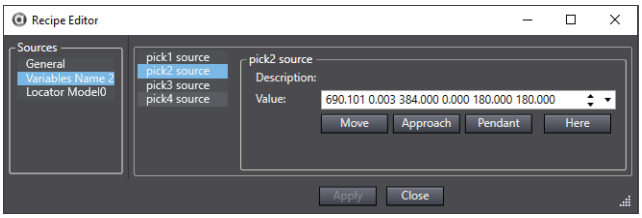
Group	Item	Description
Selected Recipe		Edit the selected Recipe.
		Clear the selected Recipe.

Group	Item	Description
Available Recipes		Makes the highlighted Recipe in the Available Recipe list the active Recipe.
		Edit the selected Recipe.
		Add a new Recipe.
		Create a copy of the selected Recipe.
		Delete a selected Recipe.
		Load a Recipe from a saved file on the PC.
		Save a Recipe to a file on the PC.

All of the buttons can also be activated through a C# script, refer to *Recipe Script Selection* on page 8-74 for more information.

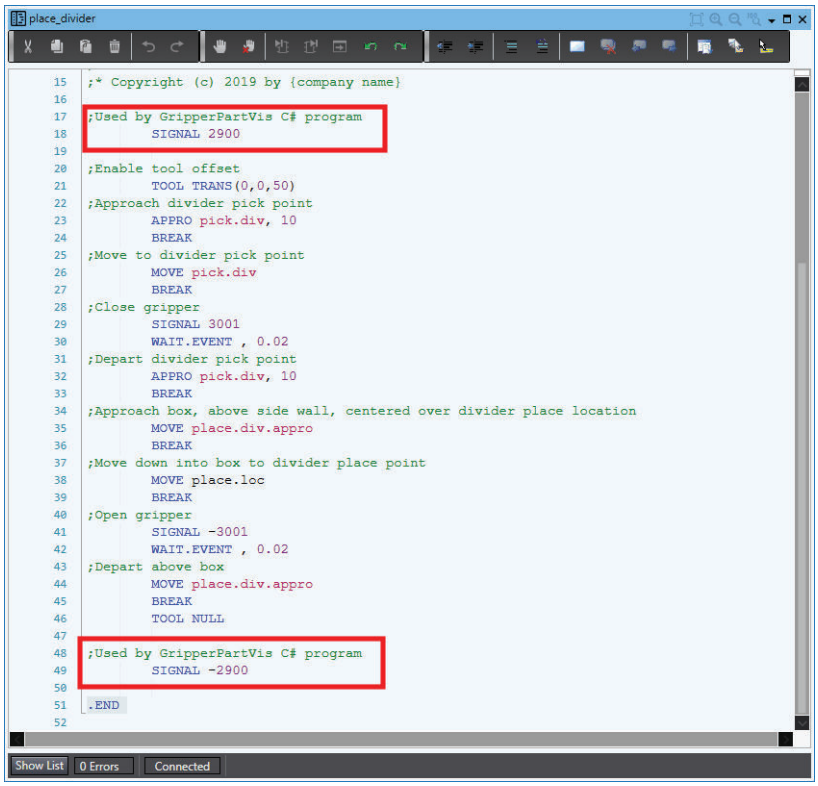
V+ Variable Source

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.



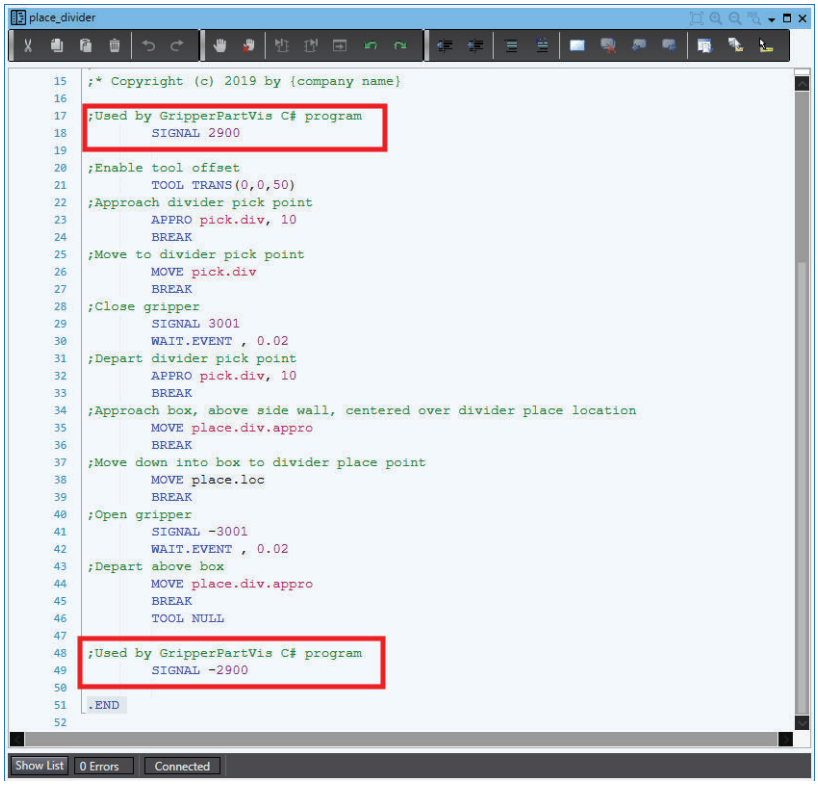
Locator Model Source

The currently trained locator model is visible and the locator model can edited or retrained .



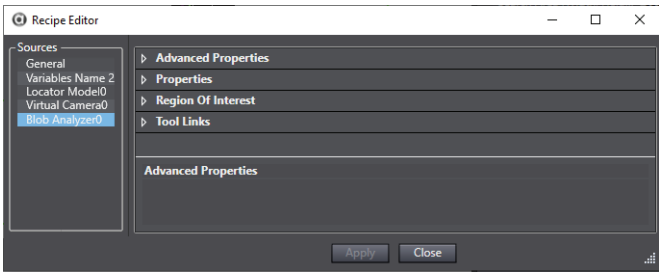
Virtual Camera Source

The acquisition properties are displayed in a list. The acquisition settings can be added, or removed as needed.



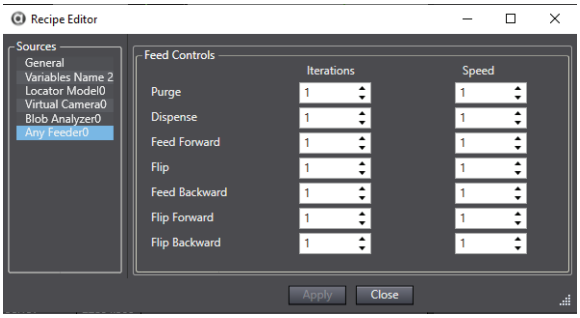
Other Vision Tool Sources

Adjust the parameters of the vision tool.



AnyFeeder Sources

Adjust the Feed Controls of the AnyFeeder.

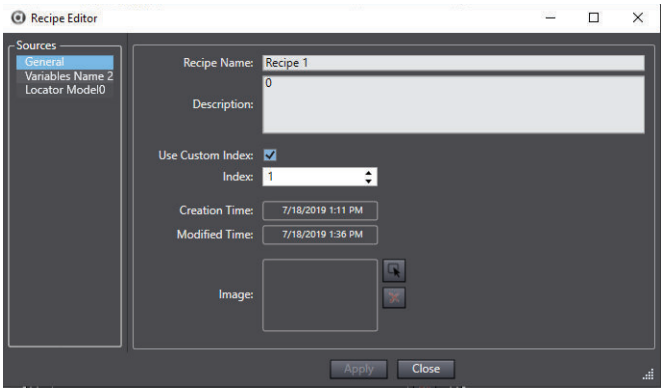


Creating a New Recipe

Use the following procedure to create a new Recipe.

1 Add a new recipe with the **Add** button ( ). A new Recipe will appear in the Available Recipes list.

2 Select the recipe and then click the **Edit** button ( ). This will open the Recipe Editor window.



3 Select the General item in the Sources list and then input the general information and settings about the Recipe.

Item	Description
Recipe Name	Provide a unique name for the Recipe.
Description	Provide a description associated with the Recipe.
Index	Select the **Use Custom Index** option to set a unique index number. This is the index of the Recipe used when accessing the Recipe through Robot Vision Manager or with a C# program. Refer to the ACE API Reference Manual (Help Menu) for more information.
Creation Time	The time the Recipe was created.
Modified Time	The last time the Recipe was modified.
Image	User-defined picture associated with the Recipe.

4 Make any adjustments to other data source items for the currently selected Recipe and then click the **Apply** button. When all data source items have been adjusted for that Recipe, click the **Close** Button. The Recipe creation procedure is complete.

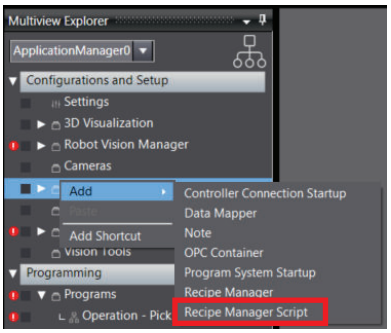
8-5-9 Recipe Manager Script

A Recipe Manager Script allows customization of C# methods invoked when certain events occur in the lifetime of a Recipe. These methods are summarized below.
Most applications that use a Recipe Manager do not require a Recipe Manager Script.

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 and the string will be displayed.
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.
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.
void BeforeSelection(Recipe recipe)	If a Recipe can be selected, this method is called before the Recipe Editor is selected.
void AfterSelection(Recipe recipe)	This method is called after the Recipe is selected.

Adding a Recipe Manager Script Object

To add a Recipe Manager Script object, right-click **Configuration**, select **Add**, and then click **Recipe Manager Script**. A new Recipe Manager Script object will be added to the Multiview Explorer.


Multiview Explorer
ApplicationManager0
Configurations and Setup
Settings
3D Visualization
Robot Vision Manager
Cameras
Add
Add Shortcut
Programming
Programs
Controller Connection Startup
Data Mapper
Note
OPC Container
Program System Startup
Recipe Manager
Recipe Manager Script

8-6 Feeders

Feeders are objects in an ACE project that represent devices used to dispense and present parts to a robot. There are three different feeder objects that can be added to the ACE project.

- AnyFeeder - a feeder object that uses serial communications with the PC running ACE software for sequence control and feedback purposes. This is typically controlled using Robot Vision Manager from a V+ program or from a Custom Vision tool.
- IO Feeder - a feeder object that uses discrete signals for control and feedback purposes. This is typically used with a Process Manager object to indicate Part and Part Target availability and can be associated in Control Sources for static Part and Part Target sources.
- iPF (Integrated Part Feeder) - a feeder object that uses Ethernet communication with the IPC (Application Manager) running ACE software for control and feedback purposes. This is typically controlled using a Robot Vision Manager from a V+ program.

All of the above feeder objects can be configured in the Application Manager device of an ACE project as described in the following sections.

8-6-1 AnyFeeder Object

AnyFeeder objects represent an integrated parts feeding system optimized to work together with vision, motion, and robots. AnyFeeder objects can be added to provide control and configuration of the parts feeder in the ACE project.

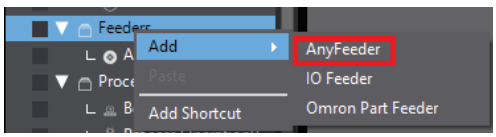
When Emulation Mode is enabled, all Feeder Function durations are emulated. Durations for error reset, initialization, operation abort, or firmware restart are not emulated because these operations are not intended to be requested during a feed cycle.

Additional Information

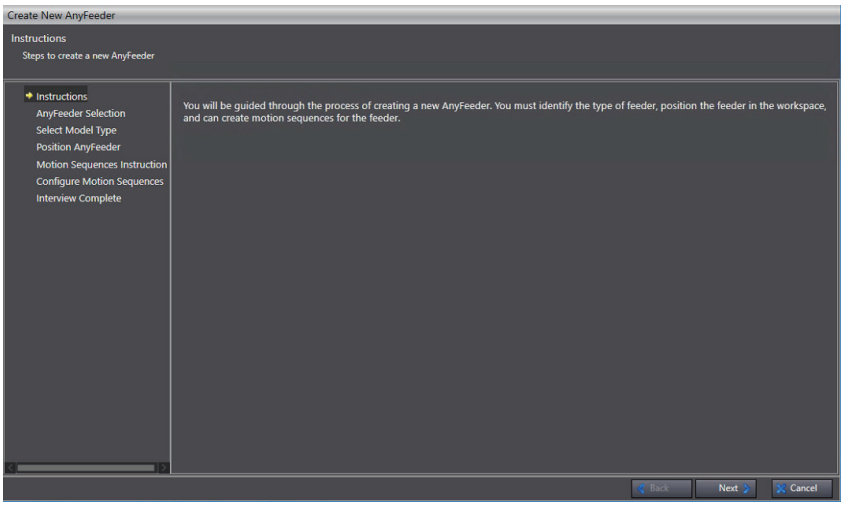
More information about AnyFeeder devices is available in the *AnyFeeders User's Guide* (Cat. No. 18876-000).

Adding an AnyFeeder Object

To add an AnyFeeder object, right-click **Feeders**, select **Add**, and then click **AnyFeeder**.



The *Create New AnyFeeder* wizard opens.



Make selections for the model type, position in the workspace, and motion sequences in the Create New AnyFeeder wizard. Click the **Finish** Button after completing all steps and then the AnyFeeder object will be added to the Multiview Explorer. Refer to the sections below for more information about wizard configuration items.

Additional Information

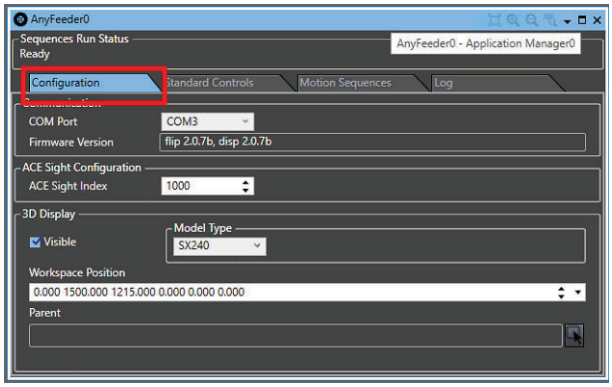
The communication port selection and connection test functions are not available while operating in Emulation Mode.

AnyFeeder Configuration

To access the AnyFeeder configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the AnyFeeder editor in the Edit Pane.
Use the information below to understand the AnyFeeder configuration items.

● Configuration Items

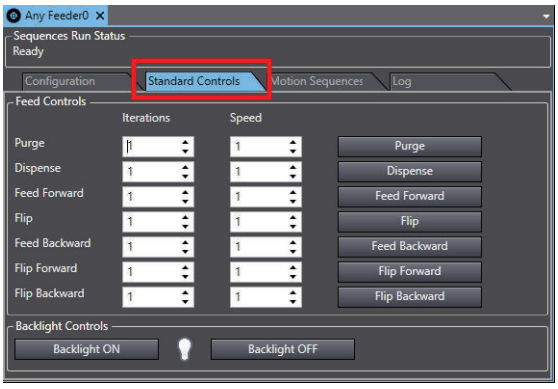
The Configuration tab contains the following items used for general configuration settings.



Group	Item	Description
Communication	COM Port	Specifies the PC serial communications port that is used to send commands to the AnyFeeder device. This is not available in Emulation Mode.
	Firmware Version	Displays the firmware version of the connected AnyFeeder device. This is not available in Emulation Mode
Robot Vision Manager Configuration	Robot Vision Manager Index	The Robot Vision Manager Index can be used to reference this configuration from a V+ program. Specify a unique number between 0 to 9999 as the Robot Vision Manager reference. When an AnyFeeder device configuration and a Robot Vision Manager sequence have the same index value, the PC will only try to run the Robot Vision Manager sequence.
3D Display	Model Type	Select the AnyFeeder Model type from the drop-down selection menu.
	Visible	Use the Visible selection box display or hide the device from the 3D Visualizer.
	Workspace Position	Enter the location of the AnyFeeder device in the workspace.
	Parent	Select a reference to a parent object.

● Standard Controls Items

The Standard Controls tab contains the following items used for manually controlling the AnyFeeder device. The buttons are used to control the AnyFeeder device as described below.



Group	Item	Description
Feed Controls	Feeder Functions	• Purge: Feed parts out of the feeder backwards. • Dispense: Move parts from the bulk container onto the feed surface. • Feed Forward: Feed parts forward. • Flip: Flip parts without moving forwards or backwards. • Feed Backward: Feed parts backward. • Flip Forward: Flip parts and move them forward. • Flip Backward: Flip parts and move the backward.
	Iterations	Specify the number of times this action is performed. There are some actions that are designed to occur at the end of all motion and will only occur once, regardless of iteration number. For example, the Dispense command causes the lip in the top bin of the AnyFeeder to retract at the beginning and then come back up at the end. This will only occur once when iteration is set to a number greater than 1.
	Speed	Set the speed of the Feeder Function. Settings can range from 0-10 where 0 is the minimum speed and 10 is the maximum speed.
Backlight Controls	Backlight ON/OFF	Turn the backlight ON or OFF. The indicator ( ) turns ON or OFF to show the state of the backlight.

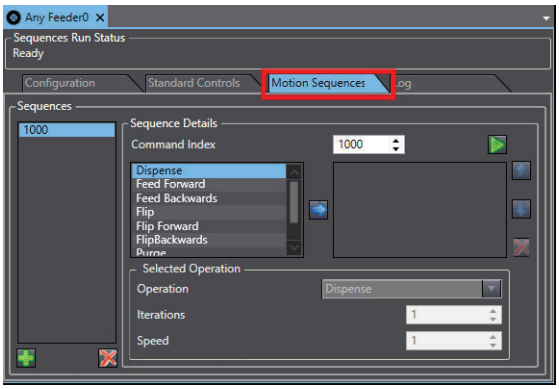
● Motion Sequence Configuration

The Motion Sequences tab shows a listing of high level motion sequences associated with the AnyFeeder device. A sequence can be defined as a collection of individual feeder functions. When a sequence is executed, all the operations are performed in the order defined in this area.
Motion sequences can be triggered through the AnyFeeder user interface, a C# program, or with a V+ program. Motion sequences are stored as Command Index numbers between 1000 to 10000. The motion sequence is referenced with this number in C# and V+ programs.

Sequences and sequence steps can be removed using the **Delete** buttons ( ).

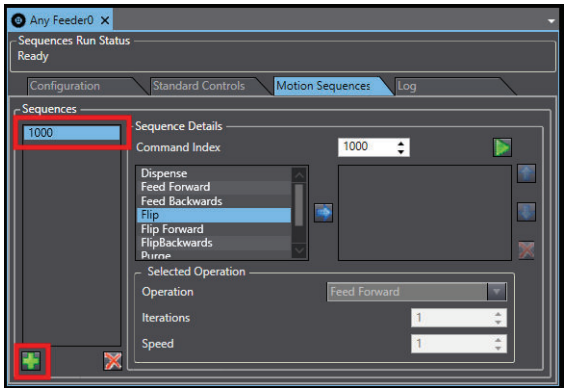
Sequence actions can be rearranged with the **Up** ( ) and **Down** ( ) buttons.

A sequence can be tested with the **Play** button ( ).



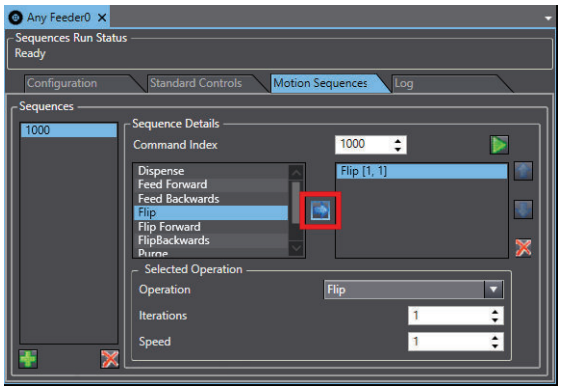
Use the following procedure to create new Motion Sequences.

1 Add a new sequence with the **Add** button ( ). A new sequence will be placed in the sequence list.



2 Adjust the Command Index value if necessary.

3 Select the first Feeder Function from the list and then click the **Select** Button ( ) to add it to the action list. The Flip function has been added in the example below.



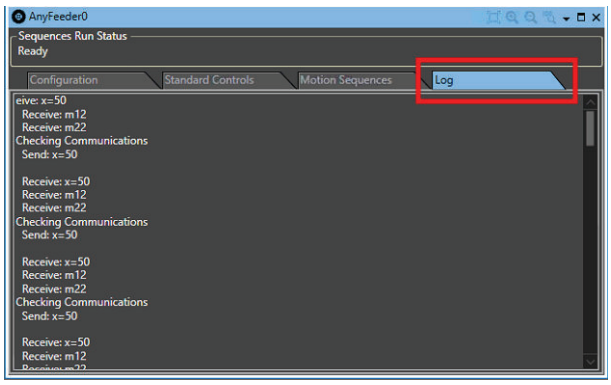
4 Select the action from the Selected Operation list and then make any necessary adjustments to Iterations and Speed.

5 Repeat steps 4 through 8 to add more actions to the sequence as needed.

6 Click the **Play** Button (▶) to execute the sequence as a test. This will cause the connected AnyFeeder to move and execute the sequence. If the sequence executes correctly, the procedure is complete.

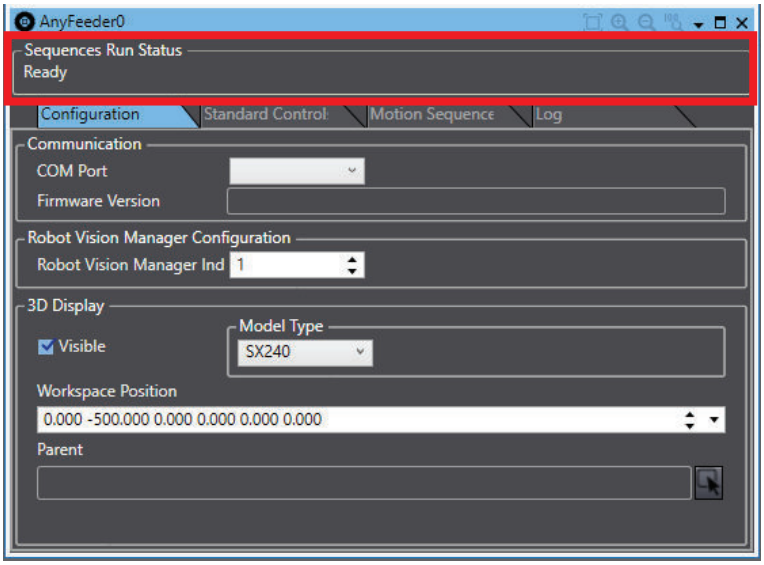
● Log Items

The log tab shows the communication history between the AnyFeeder device and ACE.



Sequences Run Status Area

The *Sequence Run Status* area displays status information.



Use the following messages and error codes displayed in the *Sequence Run Status* area to diagnose AnyFeeder problems.

Error Code	Message	Details
10	ErrorCodeMotor1Ready	Motor1 finished its movement. This value corresponds to an internal AnyFeeder Error code value.
11	ErrorCodeMotor1Busy	Motor1 is currently performing an operation.
12	ErrorCodeMotor1InvalidCommand	Motor1 received an invalid command.
13	ErrorCodeMotor1ServoError	Motor1 servo is in error.
16	ErrorCodeMotor1NotInitialized	Motor1 is not initialized.

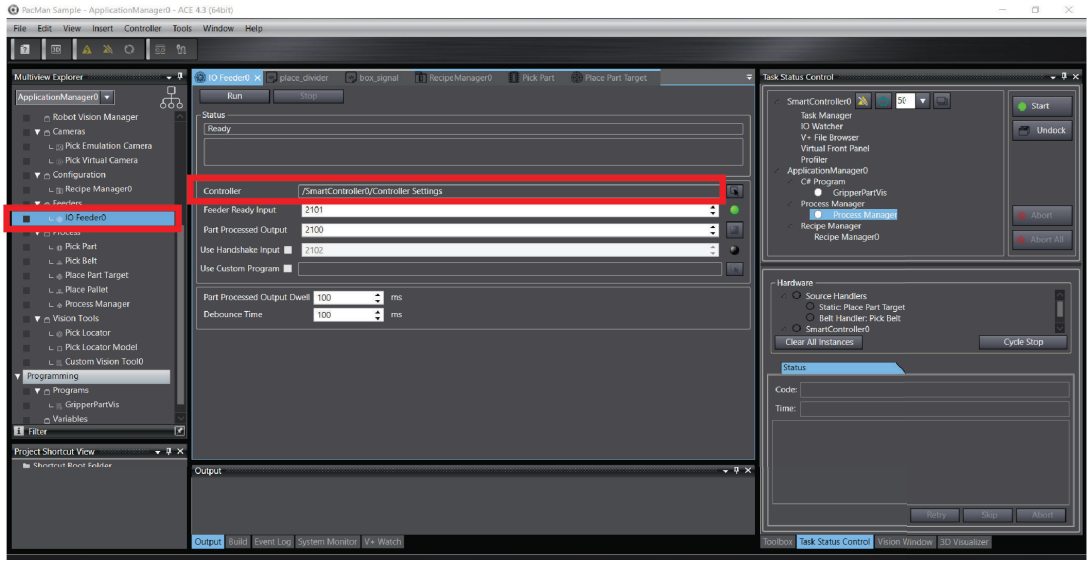
Error Code	Message	Details
17	ErrorCodeMotor1Error	Motor1 is in error.
20	ErrorCodeMotor2Ready	Motor2 finished its movement. This value corresponds to an internal AnyFeeder Error code value.
21	ErrorCodeMotor2Busy	Motor2 is currently performing an operation.
22	ErrorCodeMotor2InvalidCommand	Motor2 received an invalid command.
23	ErrorCodeMotor2ServoError	Motor2 servo is in error.
26	ErrorCodeMotor2NotInitialized	Motor2 is not initialized.
27	ErrorCodeMotor2Error	Motor2 is in error.
28	ErrorCodeTimeoutSynchro	Motor 2 did not receive the sync-signal from Motor1 to start action.
30	ErrorCodeTimeout	The AnyFeeder action was too long to complete its commanded movement.
31	ErrorCodeComPort	The current COM port is not valid.
32	ErrorCodeWriteCOM	An error happened while writing on COM port.
33	ErrorCodeReadCOM	An error happened while reading on COM port.
34	ErrorCodeInvalidComPort	The Com Port is not valid.
35	ErrorCodeComClosed	The COM port is closed.
36	ErrorCodeReplyFormat	The reply from the AnyFeeder is not well formatted.
37	ErrorCodeInvalidCommand	A command is not valid.
38	ErrorCodeNoConnection	There is no connection with the AnyFeeder device.

8-6-2 I/O Feeder Object

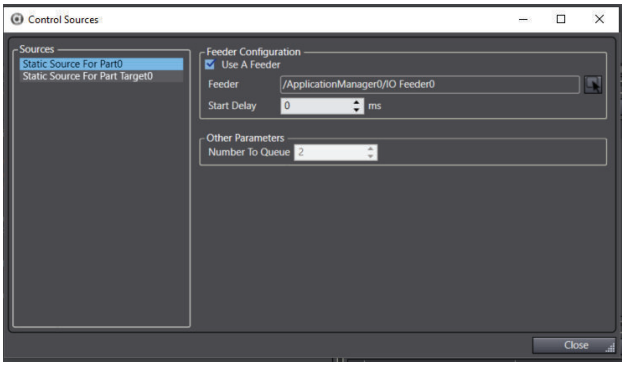
I/O Feeder objects represent generic feeder devices that are controlled with input and output signals from a connected robot controller.

Adding an IO Feeder Object

To add an IO Feeder object, right-click **Feeders**, select **Add**, and then click **IO Feeder**. A new IO Feeder object will be added to the Multiview Explorer.



On Multiview Explorer, with Process Manager opened, select the **Control Sources** Tab. Set the Sources and *Configuration*, and then close the Window.



IO Feeder Configuration

To access the IO Feeder configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This opens the IO Feeder editor in the Edit Pane.
Use the table below to understand the IO Feeder configuration items.

Item	Description
Run Button	Used to perform one test cycle of the feeder. The operation stops when the cycle has completed or if the **Stop** button is clicked before the end of the cycle. To repeat or restart the cycle, click the **Start** button again. When clicked, this button dims until the **Stop** Button is clicked (feeder test cycle has been interrupted) or the cycle has completed .
Stop Button	Stops (interrupts) the test cycle. The test cycle can be restarted by clicking the **Start** Button. This icon is dimmed until the **Start** Button is clicked (feeder test cycle has started).
Status	The Status item provides operation and error information about the I/O Feeder object. The current status is shown in the top field. A description of the status is shown in the lower field.

Item	Description
Controller	Specifies the controller that will process the feeder signals. Click the **Select** Icon ( ) to display the list of available controllers and then select a reference controller from the list.
Feeder Ready Input	Specifies the input signal that indicates the feeder is ready and available to present a part instance. When Emulation Mode is enabled, this signal is ignored, but soft signals can be substituted for testing purposes.
Part Processed Output	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. When Emulation Mode is enabled, this signal is ignored, but soft signals can be substituted for testing purposes.
Use Handshake Input	If enabled, the feeder will assert a signal indicating it has acknowledged the part processed signal. When Emulation Mode is enabled, this signal is ignored, but soft signals can be substituted for testing purposes.
Use Custom Program	The feeder interface code runs as a V+ program on the specified controller. This program can be overwritten if some custom logic needs to be applied. Select the custom program and enable this option by clicking the **Select** Icon ( ). A custom V+ program selection wizard will appear to step through the procedure.
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 detected in the ON state before it is considered logically ON.

8-6-3 iPF Object

The Integrated Part Feeder (iPF) object represents the system used to position and orient a wide range of components within a part picking application. iPF objects can be added to provide control and configuration of the system in an ACE project.
When Emulation Mode is enabled, all iPF function durations are emulated. Durations for error reset, initialization, operation abort, or firmware restart are not emulated because these operations are not intended to be requested during a feed cycle.

Precautions for Correct Use

- When the purge flap is open, do not use any other vibrations than left or right. If the constraint is not respected, the mechanism could be irreparably damaged.
- The purge option activation modify the vibration behavior. Do not forget to deactivate the purge option when you remove the purge kit.

Additional Information

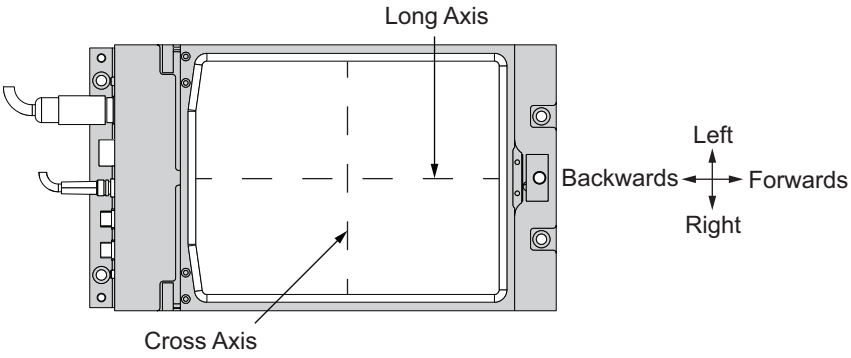
Refer to the *iPF-Series Flexible Part Feeder User's Manual* for more information.

The following operations are available with the iPF device.

- Feed Forward: Move parts in the forward direction along the picking surface.
- Feed Backward: Move parts in the backward direction along the picking surface .
- Feed Left: Move parts in the left direction along the picking surface.

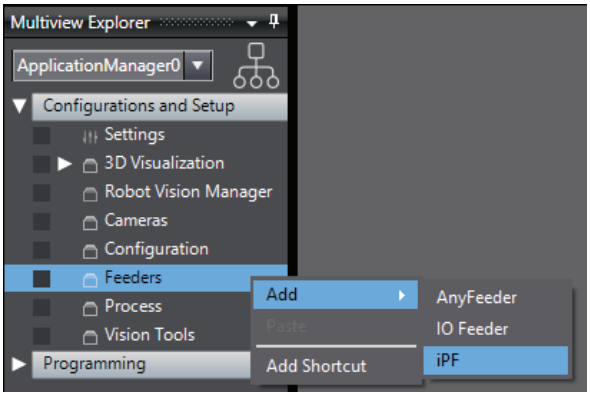
- Feed Right: Move parts in the right direction along the picking surface.
- Flip: Flip and spread parts to distribute them across the picking surface.
- Long Axis Centering: Center parts along the long axis of the picking surface.
- Cross Axis Centering: Center parts along the cross (short) axis of the picking surface.
- Dispense Hopper 1 (when a Hopper 1 is present): Dispense parts from Hopper 1.
- Dispense Hopper 2 (when a Hopper 2 is present): Dispense parts from Hopper 2.
- Purge System (when purge is enabled and a Hopper is present): Purge parts from the hopper and feeder.
- Purge Feeder (when purge is enabled): Purge parts from the feeder.
- Wait: Wait for a specified amount of time (ms) before proceeding to the next operation.

A directional reference is provided below.

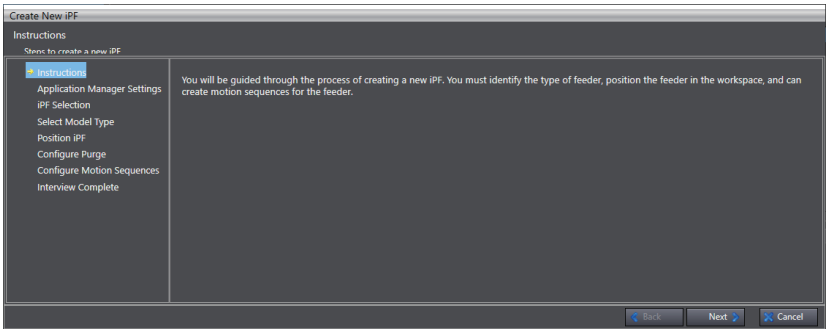


Adding an iPF Object

To add an iPF object, right-click **Feeders**, select **Add**, and then select **iPF**.



The *Create New iPF* wizard opens.

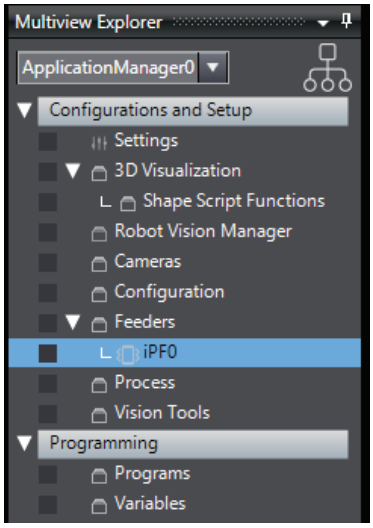


Step through the following settings in the wizard.

- Application Manager Settings

- iPF Selection
- Select Model Type
- Position iPF
- Configure Purge
- Configure Motion Sequences

After all steps in the wizard are completed, the new iPF object will be available in the Multiview Explorer.

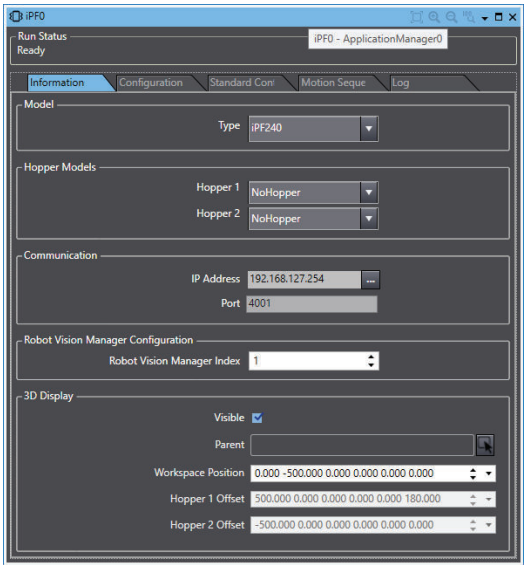


iPF Configuration

To access the iPF configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This opens the iPF editor in the Edit Pane.
Use the information below to understand the iPF configuration items.

● Information

The *Information* tab is used to set ACE project properties of the iPF object as described below.



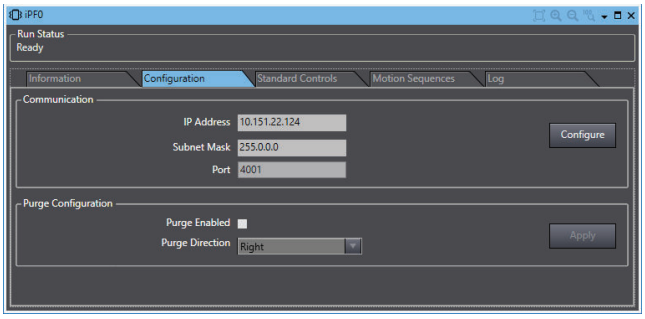
Group	Item	Description
Model	Type	iPF device model type that ACE will communicate with.

Group	Item	Description
Hopper Models	Hopper 1	Hopper model type that is present in the system and associated output on the iPF device that will be connected to the Hopper.
	Hopper 2*1	
Communication	IP Address	IP address and Port that ACE will use to communicate with the iPF device.
	Port	
Robot Vision Manager Configuration	Robot Vision Manager Index	Robot Vision Manager Index number that will be used to reference this configuration from a V+ program. Specify a unique number between 1 to 10000 as the Robot Vision Manager reference. Refer to *Robot Vision Manager User's Manual (Cat. No. I667)* for more information. ACE will not allow duplicate values for Robot Vision Sequence numbers. ACE will automatically assign the next available Robot Vision Manager sequence number when creating an iPF object to prevent duplications.
3D Display	Visible	Display or hide the device from the 3D Visualizer.
	Parent	Reference to a parent object that specifies the object the iPF device will be relative to.
	Workspace Position	Location of the iPF device in the workspace.
	Hopper 1 Offset	Position offset of the Hopper relative to the iPF.
	Hopper 2 Offset	

*1. Hopper 2 availability may be restricted by the iPF240 configuration. Refer to the *iPF-Series Flexible Part Feeder User's Manual* for more information.

● Configuration

The *Configuration* tab is used to make settings that are stored in the iPF device.



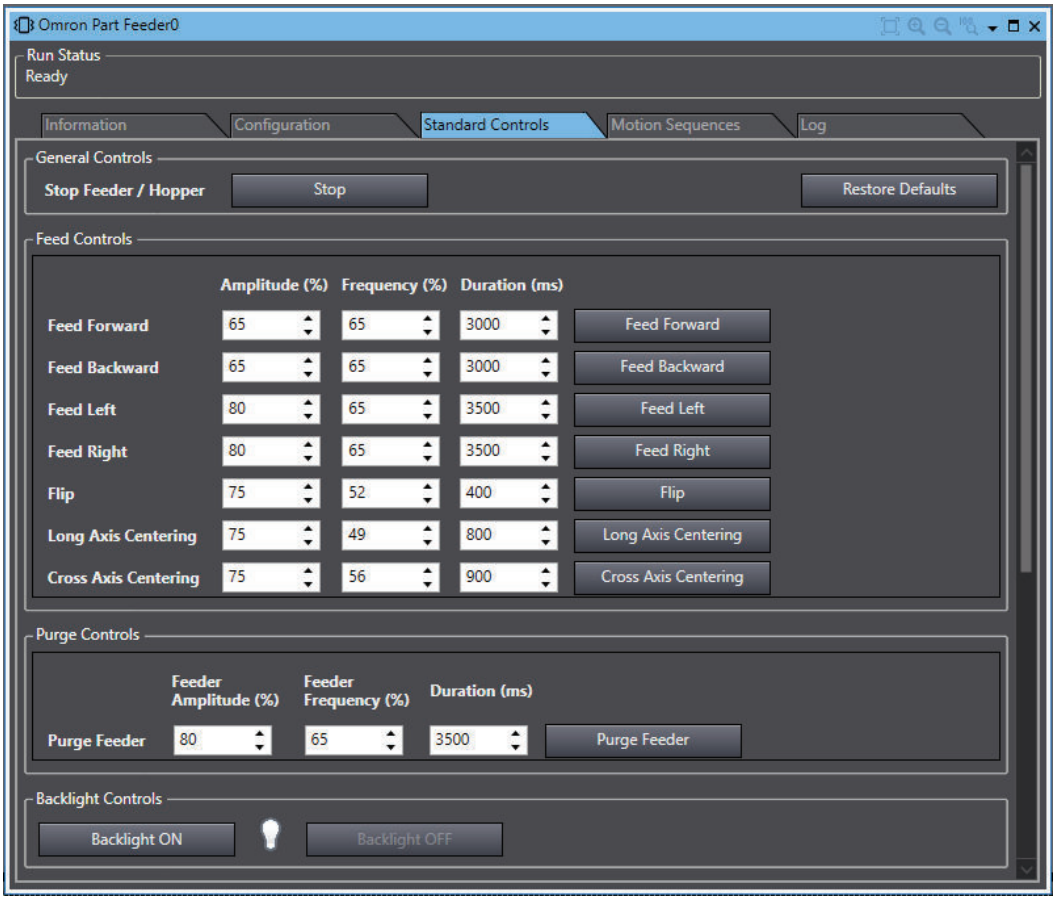
Group	Item	Description
Communication*1	IP Address	Network settings for the iPF device.
	Subnet Mask	
	Port	
	Configure Button	Change network settings for the iPF device.
Purge Configuration	Purge Enabled	Enable or disable the purge function.
	Purge Direction	Select **Right** or **Left** to set the Purge Direction.
	Apply Button	Apply settings after changes are made.

*1. Communication settings are not available when Emulation mode is active.

Standard Controls

The *Standard Controls* tab is used to manually adjust all motions and control operations that apply to the iPF object.

The values that are present in the Feed Controls area are implemented as the default values in the Motion Sequences area.

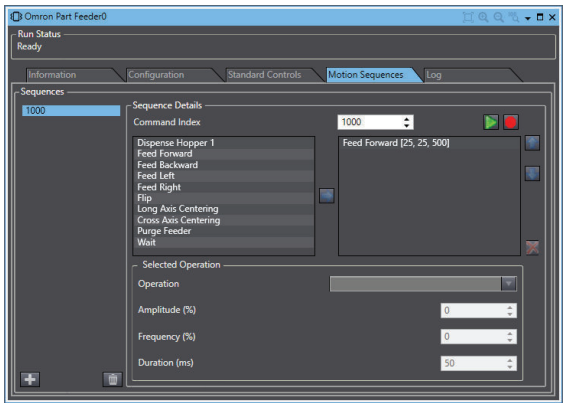


Item	Description
General Controls	Use the **Stop** button to halt all ongoing operations. Use the **Restore Defaults** button to set all feed control values to default settings.
Feed Controls	Adjust Amplitude (%), Frequency (%), and Duration (ms) for feed control motions. Use the buttons to manually execute the operation.
Purge Controls	Adjust Amplitude (%), Frequency (%), and Duration (ms) for the feeder purge function. Use the **Purge Feeder** button to manually execute the operation.
Backlight Controls	Turn the backlight ON and OFF manually.

Motion Sequences

The *Motion Sequences* tab is used to create and edit iPF sequences. Each sequence is identified by a specific Command Index (1000 to 10000). A sequence can be executed using its Command Index with a C# program, V+ program, or within the *Motion Sequences* tab.

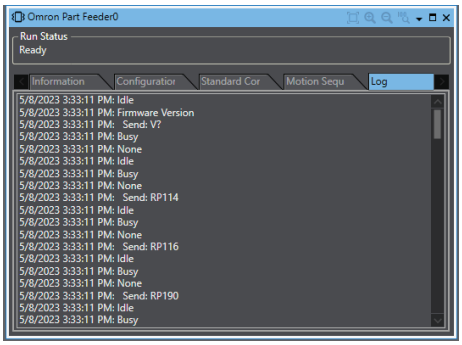
When a sequence is executed, iPF operations are performed in the order defined in this area.



Item	Description
Sequences	Create a new Sequence by clicking the Plus ( ) Button. Remove the selected Sequence by clicking the Delete ( ) Button.
Command Index	Modify the Command Index value for the currently selected Sequence.
Execute Button	Run the Sequence that is currently selected with the Execute ( ) Button.
Stop Button	Stop the sequence that is currently selected with the Stop ( ) Button.
Command Sequence Arrangement Buttons	Use the right arrow ( ) Button to add an Operation to the selected Sequence. Use the Up ( ) and Down ( ) Buttons to arrange the order of the Operation within the selected Sequence. Use the Delete ( ) Button to remove an Operation from the selected Sequence.
Selected Operation	Use the following items in the *Selected Operation* area to adjust each Command. All adjustments are unique to the selected Sequence. • Operation: Select an Operation to adjust. • Amplitude: Set the vibration amplitude for the selected Operation. • Frequency: Set the vibration frequency for the selected Operation. • Duration: Set the time duration for the selected Operation.

● Log

The *Log* tab is used to view the communication history of the iPF device.



Errors

The following table displays a list of error details and their respective corrective actions.

Additional Information

Contact your local OMRON representative for information about other errors not listed in the table below.

Error Number	Error Description	Likely Cause	Corrective Action
[00012]	No connection with iPF	A connection could not be established with the iPF device	Check the network settings and the physical network connections.
[00013]	Error toggling purge	The purge gate could not open or close.	Check the purge area for obstructions. Check the purge gate sensor for damage or misalignment.
[00014]	Purge gate closure timeout	The purge gate was not able to close in the expected time.	
[00015]	No S-power.	S-power supply not detected.	Check S-power supply and wiring.
[00017]	iPF is above the maximum allowable temperature.	Excessive heat detected.	Reduce the Flip operation frequency or duration. Check the ambient operating temperature to ensure it is within the allowable range.

8-7 Process Objects

Process objects are used to define, configure, and manage the process elements of an Pack Manager packaging application.

As Process objects are added to the Application Manager device, they can be shown in the 3D Visualizer, if the **Visible** option or **Show in Visualization** option is enabled in the object editor. For example, in addition to the robot, the 3D Visualizer shows:

- The robot Belt object windows.
- The location of latch sensors.
- The location of each camera field of view.
- Part and Part Target instances.



Additional Information

Part and Part Target instances are represented in the 3D Visualizer with different colors to indicate their current state.

- Yellow: part is allocated.
- Orange: part is not allocated.
- Light green: part target is allocated.
- Green: part target is not allocated.

Process objects listed below are described in detail in the following sections.

Part Buffer

The Part Buffer object defines an object that is an overflow area where parts can be temporarily stored when part targets are unavailable to accept more parts.

Refer to *8-7-1 Part Buffer Object* on page 8-98 for more information.

Part Target

The Part Target object defines an object that is a possible destination for a part.

Refer to *8-7-2 Part Target Object* on page 8-101 for more information.

Part

The Part object defines an object to be picked and processed to an available Part Target.

Refer to *8-7-3 Part Object* on page 8-104 for more information.

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.

Refer to *8-7-4 Belt Object* on page 8-108 for more information.

Process Manager

Process Managers are the central control point for developing packaging applications. A Process Manager is used to create complex applications without having to write any programming code. It provides access to automatically generated V+ and C# programs to customize the default behavior to meet the requirements of an application, if necessary.

The Process Manager run-time handler is the supervisory control for the entire packaging system, managing allocation and queuing for multiple controllers, robots, conveyors, parts, and targets.

Refer to *8-7-5 Process Manager Object* on page 8-117 for more information.

Allocation Script

The Allocation Script object is used to create and edit custom part-allocation programs for use with the Process Manager.

Refer to *8-7-6 Allocation Script Object* on page 8-160 for more information.

Pallet

The Pallet object defines the layout of a pallet, which can be used to pick parts from or place parts to. Refer to *8-7-7 Pallet Object* on page 8-161 for more information.

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.

Refer to *8-7-8 Vision Refinement Station Object* on page 8-167 for more information.

Process Operation

The Process Operation object defines an object that is used to add a program to an operation to expand operational functions.

Refer to *8-7-10 Process Operation Object* on page 8-189 for more information.

8-7-1 Part Buffer Object

The Part Buffer object defines a physical object that is an overflow area where parts can be temporarily stored when an output conveyor belt or feeder is unavailable to accept the parts. This buffer can hold a single part, it can be a static pallet, or just a flat surface that acts as a pallet that holds multiple parts.

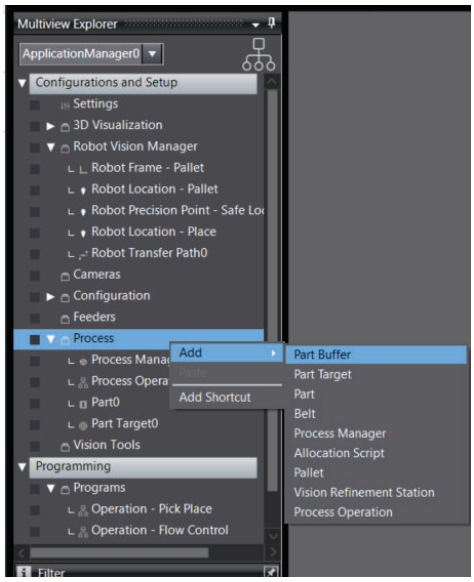
Control the access method for pallets in the following ways.

- Last In, First Out (LIFO).
- First In, First Out (FIFO).

The Pallet object editor is used to specify the pallet parameters and access method. Refer to *8-7-7 Pallet Object* on page 8-161 for more information.

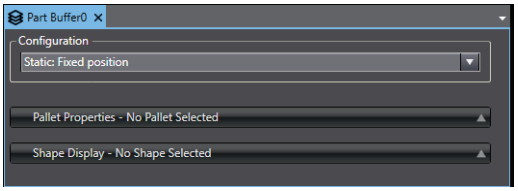
Adding a Part Buffer Object

To add a Part Buffer object, right-click **Process**, select **Add**, and then click **Part Buffer**. A new Part Buffer object will be added to the Multiview Explorer.



Part Buffer Configuration

To access the Part Buffer configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Part Buffer editor in the Edit Pane.



● Configuration Drop-down List

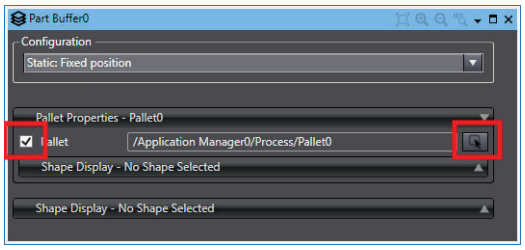
The Configuration drop-down list box is used to specify how the Part Buffer is used by the system. **Static: Fixed Position** is the only option for this item. This means parts are placed at a static location.

This object does not support dynamic Part Buffers by default and operates under the assumption that parts placed in the buffer will be available in the same position when accessed later. Be sure to consider the physical state of parts in the buffer when a Process Manager is stopped and restarted.

● Pallet Properties

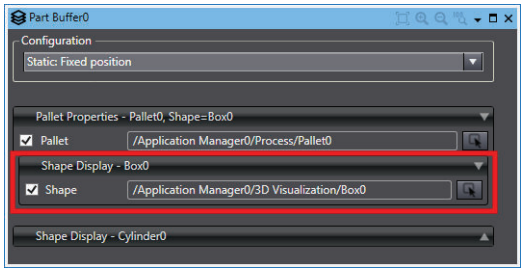
Specify a pallet that is used to hold the part(s) with the *Pallet Properties* area. Make reference to a Pallet object by checking **Pallet** and then click the **Select** Button to specify that object as shown below.

A Pallet object must already exist in the project. Refer to *8-7-7 Pallet Object* on page 8-161 for more information.



● Shape Display (Pallet)

Use the Shape Display to specify a shape to represent the Pallet in the 3D Visualizer when the Process Manager runs. Select the **Shape** Check Box and then click the **Select** Button to specify a shape.



Adding a Part Buffer to a Process

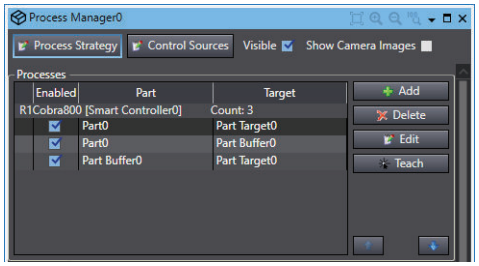
Use the following procedure to add a Part Buffer to a Process after creating and configuring the Part Buffer object.
The default Process Strategy can choose the appropriate process based on part / target availability.

1. Create a Process that includes a Robot, a Part as the Pick Configuration, and a Part Target as the Place Configuration.

2. Create a second Process that includes the original Robot and Part, and select the Part Buffer as the Place Configuration.

3. Create a third Process that includes the original Robot and Part Target, and select the Part Buffer as the Pick Configuration.

4. Use the **Up** and **Down** buttons in the Process Manager Processes area to arrange the process by priority. The process at the top of the list has the highest priority.



5. Set the Part Buffer access order. After the Part Buffer access order is set, the procedure is complete.

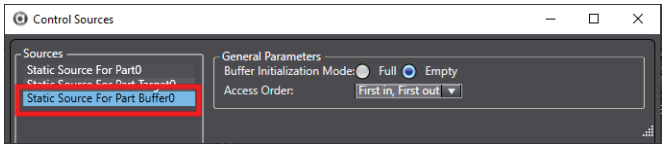
Setting the Part Buffer Access Order

When a pallet is used for the part buffer it is necessary to specify how the parts will be accessed as the buffer is being emptied. The following options are used.

- 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.

When parts are stacked (more than one layer is specified for the pallet), the access order must be set as LIFO.
Use the following procedure to set the Part Buffer access order. Refer to *Control Sources* on page 8-148 for more information.

1. In the Control Sources setting area, select the **Static Sources For Part Buffer** item from the Sources list (if the Part Buffer object has been renamed, select the corresponding item).



2 If necessary, 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.

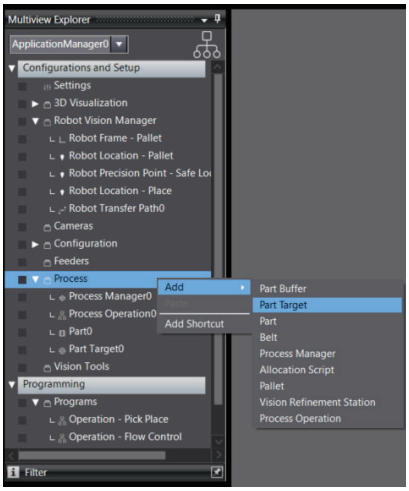
3 Select the required Access Order from the list and then the Part Buffer access order setting procedure is complete.

8-7-2 Part Target Object

The Part Target object defines a location that is a possible destination for a part. The Part Target has a configuration property that specifies how the target / instance is handled by the application.

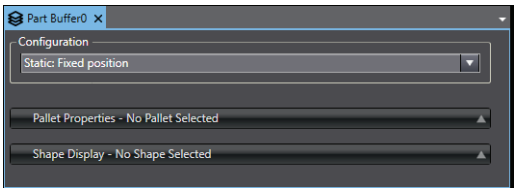
Adding a Part Target Object

To add a Part Target object, right-click **Process**, select **Add**, and then click **Part Target**. A new Part Target object will be added to the Multiview Explorer.



Part Buffer Configuration

To access the Part Buffer configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Part Buffer editor in the Edit Pane.



● Configuration Drop-down List

The Configuration drop-down list box is used to specify how the target is input to the system. The options are described below.

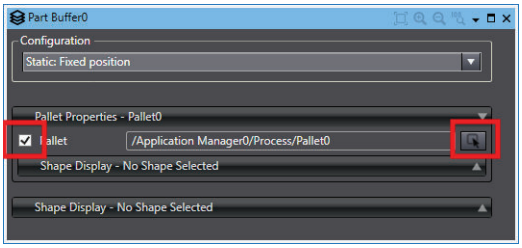
- **Belt: Latching, belt camera, or spacing interval** - targets / instances are located on a conveyor belt using latching or fixed-spacing. Vision and / or a pallet may be included in the part delivery system. Refer to *Belt: Latching, Belt Camera, or Spacing Interval Configuration* on page 8-106 for more information.
- **Static: Fixed position** - targets / instances are acquired from a static location such as a part feeder or a pallet. Refer to *Static: Fixed Position Configuration* on page 8-107 for more information.
- **Vision: Fixed camera not relative to a belt** - locations are acquired through camera that is not located over a belt. Refer to *Vision: Fixed camera not relative to belt* on page 8-107 for more information.

If the part is supplied on a belt with a camera, the **Belt: Latching, belt camera, or spacing interval** option must be selected.

● Pallet Properties

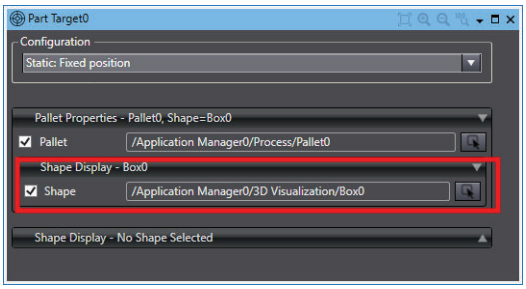
Specify a pallet that is used to hold the part(s) with the Pallet Properties area. Make reference to a Pallet object by checking **Pallet** and then use the **Select** Button to specify that object as shown below.

A Pallet object must already exist in the project. Refer to *8-7-7 Pallet Object* on page 8-161 for more information.



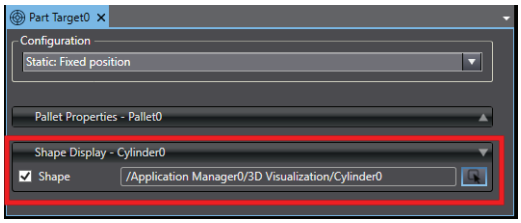
● Shape Display (Pallet Properties)

Use the Shape Display to specify a shape to represent the Pallet in the 3D Visualizer when the Process Manager runs. Select the **Shape** Check Box and then use the **Select** Button to specify a shape.



● Shape Display

Use the Shape Display to specify a shape to represent the Part in the 3D Visualizer when the Process Manager runs. Select the **Shape** Check Box and then use the **Select** Button to specify a shape.



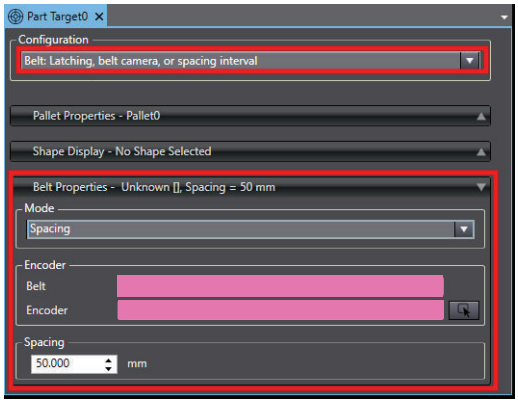
● Belt: Latching, Belt Camera, or Spacing Interval Configuration

When Belt is selected for the Part Target configuration, the operation mode can be either Vision, Latch, or Spacing.

- Vision: a vision tool is used to detect the part on the belt.
- Latch: a latch sensor is used to detect the part on the belt.
- Spacing: parts are spaced evenly on the belt.

● Belt Properties

This area is used to select the mode of the belt that is used to handle the part. Specify other details related to the belt mode selection in this area. Use the information below to make appropriate selections.



Item	Description
Belt / Encoder	Select the encoder from a list of available Process Belt Encoders. This will populate the associated Belt object automatically.
Vision Tool (Vision Mode)	Select the vision tool used to detect the part on the belt. For 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. As an option, select **Use Named Instance (select Model or enter custom result name)**and then use the **Select** Button to reference an existing Locator Model or use the **Add** Button to add a custom result name.
Spacing (Spacing Mode)	Specify the spacing in millimeters between targets / instances on the conveyor belt. It is important to recognize that there is no physical reference for where the spacing instances begin when a Process Manager is started. Spacing is convenient for emulation purposes, but in physical systems an upstream or downstream process or conveyor belt geometry may require that the spacing be synchronized with the rest of the machine. A latch sensor is recommended in this situation.

● Static: Fixed Position Configuration

When Static is selected for the Part Target configuration, the Part Target is in a fixed position. There are no additional settings to configure with this selection.

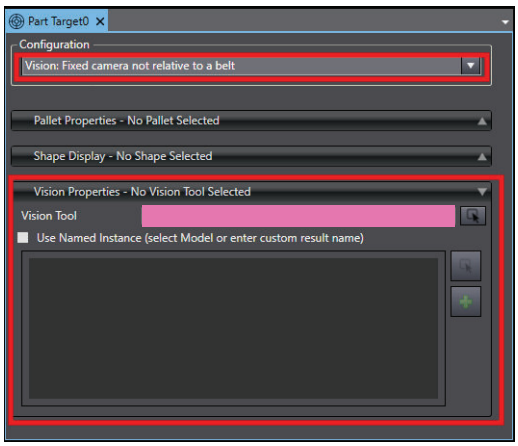
● Vision: Fixed camera not relative to belt

When Vision is selected for the Part Target configuration, a vision tool must be specified in the Vision Properties area.

● Vision Properties

This area is used to select the vision tool and optionally, the named instance that is used to acquire the part position.

Click the check box to specify a named instance and then select a Model or enter a custom result name.

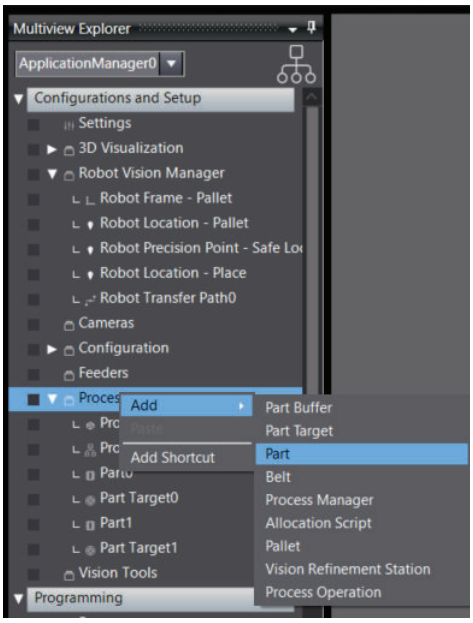


8-7-3 Part Object

The Part object defines a physical object that is input to the application for processing. The Part has a configuration property that specifies how the target / instance is input to the application.

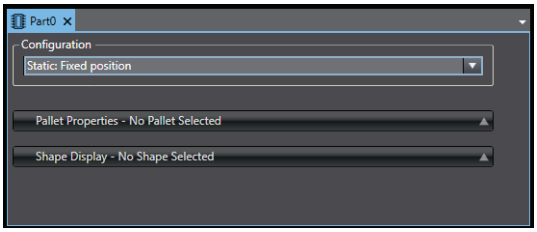
Adding a Part Object

To add a Part object, right-click **Process**, select **Add**, and then click **Part** . A new Part object will be added to the Multiview Explorer.



Part Configuration

To access the Part configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Part editor in the Edit Pane.



● Configuration Drop-down List

The Configuration drop-down list box is used to specify how the target is input to the system. The options are described below.

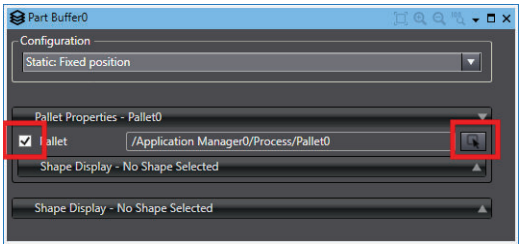
- **Belt: Latching, belt camera, or spacing interval** - targets / instances are located on a convey- or belt using latching or fixed-spacing. Vision and / or a pallet may be included in the part delivery system. Refer to *Belt: Latching, Belt Camera, or Spacing Interval Configuration* on page 8-106 for more information.
- **Static: Fixed position** - targets / instances are acquired from a static location such as a part feeder or a pallet. Refer to *Static: Fixed Position Configuration* on page 8-107 for more information.
- **Vision: Fixed camera not relative to a belt** - locations are acquired through camera that is not located over a belt. Refer to *Vision: Fixed camera not relative to belt* on page 8-107 for more information.

If the part is supplied on a belt with a camera, the **Belt: Latching, belt camera, or spacing interval** option must be selected.

Pallet Properties

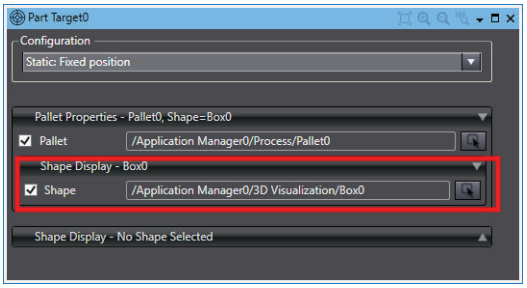
To specify a pallet that is used to hold the part(s) with the Pallet Properties area. Make reference to a Pallet object by checking **Pallet** and then use the **Select** Button to specify that object as shown below.

A Pallet object must already exist in the project. Refer to *8-7-7 Pallet Object* on page 8-161 for more information.



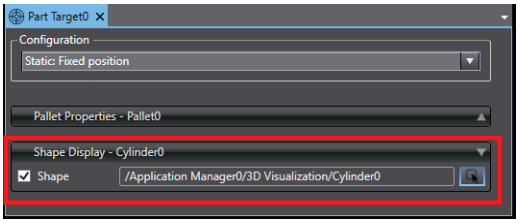
Shape Display (Pallet Properties)

Use the Shape Display to specify a shape to represent the Pallet in the 3D Visualizer when the Process Manager runs. Select the **Shape** Check Box and then use the **Select** Button to specify a shape.



Shape Display

Use the Shape Display to specify a shape to represent the Part in the 3D Visualizer when the Process Manager runs. Select the **Shape** Check Box and then use the **Select** Button to specify a shape.



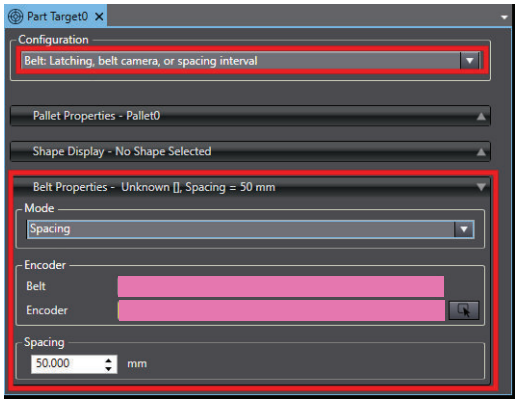
Belt: Latching, Belt Camera, or Spacing Interval Configuration

When Belt is selected for the Part Target configuration, the operation mode can be either Vision, Latch, or Spacing.

- Vision: a vision tool is used to detect the part on the belt.
- Latch: a latch sensor is used to detect the part on the belt.
- Spacing: parts are spaced evenly on the belt.

● Belt Properties

This area is used to select the mode of the belt that is used to handle the part. Specify other details related to the belt mode selection in this area. Use the information below to make appropriate selections.



Item	Description
Belt / Encoder	Select the encoder from a list of available Process Belt Encoders. This will populate the associated Belt object automatically.
Vision Tool (Vision Mode)	Select the vision tool used to detect the part on the belt. For 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. As an option, select **Use Named Instance (select Model or enter custom result name)**and then use the **Select** Button to reference an existing Locator Model or use the **Add** Button to add a custom result name.
Spacing (Spacing Mode)	Specify the spacing in millimeters between targets / instances on the conveyor belt. It is important to recognize that there is no physical reference for where the spacing instances begin when a Process Manager is started. Spacing is convenient for emulation purposes, but in physical systems an upstream or downstream process or conveyor belt geometry may require that the spacing be synchronized with the rest of the machine. A latch sensor is recommended in this situation.

● Static: Fixed Position Configuration

When Static is selected for the Part Target configuration, the Part Target is in a fixed position. There are no additional settings to configure with this selection.

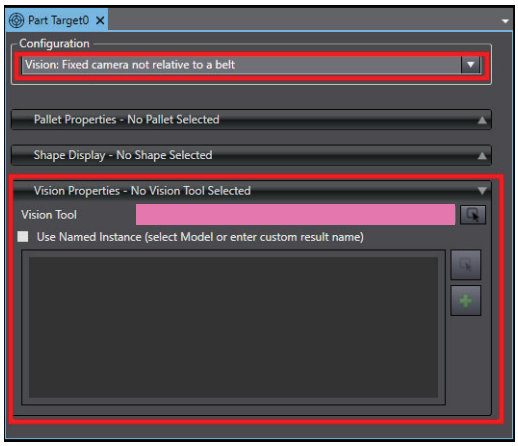
● Vision: Fixed camera not relative to belt

When Vision is selected for the Part Target configuration, a vision tool must be specified in the Vision Properties area.

● Vision Properties

This area is used to select the vision tool and optionally, the named instance that is used to acquire the part position.

An option is to specify a named instance and then select a Model or enter a custom result name.



8-7-4 Belt Object

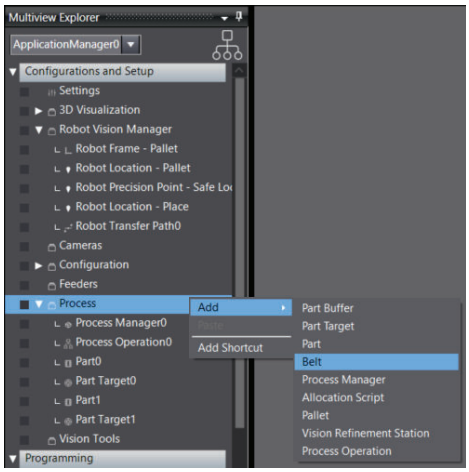
A Belt object represents a physical conveyor belt in the workcell or packaging line. A belt may be tracked by multiple robots that may be controlled by a single or multiple controllers. Belts may also be related to multiple part or part target objects.

The Belt object provides settings for Active Control, Emulation Mode behavior, workspace positioning for 3D visualization and multi-robot allocation order, and a list of associated belt encoder inputs for related controllers. This section will describe how these settings are used.

Belt objects operate differently in Emulation Mode. Refer to *Emulation Mode Differences* on page 1-9 for more information.

Adding a Belt Object

To add a Belt object, right-click **Process**, select **Add**, and then click **Belt**. The new Belt object wizard will open.



Controllers to be associated with the new Belt object must be online before adding the Belt object.

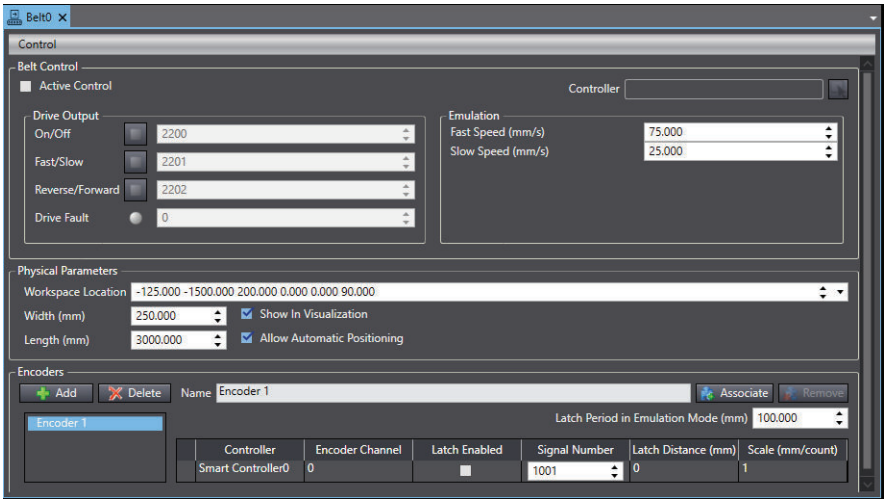
Follow the wizard steps below to define the Belt object.

- Set the belt size.
- Position the belt in the workspace.
- Select any controllers to be associated with the belt.
- Set belt control I/O signals (if the belt can be controlled).
- Select and test the encoder channel(s).

A Belt object will be added to the Multiview Explorer when the wizard steps complete.

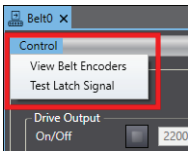
Belt Configuration

To access the Belt configuration, double-click the object in the Multiview Explorer or right-click it and select **Edit**. This will open the Belt editor in the Edit Pane.



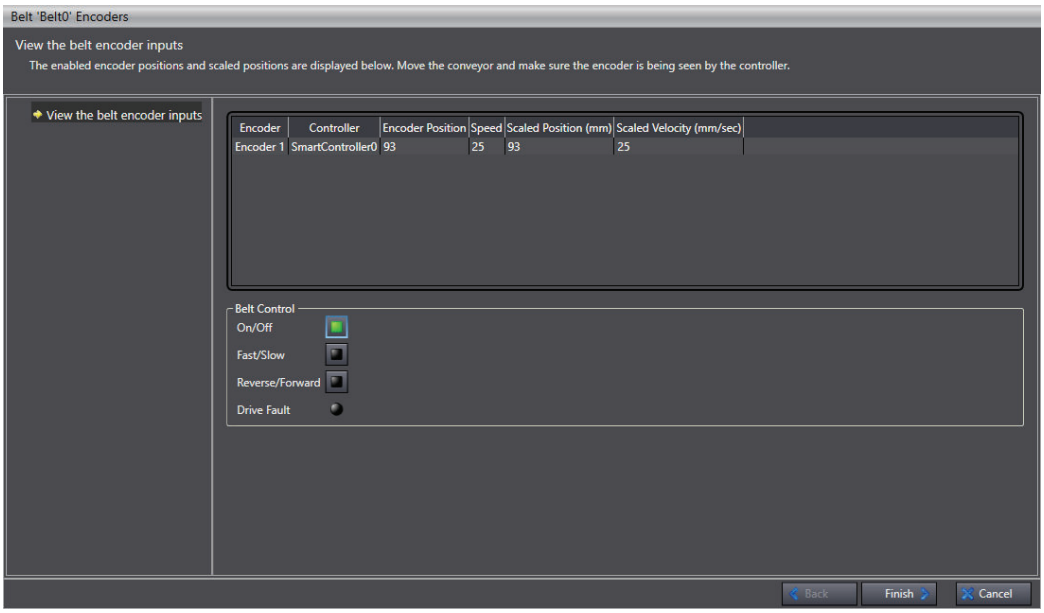
Control Menu Items

The Belt Control Menu items provide the following functions that can be used as necessary while configuring a Belt object.



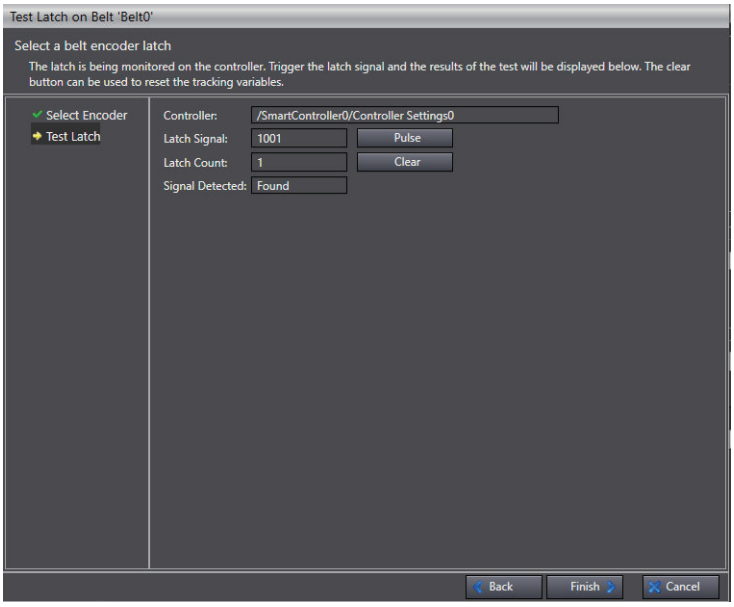
● View Belt Encoders

Selecting this menu item will display a wizard used to view the operation of the belt encoder. Select a belt encoder from the list and then activate the belt and observe the values. These values should change when the belt moves. The belt controls in the wizard are available if Active Control is enabled or ACE is in Emulation mode.

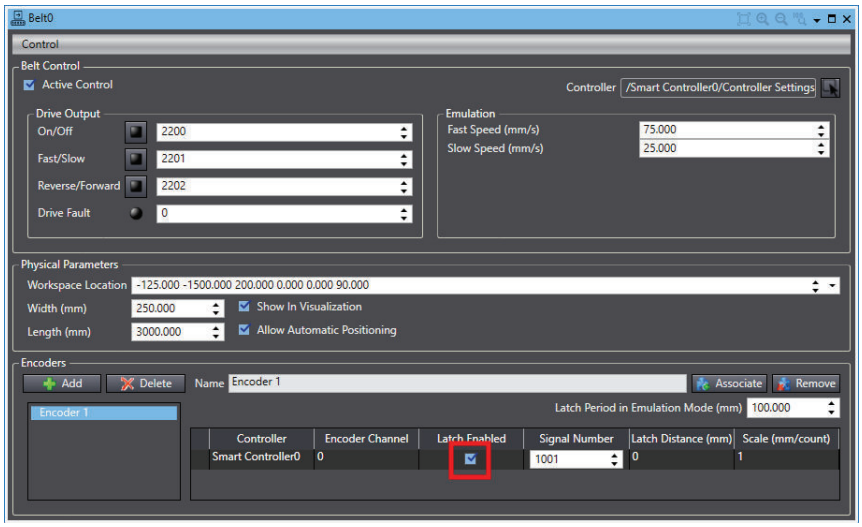


● Test Latch Signal

Selecting this menu item will display a wizard used to view the operation of the encoder latch signal. To test encoder latch, click the **Next** Button, and then use the **Pulse** Button to ensure the Latch Count increments correctly.



A Belt Encoder Latch must be configured and the latch must be enabled in the Encoders section to make the latch signal appear in the Test Latch Signal wizard. Refer to *Configure Belt Encoder Latches* on page 7-21 for more information.



Active Control

Active Control can be configured to control the belt during the calibration and teaching process. It can also optionally be set to control the belt during run time based on part or part target instance location. If the conveyor belt can be controlled by robot controller digital I/O signals, enable Active Control, select the controller, and enter the appropriate signal numbers in the Drive Output fields. Typically, if the Process Manager can stop the belt without affecting upstream or downstream processes, then the controller of the robot positioned farthest downstream is selected to control the belt. This robot is usually selected to provide all robots the opportunity to process the part or part target instance, and if an instance is not processed by any robot, the belt can be stopped to ensure all of them are processed. Refer to *Process Strategy Belt Control Parameters* on page 8-155 for more information.

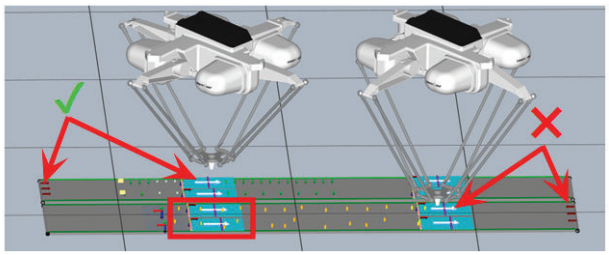
Physical Parameters

The Physical Parameters are generally set when creating the Belt object, but they can be modified as needed when the Process Manager is not executing. The Workspace Location, Width, and Length of the Belt object should be configured to closely approximate the position of the physical belt relative to robots and other workcell hardware. The accuracy of robot belt tracking behavior is dependent on the belt calibration and scale factor, not on the location of the Belt object. However, the Belt object shown in the 3D Visualizer provides a graphical representation of the conveyor and is used to understand the relative position of multiple robots along the belt for the purposes of instance allocation.
In Emulation Mode it is common to set the Belt object position before performing calibrations so that it can be used as a visual reference. On physical systems it is common to refine the position of the Belt object after performing belt calibrations.

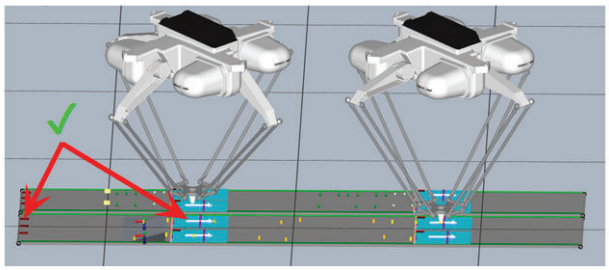
Belt Direction of Travel

In the 3D Visualizer, the Belt object displays three red arrows in the 3D Visualizer at one end of the belt that represent direction of belt travel. These red arrows must align with the white arrows that indicate direction of belt travel for each belt calibration.

The following figure shows examples of correct and incorrect belt direction of travel. Notice the lower belt has three red arrows on the right oriented in the opposite direction of the white arrows in the calibrations, which is incorrect. This results in the parts within the red box not being allocated to the upstream robot because the Belt object indicates parts should be allocated to the downstream robot first. This results in the downstream robot processing parts while all upstream robots do not pick available instances.



Incorrect belt direction of travel can be corrected by rotating the roll angle of the Belt object 180 degrees to align the direction of belt travel between the Belt object and the associated belt calibrations, and adjusting X and Y position to accurately represent the physical belt in the system. Once corrected, both robots can process instances as expected, where any instances not processed by the upstream robot can be processed by the downstream robot as shown in the following figure.

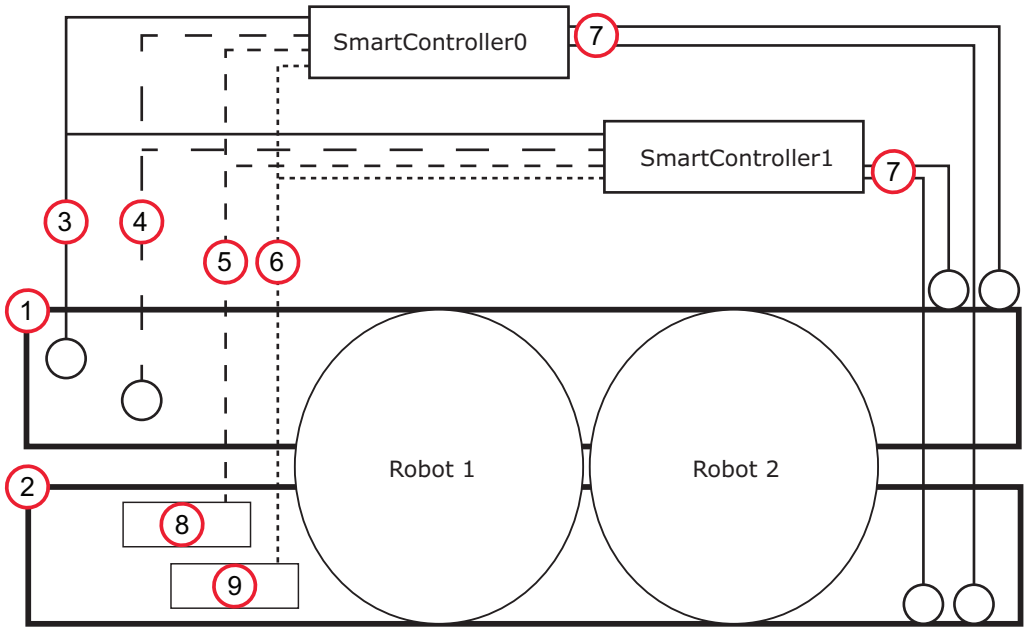


Encoders and Encoder Associations

The Belt object provides an interface for defining Encoders and Encoder Associations. A single Belt object may be used as a belt source for multiple part and part target types, and also may be associated with belt encoder inputs for multiple controllers.
Belt object Encoders are effectively virtual encoders for the purpose of constructing tracking structures and allocation limits independently for each part or part target object type.
Encoder Associations are used to understand the physical belt encoder and latch signal inputs that are wired to each robot controller that tracks the belt. Depending on the system configuration, it may be necessary to configure Encoders, Encoder Associations, or both. This section will describe the different situations that are supported.

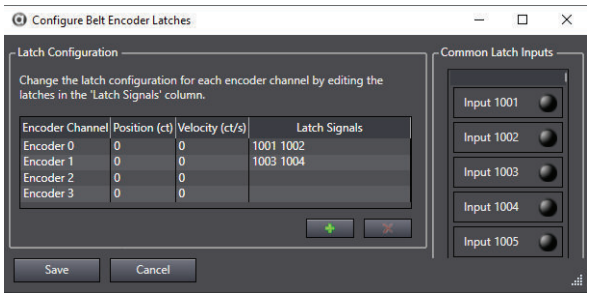
● Multi-Robot, Multi-Belt Configuration Example

Consider the system configuration shown in the following figure with two robots and two controllers (one controller per robot). The lower belt is the part picking belt with two belt-relative cameras locating two different part types (camera one locates Part1 one and camera two locates Part2). The upper belt is the part placing belt with two latch sensors used for generating two different part target type instances. A belt encoder input for each belt is required for each controller.

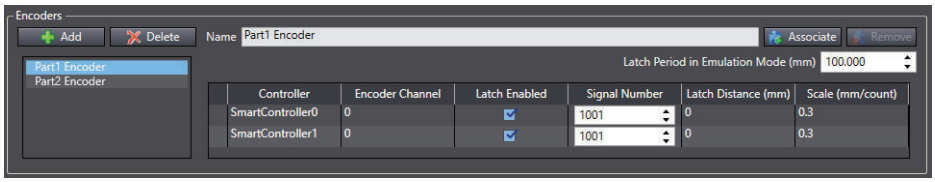


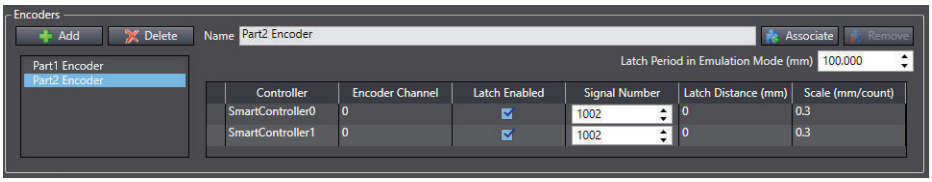
Item	Description
1	Place Belt
2	Pick Belt
3	Sensor Signal 1003
4	Sensor Signal 1004
5	Latch Signal 1002
6	Latch Signal 1001
7	Belt Encoder Inputs
8	Camera 2
9	Camera 1

With the configuration shown in the figure above, Belt Encoder Latches are configured as shown below, assuming a rising edge for each part and part target detection.
With a different configuration the falling edge could be assumed, both latches are available.

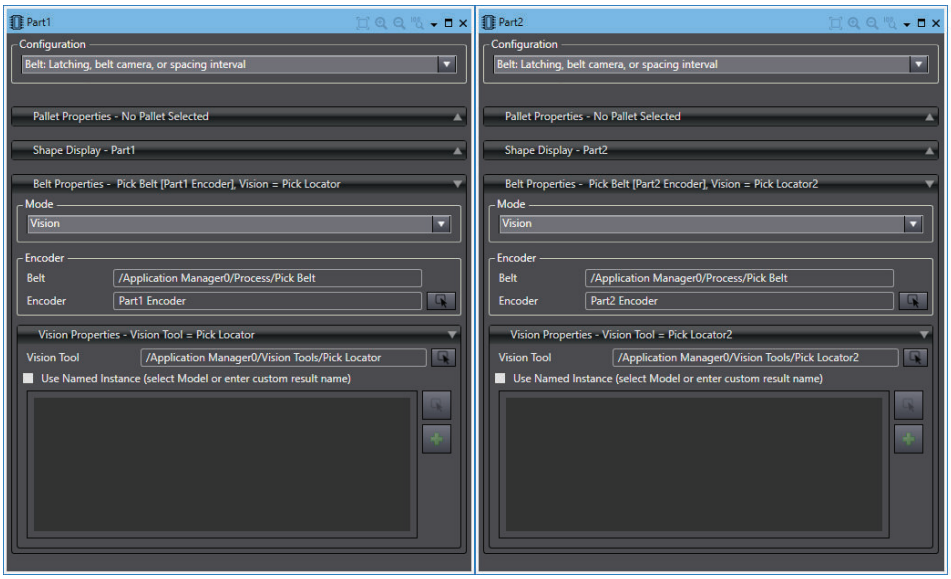


In the Belt object, a virtual encoder is needed for each part object (Part1 and Part2). Each virtual encoder will have an encoder association with the corresponding Belt object Encoder Channel of each controller, and the latch signal of the corresponding camera detecting that part object type as shown below in the figures below.





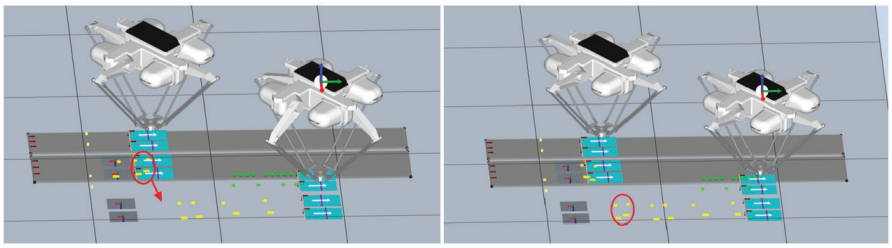
Part1 and Part2 configurations include references to their respective virtual encoders and vision tool for location as shown in the figure below.



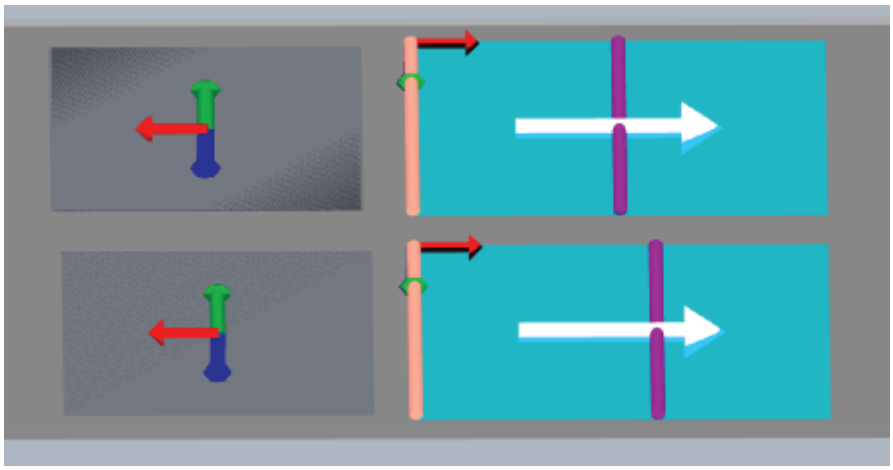
The separate virtual encoders are necessary to support independent tracking structures for the different part and part target types, including individual belt calibrations and allocation limits.
The separate encoder associations and latch signals allow Pack Manager applications to manage capturing and storing latch signals and encoder positions of every instance for each controller.
When an instance is not processed by the upstream robot and is reallocated to the downstream robot, the latch position reference is automatically changed from the SmartController0 encoder latch position to the SmartController1 encoder latch position.

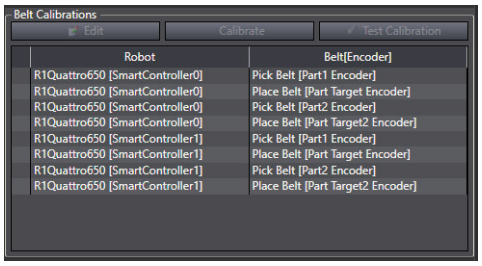
Additional Information

When an instance is allocated to the next controller and the latch reference changes, an instance can jump in the 3D Visualizer. This is often due to poor sensor calibrations or incorrect robot locations, as shown in the figure below. This can cause confusion when instances jump farther upstream when deallocated from the upstream robot. It is highly recommended to keep the 3D Visualizer orderly and represent the hardware system as closely as possible. This prevents many issues during commissioning and makes troubleshooting easier if issues arise.

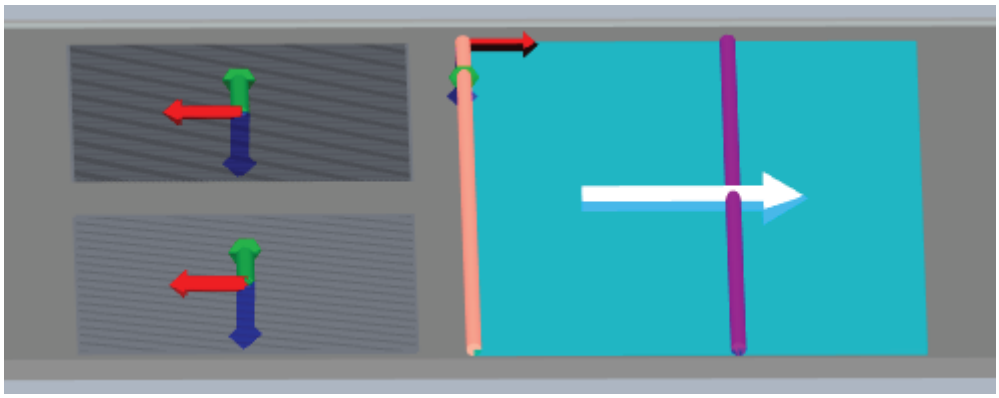


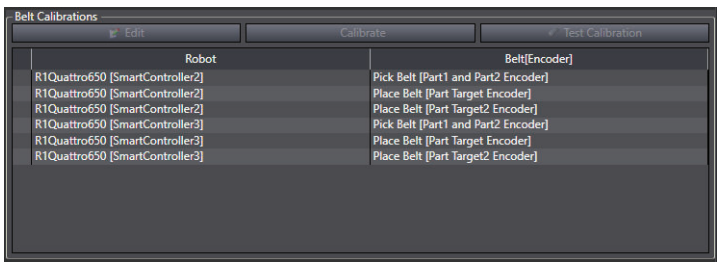
In this configuration, each camera is relative to independent belt calibrations and allocation limits as shown in the figures below. Notice that there is a Pick Belt calibration for each virtual encoder.





Alternatively, the ACE software supports configurations that require multiple cameras to be relative to a single robot-to-belt calibration and set of allocation limits as shown in the figures below, where Part1 and Part2 are configured to use a single virtual encoder. This is achieved by defining multiple latch signals for specific encoder associations and configuring the corresponding robot-to-sensor belt camera calibrations to use specific latch signal numbers. This is only valid for belt cameras. For example, multiple latch signals are not allowed if the configuration is for belt-latch configuration, such as locating multiple pallets with different latch signals. For latch and spacing calibrations, use multiple virtual encoders.

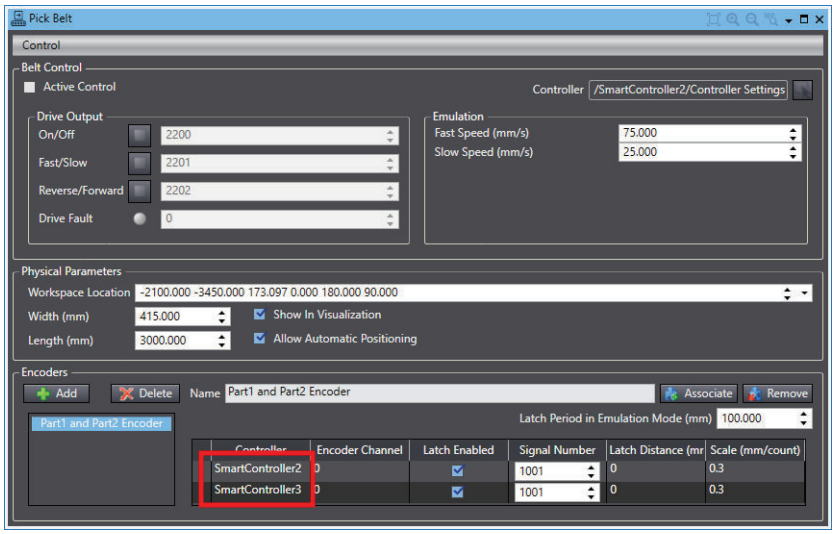




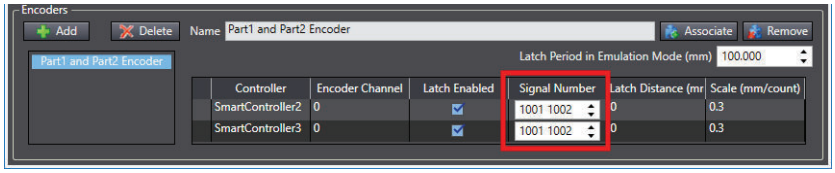
● Multiple Cameras Relative to a Single Robot-to-belt Calibration

Use the following procedure to configure multiple cameras relative to a single robot-to-belt calibration.

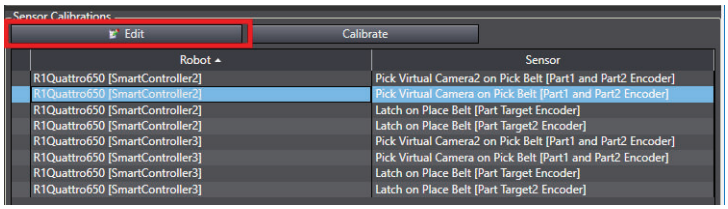
1 Open the Belt object and verify that a controller is associated with the belt encoder.



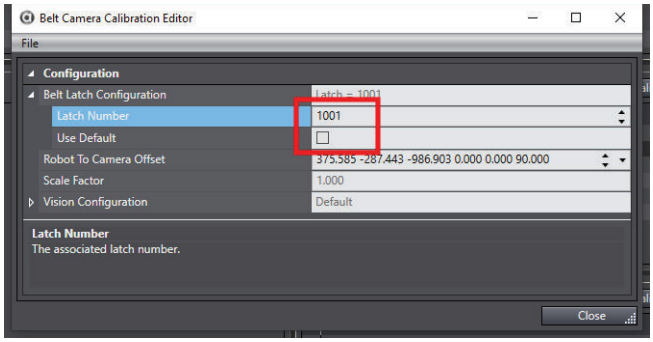
2 Specify the latch signals used by each camera in the Signal Number field. Multiple signals must be separated by a space, as shown in the following figure.



3 Locate the Robot-to-Sensor belt camera calibrations in the Process Manager. Select a Sensor Calibration and click the **Edit** Button.



4 Expand the Belt Latch Configuration parameter group, uncheck **Use Default,** and then enter the desired latch signal number associated with that camera.



5 Repeat steps 3 and 4 for the other belt camera calibrations and corresponding latch signals. Once all belt camera calibrations and latch signals are accounted for, the procedure is complete.

8-7-5 Process Manager Object

The Process Manager is the central control point for developing a process-managed application, such as an Pack Manager packaging application. The Process Manager is used create and configure complex applications without writing any programming code. For advanced development, it provides access to V+ programming controls, enabling applications to be modified or customized.
When Emulation Mode is enabled, robot-to-hardware calibrations and referencing procedures use 3D Visualization. Refer to *Emulation Mode Differences* on page 1-9 for more information.

Process Components

This section describes the Process components, which are accessed from the Process Manager object. The other application components, such as 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.

● Belts and Belt Encoders

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 electrically connected to. 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 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 Target as described below.

Part / Part Target Configuration	Vision Properties	Belt Properties	Pallet
Belt	Optional	Required	Optional
Static	Not Used	Not Used	Optional
Vision	Required	Not Used	Optional

● Part and Part Target Configuration Options

The Part and Part Target configuration defines the part pick or place requirements. The following options are available.

Part and Part Target Configuration	Description	Details
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.	A belt and encoder must be specified for use with the Part / Part Target. Then, a Belt Mode is defined that describes how the part is related to the belt. For this item, additional information is required based on the options below. **Belt Mode: Vision** • Vision properties are required. • Spacing is not used. **Belt Mode: Latch** The latch information is taken from the Belt object linked to the Belt Properties. • Vision properties are not used. • Spacing is not used. **Belt Mode: Spacing** • Vision properties are not used. • Spacing is required.
Static	The part is picked from or placed to 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.	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 vision 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.

● Part Process

A Part Process identifies a robot that can pick a Part or collection of Parts and place it at a Part Target or collection of Part Targets. 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. 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.

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.
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.
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.

● Pallets

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.

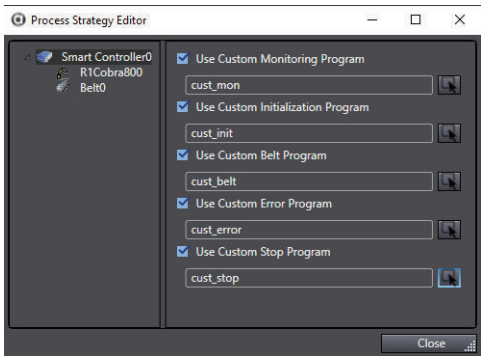
● Process Manager Customization

The process manager architecture was designed with extensibility and customization in mind. Many aspects of the product can be changed to meet custom application needs. When some aspects of a process manager application are customized, the system will create a copy of the default behavior as a starting point for the changes.
Customizations can be defined at various places in the process manager user interface. This section details provides an overview of the possible customizations and identifies where these are located in the user interface.
The primary process manager customizations are:

- Process Strategy Customization
- Controller Customizations, refer to See *Controller Customization* on page 8-170
- Robot Customizations
- Belt Monitoring
- Motions Settings: Pick and Place and Refinement

● Process Strategy 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.



Item	Description
Use Custom Monitoring Program	The default process monitoring has the following functions. • Checks for updates to process strategies. • Handles belt monitoring. • Monitors parts and part targets. The default V+ monitoring program can be copied for editing, or select an existing program.
Use Custom Initialization Program	The default initialization program that executes before the control programs (robot, belt, process strategy, etc.) are started. This can be used to initialize system switches, parameters, and variables or execute V+ programs that need to be started in parallel to the Process Manager. The default V+ initialization program can be copied for editing, or select an existing program.
Use Custom Belt Program	The default belt program monitors the speed / ON / OFF status of all belts. The default V+ belt program can be copied for editing, or select an existing program. Most applications do not require a Custom Belt monitoring program and can be sufficiently controlled using Process Strategy Belt Control Parameters as described in this section.
Use Custom Error Program	The default error program handles the processing and reporting of errors during the execution of a process. The default V+ error program can be copied for editing, or select an existing program. This program can be used to automate error handling that are reported to the PC by default. All Process Manager V+ program error handling will lead to this program. Use this program to automate error handling of errors that are reported to the PC by default. This program will check if any user-defined error responses exist in the Process Strategy - Robot parameters.
Use Custom Stop Program	The custom stop program can be used to perform certain operations after the application has stopped. The default V+ stop program can be copied for editing, or select an existing program.

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. Some examples are listed below.

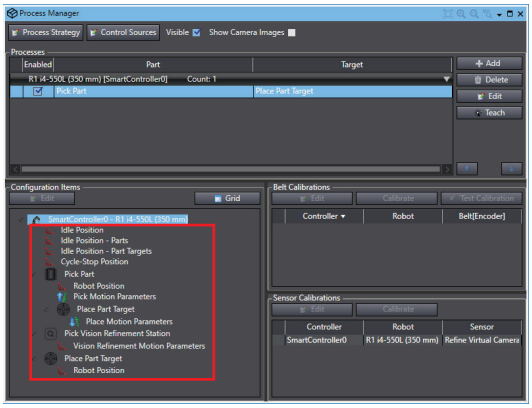
- Each robot will need an idle position.
- For each part that is picked, motion parameters describing the approach height, depart height, motion configuration, and offset at the pick location must be defined.

- 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.

● Motion Parameters for Pick and Place

When a pick and place process is defined, the software will create motion parameters that are used when processing the parts and targets. These parameters can be adjusted within the Process Manager's *Configuration Items* area if necessary.



Refer to *Configuration Items* on page 8-141 for more information.

Sources

A Source is an object that interacts with the hardware and discovers Part Instances and Part Target Instances. The Process Manager analyzes the configuration of Part Processes in order to determine what Sources are needed to control the hardware. A Source is allocated for each of the following 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 is used 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.
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 run-time 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 / 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 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 functionality 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 can be used in the allocation of Part Instances.

Line Balancing

A Process Strategy is invoked to determine how to allocate the Part / 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.

● Line Balancing and 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 function 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, as described below.

- **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

If required, the system enables the use of C# to define Process Strategies within the application.

● Controller Software

The application is split into the following two sections.

- A series of V+ programs that are responsible for picking and placing an instance.
- A series of V+ programs responsible for managing the queue of parts and communicating with the PC.

● Controller Software

The application is split into the following two sections.

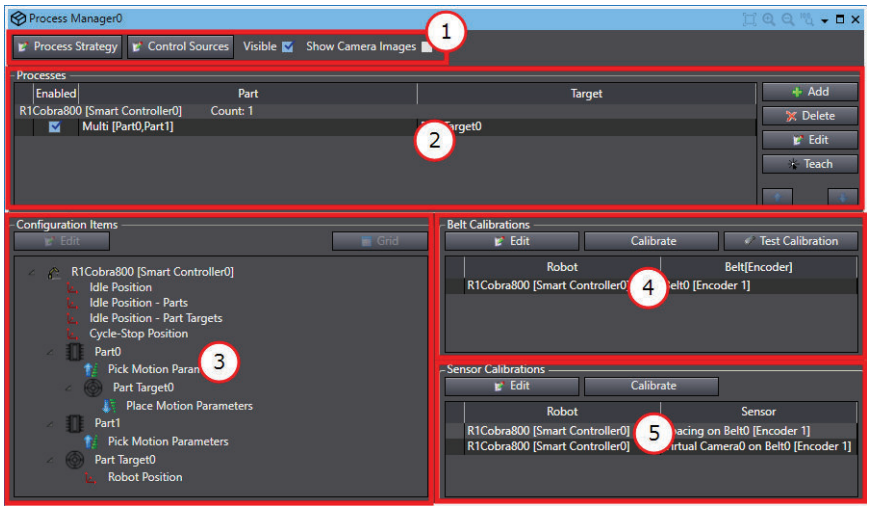
1. A series of V+ programs that are responsible for picking and placing an instance.
2. A series of V+ programs responsible for managing the queue of parts and communicating with the PC.

● Robot Control Code

The V+ program 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 Manager Editor

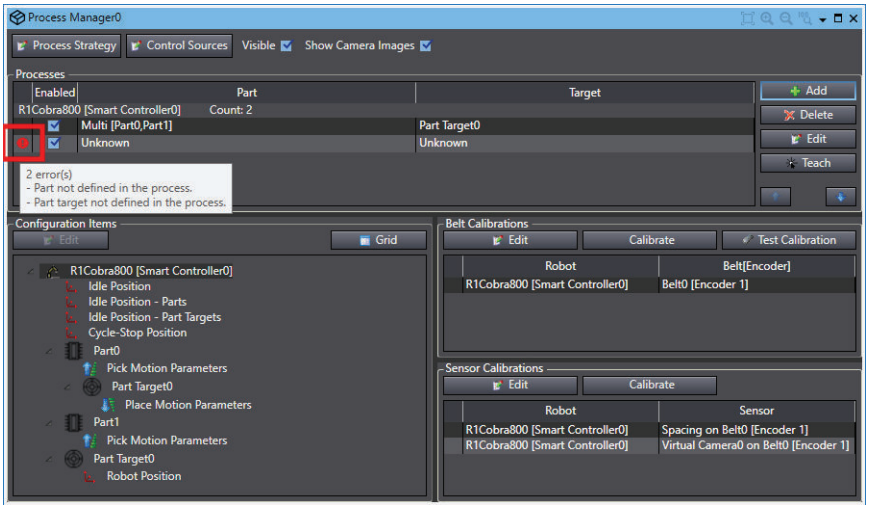
Use the following information to understand the Process Manager Editor area.



Item	Name	Description
1	Editor Functions	**Process Strategy** Button (Process Strategy) Displays the Process Strategy editor, which defines the pick / place strategy for the Process Manager. Refer to *Process Strategy* on page 8-151 for more information.
		Control Sources Button (Control Sources) Displays the Control Sources editor, which defines the properties for part and target hardware sources, as required by the selected workcell configuration. Refer to *Control Sources* on page 8-148 for more information.
		Visible Check Box Display or hide all Process Manager calibrations and instances in the 3D Visualizer.
		Show Camera Images Check Box Display or hide the camera image in the camera calibration / field of view shown in the 3D Visualizer.
2	Processes List	Displays a list of defined processes. Refer to Processes in the *8-7-5 Process Manager Object* on page 8-117 for more information.
3	Configuration Items Editors	Provides access to editors for all motion parameters, offsets, and static frames that must be provided for the processes defined in the Process List. Refer to *Configuration Items* on page 8-141 for more information.
4	Belt Calibrations Editor	Lists all Robot-to-Belt calibrations required for the defined processes. Also provides access to edit belt allocation limits. Refer to *Belt Calibrations* on page 8-133 for more information.
5	Sensor Calibrations Editor	Lists all Robot-to-Camera, Robot-to-Sensor, and Spacing calibrations required for the defined processes. Refer to *Sensor Calibrations* on page 8-138 for more information.

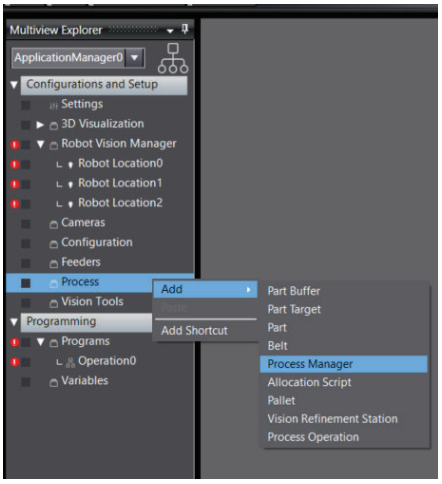
Process Manager Configuration Errors

If there is a configuration error, an alert icon ( ) displays in the corresponding item. Hover the cursor over the icon to display a message that describes the error as shown below.



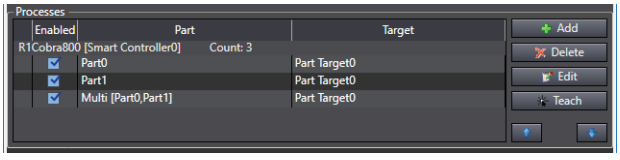
Adding a Process Manager Object

To add a Process Manager object, right-click **Process**, select **Add**, and then click **Process Manager**. A new Process Manager object will be added to the Multiview Explorer.



Processes in the Process Manager Object

This section describes the Processes Editor section of the Process Manager Object. This area defines the elements (robot, pick location, place location) for a particular process that will be controlled by the Process Manager.

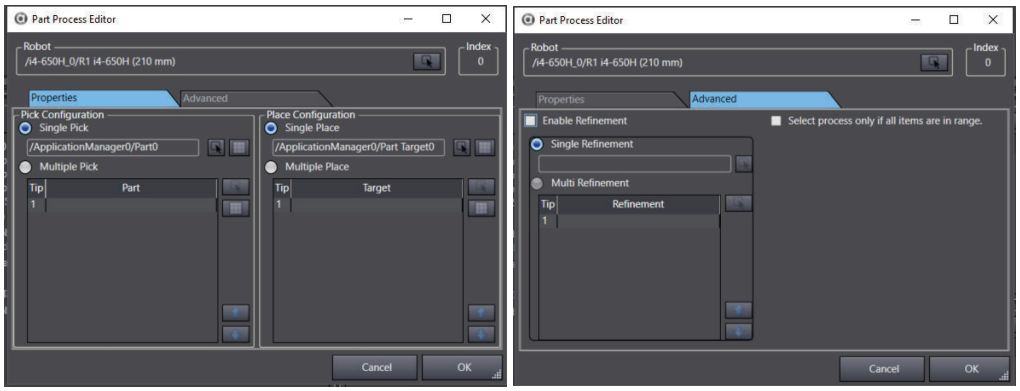


Item	Description
Up/Down Buttons ( )	Sets the priority for the selected process. The process at the top of the list receives the highest priority. Refer to *Changing Process Priority* on page 8-130 for more information
Enabled Check Box	If checked, the process is enabled for use.
Part	The Part(s) specified for the process. Double-click this item or select the process and then click the **Edit** button to change it.
Target	The Part Target(s) specified for the process. Double-click this item or select the process and then click the **Edit** Button to change it.
Add Button (Add)	Add a new process to the Processes list.
Delete Button (Delete)	Remove the selected process from the Processes list.
Edit Button(Edit)	Edit the selected process. Refer to *Part Process Editor* on page 8-126 for more information.
Teach Button (Teach)	Teach the selected process. Refer to *Teaching a Process* on page 8-131 for more information.
Alert Icon ( )	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 for a description of the reason(s) for the alert.

Part Process Editor

The Part Process Editor is used to specify the items used in the process. Access the Part Process Editor by double-clicking an existing process, or by clicking the **Add** Button for a new process.

Use the **Pallet Slot Selection** Button ( ) to select the pallet slots the robot can access for pick or place configurations. This button is only available when a Pallet object is configured for the selected part / target.



● Robot Reference

The Robot reference is used to specify the robot that will be used for the pick-and-place process. Click the **Select** Button ( ) to open the Select a Reference Dialog Box, which is used to create or specify the robot that will be used in the pick-and-place process.

● Index

This item displays the index number of the process, which can be referenced in V+ programs and C# programs.

● Properties Tab: Pick Configuration Group

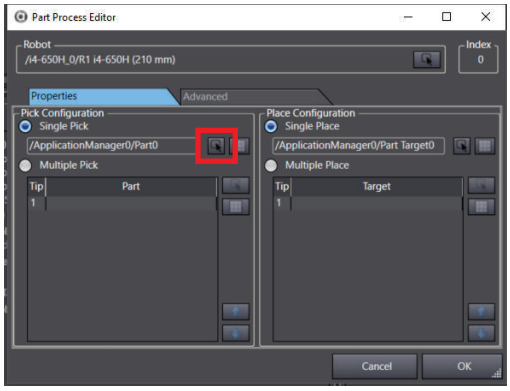
The Pick Configuration group is used to specify the single or multi-pick Part items for the pick-and-place process, as follows.

Single Pick/Place and Multiple Pick/Place can be used together. For example, to pick multiple parts individually and then place them all at the same time, use Multiple Pick and Single Place.

■ Single Pick

Select this item for a single-pick application where only one pick motion is performed. Use the **Select** Icon to browse for the part that will be picked.

When Single Pick / Place configuration is used with an I/O EndEffector that has multiple tips defined, the center line Tip Offset is applied. This is the average of all Tip Offsets defined in the I/O EndEffector object. Refer to *7-5-2 I/O EndEffectors* on page 7-43 for more information.

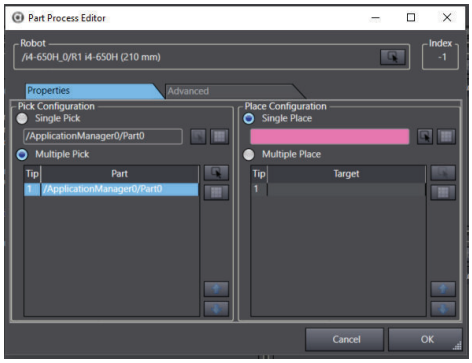


■ Multiple Pick

Select this item for a multiple-pick application where multiple parts will be picked before placement at a target. When multi-pick is enabled, the available tip indexes will be provided for the selected robot I/O EndEffector. Use the **Select** Icon to browse for the part that will be picked by each tip.

Use the **Up/Down** Buttons to change the order of the tip processing.

Multiple Tips must be defined in the robot's I/O EndEffector configuration. The Process Manager will access the I/O EndEffector object specified in the *Selected End-Effector* property of the robot object.

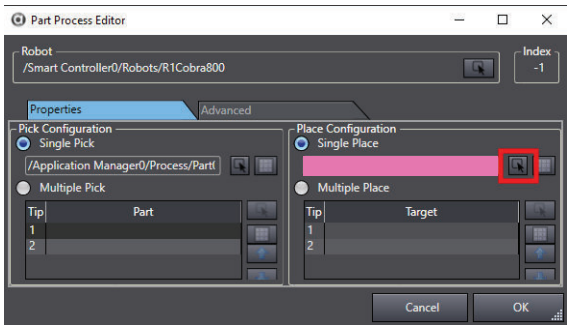


● Properties Tab Place Configuration Group

The Place Configuration group is used to specify the single or multi-place Part Target items for the pick-and-place process as follows.

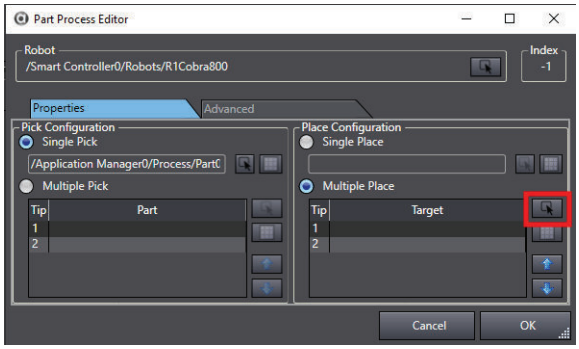
■ Single Place

Select this item for a single-place application where only one place-motion is performed. Use the **Select** Icon to browse for the part target where the part will be placed.



■ Multiple Place

Select this item for a multiple-place application where multiple place-motions are performed.
Use the **Select** Icon to browse for the part that will be picked.
Use the **Up/Down** Buttons to change the order of the tip processing.
Multiple Tips must be defined in the robot's IO EndEffector configuration. The Process Manager will access the IO EndEffector object specified in the *Selected End-Effector* property of the robot object.



● Advanced Tab: Enable Refinement

The Enable Refinement option is used to perform a position refinement operation over an upward-facing camera to improve part placement accuracy. This is used when placement error needs to be smaller than error introduced during the pick operation.
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. Refer to *8-7-8 Vision Refinement Station Object* on page 8-167 for more information.

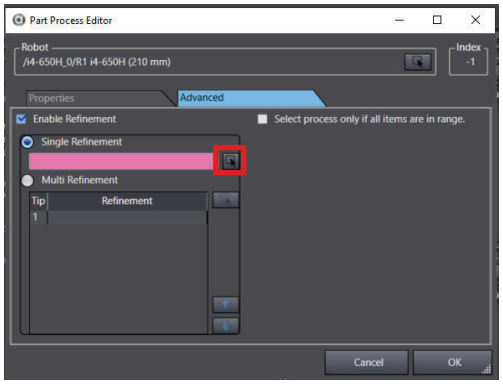
■ Single Refinement

Select this item for a single-vision-refinement application under the following conditions.

- Only one part at a time is being moved by the robot.

- 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 **Select** Icon to browse for the vision refinement station.

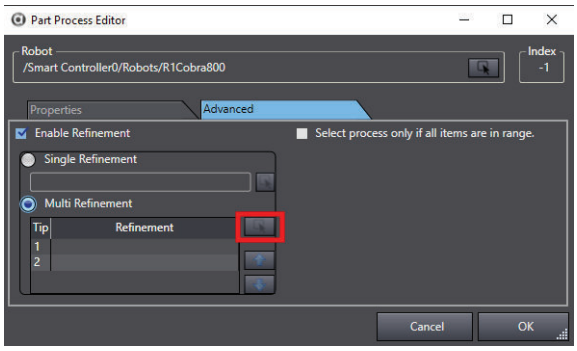


■ Multi Refinement

Select this item for a multiple-tip-gripper applications. Each tip will be moved individually to the specified vision refinement station.

Use the **Select** Icon to specify the tool tip and corresponding vision refinement.

Use the **Up/Down** Buttons to change the order of the tip processing.

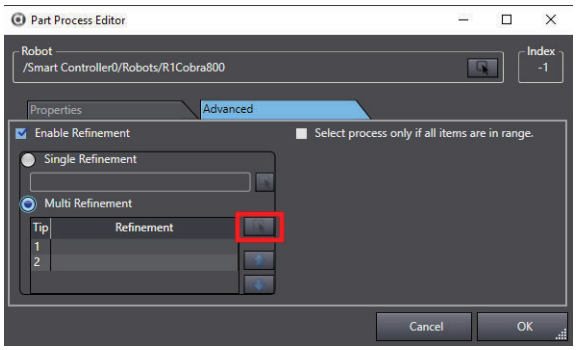


■ Select Process Only if All Items are in Range

The **Select process only if items are in range** Check Box tells the system to only select this process if all parts and targets are in range of the robot. This option is typically disabled, but may be useful in multi-process applications. Select this Process only if the required Part(s) and Part Target(s) are within range.

This may not be sufficient in parallel-flow configurations with many Parts and Part Targets. In this case, consider part and target filtering to reduce the number of instances the robot has to process through when checking position of all instances.

This is typically not necessary, but can be helpful in multi-process configurations. For example, in a situation where Process A has higher priority than Process B, Process A Part is further up-stream relative to Process B Part, but Process B Target is further downstream than Process A Target, the system can be optimized by selecting Process B when both part and target are already within range rather than selecting process A by priority and waiting for Part A to come into range. During this waiting time, Process B Target may move downstream and out of range.

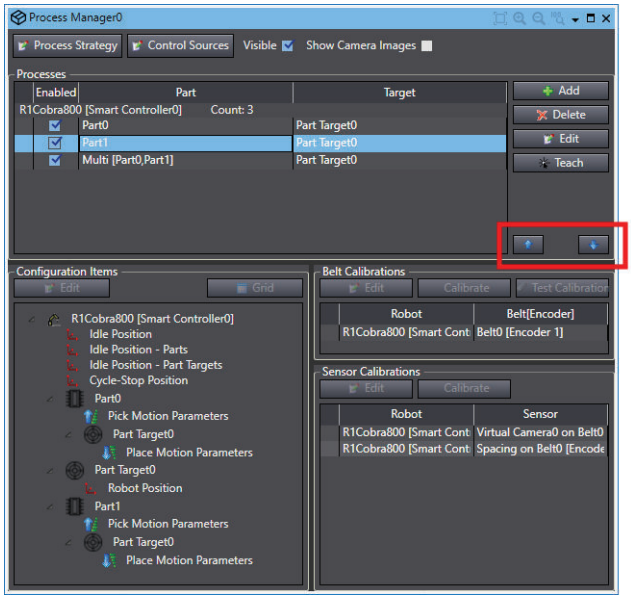


Changing Process Priority

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 list receives highest priority.

Change the priority for a process by using the **Up / Down** Buttons on the Process list editor, shown in the following figure.

In addition to the arrows, the process priority are changed through the Process Selection Mode setting of the Robot Parameters in the Process Strategy Editor area. Refer to *Process Strategy Robot Parameters* on page 8-152 for more information.



For example, there are 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 the first process 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.

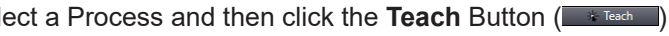
However, changing the order (the priority), gets a different behavior as follows.

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 Process is the teach function. This must be performed after all calibrations are complete. Ideally, all calibrations would be performed as accurately as possible with a calibration pointer and disk. After calibrations are complete, the process end effector can be installed to teach the process. Every step of the process is performed one at a time to capture any necessary motion offsets. Select a Process and then click the **Teach** Button (

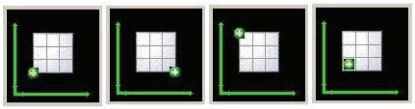
) to open the Teaching Process wizard.

The following process illustrates the steps for a basic pick-and-place application. The steps in the wizard will vary depending on the application and types of Parts and Part Targets that are defined in the application.



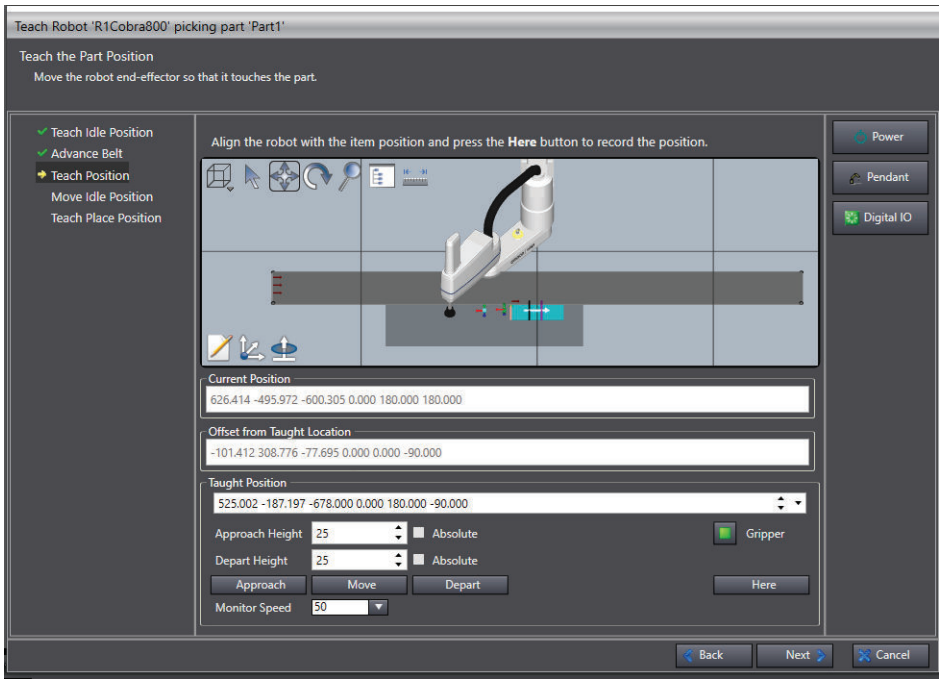
Additional Information

If a Pallet object is used for a Part or Part Target, additional screens will open 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. Refer to *8-7-7 Pallet Object* on page 8-161 for more information.



Use the Teaching a Process wizard to make selections for pick, place, and idle robot positions.

When the **Absolute** Option is selected, the approach or depart heights will be an absolute Z coordinate in robot world coordinates. Ensure that the value is in-range and safe. Otherwise, the robot could move to an unintended location or out of range, which may damage the robot or other equipment in the workcell.



Using Multiple Process Manager Objects

The ACE project can share robot-to-hardware calibration information between multiple Process Manager objects. If the same robot and hardware are used for each process, create multiple Process Manager objects in the ACE project without having to repeat the calibrations.
For example, assume setting up an ACE project to handle the packaging of various fruits. There are three fruits 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, use the following procedure.

1. Create a Process Manager object for apples named Apple Packing.
2. Add the Belt, Sensor, Part, and Part Target objects.
3. Perform the Robot-to-Belt calibration.
4. Perform the Robot-to-Sensor calibration.
5. Optionally, edit the Process Strategy as needed.
6. Optionally, edit the Control Sources as needed.
7. Teach the Process.
8. Optionally, edit the Configuration Items, as needed.
9. Add a second Process Manager object for oranges named Orange Packing.

10 Add a new Part and/or Part Target if the pick or place requirements are different than those for apples. If the Part and/or Part Target use the same robot, sensor, and belt objects that were used with the Apple Packing process, there is no need to repeat the calibrations.

11 Optionally, edit the Process Strategy and Control Sources as needed.

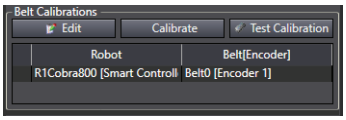
12 If a new Part and/or Part Target was added for Orange Packing, teach the Process.

13 Optionally, edit the Configuration Items as needed.

14 Repeat steps 9 to 13 for Peaches Packing. After all Process Manager objects are added, the procedure is complete.

Belt Calibrations

This section describes the Belt Calibrations list in the Process Manager. The Belt Calibrations area defines the robot-to-belt calibrations required by the defined Processes in the Process list. Refer to *Processes in the Process Manager Object* on page 8-125 for more information.



Item	Description
Edit Button (Edit)	Click to edit the selected belt calibration. Refer to *Editing the Belt Calibration Parameters* on page 8-134 for more information.
Calibrate Button (Calibrate)	Click to begin the belt calibration procedure. Refer to *Belt Calibration Wizard* on page 8-136 for more information.
Test Calibration Button (Calibrate)	Click to test the current belt calibration. This button is not available until the belt has been calibrated. Refer to *Testing the Belt Calibration* on page 8-136 for more information.
Robot	Specifies the robot for the belt calibration. Double-click this item or click the **Edit** Button to display the Belt Calibration Editor. Refer to *Editing the Belt Calibration Parameters* on page 8-134 for more information.
Belt [Encoder]	Specifies the belt and encoder for the belt calibration. Double-click this item or click the **Edit** Button to display the Belt Calibration Editor. Refer to *Editing the Belt Calibration Parameters* on page 8-134 for more information.

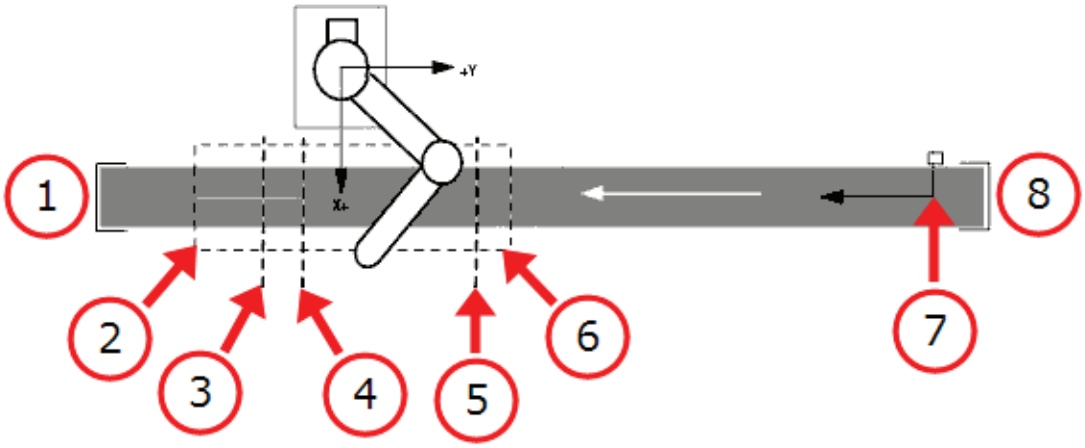
● Creating a Belt Calibration

When a belt calibration is required, the Process Manager displays the Belt object name with an **Alert** Icon in the Belt Calibrations list. The belt is calibrated using the Belt Calibration wizard, which is accessed from the **Calibrate** Button in the Belt Calibrations area. After the belt is calibrated

using the wizard, values such as allocation limits, upstream limits, dynamic wait line, process limit, and horizontal filtering limits can be manually edited.

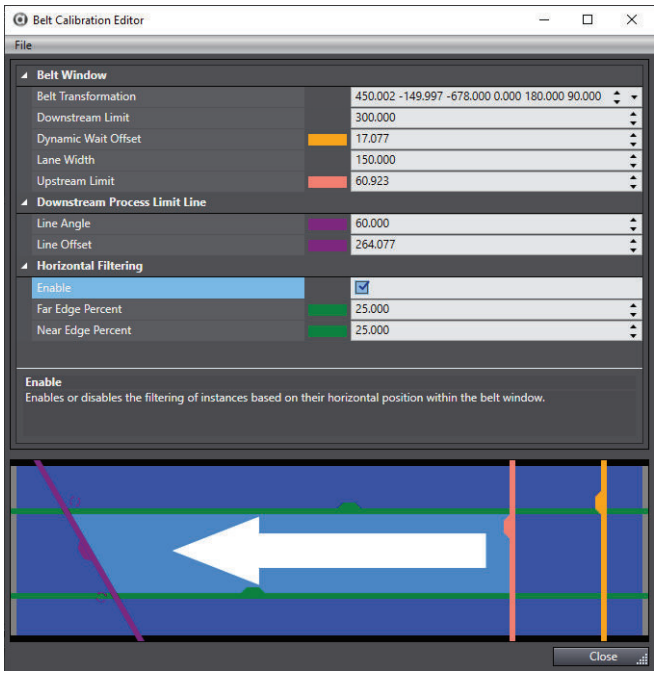
● Editing the Belt Calibration Parameters

After the belt has been calibrated using the Belt Calibration wizard, the stored belt calibration parameters and allocation limits can be manually edited, such as upstream limit, downstream limit, and downstream process limit. These parameters are edited using the Belt Calibration Editor. The following figure illustrates several of the Belt Calibration Editor items in a typical workcell.
The belt must be calibrated using the wizard before the values can be manually edited.



Item	Description
1	Downstream
2	Downstream Limit
3	Process Limit
4	Belt Stop Line
5	Dynamic Wait Line
6	Upstream Limit
7	Object Sensor Origin
8	Upstream

To access the Belt Calibration editor, click the **Edit** button in the Belt Calibrations group. The Belt Calibration editor will open.
Adjust the parameter values or use the graphical representation to reposition the lines accordingly.



Group	Item	Description
Belt Window	Belt Transformation	The transformation describing the location of the belt window relative to the origin of the robot. This location also defines the upstream tracking limit of the belt window. The X-axis of the belt transformation is often referred to as the belt vector.
	Downstream Limit	Downstream window limit (mm from the belt frame origin along the belt vector). The robot will not track beyond this limit.
	Dynamic Wait Offset	Distance along belt vector (mm from belt transformation origin) where robot will wait for a part or target that is currently upstream of the upstream limit. Once the part or target reaches the Upstream Limit, the robot will approach the part. This may be upstream or downstream of the Upstream Limit depending on the motion parameters and path of travel to a target.
	Lane Width	The width of the belt window starting from the belt transform, pointing in the positive Y-direction.
	Upstream Limit	Upstream pick limit (mm from belt transformation along the belt vector).

Group	Item	Description
Downstream Process Limit Line	Line Angle	The angle of the downstream process limit line.
	Line Offset	Downstream process limit. If a robot has initiated a move to an instance that is upstream or at this limit, it will proceed with that motion unless a belt window violation occurs at the downstream limit of the belt window. However, if an allocated instance crosses the downstream process limit line before a robot initiates a motion to it, then it will be unallocated from this robot queue. The value must be between the upstream limit and the downstream limit. A good initial value can be calculated by subtracting the distance the belt will travel during a pick or place operation from the Downstream limit (length of belt window), and then subtract an additional 15-25mm. If the process limit is any closer to the downstream limit, the process will likely experience belt window violations.
Horizontal Filtering	Enable	Enables or disables the filtering of instances based on their horizontal position within the belt window. When this is enabled, the pick area is limited to a subset of the width of the belt window. Different robots can be forced to pick in different horizontal regions (lanes) of the belt. For example, consider the conveyor belt as a three-lane highway (as shown in the previous figure), robot one can be filtered to pick from the near one-third of the belt window, robot two filtered to pick from the middle one-third of the belt window, and robot three filtered to pick from the far one-third of the belt window.
	Far Edge Percent	The distance from the far edge of the conveyor where the robot cannot process.
	Near Edge Percent	The distance from the near edge of the conveyor where the robot cannot process.

● Belt Calibration Wizard

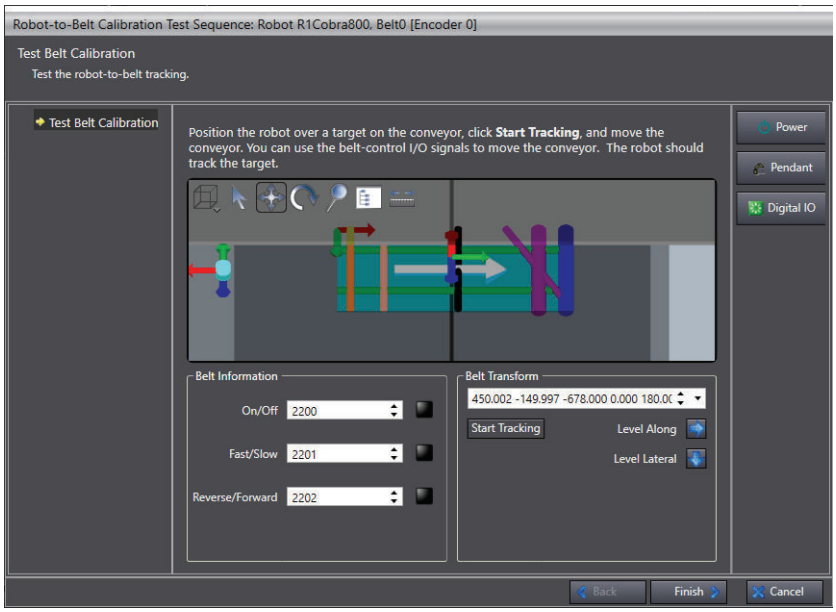
The Robot-to-Belt Calibration wizard provides an interactive series of steps that guide users through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and used to populate the fields in the Belt Calibration Editor.

Click the **Calibrate** Button to begin the Belt Calibration wizard.

This feature operates differently in Emulation Mode. Refer to *1-4-1 Emulation Mode* on page 1-9 for more information.

● Testing the Belt Calibration

The Test Belt Calibration page is used to test the current robot-to-belt calibration. Click the **Test Calibration** Button to begin the Belt Calibration test.



Use the following procedure to test the robot-to-belt calibration.

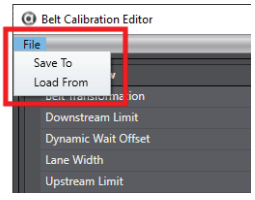
1. Make sure the belt is turned OFF so the belt is not moving.
2. Place a part on the belt.
3. Position the robot tool tip so that it is just above the center of the part.
4. On the Test Calibration page, click the **Start Tracking** Button.
5. Start the conveyor so the belt is moving. The robot should track the target location until it leaves the tracking region.
6. Confirm the correct tracking and click the **Stop Tracking** Button to stop the tracking.
7. Click the **Next** Button to proceed. The Robot-to-Belt Calibration wizard closes and the procedure is completed.

Additional Information

The distance between the robot tool tip and the part on the belt should remain constant while tracking. If not, the calibration procedure should be executed again.

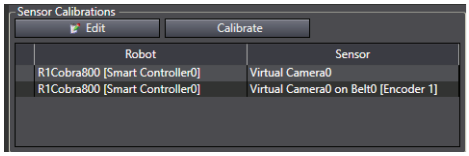
● Saving and Loading a Belt Calibration

After a calibration has been completed, the data can be saved by selecting **File** > **Save To** on the calibration editor menu. Load a previously-saved calibration file by clicking **File** and then **Load From** on the calibration editor menu.



Sensor Calibrations

The Sensor Calibrations area defines the robot-to-sensor calibrations for the selected workcell process. These should be performed after Robot-to-Belt calibrations. Refer to *Processes in the Process Manager Object* on page 8-125 for more information.



Item	Description
Robot	The robot specified for the belt calibration. Double-click this item or click the **Edit** Button to display the Sensor Calibration Editor. Refer to *Editing the Calibration Parameters* on page 8-138 for more information.
Sensor	The sensor specified for the process. Double-click this item or click the **Edit** button to display the Sensor Calibration Editor. A sensor can be any of the following, depending on the type of configuration selected for the Part or Part Target. • Belt camera. • Fixed camera (downward-looking camera). • Latch sensor. • Spacing reference. • Refinement camera (upward-looking camera). Refer to *Editing the Calibration Parameters* on page 8-138 for more information.
Edit Button (Edit)	Click to edit the selected belt calibration. Refer to *Editing the Calibration Parameters* on page 8-138 for more information.
Calibrate Button (Calibrate)	Click to teach the selected process. Refer to *Editing the Calibration Parameters* on page 8-138 for more information.

● 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.

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.

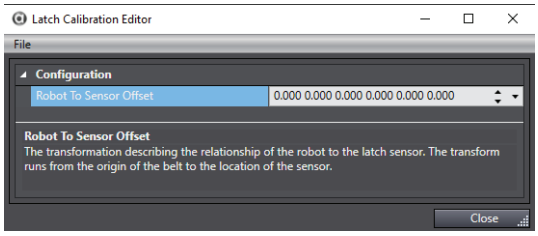
● Editing the Calibration Parameters

After the sensor is calibrated through the Sensor Calibration wizard, the stored sensor-calibration parameters, such as the robot-to-sensor offset, can be manually edited. These parameters are edited using the sensor Calibration Editor. To access the sensor Calibration Editor, select a sensor and then click the **Edit** Button in the Sensor Calibrations group. The Sensor Calibration Editor opens.

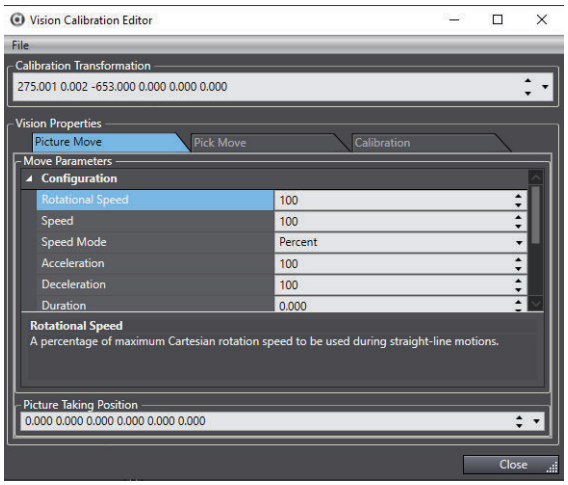
The sensor must be calibrated, or loaded from a previously-saved calibration data file, before the values can be manually edited.

The Sensor Calibration Editor contains the sensor properties configuration parameters. These are used to configure various settings of the selected sensor.

The following figure shows the Latch Calibration Editor, which has one property for controlling the calibration offset.



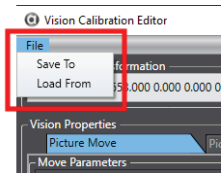
The following figure shows the Vision Calibration Editor, which contains a calibration offset along with additional parameters for controlling the robot motion during the picture-taking and part-pick operations of the automated hardware calibration. These are not used during run time when the robot is performing the process (run time motion parameters will be found in configuration items).



● Saving and Loading a Sensor Calibration

After a calibration has been completed, the data can be saved by selecting **File** and then **Save To** on the calibration editor menu.

Load a previously-saved calibration file by clicking **File** and then **Load From** on the calibration editor menu.



● Calibration Wizard - Automatic Versus Manual Calibrations

The calibrations can be performed using either the automatic calibration (preferred method) or the manual calibration procedure. In the automatic calibration procedure, the user teaches 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, the robot is moved 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.

It is recommended using the calibration wizard, in order to obtain the optimum performance from the system.

The manual calibration procedure is available for the Fixed Camera and Refinement Camera calibrations.
Some calibrations operate differently in Emulation Mode. Refer to *1-4-1 Emulation Mode* on page 1-9

● Using the Sensor Calibration Wizard

The Sensor Calibration wizard provides an interactive series of steps that guide a user 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.

● Calibration Types

The Sensor Calibrations area defines all of the calibration types that are used in the project. These calibration types are described below.

■ Spacing Reference Calibration

The Spacing Reference Calibration wizard configures the positioning of parts along a belt at defined intervals. This process is performed by setting a stationary point along the belt from which the instances will be generated. This point should be calibrated outside of the belt window to avoid difficulties in allocation. If creating spacing instances for multiple robots, the spacing calibrations must reference the same upstream position.

■ Refinement Camera Calibration

The Refinement Camera Calibration wizard is functionally similar to the Fixed Camera Calibration. Refinement Camera Calibrations require the robot to be able to pick up a part. Calibration pointers will not be helpful in this scenario.

■ Belt Latch Calibration

Belt 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 on a conveyor belt. This synchronizes the rest of the system with the position of the object.

■ Belt Camera Calibration

The Belt Camera Calibration wizard configures the positioning of a camera with respect to a robot when a belt object is present. It includes controls for moving the belt, indicators to show if the belt is ON, and fields for speed, position, and direction. Depending on the step in the calibration wizard, it also shows the 3D visualization of the vision window for the associated camera.
Refer to *Robot-to-Camera Calibration* on page 1-24 for a more detailed explanation of the calibration process.

■ Fixed Camera Calibration

The Fixed Camera Calibration wizard configures the positioning of a camera with respect to a robot when both the camera and the surface in the field of view are stationary. The wizard will show the 3D visualization of the camera vision window at various steps in the process.

Depending on the application, the wizard will end with manual or automatic configuration. Automatic calibration assumes that the pick surface is parallel to the tool plane. If the pick surface is not parallel to the tool plane, the parameters should be adjusted so manual calibration is performed instead.

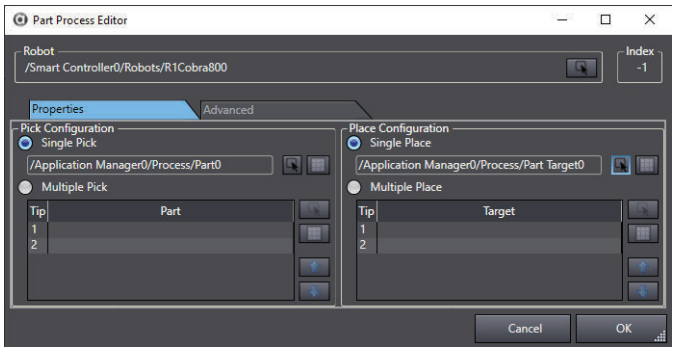
Refer to *Robot-to-Camera Calibration* on page 1-24 for a more detailed explanation of the calibration process.

Configuration Items

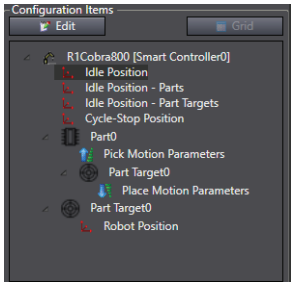
The Configuration Items area defines the workcell items and the relationships between those items that are associated with a particular workcell configuration. The Configuration Items area also allows quick access to the robot position (location) editors for a particular item, such as the idle, part 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 area. For example, a basic pick and place application would look like this in the Part Process editor:



The corresponding Configuration Items area has the following appearance.



● Configuration Item Area Structure and Features

The Configuration Items are arranged in a tree structure to show the relationships between the work cell items. The Configuration Items group contains the following features.

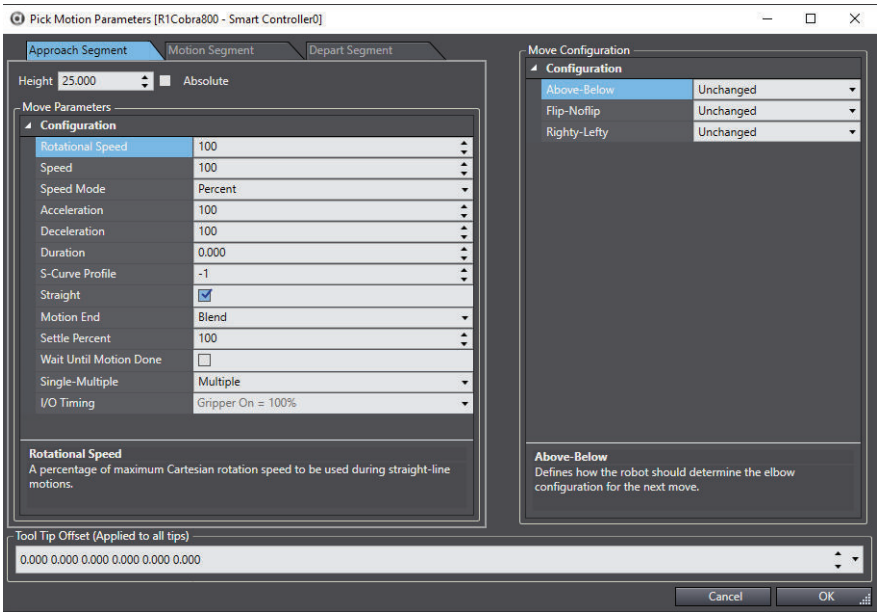
- Expand or collapse a tree branch by clicking the arrow icons next to an item name.
- Double-click any of the Position objects (idle or robot) or select the Position object and click the **Edit** Button to open the location editor that can be used to manually enter the object location.
- Click the **Grid** Button (Grid) and use the Motion Sequence Grid Editor to edit the motion parameters and offset locations (a robot must be selected in the list to enable the Grid button). Refer to *Motion Sequence Grid Editor* on page 8-147 for more information.

Location Editors

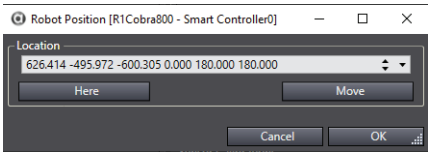
There are two types of location editors: a simple editor, to enter location information and an enhanced position editor, which contains additional sections such as Move Parameters, Move Configuration, Approach/Depart Parameters, etc.

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. Refer to *Enhanced Location Editor Parameters* on page 8-142 for more information.

The Location Editor title bar indicates the type of parameters being edited.



The Robot Position editor shown below is a simple position editor. This uses entered or taught location information for a static fixed position frame that does not require a robot-to-belt or robot-to-sensor calibration.



Enhanced Location Editor Parameters

Use the editor's parameter input fields to adjust the Move Parameters and Move Configuration for the approach, motion, and depart segments. Use the following examples of various enhanced location editor parameter grids to understand the editor functions.

- **Move Configuration Area**

  These parameters control the configuration of the robot at the selected location. For example, if a workcell contains a SCARA robot and it needs to be in a lefty configuration, set the Righty-Lefty parameter to Lefty.

- **Move Parameters for Approach/Motion/Depart Segments**

  These parameters control how the robot moves to and from the selected Part, Vision Refinement Station, or Part Target location. They allows the fine-tuning of the robot's motion. 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.

- **Absolute Check Box and Height**

The Height input field is used to enter a value for the approach and depart segments. The **Absolute** Check Box allows users to enter absolute height values for the Approach and/or Depart motion segments. Enter positive or negative values, as needed.
When the Absolute option is selected, 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.

- **I/O Timing Parameter**
  An I/O Timing parameter is included which controls the open/close timing of the gripper during each part of the motion segment. The I/O Timing (Gripper On) can use either a percent value, a distance value, or a time value as shown in the following figures.
  For example, setting the value to 25 mm, the gripper will activate at 25 mm from the pick position. Setting it to 25%, the gripper will activate at 25% of the distance from the approach start to the pick position. The time value allows the gripper timing (in milliseconds) to be set.

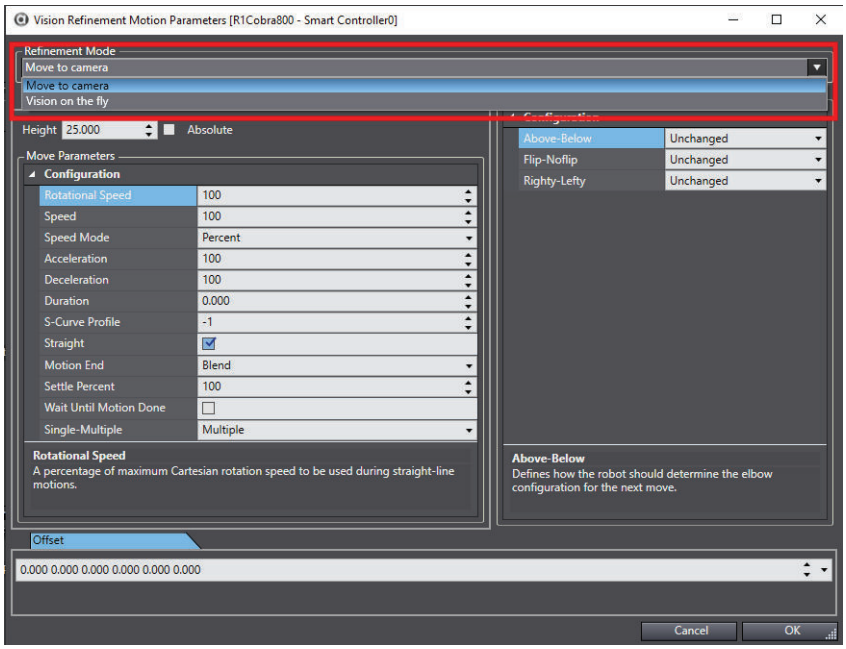
Additional Information

The distance value is useful when accessing pallets with multiple layers and an absolute approach height has been specified.

- **Use Custom Program Option (Motion Segment Tab)**
  Use **Use Custom Program** Check Box to specify a custom V+ program that controls the motion segment. To use this option, select the check box and then click the **Select** icon ( ) to display the Robot Motion Sequence Program wizard.

● Vision Refinement Motion Parameters

The Vision Refinement Motion Parameters specify 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.
  The **Offset** Tab allows gripper (tool) offset editing.
- **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.

The **Start Location** and **End Location** Tabs are used to edit the start and end points for the robot path through the camera's field of view. The **Trigger Percent** Tab allows editing where the picture request is triggered as a percentage of the robot motion, from Start Location to End Location.

The **Vision on the fly** mode can provide faster throughput, but may require more lighting and shorter Exposure Time compared to the **Move to camera** mode. A robot position latch should be used for Vision on the fly mode.

Idle Positions

Idle Positions are staging positions between picking and placing operations. They are initially defined by the teach process to be the same location, or can be manually taught to be different locations. If the Wait Mode is set to **Move to idle position**, then the following descriptions apply. If not, the Idle Position - Parts and Idle Position - Part Targets locations are not used.



Additional Information

The Wait Mode setting can be found in the Process Strategy area. Refer to *Process Strategy* on page 8-151 for more information.

● Idle Position

This is the location the robot uses when no Process is selected or when the Process Manager is aborted or stopped.

● Idle Position - Parts

This is the location the robot uses when it is waiting to pick a part, typically when no parts are available.

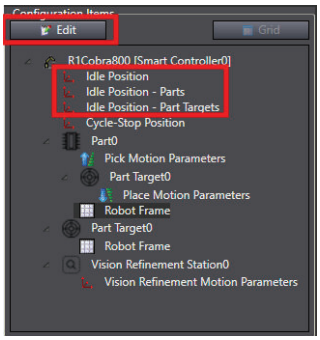
The Idle Position - Parts location is not associated with a specific process. If there are multiple Part sources in different areas of the work envelope, consider setting the Idle Position - Parts location in an area between the Part source locations and not near one specific Part source.

● Idle Position - Parts

This is the location the robot uses when it has picked a part and is waiting for a target to become available.

The Idle Position - Part Targets location is not associated with a specific process. If there are multiple Part Target sources in different areas of the work envelope, consider setting the Idle Position - Part Targets location in an area between the Part Target source locations and not near one specific Part Target source.

To access the Idle Position editor, double-click the **Idle Position** Configuration Item or select the item and click the **Edit** Button. The Idle Position editor will open.



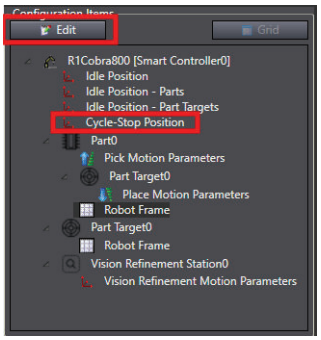
Cycle-Stop Position

The Cycle-Stop Position is a location that is used when the process Cycle-Stop is requested. The Cycle-Stop can be requested with one of the following methods.

- OPC
- Data Mapper
- C# program
- Clicking the **Cycle Stop** Button in the Task Status Control area

Using this method will result in the robot finishing the current process and then waiting for the Cycle-Stop request to be released.

To access the Cycle-Stop Position editor, double-click the **Cycle-Stop** Configuration Item or select the item and click the **Edit** Button. The Cycle-Stop Position editor will open.



Robot Frames

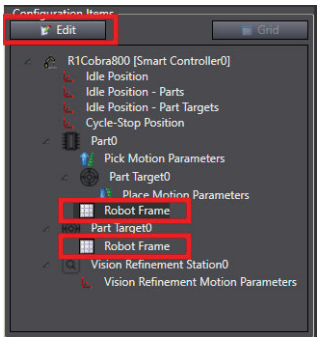
Robot frames (also known as reference frames) are useful because they allow users to teach locations relative to the frame. If the location of the frame changes in the workcell, simply update the frame information to reflect its new location. It is then possible to use any locations created relative to that frame without further modifications.

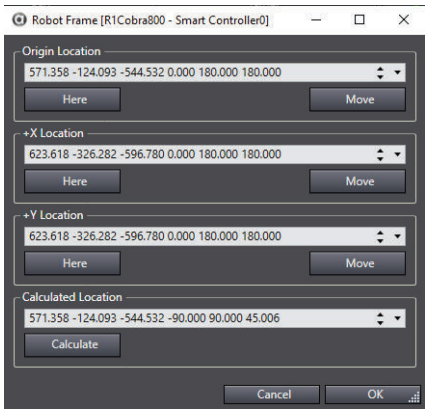
A process pallet is typically taught relative to a reference frame. This avoids the problem of teaching many individual pallet positions and then having to reteach all of those positions if the pallet moves for some reason. Instead, the pallet is taught relative to a frame. If the pallet moves in the 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. To access the Robot Frame editor, double-click the **Robot Frame** Configuration Item or select the item and click the **Edit** button.

Additional Information

Robot Frames are only available when a part or target object is configured as **Static: Fixed Position** and a Pallet object is defined in that object.





● Manually Define Robot Frames

Robot Frames will typically be defined during the teach process wizard, but can be manually defined as well. Use the following procedure to manually define a Robot Frame.

1 Teach the Origin Location.
Use the V+ 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) and then click the **Here** Button to record the position.

2 Teach the +X Location.
Use the V+ 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. Optimum results will be obtained by using a point as far away from the origin as possible. Then, click the **Here** Button to record the position.

3 Teach the +Y Location.
Use the V+ 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. Optimum results will be obtained by using a point as far away from the origin as possible. Then, click the **Here** Button to record the position.

4 Click the **Calculate** Button to calculate the position of the robot frame relative to the robot.

5 Click the **OK** Button to close the Robot Frame editor and complete this procedure.

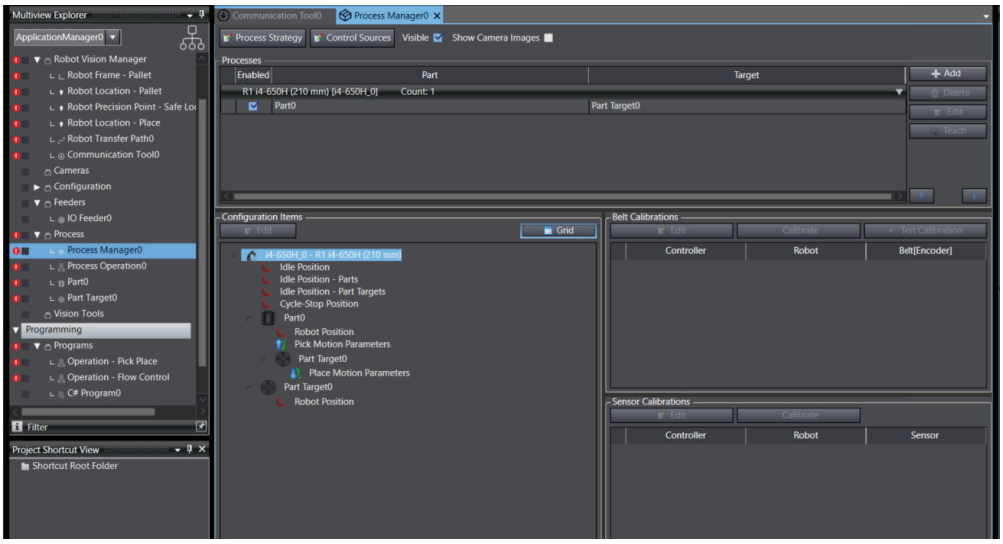
Motion Sequence Grid Editor

The Motion Sequence Grid Editor provides a grid / table interface that allows access and individual editing of the motion parameters used to optimize cycle time. Also, the multiple speed / acceleration / deceleration parameters can be changed at the same time. Click on one of the necessary fields and drag the cursor across the others. Enter a new value to change all of the selected fields.

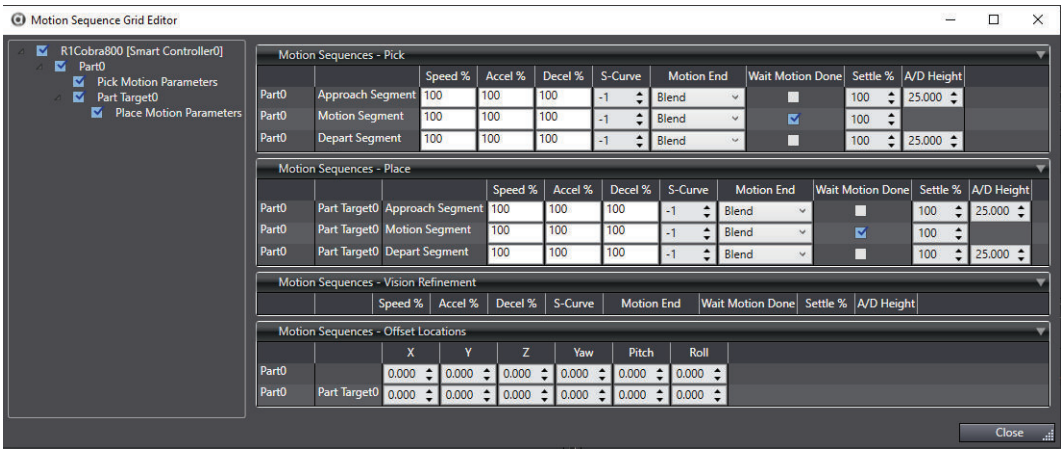
- Pick motion parameters.
- Place motion parameters.
- Vision refinement parameters.
- Offset location parameters.

If parameters change significantly, I/O timing in the Pick and Place Motion Parameter editors may need to be adjusted. Otherwise, faster motions and blending may lead to missed picks or poor placement.

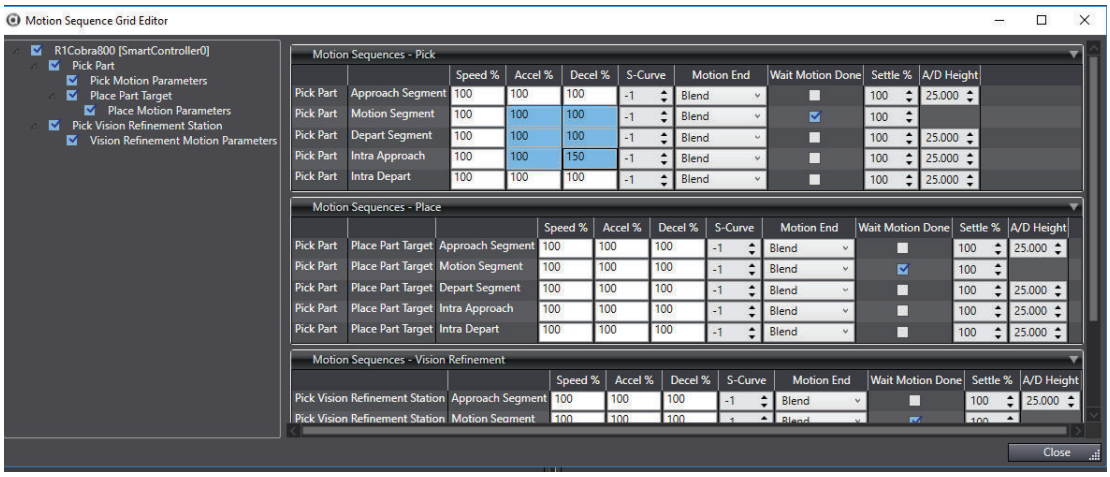
To access the grid editor, double-click on the Process Manager in Multiview Explorer, opening the editor view.



Select the robot object in the Configuration Items group and then click the **Grid** Button (Grid). The Motion Sequence Grid Editor opens.



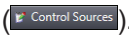
The left pane is used to select the items to display for editing. The right pane contains the editing parameters by group.



Control Sources

The Control Sources editor provides access to parameters that affect Part and Part Target sources for the defined processes. Sources are responsible for providing instances to a process. These are automatically created based on the Part and Part Target object configuration property. There are three types of Sources:

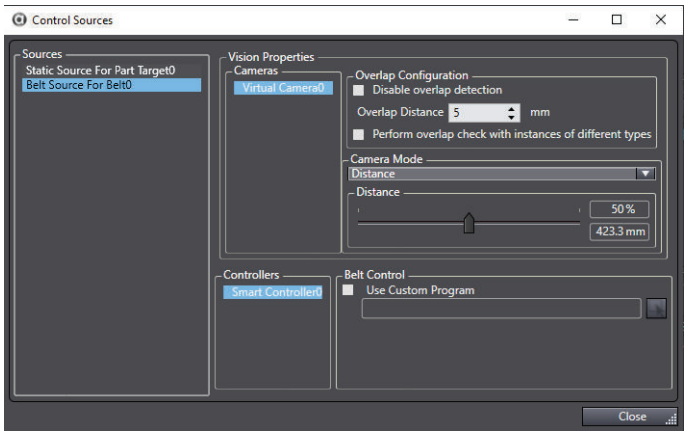
- Belt Control Sources (includes belt-relative cameras)
- Static Control Sources
- Vision Control Sources (not belt-relative)

To access the Control Sources Editor, click the **Control Sources** Button (

).

● Belt Control Sources

This section describes the Control Sources Editor when a Belt source is selected.



■ Camera Objects

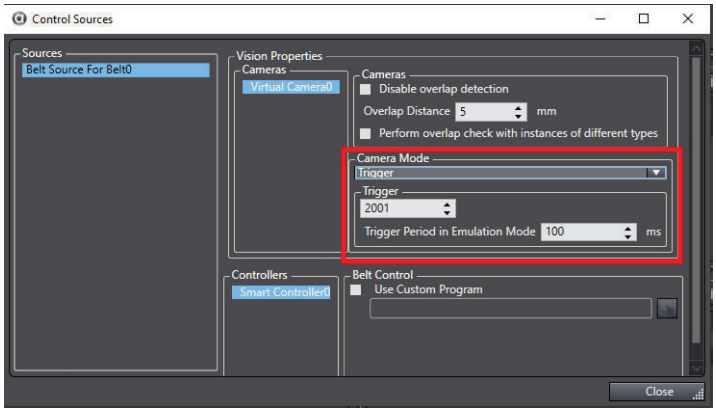
Camera objects can be added to the ACE project to provide a method for obtaining an image to be used for processing. A camera object can be either a physical camera, virtual camera, or an emulated camera. Multiple camera objects can be placed in the ACE project.

Use the information in this section to understand the different camera object types and their configurations in an ACE project.

To add a camera object to the Application Manager in the Multiview Explorer, right-click **Cameras**, select **Add** and then choose a camera type. A new camera object will be added to the Multiview Explorer.
When adding an Emulation camera for a Basler Camera, Sentech Camera, Phoxi 3D camera or Custom Device, there is an additional option to add a Virtual Camera. A Virtual Camera is typically required for most applications.

■ Camera Mode

There are three camera modes available: Distance, Trigger, and Time that are described below.
When **Distance** is selected, the Field of View Picture interval control is enabled. This control is used to adjust the picture interval relative to belt travel. The setting is displayed as a percentage of the field of view, and in millimeters (mm) as calculated from the calibration in the selected virtual camera.
When **Trigger** is selected, the Trigger Signal control is enabled. This 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. For example, this can be used in an application where an image only needs to be acquired when an object activates a sensor if it is below the camera. In this case, the trigger signal is wired to the robot controller and should not be confused with applications that require triggering the camera directly. Triggering a camera directly is configured in the Virtual Camera object Acquisition Settings.
In Emulation Mode, Trigger mode will use the **Trigger Period in Emulation Mode** distance value specified. This is used to simulate the trigger occurring based on the specified distance of belt travel.



When**Time** is selected, an image will be requested on the specified time interval.

■ 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 images that have already been added to the queue of instances to process.
If **Disable Overlap Check** is selected, all overlap checking is disabled. When this option is selected, the remaining Overlap Configuration items are not available. If a part is located in multiple images, the robot will attempt to pick at the same belt-relative position multiple times. If this occurs when Overlap Configuration is not disabled, consider increasing the Overlap distance.
If the newly-located part is within the specified Overlap Distance of a previously located part (accounting for belt travel), it is assumed to be the same part and will not be added as a duplicate new instance.

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.

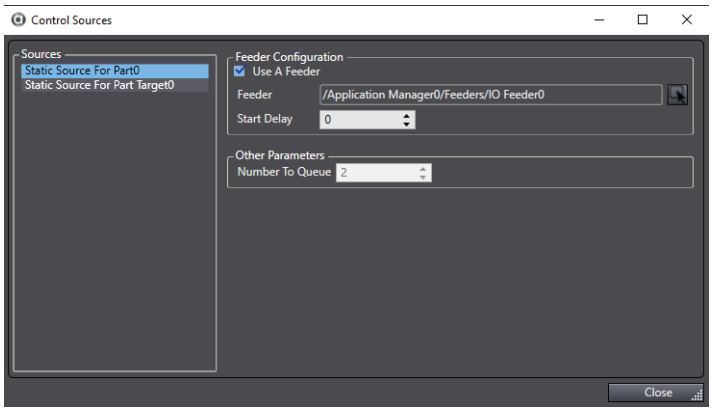
■ Use Custom Program

The default belt program (pm.belt.control) is optimized for performance of all default Process Manager configurations and flexible functionality. The Controllers list will display all controllers associated with the selected Belt object. Each controller executes a V+ program for monitoring and updating encoder position, belt velocity, image requests, latches, and instance information for all instances allocated to that controller.

Occasionally, applications require customization of this program. For example, there may be a need to sacrifice available controller processing time to achieve more frequent latch reporting or image requests. In these cases, select **Use Custom Program** and then edit the default program accordingly. The same modifications may need to be done to the belt program on each controller depending on application requirements.

● Static Control Sources

This section describes the Control Sources Editor when a **Static Control Source** is selected.



Static Sources are used for Part and Part Targets that are not related to belts or cameras.

If an IOFeeder is not enabled, then the PC generates instances for these sources at the Robot Frames defined in configuration items. Each time the controller has emptied the queue of instances, the PC will generate another set of instances and pass them to the controller. The quantity of instances generated is set by the Number to Queue property. The default value is two instances to overcome any disturbances in the communication flow between the PC and controller.

Alternatively, select **Use A Feeder** and choose an IOFeeder object that controls when parts are generated. When this is selected, another V+ program is executed for monitoring feeder activity. This can be used for individual parts or pallet configurations. For example, associate an IOFeeder with a target pallet source to use an input signal from a sensor to indicate when a box is present to be filled.

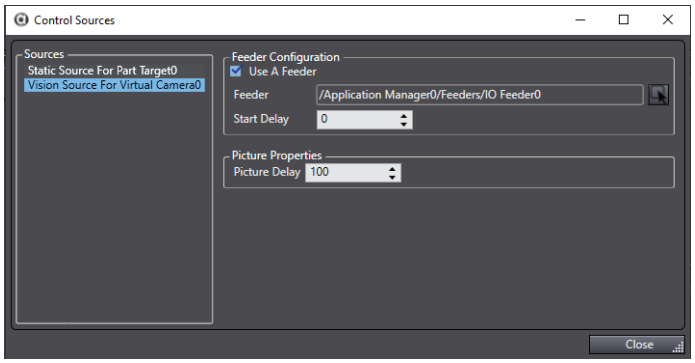
When the Feeder Configuration option is enabled, the **Number To Queue** parameter is disabled.

Use the **Feeder** selection to specify a feeder object. Click the **Select** Button (▣) and select the feeder object.

Use the **Start Delay** selection to specify 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.

● Vision Control Sources

This section describes the Control Sources Editor when a **Vision Source** is selected.



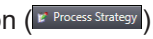
Vision Sources are used for fixed-mounted cameras. They can be associated with I/OFeeder objects, similar to Static Sources.

If a Feeder is not enabled, Vision Sources will trigger a new image to be taken when the last instance of the previous image has been processed, delayed by the Picture Delay (in milliseconds). This delay can be used to ensure the robot has moved out of the pick / image area before a new image is requested because the last part instance is considered processed once the pick operation has completed without error.

Process Strategy

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. The Process Strategy Editor provides access to the following parameters editors.

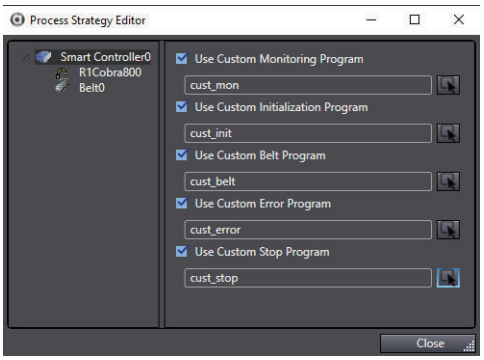
- Controller Parameters. Refer to *Process Strategy Controller Parameters* on page 8-119 for more information.
- Robot Parameters. Refer to *Process Strategy Robot Parameters* on page 8-152 for more information.
- Belt Control Parameters. Refer to *Process Strategy Controller Parameters* on page 8-119 for more information.

To access the Process Strategy editor, click the **Process Strategy** Button (

). The appropriate editor is shown based on the object selected in the left pane of the Process Strategy Editor.

● Process Strategy Controller Parameters

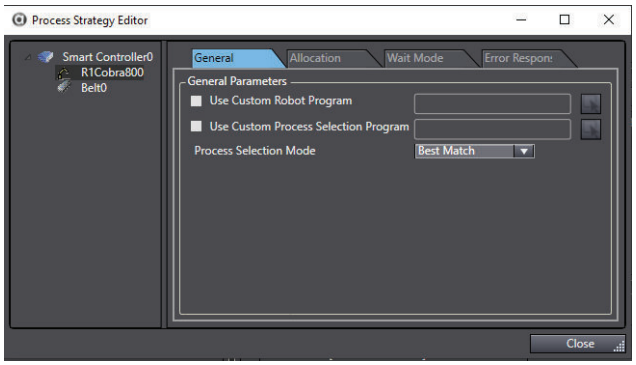
The Controller Parameters are displayed when the controller is selected in the Process Strategy Editor. The Controller Parameters group is used to specify custom V+ programs for the selected controller.



Item	Description
Use Custom Monitoring Program	The default process monitoring has the following functions. • Checks for updates to process strategies. • Handles belt monitoring. • Monitors parts and part targets. Copy the default V+ monitoring program for editing, or select an existing program. Most applications do not require a Custom Monitoring Program
Use Custom Initialization Program	The default initialization program that executes before the control programs (robot, belt, process strategy, etc.) are started. This can be used to initialize system switches, parameters, and variables or execute V+ programs that need to be started in parallel to the Process Manager. Copy the default V+ initialization program for editing, or select an existing program.
Use Custom Belt Program	The default belt program monitors the speed / ON / OFF status of all belts. Copy the default V+ belt program for editing, or select an existing program. Most applications do not require a Custom Belt monitoring program and can be sufficiently controlled using Process Strategy Belt Control Parameters as described in this section.
Use Custom Error Program	The default error program handles the processing and reporting of errors during the execution of a process. Copy the default V+ error program for editing, or select an existing program. This program can be used to automate error handling that are reported to the PC by default. All Process Manager V+ program error handling will lead to this program. Use this program to automate error handling of errors that are reported to the PC by default. This program will check if any user-defined error responses exist in the Process Strategy - Robot parameters.
Use Custom Stop Program	The custom stop program can be used to perform certain operations after the application has stopped. Copy the default V+ stop program for editing, or select an existing program.

● Process Strategy Robot Parameters

The Robot Parameters are displayed when the robot is selected in the Process Strategy Editor. There are four tabs of robot parameters: the *General Parameters* Tab, the *Allocation* Tab, the *Wait Mode Parameters* Tab, and the *Error Response* Tab that are described below.



Group	Item	Description
General	Use Custom Robot Program	Allows specifying a custom V+ main robot-control program. For example, this allows customizing exactly when a given robot executes each process. Copy the default V+ program for editing, or select an existing program.
	Use Custom Process Selection Program	Allows specifying a custom V+ process-selection program. For example, this would this allows customizing the selection and ordering of the processes. Copy the default V+ program for editing, or select an existing program.
	Process Selection Mode	Specifies the process selection mode used by the robot. • 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 in 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 and 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 timeout, it will move to the next process for possible selection.

Group	Item	Description
Allocation	Part or Target Filtering Mode	Specifies how instances are identified for processing by this robot. The filtering process occurs before the allocation process. • No Filtering: All instances are sent to the robot. • Pick / Skip Instances: Processes a certain number of instances then skips a certain number of instances. • Even Distribution: Distributes instances by allocating an equal percentage to all robots. • Percentage of Instances: Allocate a percentage of instances evenly across the range of available instances. • Skip Rate: The robot should process and skip instances to maintain a balanced flow and robot work load in a multi-robot system. • Relative Belt Position: Process instances that are separated by a certain distance. • Pick / Skip Pallets: Process a certain number of pallets then skip a certain number of pallets.
	Queue Size	Specifies the queue size for the robot. Each robot has a queue size, which represents the maximum number of Part or Part Target instances that can be sent to the robot. Each part and target is associated with a different queue for a given robot. This parameter defines the size of each queue. This parameter has a default value of 10. To change the parameter, enter the new value into the Queue Size field, or adjust the value using the up/down arrows.
	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. This can be considered the maximum distance upstream that an instance can be allocated to the robot.
	Allocation Limit	Specifies the distance upstream from the Process Limit Line that a part instance must be to allocate that part instance to the robot. This can be considered the minimum distance upstream that an instance must be for allocation. • If set to zero, instances will be allocated if they are at or upstream of the process limit line. • If greater than 0, it represents the upstream distance (in mm) from the process limit line an instance must be for it to be allocated. This parameter is useful in the case of fast-moving belts where the robot needs additional look ahead to process instances. For example, if a part pallet has four instances and the robot queue size is two, when the robot picks instance one, there is now space in the queue for one more instance. If instance three is upstream of the allocation limit, it will be allocated but if it is downstream of the allocation limit, (>0) it will not be allocated. This prevents allocation of instances that are in the downstream portion of the allocation area.
	Use Custom Allocation Script	Specifies 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. If this option is enabled, click the **Select** Button () to select the desired Custom Allocation Script. Refer to the *ACE Reference Guide - Custom Allocation Scripts* for more information.

Group	Item	Description
Wait Mode	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 Position	Causes the robot to move to the idle position after the specified **After waiting 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. This must be enabled for the Idle Position Parts and Idle Position Part Targets to be used. *Idle Positions* on page 8-144 for more information.
	Use Custom Wait Program	Allows specifying 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. This program starts with the logic specified by one of the selections above. Copy the default V+ program for editing, or select an existing program.
	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.
Error Responses	Output Signal - On Error	Defines a digital signal to assert when an error is detected for the selected robot.
	Error Listing (use the **Add**, **Delete**, and **Up** / **Down** buttons to arrange the list).	Specifies how specific error conditions are to be handled. By default, all errors will be reported and the software will wait until a users responds to the error. If an error condition is defined, it will override this default error handling. It can be used to define automatic handling of the following errors. • Single Error Code: User-defined error code. • Range of Error Codes: User-defined error code range. • Belt Window Access Error: Belt window violations. • Robot Power Errors: Problems with power being enabled or enabling power. • Gripper Errors: All gripper actuations. • All Errors: All other errors. When an error is detected for the robot, it will process the error listing handlers defined 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 Task Status Control. Alternatively, create a Custom Error Program. Refer to *Process Strategy Controller Parameters* on page 8-151 for more information.

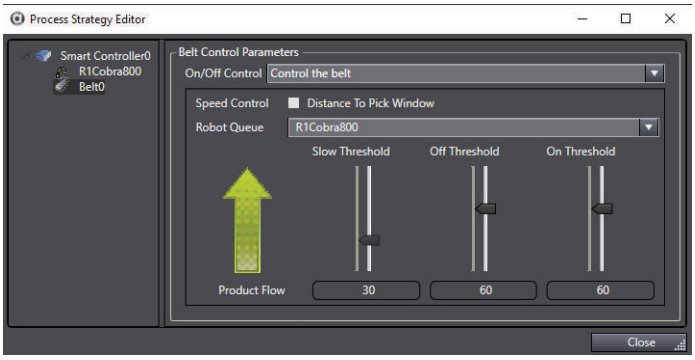
● Process Strategy Belt Control Parameters

The Belt Control Parameters are displayed when the belt is selected in the Process Strategy Editor. Belt Control Parameters are only available when the following items are configured.

- Active Belt Control is enabled in the Belt object configuration.
- A controller is selected in the Belt object configuration.
- A defined process includes a Part or Part Target that references the Belt Object.

The Belt Control Parameters group as 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.



Item	Description
On/Off Control	Specifies the ON / OFF control of the belt. There are three selections available: • **Do not control the belt**: Use this option if there are output signals that can control the belt, without the Process Manager to control the belt during run time. If the belt control is provided by a PLC and output signals are not able to control the belt, disable Active Control in the Belt object. • **Leave the belt always ON**: The belt is turned ON when the process starts and OFF when the Process Manager is stopped. • **Control the belt**: (Default) The belt is controlled based on thresholds described below. The belt is automatically turned OFF when the Process Manager stops the application.
Speed Control	If this is selected, use the Slow Threshold control to adjust the conveyor speed threshold based on how full the robot queue is. 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 Robot Parameters group). The selected robot will typically be the most downstream robot if multiple robots service the same belt. If parts get to the last robot, it needs to slow / stop the conveyor to ensure all Parts or Part Targets are processed.
Slow, Off, and On Thresholds	These thresholds are used to control the belt based on instance position. This is useful for preventing the belt from feeding the robot faster than the robot can pick the parts, or preventing not-processed Part or Part Targets from passing the most downstream robot. **Slow Threshold**: Specifies the point in the belt window for slowing the conveyor if parts reach that point. For example, 50% means that if a part reaches the midpoint of the belt window, the conveyor will be slowed. **Off Threshold**: If an instance reaches this threshold, the belt will be turned OFF. This is set as a percentage from upstream belt window limit to downstream process limit and is visualized as a black line in the 3D Visualizer (in the belt window for the selected robot). **On 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. This is set as a percentage from upstream belt window limit to downstream process limit and is visualized as a green line in the 3D Visualizer (in the belt window for the selected robot).
Product Flow	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 Downstream Process Limit.

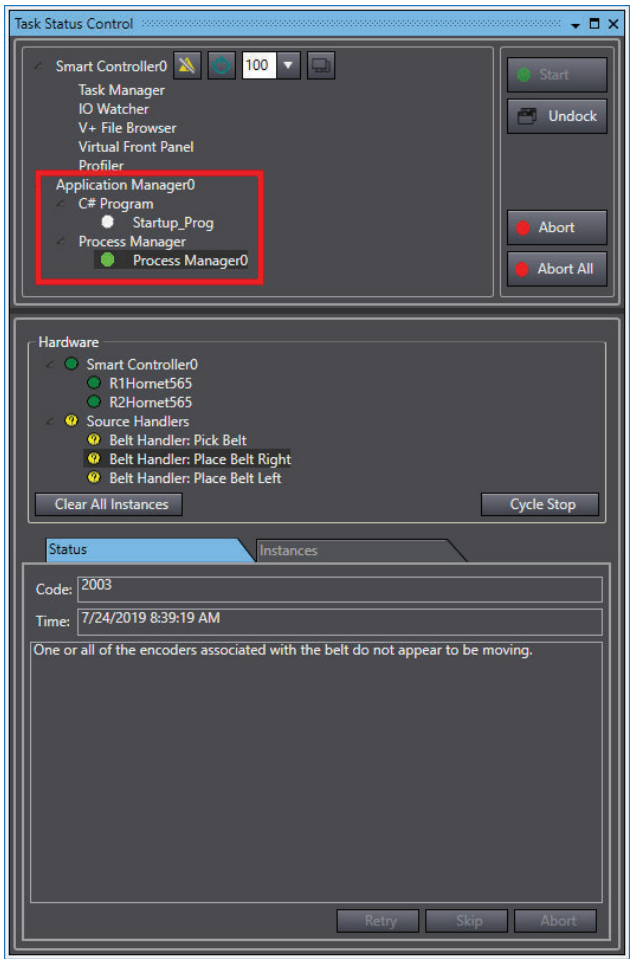
Process Manager Control

The Process Manager Control is used to start and stop a process-managed application, such as a Pack Manager packaging application.

● Using the Process Manager Control

The Task Status Control interface is used to monitor and control Process Manager objects in the ACE project. A Process Manager item in the Task Status Control area is used to select the Process Manager object, start and stop the selected application, and view status and instances on the application while it is operating.

Process Manager control items are added to the Task Status Control area as shown in the following figure. Select a *Process Manager* control item to view the Hardware and Application information areas. Refer to *5-9-11 Application Manager Control* on page 5-49 for more information.



● Process Manager Tasks

Process Manager tasks are displayed under the Application Manager group in a tree view. Tasks are grouped by type (C# program, Process Manager, etc.). When a task is selected, the following functions become available.

Additional Information

Double-clicking a Process Manager task in the Task Status Control interface will open the item in the Edit Pane.

Function	Description
Start 	Executes the selected task.
Undock 	Undocks the selected task's hardware and application information area. Close the undocked window to restore the view.
Abort 	Stops execution of the selected task.
Abort All 	Stops execution for all tasks.

Tasks are marked with icons to indicate operational status.

Icon	Description
	The task is running.
	The task is stopped/aborted.
	The task is idle.

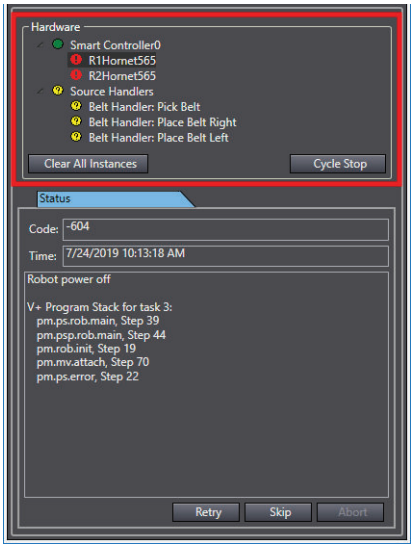
● Hardware Information Area

The Hardware information area displays the hardware items and their status for the selected Process Manager task. Use the information below to understand the functions of the hardware information area.

When a robot is waiting (for example, waiting for Parts or Part Targets to arrive or because of a cycle stop request), a yellow warning condition is displayed on the Process Manager control. Selecting the item in the Hardware Information area will display additional information in the status and instance tabs below.

Additional Information

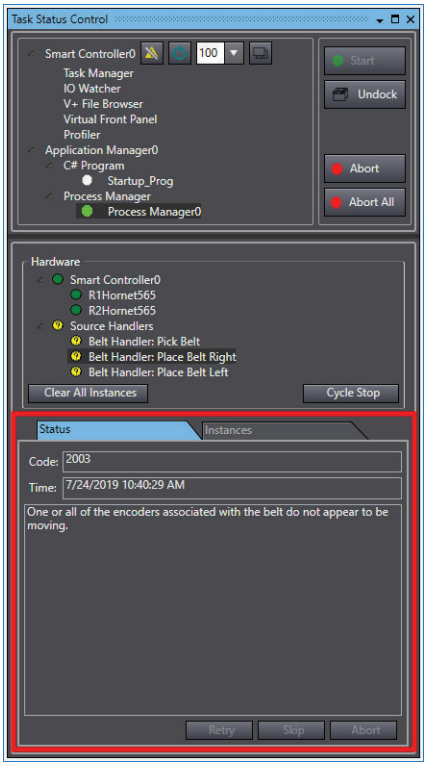
Some items on the Hardware list are in Error and Warning states until the Process Manager establishes communications with and initializes those items.



Item	Icon/Button	Description
Idle State icon		The item is idle.
Operating State icon		The item is operating.
Warning State icon		A warning condition is present for the item.
Error State icon		An error condition is present for the item. The specific warning message is displayed on the Status tab. Refer to *Status Tab* on page 8-160 more information.
ClearAll Instances button		Clears all Part and Part Target instances from the system.
Cycle Stop button		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, Robot is waiting because of a cycle stop request. is displayed in the Status tab. Resume the robot(s) and process operations by clicking the **Cycle Stop** button. The **Cycle Stop** button provides the capability to implement a variety of situations. For example, click the **Cycle Stop** button and leave the system running. When the system is in this state, all tracking is enabled. Therefore, either click the **Abort** button which stops everything or simply resume the current cycle by clicking the **Cycle Stop** button again.

● Application Information Area

The Application information area displays feedback on the operation of the item selected in the Hardware area. The Application information area has a Status tab and an Instances tab which are described below.



■ Status Tab

This tab displays information on the status of the control components driving the process. It shows the hardware in the system and the status of the selected item.

Refer to *Section 9 Troubleshooting* on page 9-1 for more information about status codes and messages.

The Status tab includes the following information.

- **Code**: Displays the error number for the message.
- **Time**: Displays the controller time when the error occurred.
- **Message 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) or displays general status and error message text.
- **Retry, Skip, Abort** Buttons: These buttons and their functions are enabled when in an error state and return the corresponding response. Refer to *Custom Error Programs* on page 8-170 for more information.

■ Instances Tab

The Instances tab displays information on the parts and part targets that are associated with each control source. The **Clear** Button removes all instances from the selected source. To remove all instances from all sources, use the **Clear All Instances** Button in the Hardware section of the Process Manager Control area.

8-7-6 Allocation Script Object

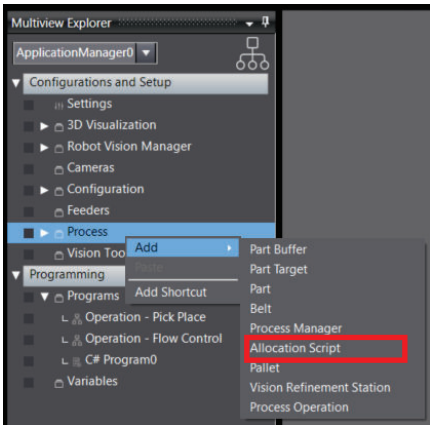
The Allocation Script object is used to create and edit custom part allocation programs for use with the Process Manager.

The Allocation Script object provides two different entry points. One entry point is used for allocating non-belt-relative instances to a robot system. The other entry point is used for allocating belt-relative

instances to a robot system. These programs can manipulate the instances 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.
Allocation Script objects are edited with the C# program editor. Refer to *6-5 Application Manager Programming* on page 6-19 for more information.

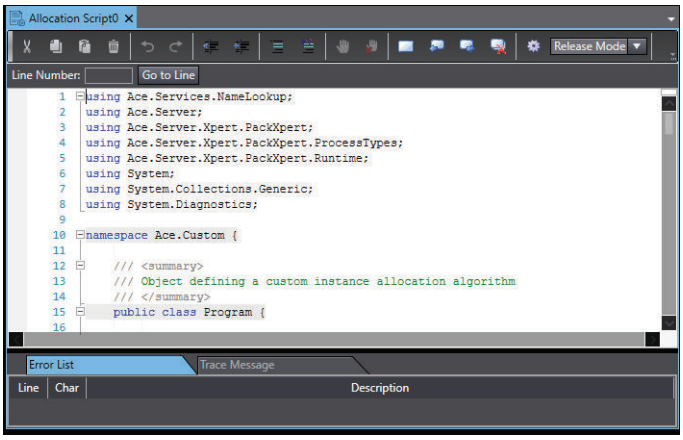
Adding an Allocation Script Object

To add an Allocation Script object, right-click **Process**, select **Add**, and then click **Allocation Script**. A new Allocation Script object will be added to the Multiview Explorer.



Allocation Script Configuration

To access the Allocation Script configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Allocation Script editor in the *Edit Pane*.



Refer to the ACE API Reference Manual (Help Menu) for examples of Allocation scripts.

8-7-7 Pallet Object

The Pallet object defines the layout of a pallet which can be used to pick parts from or place parts to. The Pallet object defines the dimensional information only (three-dimensional pallets are supported). When linked to a frame, it will position the pallet in Cartesian space.

Additional Information

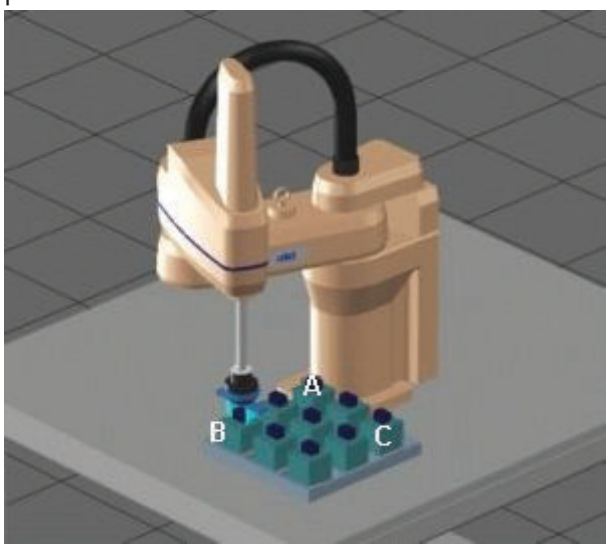
When used with a camera or belt, the camera or belt will be configured to locate the origin of the pallet, not the parts in the pallet.

Defining a Pallet Layout

Define a pallet layout by teaching points for the pallet, such as the pallet origin, a point along the pallet X-axis, and a point along the pallet Y-axis. See the following figure for an example.

Additional Information

The points labeled in the figures are only for example. 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.



Item	Description
A	Pallet Origin
B	A point along the pallet X-axis
C	A point along the pallet Y-axis

For example, assuming a 40 mm part spacing, the 3 x 3 pallet in the previous figure would be defined as follows.

Pallet Properties	Location Components					
	X	Y	Z	Yaw	Pitch	Roll
Pallet Origin (A)	220.0	220.0	54.0	0.0	180.0	0.0
Position Along X-axis	300.0	220.0	54.0	0.0	180.0	0.0
Position Along Y-axis	220.0	300.0	54.0	0.0	180.0	0.0

Also, define the following for each Pallet object as described in this section.

- Access order
- Number of parts and part spacing on the X-axis

- Number of parts and part spacing on the Y-axis
- Number of parts and part spacing on the Z-axis

Teaching a Pallet

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 or place parts from or to positions in the pallet.
The initial pallet teaching process occurs in the Process Manager object configuration during calibration or process teaching (depending on the application needs). Values obtained during the teaching process can be changed. Refer to *8-7-5 Process Manager Object* on page 8-117 for more information.

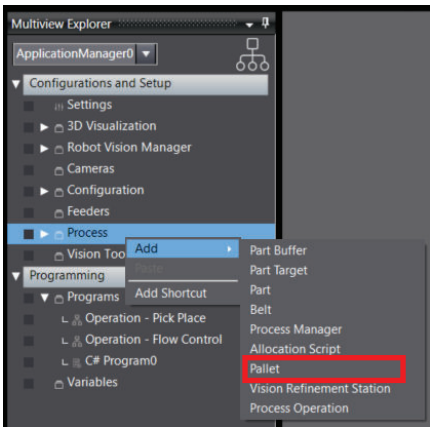


Additional Information

When teaching locations, check the gripper orientation relative to the part to make sure it is unchanged. This will ensure that the parts are picked and placed in the correct orientation.

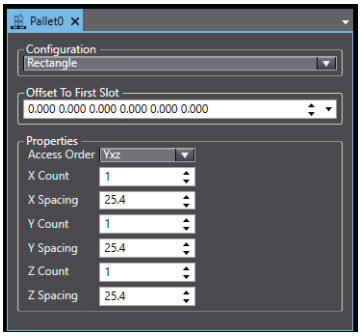
Adding a Pallet Object

To add a Pallet object, right-click **Process**, select **Add**, and then click **Pallet**. A new Pallet object will be added to the Multiview Explorer.



Pallet Configuration

To access the Pallet configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Pallet editor in the *Edit* Pane.



The Pallet editor provides an interface for setting various pallet-related parameters, such as the pallet configuration, location, and properties. The ACE software calculates the individual positions based on the input data and defines the positions in the Pallet object. This allows users to define individual X, Y, and degree positions of each slot and define circular pallets or pallets with offset rows.



Additional Information

The pallet is created as a separate object that can be linked with a frame. Refer to *Robot Frames* on page 8-145 for more information.

● Configuration Drop-down List

The Configuration drop-down list box is used to specify the type of pallet being used.
Rectangle: parts are arranged on the pallet in rows and columns.
Custom: parts are arranged in rectangular or radial pattern. For a rectangular pallet, specify the offsets, spacing, and part counts for X, Y, and Z. For radial pallets, specify the start angle, angle spacing, part count, and radius.

● Offset to First Slot

The Offset to First Slot setting defines the origin of the pallet to reference all slot positions.

● Properties Area

When a rectangular pallet configuration is selected, use the Properties area to specify the access order, part count and part spacing for X, Y, and Z.
When a custom pallet configuration is selected, this area changes to a table that contains information collected from the Add Pattern Dialog Box. Refer to *Custom Pallet Configuration Settings* on page 8-164 for more information.

● Rectangular Pallet Configuration Settings

Use the information below to make the settings for a rectangular pallet configuration.

■ Access Order

The Access Order property defines how the robot will access the pallet. For example, if an access order of Yxz is selected, the robot will begin accessing the pallet positions with the first Y-row. After the Y row is finished, it will move to the next row in the X-direction. After all X-direction rows are accessed, it will move in the z-direction to access the next row.

■ X, Y, Z Count

This defines the number of slots on the X, Y, and Z axes.

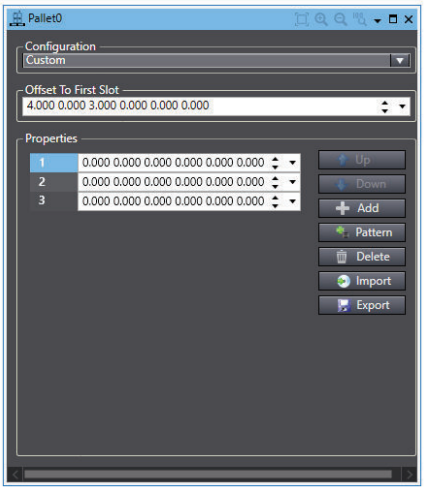
■ X, Y, Z Spacing

This defines the slot spacing for the X, Y, and Z axes.

● Custom Pallet Configuration Settings

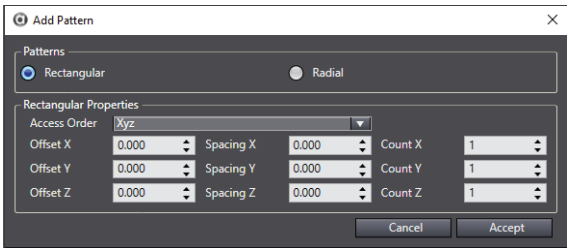
A custom Pallet is typically used for irregular slot arrangements. Use the custom Pallet configuration to define each slot position. For example, if a pallet is 3 x 3 x 2, it will have 18 slot position items defined in the *Properties* area of the custom Pallet object as shown below.
Define individual slot positions manually, using the **Add** Button or automatically using the **Pattern** Button as described below.

When the Pallet has no pattern, use the *Add* button to define individual slot positions.



Use the **Add, Delete, Up and Down** buttons to create and arrange each Pallet slot location.

■ Adding a Custom Pattern

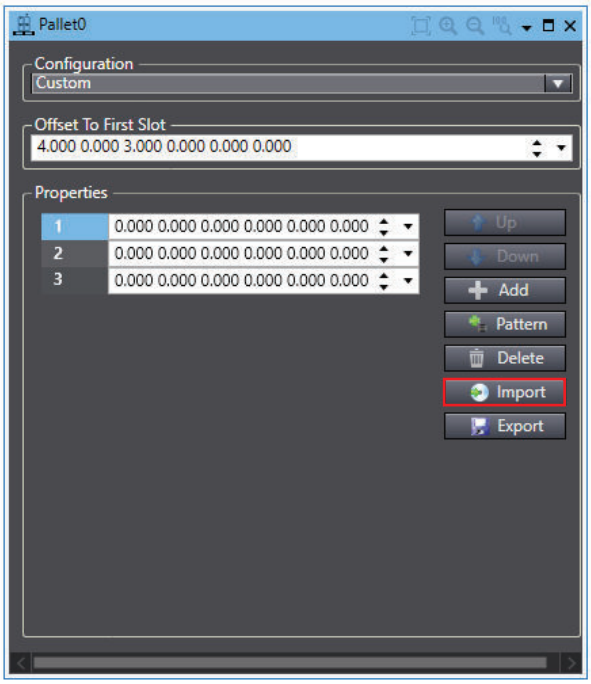


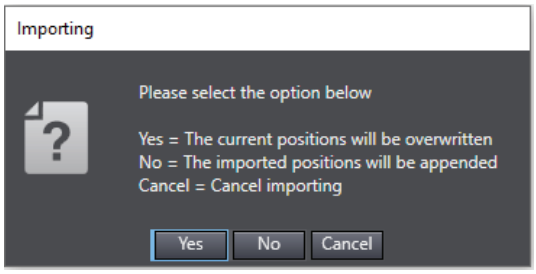
Click the **Pattern Button** (Pattern) to define the custom Pallet using the **Add Pattern** Dialog Box and then choose **Rectangular** or **Radial**.

- **Rectangular Properties**: Set the X,Y and Z offset, spacing, and count for the entire Pattern. When the Rectangular Properties are set, click the **Accept Button** and the Custom Properties list will be populated accordingly.
- **Radial Properties**: Set the Start Angle, Angle Spacing, Count, and Radius for the entire Pattern. When the Radial Properties are set, click the **Radial PropertiesAccept Button** and the Custom Properties list will be populated accordingly.

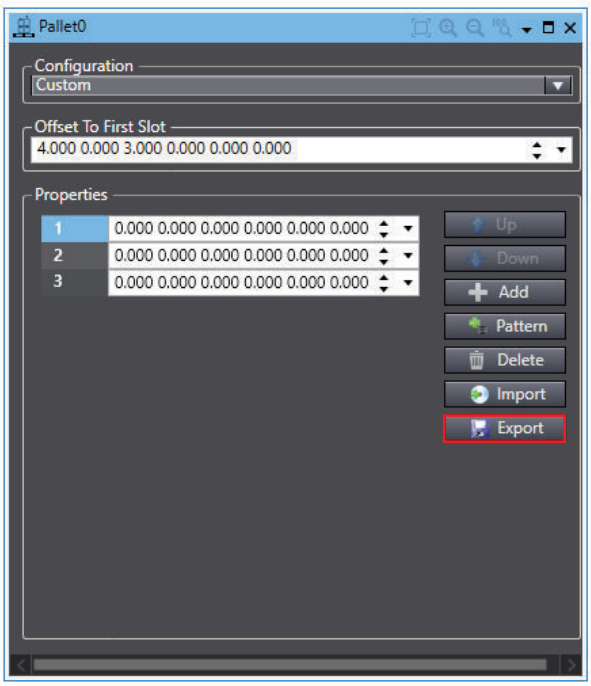
■ Importing and Exporting Custom Properties

To import custom Pallet properties from a .txt file, click the **Import** Button. A window will appear detailing the Import options.

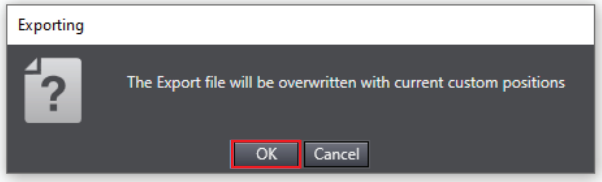




Clicking **Yes** will overwrite the current positions for the selected pallet. Clicking **No** will append the imported positions to the current list of positions for the selected pallet.
To export custom Pallet properties to a .txt file, click the **Export** Button.



You can either export to a new file or overwrite an existing file with the exported properties.To overwrite an existing file, select the file you would like to overwrite from the file browser and click **Open**. A window will appear, select **OK**to overwrite the file



Pallet Visualization

Use the 3D Visualizer to select a shape to represent the pallet. The shape is specified on the Part or Part Target object editor. The shape can be selected from a box, cylinder, or Refer to *8-7-2 Part Target Object* on page 8-101, and *8-7-3 Part Object* on page 8-104 for more information.

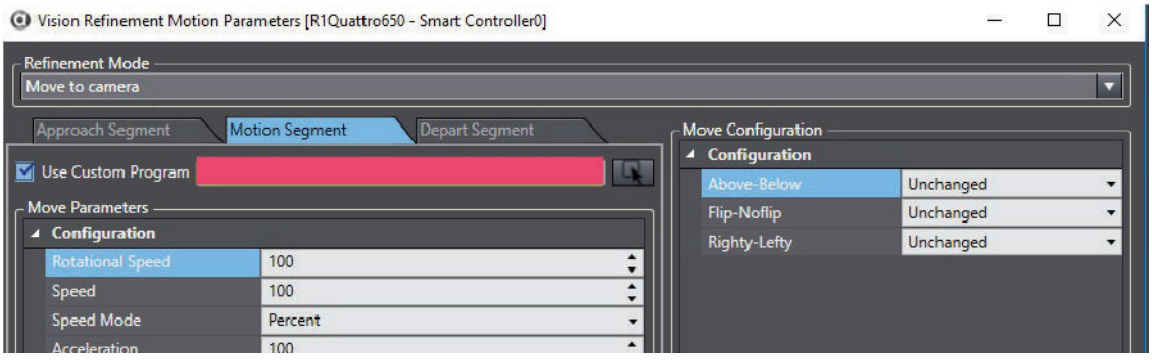
8-7-8 Vision Refinement Station Object

The Vision Refinement Station object defines a location with a camera to improve the part-to-gripper orientation for more accurate placement of the part.

Additional Information

The following information assumes that a physical camera has been installed and calibrated, a virtual camera has been created, and a vision tool and model created.

When the refinement is specified, the instance will be located with a camera after the pick operation and before the placement operation.
The program that defines the motions of a refinement operation can also be overridden if the default motions are not sufficient. The refinement configuration item defines all the parameters used when re-fining parts in the gripper before placing at a target. The motion sequence logic can be overridden.



When the refinement program is customized, the system starts by creating a copy of the default refinement sequence. The default refinement sequence can be broken down into several steps:
If normal camera refinement is enabled

- Calculate the Approach, Move, and Depart Positions
- Perform the Approach Sequence
- Perform Move to Destination
- Issue Camera Refinement Operation
- Perform the Depart Sequence

If Vision-on-the-Fly refinement is enabled

- Calculate the Start and End Positions
- Move to the Start Position
- Start Moving to the End Position
- Trigger the Refinement Operation

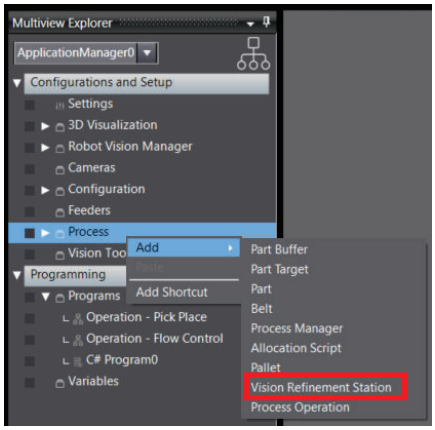
The definition of the default refinement program header example is shown below.

```
.PROGRAM cu.refinement(tsk.idx, grip.idx, cal.idx, cfg.idx, $refine, vals[], calc.offset, sts)

; ABSTRACT:  Perform a vision refinement operation.
;
; INPUTS:       tsk.idx         Index of the task
;               grip.idx        Gripper index to use
;               cal.idx         Index of the camera calibration
;               cfg.idx         Index of the vision refinement configuration
;                                 item parameters
;               $refine         The name of the refinement operation
;               vals            The motion configuration array
;
; OUTPUTS:      calc.offset     The calculated position of the part
;                                 relative to the gripper.
;               sts             Status of the operation:
;                                 pm.tsk.success = Successful operation
;                                 pm.tsk.retry   = Retry the operation
;                                 pm.tsk.skip    = Skip the operation
;                                 pm.tsk.abort   = Abort the operation
;
; SIDE EFFECTS: None
;
;* Copyright (c) 2009-2010 by Adept Technology, Inc.
```

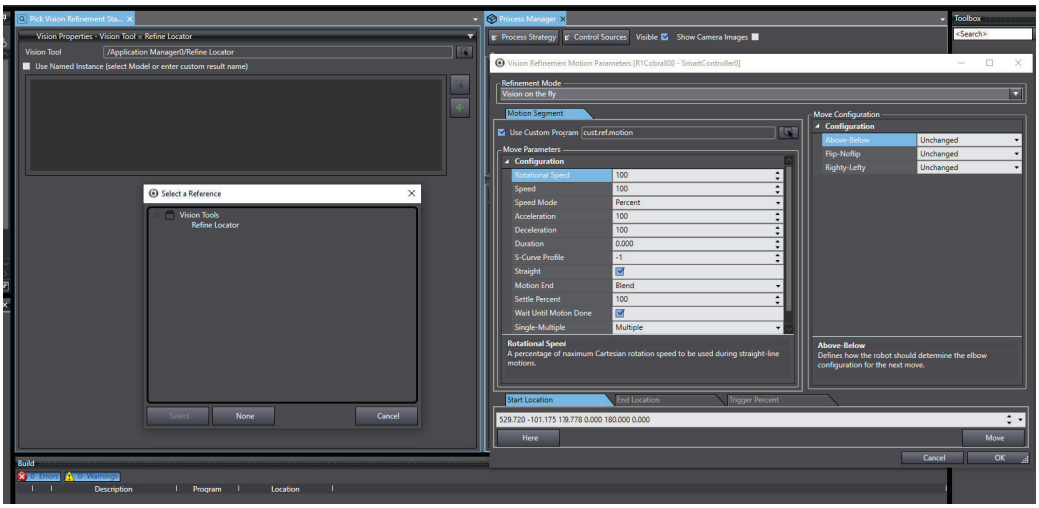
Adding a Vision Refinement Station Object

To add a Vision Refinement Station object, right-click **Process**, select **Add**, and then click **Vision Refinement Station**. A new Vision Refinement Station object will be added to the Multiview Explorer.



Vision Refinement Station Configuration

To access the Vision Refinement Station configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This will open the Vision Refinement Station editor in the Edit Pane.



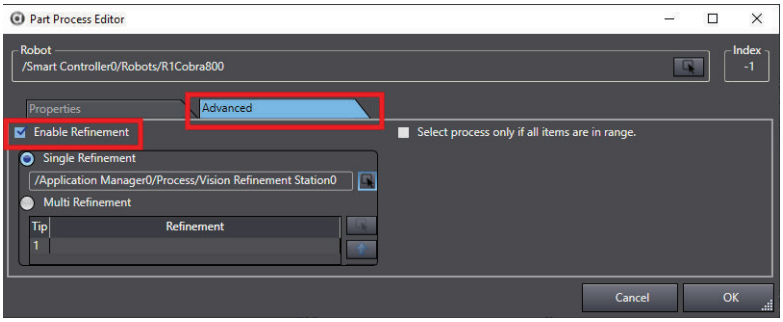
● Vision Properties

The Vision Refinement Station only has a single configuration item. Use the Vision Properties drop-down to specify the vision tool that will be used to locate the part in the gripper.

As an option, select **Use Named Instance** (select **Model** or enter custom result name)and then click the **Select** Button () to reference an exiting Locator Model or click the **Add** Button () to add a custom result name. For 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.

Adding the Vision Refinement Station to a Process

After the Vision Refinement Station is created, 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. Refer to *Process Manager Editor* on page 8-123 for more information.



Editing Motion Parameters

After adding the Vision Refinement Station to the pick-place process, the motion parameters for the station can be edited. This is done using Vision Refinement Motion Parameters that are accessed from the *Configuration*

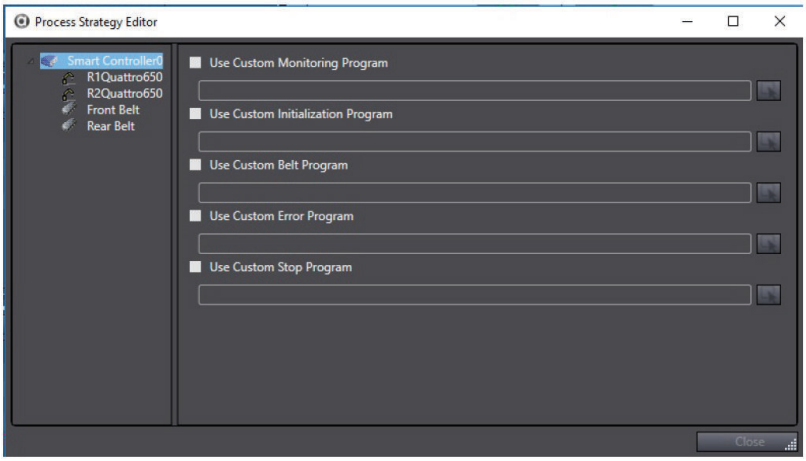
Items group. Refer to *Vision Refinement Motion Parameters* on page 8-143 for more information.

8-7-9 Process Manager Customization

The Process Manager is the central control point for developing a process-managed application, and custom programs for managing applications.

Controller Customization

The general controller parameters are displayed when the controller node is selected in the process strategy editor. The Controller can then be customized for the identified objects by selecting a specific check-box, as shown below. Refer to *Process Strategy Controller Parameters* on page 8-119 for more information.

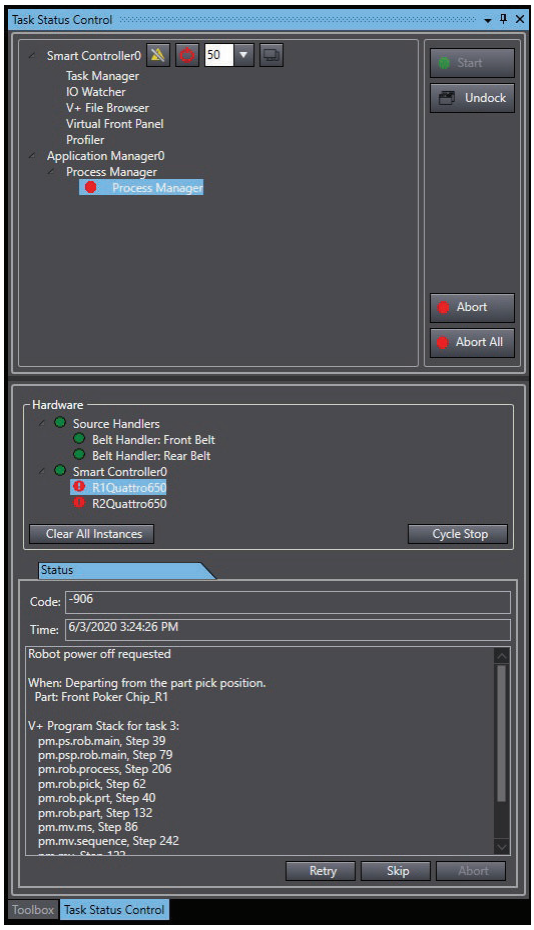


Custom Error Programs

The Process Manager runtime control will display errors that are reported during the execution of a Process Manager application. Many of these errors are generated from the V+ programs that control the robot and monitor other aspects of the hardware on the robot controller. Controller generated error handling can be customized to identify those errors that are displayed on the user interface.

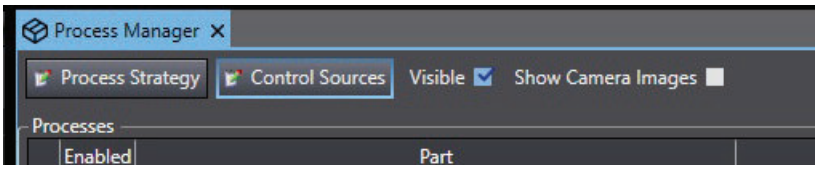
● Error Display

The process manager runtime display will show errors for all hardware sources, controllers, and robot stations, as shown in the following figure.

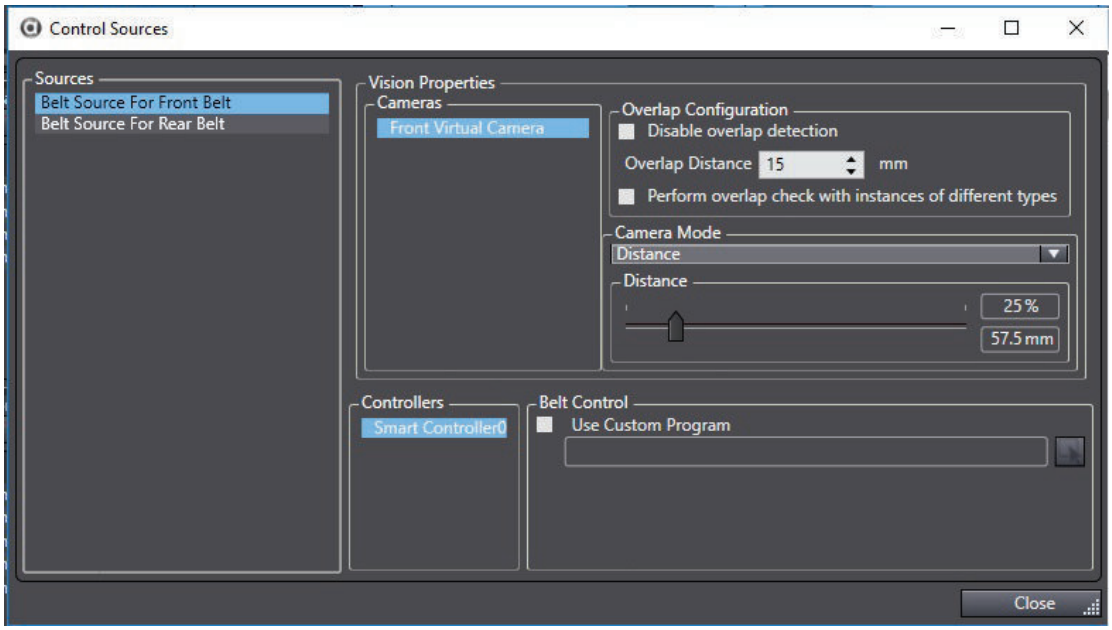


Custom Belt Monitoring

When a belt is used to present part or target instances to a robot, the software creates a default V+ pro-gram to manage the belt operation. The settings associated with the belt monitoring can be viewed under the Control Sources section in the process manager editor.



When selected, the available control sources are presented in a list on the left. Select the belt source for a given belt object in the process configuration, to see a page similar to the following figure. Select-ing the *Use Custom Program* option will enable the custom configuration.



Depending on the specific configurations, different settings will be available. The V+ program can monitor the status of a belt, and any associated latches, cameras, and spacing configurations. The program also reports the status of these items to the PC.

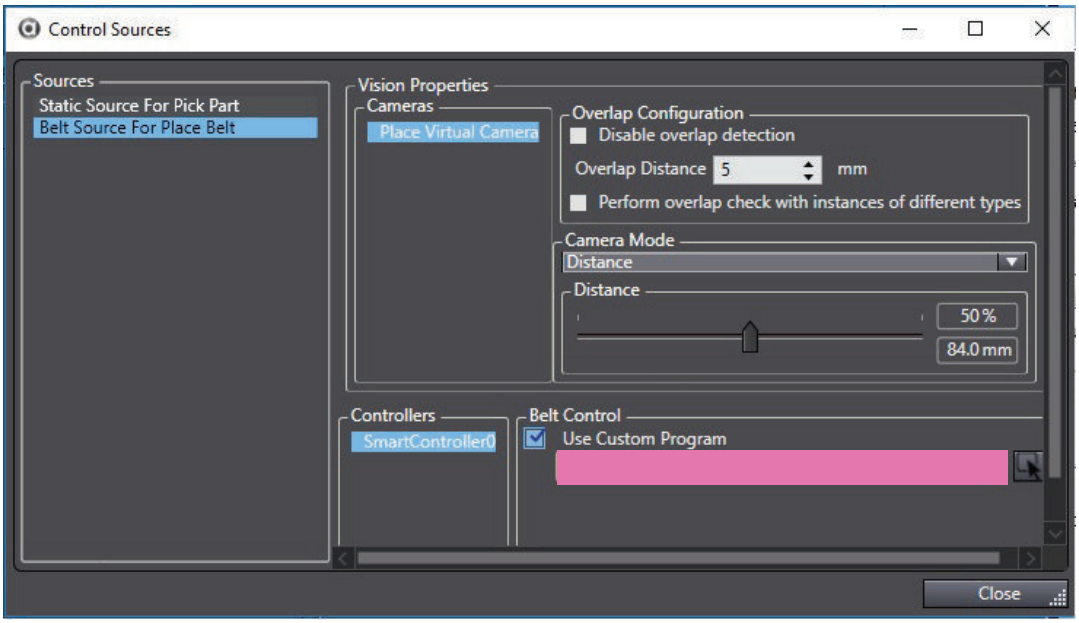
Custom Belt Control

A part or target can be configured for presentation on a belt in one of three modes:

- Belt Camera
- Latch Input signal
- Instances created at a spacing interval

To handle these 3 cases, the process manager must download and run a V+ program. This program must monitor and control when parts and targets are created based on feedback from the belt encoders. As the various belt encoders change position, the program will send messages to the PC triggering the creation of parts and targets.

If the default behavior of the belt program is not sufficient, then the user can customize the default belt control program. The customization can be done in the control source editor. To open the editor, select **Use Custom Program**, as shown in the following figure.



The belt control program is a rather long program. To understand the behavior, it can best to break it into separate sections:

- Initialization
- Monitor encoder motion
- Monitor spacing points
- Monitor hardware latches
- Monitor camera points

● Monitor Encoder Motion

Once the belt control finishes initializing control variables, it will monitor the encoder position and velocity. After it checks all of the encoders, it will periodically send the current encoder position and velocity to the PC using the call pm.ace.blt.pos, as shown in the following figure.

```
DO

    IF (TIMER(-3) < scan.time) AND (TIMER(-3) < send.time) THEN
        WAIT
        NEXT
    END

    CALL pm.blt.init(blt.idx)            ; Monitor belt tracking structures from the PC
    CALL pm.blt.chk.dflt(blt.idx)        ; Check drive fault status

    ; Read the current status of the encoders
    ; Periodically update the PC with the encoder velocity. If a 5%
    ; change in encoder velocity is detected, then send immediately

    send.encoder = (TIMER(-3) > send.time)
    FOR i = 0 TO pm.belt[blt.idx,pm.bcf.encoder]-1
        belt.num = pm.belt[blt.idx,pm.bcf.encoder,i]
        CALL pm.read.encoder(belt.num, encoder.value[blt.idx,i], encoder.vel[blt.idx,i])
        IF (last.enc.vel[blt.idx,i] <> encoder.vel[blt.idx,i]) THEN
            IF (last.enc.vel[blt.idx,i] == 0) OR (encoder.vel[blt.idx,i] == 0) THEN
                send.encoder = TRUE
            ELSE
                IF ABS((last.enc.vel[blt.idx,i]-encoder.vel[blt.idx,i])/encoder.vel[blt.idx,i]) > 0
                    send.encoder = TRUE
                END
            END
        END
        last.enc.vel[blt.idx,i] = encoder.vel[blt.idx,i]
    END

    IF (send.encoder) THEN
        FOR i = 0 TO pm.belt[blt.idx,pm.bcf.encoder]-1
            belt.num = pm.belt[blt.idx,pm.bcf.encoder,i]
            CALL pm.ace.blt.pos(blt.idx, belt.num, encoder.value[blt.idx,i], encoder.vel[blt.idx,i]
        END
        send.time = TIMER(-3)+send.interval
    END

    IF (TIMER(-3) < scan.time) THEN
        WAIT
        NEXT
    END
    scan.time = TIMER(-3)+scan.interval
```

● Monitor Spacing Points

In the initialization section, tracking variables are initialized for the spacing reference points. When a spacing part or target is defined, the user specifies the distance between the generated instances. This spacing parameter is used, along with the scale, to determine how much the conveyor must move before a part is generated. After the conveyor moves the specified distance from the reference point, it will send a message to the PC to create an instance using pm.ace.blt.sp, as shown in the following figure.

```
; Look through the spacing reference points. If the encoder
; has reached a reference point, send the reference to ACE.
; Loop and make sure to send all reference points.

FOR i = 0 TO pm.belt[blt.idx,pm.bcf.sp.obj]-1

    idx = pm.belt[blt.idx,pm.bcf.sp.enc,i]
    encoder = encoder.value[blt.idx,idx]
    scale = pm.belt[blt.idx,pm.bcf.scale,idx]

    DO

        CALL pm.blt.travel(encoder, space.ref[blt.idx,i], distance)
        distance = distance*scale

        IF distance >= pm.belt[blt.idx,pm.bcf.spacing,i] THEN

            ; Update the reference position and handle
            ; encoder rollover.

            space.ref[blt.idx,i] = space.ref[blt.idx,i]+(pm.belt[blt.idx,pm.bcf.spacing,i]/scal
            CALL pm.blt.convert(space.ref[blt.idx,i])

            $spacing.item = $pm.belt[blt.idx,pm.bcf.sp.obj,i]
            CALL pm.ace.blt.sp(blt.idx, $spacing.item, space.ref[blt.idx,i], "")

        END
    UNTIL distance < pm.belt[blt.idx,pm.bcf.spacing,i]
END
```

Associate a custom tag with instances created by the spacing event by modifying the call to pm.ace.blt.sp to pass in a tag for the last argument.

● Monitor Hardware Latches

The program will go through the list of encoders that need to be monitored for latches. When a latch is detected, it will check the latched encoder position against the minimum latch distance that was defined in the belt editor. If the latch distance is sufficient, the latch information is sent to the PC using pm.ace.blt.ltch., as shown in the following figure.

```
        ; Look through the latch items. If a latch had been seen
        ; previously, then ensure the new latch is spaced far enough
        ; apart from the previous latch.

        FOR i = 0 TO pm.belt[blt.idx,pm.bcf.latch]-1

            idx = pm.belt[blt.idx,pm.bcf.latchenc,i]
            encoder = pm.belt[blt.idx,pm.bcf.encoder,idx]+1
            scale = pm.belt[blt.idx,pm.bcf.scale,idx]

            latch.num = LATCHED(-encoder)
            WHILE (latch.num) DO

                ; Get the index associated with the latch that was
                ; found and read the latch. The latch may not be one we are
                ; monitoring for. In that case, ignore the latch

                CALL pm.blt.gltchix(blt.idx, latch.num, idx)
                latch.value = DEVICE(0,encoder-1,sts,4)

                IF (idx >= 0) THEN

                    ; If latch distance checking is in effect, we must check
                    ; if the distance between latches is large enough.

                    IF latch.ref.chk[blt.idx,idx] THEN
                        CALL pm.blt.travel(latch.value, latch.ref[blt.idx,idx], distance)
                        scale = pm.belt[blt.idx,pm.bcf.scale,i]
                        distance = distance*scale
                        IF (distance < pm.belt[blt.idx,pm.bcf.latchdst,idx]) THEN
                            latch.num = 0
                        END
                    END

                    ; If there is a latch to report, send to the PC.
                    ; Update the latch reference for latch distance
                    ; checking when the next latch is detected.

                    IF latch.num THEN
                        IF (sts >= 0) THEN
                            CALL pm.ace.blt.ltch(blt.idx, encoder-1, latch.num, latch.value, "")
                            latch.ref[blt.idx,idx] = latch.value
                            latch.dist[blt.idx,idx] = 0
                            latch.ref.chk[blt.idx,idx] = TRUE
                        END
                    END
                END

                latch.num = LATCHED(-encoder)
            END
```

Some applications may want to change the logic for when latches are reported. For example, an application may require only sending every-other latch so a robot only processes 50% of the instances generated from the latch.
Associate a custom tag with instances created by the latch event by modifying the call to pm.ace.blt.ltch to pass in a tag for the last argument.

● Monitor Camera Points

In the initialization section, tracking variables are initialized for the camera reference points. When a belt-camera based part or target is referenced in the process manager, the user can specify if the picture is triggered by conveyor motion or a digital signal. When conveyor motion is specified, it is necessary to specify a camera picture percent. This is used, along with the scale and the average field of view size, to determine how much the conveyor must move before a camera picture request is generated. After the conveyor moves the specified distance from the reference point, it will send a message to the PC to perform a picture using pm.ace.blt.cam.
If a signal trigger is specified, pictures will be taken when the signal is triggered.

```
; Look through the belt camera items

FOR i = 0 TO pm.belt[blt.idx,pm.bcf.cameras]-1

    take.picture = FALSE

    ; If operating in picture distance mode, then calculate how far the encoder
    ; has moved and see if a picture needs to be taken.

    IF pm.belt[blt.idx,pm.bcf.pic.md,i] == pm.bc.pmd.dist THEN

        idx = pm.belt[blt.idx,pm.bcf.cam.enc,i]
        encoder = encoder.value[blt.idx,idx]
        scale = pm.belt[blt.idx,pm.bcf.scale,idx]

        CALL pm.blt.travel(encoder, cam.ref[blt.idx,i], distance)
        distance = distance*scale

        inc = pm.belt[blt.idx,pm.bcf.cam.fov,i]*pm.belt[blt.idx,pm.bcf.fov.intv,i]

        IF distance > inc THEN

            take.picture = TRUE

            ; Update the reference position and look for
            ; rollover. If the camera gets behind by 2x the
            ; distance, set the reference based on the encoder
            ; position. The belt is moving faster than the
            ; camera can process.

            IF distance > 2*inc THEN
                cam.ref[blt.idx,i] = encoder-(inc/scale)
            ELSE
                cam.ref[blt.idx,i] = cam.ref[blt.idx,i]+(inc/scale)
            END
            CALL pm.blt.convert(cam.ref[blt.idx,i])

        END
    ELSE
        ; Operating in trigger mode. If the trigger signal goes on, then we need to take a pict

        IF SIG(pm.belt[blt.idx,pm.bcf.pic.trig,i]) THEN
            take.picture = TRUE
        END
    END

    IF take.picture THEN

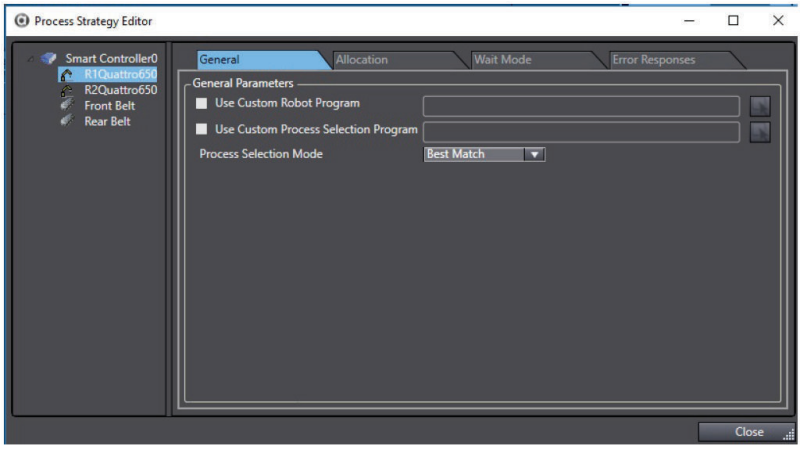
        $spacing.item = $pm.belt[blt.idx,pm.bcf.cameras,i]
        remote = pm.belt[blt.idx,pm.bcf.cam.isrm,i]
        CALL pm.ace.blt.cam(blt.idx, $spacing.item, remote, cam.ref[blt.idx,i])

        ; If we are in trigger mode, then we need to wait until the signal goes low
        ; But we check if the process manager has not been stopped
        ; If we are in emulation mode, we will simply reset the trigger with no wait

        IF (pm.belt[blt.idx,pm.bcf.pic.md,i] == pm.bc.pmd.trig) THEN
            IF sv.emulate.mode THEN
                SIGNAL -pm.belt[blt.idx,pm.bcf.pic.trig,i]
            ELSE
                DO
                    CALL pm.chk.stat(run)
                    WAIT
                UNTIL NOT ((SIG(pm.belt[blt.idx,pm.bcf.pic.trig,i]) AND run == TRUE))
            END
        END
    END
END
```

Robot Customizations

The robot parameters are displayed when a robot node is selected in the process strategy editor. There are many parameters associated with a robot. These settings are further broken down into a variety of categories represented by the four tabs in the display: *General*, *Allocation*, *Wait Mode* and *Error Responses*.



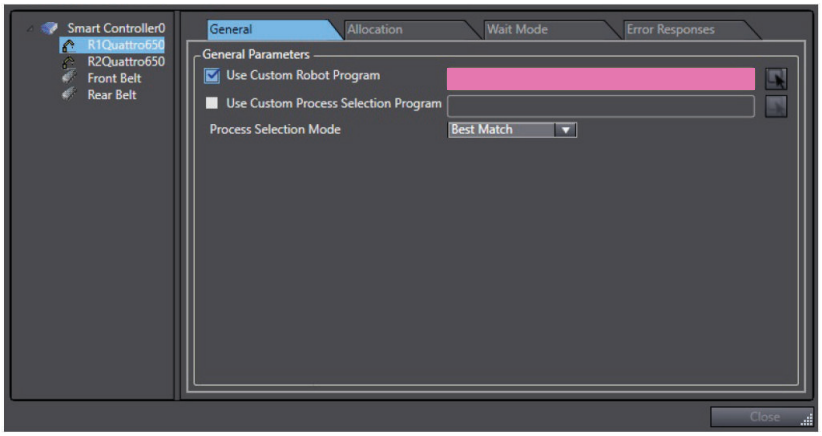
There are two available parameters for the *General* Tab; *Use Custom Robot Program* and *Use Custom Process Selection Program*.

● Custom Robot Process Selection Program

The process manager application software allows users to take complete control over the behaviors of a robot by customizing the main program that governs the robot top level behaviors. When a custom robot program is created, a copy of the default program is used as the basis for the custom program.

The default main robot program will call a process selection program and then execute that process. However, some applications may benefit from customizing the main robot program to force the selection of a process directly based on some custom condition or bypass the concept of processes entirely and command the robot to pick directly from the part and target queues. These two possibilities are detailed below.

The robot program can be customized in the process strategy editor.



Use the following customization steps to change the selection of processes. Below is an example of a program that does the following:

- Initializes the robot program moving it to an idle position.
- Looks up the process index for 2 processes defined in the process manager.
- Command the robot to execute the first process 3 times followed by 1 execution of process two.
- Check to see if the stop button was pressed.

```
.PROGRAM cu.robot2()

; ABSTRACT: Custom robot program for the Process Manager
; Process Strategy.

        AUTO REAL task.idx, i, sts
        AUTO REAL proc.idx.1, proc.idx.2

        tsk.idx = TASK()

        ; Initialize the robot
        CALL pm.rob.init(tsk.idx)
        CALL pm.rob.idlemd(tsk.idx, FALSE, 0, "")

        ; Get the index associated with the first two processes in the process manager
        CALL pm.ps.map.idx(0, proc.idx.1)
        CALL pm.ps.map.idx(1, proc.idx.2)

        ; Control loop
        DO

            ; Execute the first process 3 time
            FOR i = 1 TO 3
                CALL pm.rob.process(tsk.idx, proc.idx.1, sts)
            END

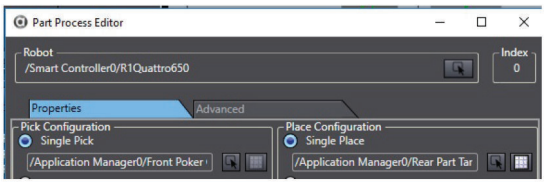
            ; Execute the second process
            CALL pm.rob.process(tsk.idx, proc.idx.2, sts)

            ; Check process status
            CALL pm.chk.stat(is.running)

        UNTIL is.running == FALSE

        RETURN
.END
```

The process index used in the custom robot program in the call to pm.ps.map.idx is the process index associated with the process on the PC. This process index can be located in the *Part Process* Editor in the *Process Manager Editor*.



The program pm.ps.map.idx converts the process index on the PC to an index relative to the current robot task. It is this converted index that must be used when calling pm.rob.process.

■ Access Queues Customization

Accessing queues is an alternative to using process selection to customize the robot behaviors. A custom application can directly access the part and target queues without using processes. There are times where the options allowed in the confines of a process are to restrictive. Using this technique, custom logic can be created without having to specify low level motions of the robot.

Below is an example of a program that does the following:

- Initializes the robot program moving it to an idle position.
- Looks up the queue numbers for a part and a part target.
- Command the robot to pick a part and then perform a placement of that part at a part target.
- Check to see if the stop button was pressed.

```
.PROGRAM cu.robot3()

; ABSTRACT: Custom robot program for the Process Manager
; Process Strategy.

        AUTO REAL tsk.idx, is.reset, sts
        AUTO REAL part.idx, target.idx

        tsk.idx = TASK()

; Initialize the robot

        CALL pm.rob.init(tsk.idx)
        CALL pm.rob.idlemd(tsk.idx, FALSE, 0, "")

; Get the part and target indexes

        CALL pm.prt.get.idx(tsk.idx, "/Process/Part - Spacing 2", part.idx)
        CALL pm.trg.get.idx(tsk.idx, "/Process/Target 2 - Static", target.idx)

; Control loop

        DO

            ; Perform a pick
            CALL pm.rob.pick(tsk.idx, part.idx, -1, pm.ms.mv.norm, , is.reset, sts)

            ; Perform a place
            CALL pm.rob.place(tsk.idx, target.idx, part.idx, -1, pm.ms.mv.norm, , is.reset, sts)

            ; Check to see if the process manager is still running
            CALL pm.chk.stat(is.running)

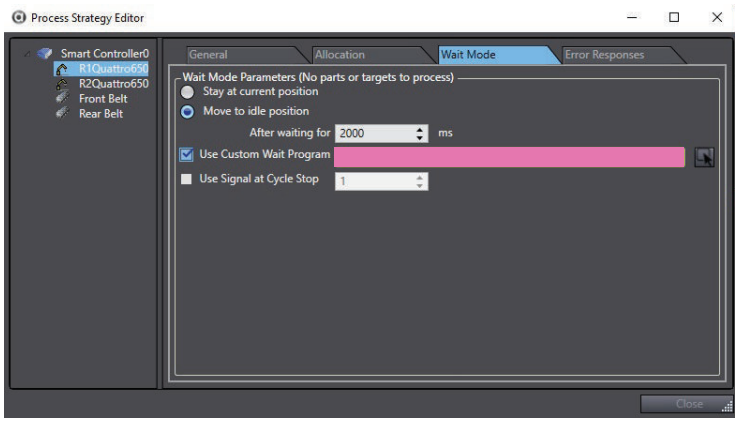
        UNTIL is.running == FALSE

        CALL pm.rob.clear(tsk.idx, sts)

        RETURN
.END
```

■ Custom Robot Wait Programs

When the robot does not have product available, it enters idle mode. The behavior of the robot is selectable for an idle waiting mode:



By default there is the option to leave the robot at the current position or move it to the idle position after waiting a certain amount of time. The robot wait program can be customized. When it is customized, start with a copy of the default program:

```
.PROGRAM cu.rwait(rob.idx, $type, time, done)

; ABSTRACT:  Method called when the robot does not have a part or target
;       available. This program checks to see if the robot needs to be
;       moved to the idle location or should stay at the current location.
;
;       This can be called if no process is available -or- a process is
;       selected but the expected part or target is not in the robot queue.
;
;       This program will be called iteratively. This program should do a
;       check to see if the robot should be moved then return. If it issues
;       the move, it should set the done parameter to TRUE.
;
; INPUTS:       rob.idx         Index of the robot
;               $type           Type of object the robot is waiting for
;                                 or empty ("") if no process has been selected
;               time            Time that has elapsed. Units are in ms.
;
; OUTPUTS:      done            Has the robot been moved? Setting this to TRUE
;                                 will indicate the robot has been moved to the
;                                 waiting position.
;
; SIDE EFFECTS: None
;
;* Copyright (c) 2007-2009 by Adept Technology, Inc.
```

The default program gets called while the robot is waiting. By default, this program implements the logic described above. If *Move to Idle* is enabled, it will monitor the elapsed time and move the robot to the idle position as needed. This can be seen in the body of the program, in the following figure.

```
        AUTO REAL move.to.idle, values[4]

; Extract the parameters for the waiting

        CALL pm.psp.rob.pars(rob.idx, values[])

; See if we need to move to idle

        IF (values[psp.rob.mvidle]) THEN
            IF (time > values[psp.rob.delay]) THEN
                move.to.idle = TRUE
            ELSE
                move.to.idle = FALSE
            END
        ELSE
            move.to.idle = FALSE
        END

; Perform the move to idle position, if needed

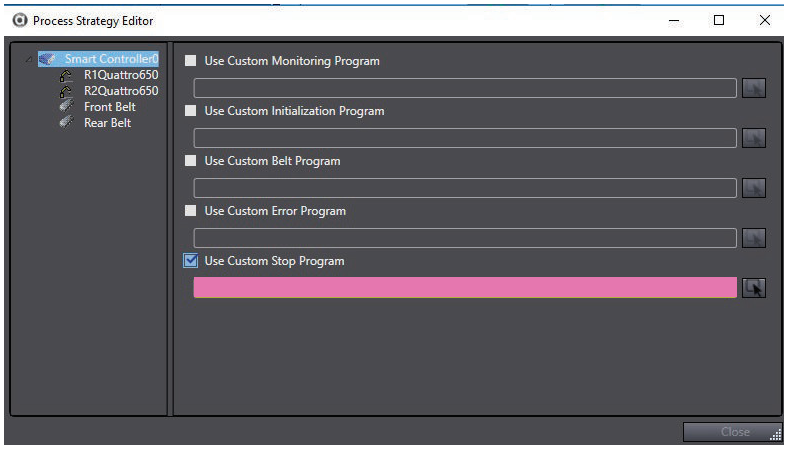
        IF (move.to.idle) THEN
            CALL pm.mv.idle(rob.idx)
            done = TRUE
        END

        RETURN
```

■ Customizing Stop Behavior

When the process manager application is stopped, the system will perform a stop procedure ending the V+ tasks used by the runtime.

The user can define a V+ program that is called when the process manager is in the process of stopping through the following menu:



The default stop program looks similar to the following figure.

```
.PROGRAM cu.stop(tasks.stopped)

; ABSTRACT:  Perform operations required when the application is stopped.
;       This method will be called before tasks have been stopped and
;       after the tasks have been stopped.
;
;       This program will be allowed to run for at most 500 ms. It should only
;       perform operations that will be completed within that 500 ms time limit.
;
; INPUTS:       tasks.stopped   Have the tasks been stopped?
;
; OUTPUTS:      None
;
; SIDE EFFECTS: None
;
;* Copyright (c) 2007-2008 by Adept Technology, Inc.

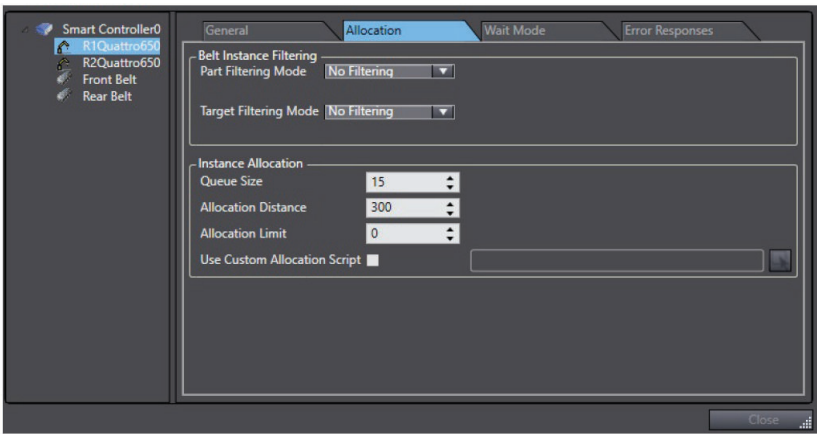

; Check the belts to see if they need to stop running

        IF (tasks.stopped) THEN
            CALL pm.psp.belt(pm.ps.tsk, FALSE)
        END

        RETURN
.END
```

● Robot Allocation

The parameters in the allocation section control how part and target instances are allocated to the associated robot:



■ Custom Allocation Script

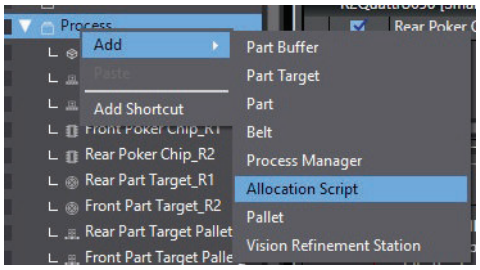
The process manager allows for customization of the allocation logic to change from the default allocation strategy to one that is effective for specific needs.

The default allocation strategy is designed so that each robot will pick as many instance as possible and instances are ordered from farthest downstream to farthest upstream. Any instances not processed by a given robot are placed in the next robot queue that is capable of processing the instance.

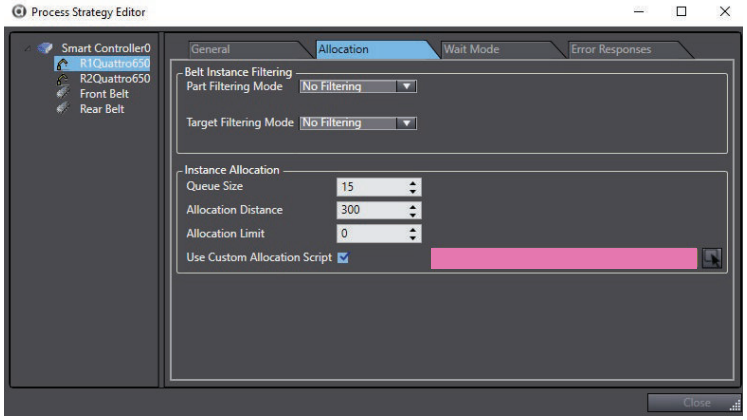
The following are two examples of situations that would require a custom allocation script.

- Instances need to be picked up in rows perpendicular to the flow of parts. In this case, instances may need to be grouped into rows then sorted left-to-right.
- Instances need to be load balanced between a certain number of robots.

An allocation script can be added to the workspace by clicking **Process**, then **Add** and selecting **Allocation Script**, as shown in the following figure.



Once created, the script must be associated with each robot that will use the script.



Each robot maintains a queue size which defines how many instances of a given kind can be send to a robot controller. This queue size parameter is not the total number of instances, it is the total number of instances of each part or target that can be sent. The robot controller

maintains a different queue for each possible part or target that a robot can process. This parameter defines the queue size for each of these queues.

After the default allocation script is created, initialize it to use the logic of the default allocation algorithm. There are 4 different methods that can be customized:

- Initialize
- Allocate Non-Belt Instances
- Allocate Belt Instances
- Notification as Instances are Processed

■ Default Allocation Script Initialize Method

The Initialization method is called when the process manager is started. Each robot that is associated with the script will used to call the initialize method:

```
/// <summary>
/// Initializes the allocation algorithm at the start of runtime processing.
/// </summary>
/// <param name="station">The station.</param>
public void Initialize(RobotStation station) {
    Trace.WriteLine("Initializing for: "+station.Robot.FullPath);
}
```

■ Allocate Non-Belt InstancesObject

The data requirements and allocation logic for belt based instances are very different than for non-belt based instances. As such, each class of instances is handled by a separate method. The non-belt instance method is called to allocate static and non-belt camera instances. The default behavior is to allocate the instances to the robot as long as there is space in the queues. The two non-belt instances are indicated below.

- availableInstances: Input parameter containing the list of all instances that could be allocated.
- instancesToAllocate: Output parameter containing the instances that should be sent to the robot controller.

It is the job of the AllocateNonBeltInstances to move instances from the availableInstances list to the instancesToAllocate list.

```
/// <summary>
/// Allocates the non-belt relative instances to a robot station.
/// </summary>
/// <param name="station">The station.</param>
/// <param name="processType">The type of object being allocated.</param>
/// <param name="queueSize">The queue size of the robot station.</param>
/// <param name="availableInstances">The available instances.</param>
/// <param name="instancesToAllocate">The instances to allocate to the station.</param>
public void AllocateNonBeltInstances(RobotStation station, IProcessType processType, int queueSize, Locat

    // If no instances, do not process
    if (availableInstances.Count == 0)
        return;

    // Get the number of items in the robot queue for that type
    int numberInQueue = station.GetInstancesInProcess(processType);

    // Add as many items as we can to the queue
    foreach (LocatedInstance instance in availableInstances) {

        // Make sure there is room to add
        if (numberInQueue >= queueSize)
            break;

        // Allocate the part to the station
        instancesToAllocate.Add(instance);

        // Note that one more has been added
        numberInQueue++;

    }
}
```

■ Allocate Belt Instances

The default behavior for allocation of belt-based instances is similar to non-belt instance handling from the standpoint that instances are allocated to a robot as long as the queue can take more instances.

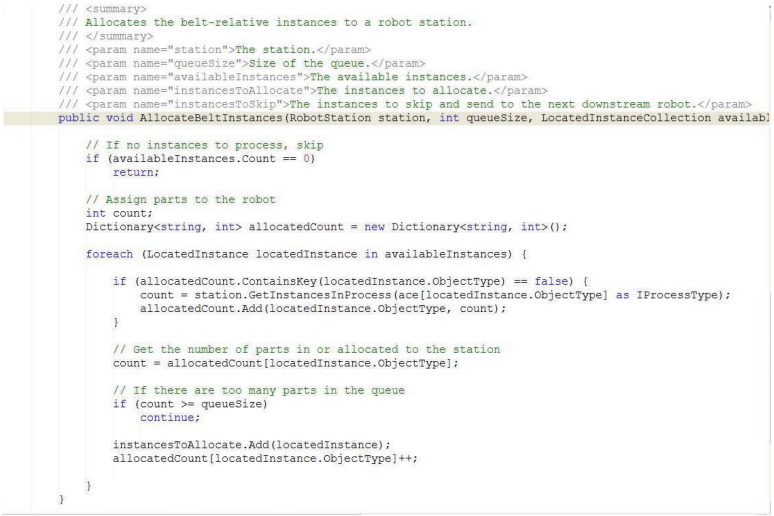
The belt instances allocation method is passed three different arrays of note:

- availableInstances: Input parameter containing the list of all instances that could be allocated.

- instancesToAllocate: Output parameter containing the instances that should be sent to the robot controller.
- instancesToSkip: Output parameter containing the instances that should be sent to the next down-stream robot capable of processing the instance.

It is the job of the AllocateBeltInstances to move instances from the availableInstances list to either the instancesToAllocate list or the instancesToSkip list. The availableInstances list is an ordered list where the first instance in the list is the farthest downstream instance that is still capable of being processed by the robot station.

If an instance is never allocated by the AllocateBeltInstances method, the instance will automatically be moved to the queue of the next downstream robot once the instance passes out of the robot belt window.



```
/// <summary>
/// Allocates the belt-relative instances to a robot station.
/// </summary>
/// <param name="station">The station.</param>
/// <param name="queueSize">Size of the queue.</param>
/// <param name="availableInstances">The available instances.</param>
/// <param name="instancesToAllocate">The instances to allocate.</param>
/// <param name="instancesToSkip">The instances to skip and send to the next downstream robot.</param>
public void AllocateBeltInstances(RobotStation station, int queueSize, LocatedInstanceCollection availabl

    // If no instances to process, skip
    if (availableInstances.Count == 0)
        return;

    // Assign parts to the robot
    int count;
    Dictionary<string, int> allocatedCount = new Dictionary<string, int>();

    foreach (LocatedInstance locatedInstance in availableInstances) {

        if (allocatedCount.ContainsKey(locatedInstance.ObjectType) == false) {
            count = station.GetInstancesInProcess(ace[locatedInstance.ObjectType] as IProcessType);
            allocatedCount.Add(locatedInstance.ObjectType, count);
        }

        // Get the number of parts in or allocated to the station
        count = allocatedCount[locatedInstance.ObjectType];

        // If there are too many parts in the queue
        if (count >= queueSize)
            continue;

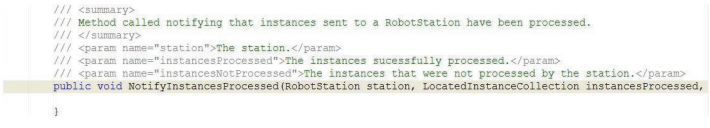
        instancesToAllocate.Add(locatedInstance);
        allocatedCount[locatedInstance.ObjectType]++;

    }
}
```

■ Notify Instance Processed

After an instance is sent to a RobotStation, the robot will either respond indicating that the instance was processed successfully or the instance was not processed. For example, if a belt-based instance was picked by the robot, it would respond to the PC that the instance was successfully processed. However, there are times where the robot was not able to pick all the instances on the conveyor because there are too many given the belt speed. In this case, the instances are robot responds to the PC that those instances were not processed.

As robots report which instances are processed and not processed, the NotifyInstanceProcessed method is called. An example is shown in the following figure.



```
/// <summary>
/// Method called notifying that instances sent to a RobotStation have been processed.
/// </summary>
/// <param name="station">The station.</param>
/// <param name="instancesProcessed">The instances sucessfully processed.</param>
/// <param name="instancesNotProcessed">The instances that were not processed by the station.</param>
public void NotifyInstancesProcessed(RobotStation station, LocatedInstanceCollection instancesProcessed,

}
```

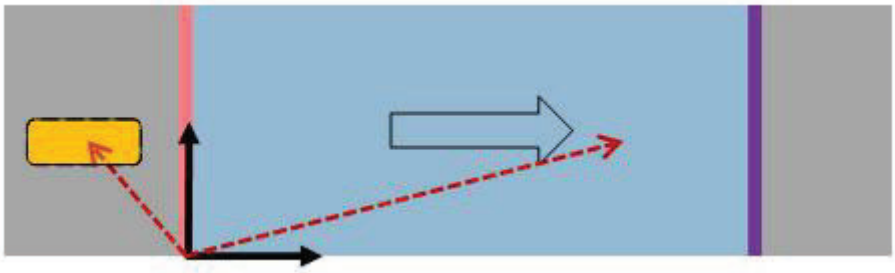
■ Located Instance

The LocatedInstance class represents an instance that can be processed by a robot. There are many properties which can be useful when defining if an instance should be processed. These properties include:

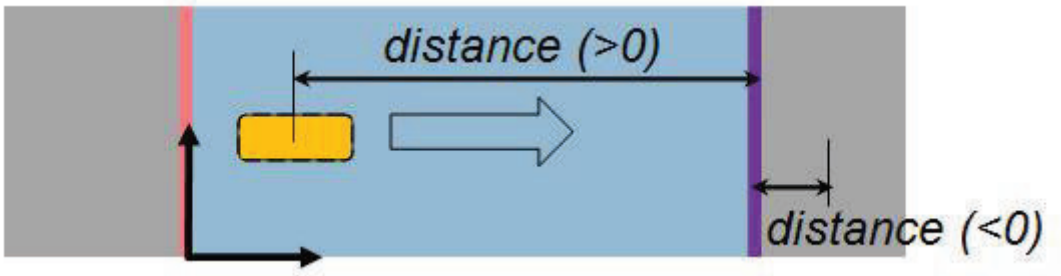
- BeltRelativePosition
- DistanceToPickLine
- DistanceToUpstreamWindow
- Handler

- Location
- ObjectType
- PalletIndex
- PalletInstance
- GetControllerLatchPosition

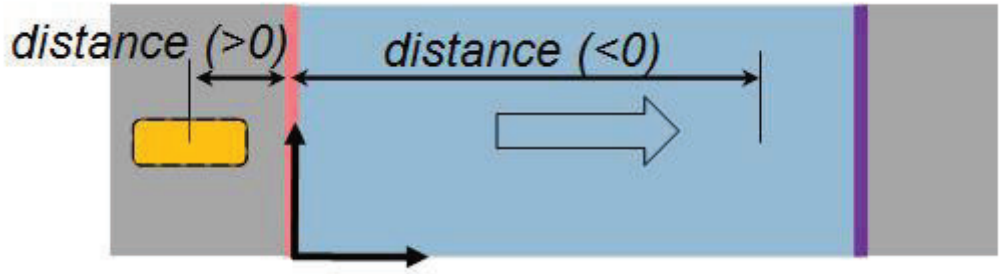
The BeltRelativePositon property gives the position of the Instance relative to the belt frame and varies with time. A visual example is shown in the following figure.



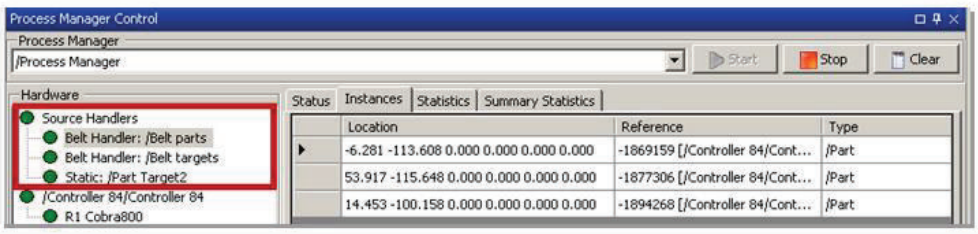
The DistanceToPickLine property gives the distance between the instance and the pick line limit and varies with time. A visual example is shown in the following figure.



The DistanceToUpstreamWindow property gives the distance the upstream line and the instance and varies with time. A visual example is shown in the following figure.



The Handler gives the source handler the generated instance. A visual example is shown in the following figure.



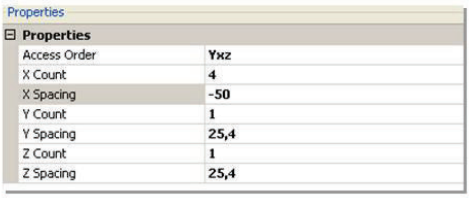
The Location property gives the location of the instance in its locator frame.
The ObjectType gives the name of that object type as a string, as shown in the following figure.

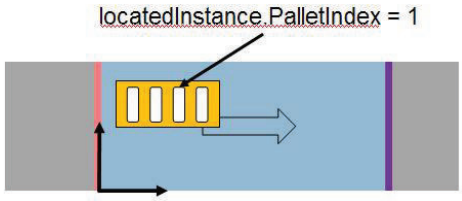
```
Part myPart = (Part) ace["/process1/Part"];
PartTarget myTarget = (PartTarget) ace["/process1/Part Target"];

if (locatedInstance.ObjectType == myPart.FullPath)
    ace.AppendToLog("this is my part");

if (locatedInstance.ObjectType == myTarget.FullPath)
    ace.AppendToLog("this is my target");
```

The PalletIndex property is used only if the instance relates to a pallet and it defines the index of the pallet slot. This is shown in the following figure.





The PalletInstance is used only if the instance is related to a pallet and has a Global Unique Identifier (GUID) of the pallet associated with this instance

There are also several methods that can be used in the process of allocating instances, such as GetControllerLatchPosition: which gets the latched value of the instance (reference).

■ LocatedInstance Tag Property

There is one additional property that can be very useful when synchronizing a custom allocation script with a C# custom vision tool and/or robot behavior in V+. When a custom vision tool executes, a VisionTransform collection is returned as an output of the Main method. The VisionTransform class has a Tag property which is of type object. Use this property to tag a vision instance in the custom vision tool and retrieve this tag from an allocations script.

For example, the custom vision tool might look like the following figure.

```
foreach (BlobResult res in myBlob.Results){

    // declare a List used to tag the instance
   List<float> myTag = new List<float>();
   // fill that List
   myTag.Add(res.Area);
   myTag.Add(res.GreyLevelMean);
   // tag the vision result
   res.Position.Tag = myTag;
   // add the vision result in Results of the custom vision tool
   results.Add(res.Position);
}
```

From within the allocation script, access the information as shown in the following figure.

```
List<float> tagInstance = (List<float>) locatedInstance.Tag;
float instanceArea = tagInstance[0];
float instanceMeanGrey = tagInstance[1];
```

Additionally, if the tag is defined for a LocatedInstance, the tag will be converted to a string representation and send to V+ as part of the instance information. From within a custom motion sequence program, the tag can be extracted as shown below.

```
.PROGRAM cu.mv.sequence(tsk.idx, $type, blt.idx, reference, pal.idx, grip.idx, grip.state, pos, vals[], sts)

; ABSTRACT:  Perform a move defined by a motion sequence.
;
; INPUTS:       tsk.idx         Index of the task
;               $type           Type of item being operated on
;               blt.idx         Belt number to move to
;                                 0   = Non-belt relative move
;                                 > 0 = Belt relative move
;               reference       Encoder/reference position associated
;                                 with the belt.
;               pal.idx         Instance pallet index
;               grip.idx        Gripper index to use
;               grip.state      Expected gripper state
;               pos             Location of the object/type
;               vals            The motion configuration array
;
; OUTPUTS:      sts             Status of the operation
;                                 pm.tsk.success = Successful operation
;                                 pm.tsk.retry   = Retry the operation
;                                 pm.tsk.skip    = Skip the operation
;                                 pm.tsk.abort   = Abort the operation
;
; SIDE EFFECTS: None
;
;* Copyright (c) 2007-2010 by Adept Technology, Inc.

        AUTO REAL at.wait, belt.win.idx, code, distance, offset
        AUTO REAL resp, rob.num
        AUTO $cust.program, $tag
        AUTO LOC appro.pos, depart.pos, grip.trans, position

        sts = pm.tsk.success

        CALL pm.rob.gettag(tsk.idx, $tag)
        TYPE "Tag = "+$tag

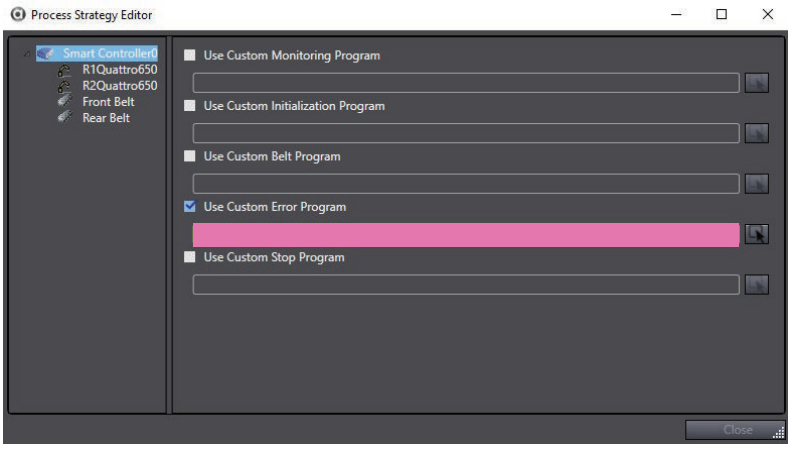
; Calculate the position, factoring in the gripper transformations

        ; Initiate the gripper selection
```

V+ Error Customization

When an error is generated on a controller, a single program on the robot is called to handle the error. This program makes a determination if an error is to be send to the PC to display or if the error should be automatically handled without user notification.

This program can be customized on a controller by controller basis using the GUI shown in the following figure.



When the program is customized, the user will start with a copy of the default error handling program. The following figure is a copy of the default program.



```
.PROGRAM pm.psp.error(tsk.idx, code, code2, resp.mask, #stack[], resp)

; ABSTRACT:  Report errors to the PC. This program can filter
;       errors and change the reported code or handle errors
;       without reporting to the user interface.
;
; INPUTS:       tsk.idx         Task index for error reporting
;               code            Error code/status of the operation
;               code2           Variable portion of error code.
;               resp.mask       Available responses
;               #stack[]        V+ program stack information.  Indices
;                                 of 0 and 1 are valid.
;
; OUTPUTS:      resp            Possible error responses:
;                                 pm.tsk.success = The operation has
;                                     completed successfully.
;                                 pm.tsk.skip = Stop processing current
;                                     instance and move to next operation.
;                                 pm.tsk.retry = Retry processing of the
;                                     current instance.
;                                 pm.tsk.abort = Stop processing the
;                                     current process and clear the gripper.
;                                 pm.tsk.next = Move to the next instance
;                                     in the queue.
;                                 pm.tsk.fail = The operation has
;                                     not completed due to an error.
;
; SIDE EFFECTS: None
;
;* Copyright (c) 2007-2011 by Adept Technology, Inc.

        AUTO REAL is.grip.error

; If a belt window violation occurs, return with the response
; indicating the operation will be aborted.

        IF code == pm.err.blt.viol THEN
            resp = pm.tsk.abort

            ; To supress the movement to the idle position
            ; and clearing of the gripper, set the gripper
            ; clear state to FALSE by uncommenting the following
            ; line. When uncommented, the robot will stay at the
            ; current location and the gripper I/O will not be changed.
            ; CALL pm.rob.clear.st(tsk.idx, FALSE)

            GOTO 100
        END

; If an end effector error occurs, return with the response
; indicating the operation failed.

        is.grip.error = (code == pm.err.no.gr.cl) OR (code == pm.err.no.gr.op) OR (code == pm.err.inv.grip)
        is.grip.error = is.grip.error OR (code == pm.err.no.gr.ex) OR (code == pm.err.no.gr.rt)

        IF (is.grip.error == TRUE) THEN

        ; In the case of a gripper error, we report the pm.tsk.fail response.
        ; Part or target on which the error occured will be marked as failed.

            resp = pm.tsk.fail
            GOTO 100
        END


; Report the error

        CALL pm.error(tsk.idx, code, code2, pm.tsk.default, #stack[], resp)

   100  RETURN
.END
```

The error program will be called and passed the task number that has the error, the error code, and other details of the error. The program must pass back a status code the describes the response on how the error is to be handled in the resp parameter. The possible values are either Response or Description.

Code	Definition
pm.tsk.success	The operation has completed successfully.
pm.tsk.skip	Stop processing current instance as if it was properly processed and move to the next operation.
pm.tsk.retry	Retry processing of the current instance.
pm.tsk.abort	Stop processing the current process and clear the gripper.
pm.tsk.next	Stop processing the current instance and move to the next instance in the queue.
pm.tsk.fail	The operation failed.

The default program will also check to see if an error code indicates an error from the gripper operation. If this is true, the error program will automatically return a pm.tsk.fail response. Doing so, the part or target that was being processed will be marked as failed causing the system to remove it from queues and not pushing it to the next robot on the conveyor : indeed we assume it may have moved when we tried to process it.

If the error is not a belt window violation, the default program will call pm.error, which will send the error to display on the process manager runtime control. When called, the error is presented to the user and any response will be returned.

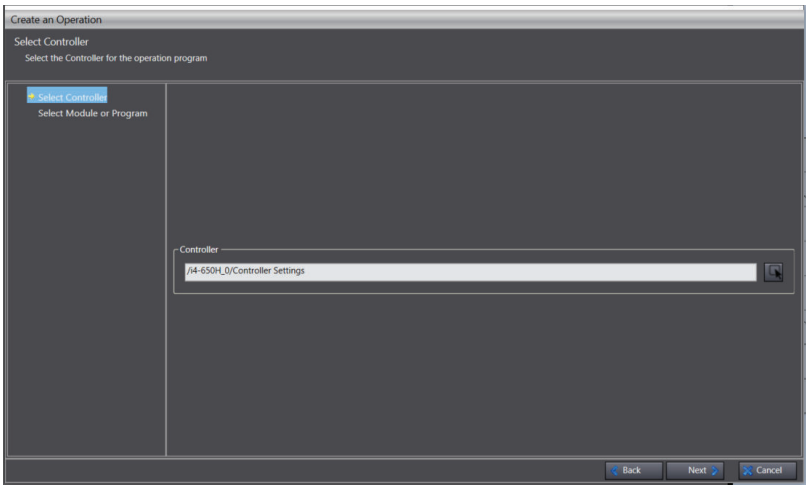
Some applications may want to display errors in a different way. For example, if a user wants to display an error using an I/O panel with lights and monitor for button presses, the application should not call the pm.error program. Rather, it should turn on I/O signals to indicate an error has been detected and then monitor for the press of a button through digital input points.

8-7-10 Process Operation Object

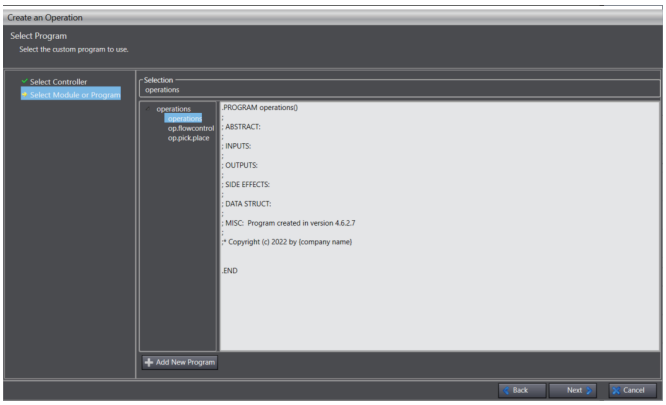
The Process Operation object adds function programs to the process manager.

Adding a Process Operation Object

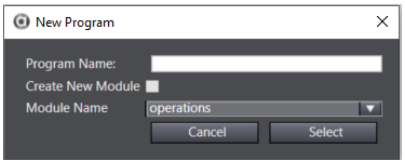
To add a Process Operation object, right-click **Process**, select **Add**, and then click **Process Operation**. A Create an Operation window opens. Select the controller the operation will apply to, and click **Next**.



Accept the default option, or use the selection button to choose another controller, then click **Next**. In the next panel, select the available program.



Click the **Add** button to name and select a program. Refer to the *V+ User's Manual (Cat. No. I671)* for more information about V+ program naming rules and requirements.



Once the available program is selected, click the **Next** Button to complete the process operation addition.

8-8 Vision Tools

ACE software provides an extensive set of vision tools for basic and complex applications.
These tools can perform the following functions.

- Image processing and filtering.
- Coarse and fine location.
- Position refinement.
- Inspection.
- Text.
- Code reading.

Each tool requires configuration to determine what data to collect, how to interpret the data, and the location of that data within the field of view. ACE software provides the ability to specify this information, either by making menu selections or by using the mouse to manipulate the tools directly in the Vision Window.
Tools are categorized as follows.

- Finder Tools

Finder Tools create a vectorized description of objects or object features and typically return coordinate results. These are used to identify features in image sources and provide locations of objects to be picked.

- Inspection Tools

Inspection Tools are often used in conjunction with Finder Tools to inspect objects that have been located. They rely on the analysis of pixel information and are designed to check various aspects of a detected object or feature, such as color deviation, defects, or product density.

- Reader Tools

Reader Tools are used to return character text string data from codes and text in an image.

- Calculation Tools

Calculation Tools allow the creation of new entities in an image that can be user-defined or based on existing entities.

- Image Process Tools

Image Process Tools provide various operations and functions for the analysis and processing of images.

- Custom Tools

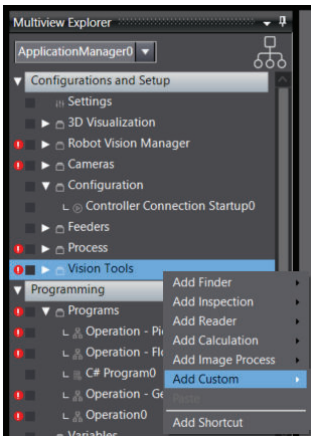
Custom Tools allow the user to more directly control the way an image or tool is processed.
The following table shows all vision tools provided in the ACE software, their respective categories, and a brief summary of their functions.

Category	Tool	Description
Finder Tools	Arc Finder	Identifies circular features on objects and returns the coordinates of the arc center, the angle of separation between the two ends, and the radius.
	Blob Analyzer	Processes information within a region of interest to identify groups of similarly-colored pixels, called blobs, and returns the position and size of each.
	Gripper Clearance	Uses histograms to identify regions of clearance around detected objects.
	Gripper Clearance 3D	Uses histograms to identify clearances within a three dimensional space.
	Labeling	Searches the image for areas of a user-defined color or color range and returns the results. Multiple colors or ranges can be searched.
	Line Finder	Identifies linear features on and returns the endpoint coordinates and line angle.
	Locator	Identifies instances of objects defined by a Locator Model within a region of interest. This tool is customizable and accurate, but slower than Shape Search3.
	Locator 3D	Identifies objects based on specified geometry after a specific part 3D model has been uploaded into the Locator Model.
	Locator Model	Defines the geometry of a model used by a Locator tool to identify instances.
	Shape Search3	Identifies instances of objects defined by a Shape Search3 Model within regions of interest. This tool is fast, but cannot be customized as much as the Locator tool.
	Shape Search3 Model	Defines the geometry of the model used by a Shape Search3 tool to identify instances.
Inspection Tools	Arc Caliper	Identifies one or more edge pairs in an arc-shaped or circular area and measures distance between the edges of each pair.
	Arc Edge Locator	Identifies an edge or set of edges that meet user-defined criteria within an arc-shaped or circular area.
	Caliper	Identifies one or more parallel edge pairs and measures distance between the edges of each pair.
	Color Data	Finds the average color within a region and analyzes its color variation and deviation from a specified reference color.
	Edge Locator	Identifies an edge or set of edges that meet user-defined criteria within a rectangular region.
	Feeder Histogram	Calculates product density in specified regions of interest. This tool is often used with the AnyFeeder object in regions associated with the dispense, pick, and front zones.
	Image Histogram	Computes gray-level statistics within a specified region of interest.
	Inspection	Judges the results of a tool and determines if they fall within a series of user-defined categories and filters.
	Precise Defect Path	Performs differential processing on an image to detect defects and contamination on a line or along an edge.
	Precise Defect Region	Performs differential processing on an image to detect defects and contamination within an area.

Category	Tool	Description
Reader Tools	Barcode	Reads bar codes in an image and acquires text string data.
	Data Matrix	Reads data matrices in an image and acquires text string data.
	OCR	Identifies text characters in image and returns them as text data. Custom characters can also be registered to an OCR Dictionary for future identification.
	OCR Dictionary	Stores dictionary data that OCR can use to identify text characters.
	QR Code	Reads QR or Micro QR code in an image and acquires text string data.
Calculation Tools	Calculated Arc	Calculates an arc based on a specified calculation mode and returns the encompassing circle.
	Calculated Frame	Calculates a frame based on a specified calculation mode and returns coordinates and orientation.
	Calculated Line	Calculates a line based on a specified calculation mode and returns coordinates and dimensions.
	Calculated Point	Calculates a point based on a specified calculation mode and returns coordinates.
Image Process Tools	Advanced Filter	Filters or alters an image using one of a variety of filter libraries, such as Background Suppression, Erosion / Dilation, and Color Gray Filter. This tool can be combined with other tools (including other Advanced Filters) to modify an image to the necessary specifications.
	Color Matching	Filters and analyzes areas of a specified color or color range in images.
	Image Processing	Filters or alters a gray scale image using one of a variety of filters, including logical and arithmetic calculations. This tool can be combined with other tools (including other Image Processing tools) to modify an image to the necessary specifications.
	Image Sampling	Extracts an area of an image and outputs it as a new image source.
	Position Compensation	Converts all pixels outside a rectangular region to black and outputs the result as a new image.
Custom Tools	Custom Vision Tool	Allows specifying a program to be called when executed. This tool can execute other tools and return a custom set of results.
	FH Tool	Enables configuration and calibration of an FH-series 2D vision system connected to ACE.
	Remote Vision Tool	Enables operation of configured vision tools by creating a vision server instance.

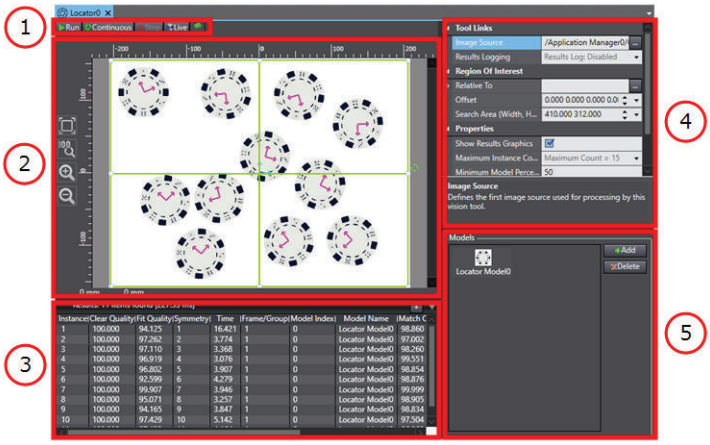
8-8-1 Adding Vision Tools

To add a new vision tool, right-click on the **Vision Tools** object in the Multiview Explorer, expand one of the categories and then select one of the tools, as shown in the figure below.

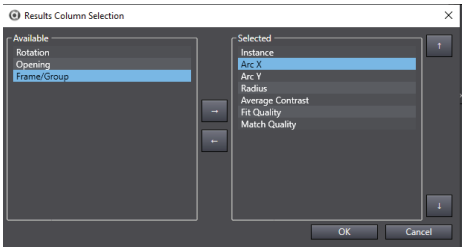


8-8-2 Vision Tool Editor

Each vision tool is configured using its corresponding object editor in the Edit Pane. In general, most of the tool editors have a similar configuration that can be split into five sections as described below.



Item	Name	Description
1	Execution Buttons	This area provides direct control of the tool with the following buttons. • **Run**: Run the tool once. This is available with most vision tools. • **Continuous**: Run the tool continuously. This is available with most vision tools. • **Stop**: Stop the tool or live operation. Can only be selected if Live or Continuous has been activated. This is available with most vision tools. • **Live**: Send images without running the tool. This is available with most vision tools. • **Auto Tuning**: Automatically sets parameters based on pixel data. This functions differently depending on the tool. See those tools for more information. This is available with the Barcode, Color Data, Data Matrix, OCR, and QR Code tools. • New Image button: Loads a new image into the tool. The model will need to be retrained. This replaces the Run button in the respective tools. This is available with the Locator Model and Shape Search 3 Model tools. • **Register to OCR Dictionary** button: Registers a character to the OCR Dictionary. This is available with the OCR Dictionary tool.

Item	Name	Description
2	Image Display	This area displays the current image from the camera. This will also include any required graphics or controls. For example, the Locator tool displays markers for each identified instance within the green region of interest, which can be modified as needed.
3	Results	This area displays the output of the tool . In most cases, this contains returned data organized in columns. The columns can be chosen by clicking the **Plus** button () which will display the *Results Column Selection* Dialog Box. Use this dialog box to select and arrange the data displayed in this area.  If the tool is in the Image Process category, it returns an image instead of data. The Results section of the tool editor will either show the result image, such as in Advanced Filter, or be nonexistent, such as in Color Matching. In the latter case, the Vision Window must be opened to view the resultant image.
4	Properties	This area displays the tool parameters. The parameters can be modified to change tool operation.
5	Additional Information Pane	This area provides any additional information about the tool. This is only found in some of the tools and generally utilizes something specific to only that tool.

8-8-3 Region of Interest

Most tools use regions of interest to define the location where the tool will be executed. Some tools allow multiple regions based around a single reference point, but most use a single rectangular region in which to execute the operation. In both cases, the region or regions are outlined in green in the tool Vision Window.
Regions of interest can be modified in two ways:

1 Clicking and dragging the green outline or its nodes. Dragging the nodes will re-size the region while dragging the outline itself will translate it. In some tools, the regions have a node for rotation.

2 Adjusting the parameters in the properties. All tools with regions of interest have a Region Of Interest section in the properties that governs the size, location and, in some cases, behavior of the region. The location and size of the region are typically governed by the Offset and Search Region properties, but the property names may vary.



Additional Information

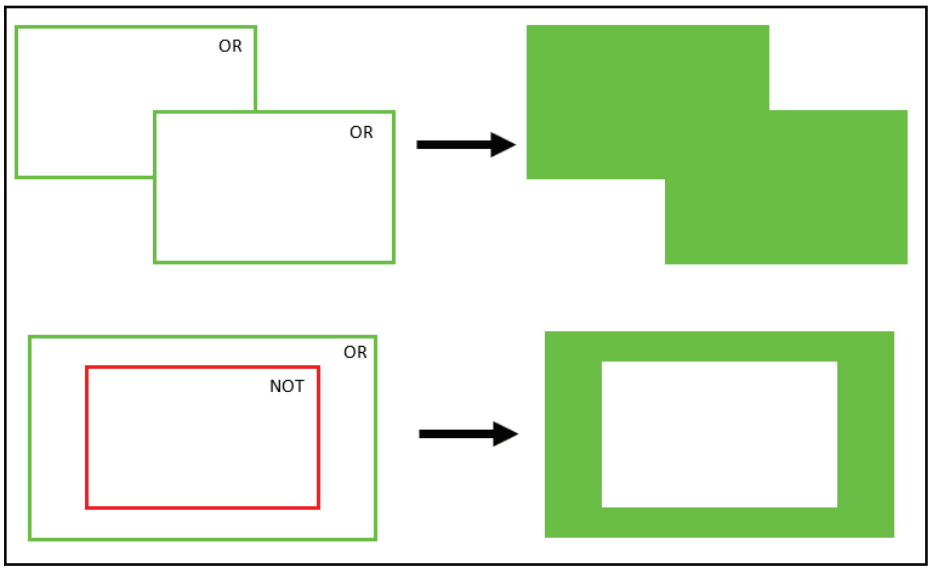
The region orientation of some tools can only be controlled with the Offset property.

Tools with Multiple Regions of Interest

Tools with multiple regions of interest can be manipulated to allow different behavior in individual regions. This is achieved by modifying two properties as described below.

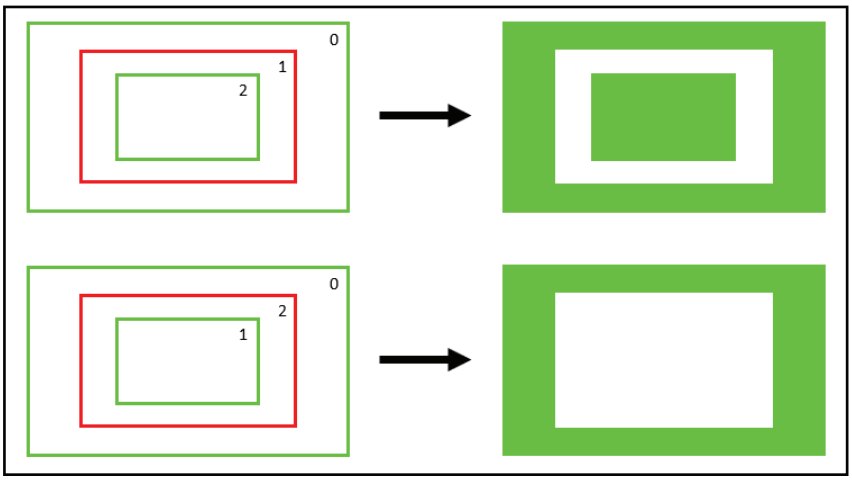
Overlap

This defines the region as either inclusive or exclusive by setting it to OR or NOT, respectively. An inclusive region returns all detected contours or instances within its boundaries. An exclusive region hides all detected contours or instances within its boundaries. The following figure shows how individual regions are registered in the figure. The original regions are displayed on the left and the resulting processed region is displayed on the right, where the green area shows what parts are read. For example, in the second set of regions, the NOT region eliminates a section of the full region, resulting in a rectangular section that is not read.



Z-Order

This defines the order in which regions are processed. Each region has its own property that defines its order in the Z-direction of an image. Since the image is two-dimensional, this value is used to determine which regions are processed first. Regions are processed in ascending order. This can be seen in the regions in Figure 1-4where the regions are processed according to their written Z-Order. In the first example, the results of all three regions can be seen, since no region is entirely blocked. However, in the second example, the smallest region is hidden because the red region has a higher value than it does, according to Z-order.



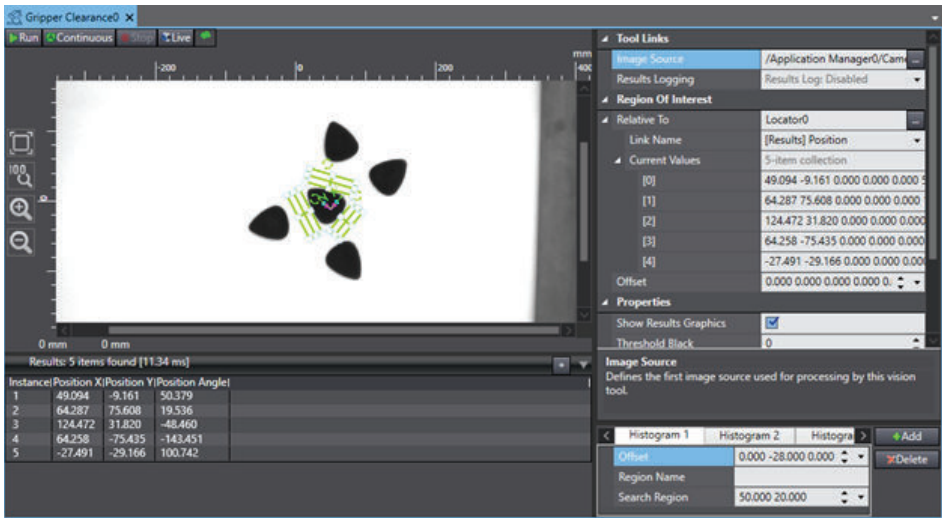
Relative To Parameter Details

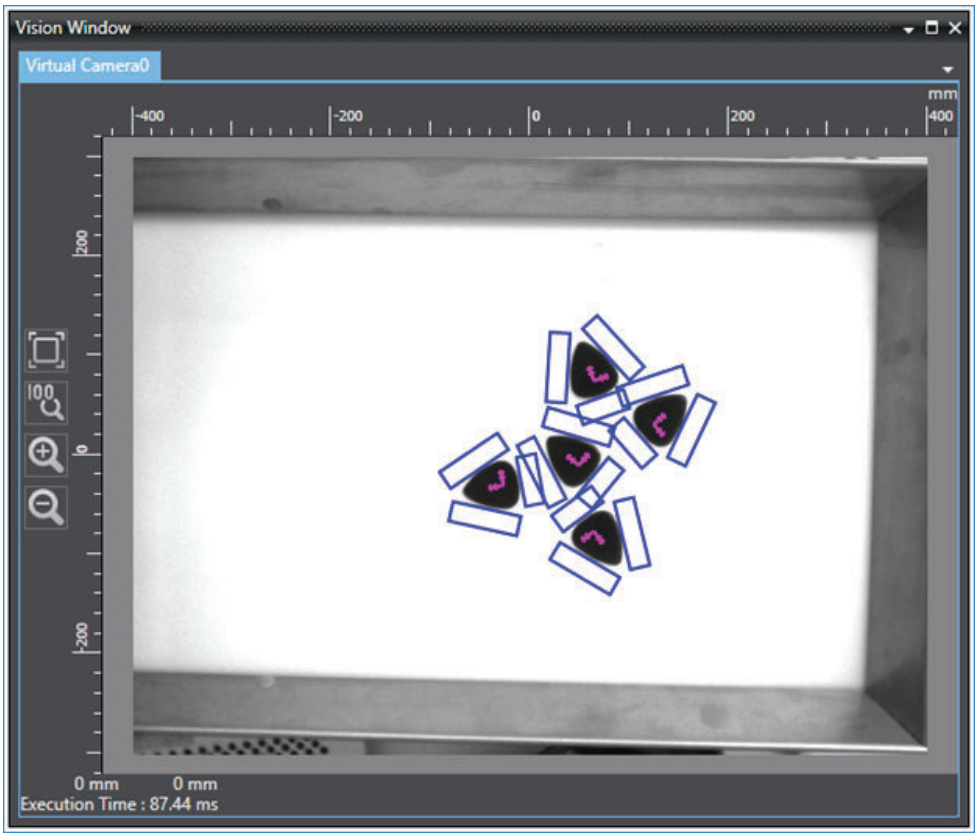
Most of the tools with regions of interest can be set relative to another tool. A vision tool is ordinarily executed with respect to the origin of the image, but when it is set relative to another tool, it will instead execute with respect to the result values of that tool. This will cause the secondary tool to execute once for every input value, unless it is otherwise constrained.

To create this relationship, set the primary tool as the Relative To property of the secondary tool in the editor of the secondary tool. In this way, the output values of the primary tool become the input values of the secondary tool. In the following figure, the Gripper Clearance tool is set relative to a Locator tool and is able to position histograms around all of the objects by creating new reference points with respect to the Locator results. The input locations are shown under the Current Values section in the properties.

Some tools display all instances in the image display. Some only display the region created with respect to the first instance and return the remainder in the Results section. The additional instances can be viewed in the ACE Vision Window as long as the property *Show Results Graphics* is enabled by selecting the check box in the *Properties* panel.

A tool set relative to another tool can be used to create a Robot Vision Manager Sequence. Refer to *8-3-7 Vision Sequence* on page 8-27 for more information about creating a Robot Vision Manager Sequence.





8-8-4 Color Spaces

The term color space refers to numeric values (or percentages) of the visible color spectrum, specifically organized to allow both digital and analog representations of color. A color space is a spectrum or range of colors with specific values.
This section describes color spaces, color values, and how to define colors by those values.

HSB and RGB Color Spaces

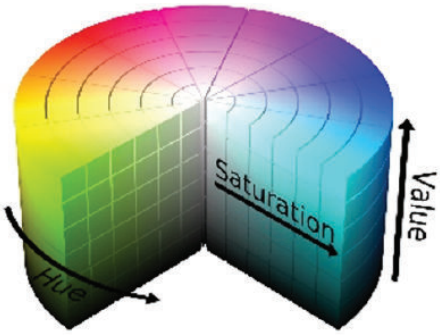
The two different types of color spaces are described below.

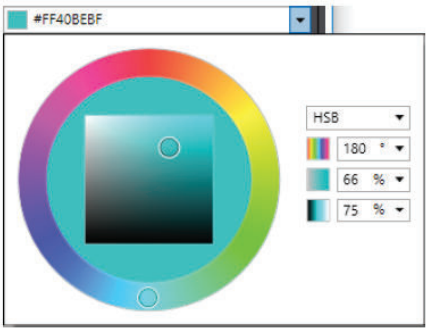
● Hue, Saturation, and Brightness (HSB) Color Space

The HSB color space arranges colors of each hue radially around a central axis of basic colors, from white at the top to black at the bottom. Hue values are set in degrees from 0 to 360. Saturation and brightness are set in percentages from 0 to 100%.
Hue is the quality of color perceived as the color itself. The hue is determined by the perceived dominant wavelength, or the central tendency combined wavelengths within the visible spectrum.
Saturation is the purity of the color, or the gray in a color. For example a high saturation value produces a very pure, intense color. Reducing the saturation value adds gray to the color.
Brightness is the amount of white contained in the color. As the value increases, the color becomes lighter and becomes more white. As the luminance value decreases the color is darker and becomes more black.

Additional Information

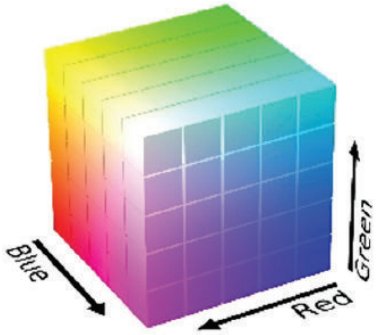
HSB is also referenced as HSL (Hue, Saturation, Luminance) and HSV (Hue, Saturation, Value) in the ACE software.

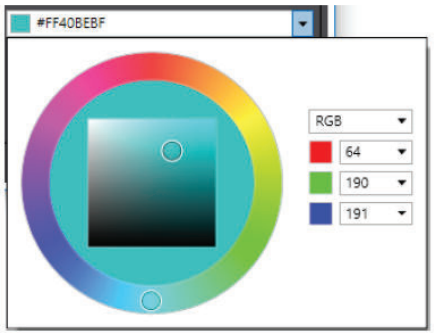




● Red, Green, and Blue (RGB) Color Space

The RGB color space uses combinations of red, green, and blue to create all colors. Red, green, and blue values are expressed with a range of 0 to 255.





Additional Information

ACE software also accepts the hexadecimal color value in the color input field.

● HSB and RGB Color Values

Settings for items such as color filters and reference colors can be adjusted with HSB, RGB, or hexadecimal values. The following table provides example values for common colors.

Color	HSB Values	RGB Values	Hexadecimal Values
White	0, 0, 100	255, 255, 255	#FFFFFF
Black	0, 0, 0	0, 0, 0	#000000
Gray	0, 0, 50	127, 127, 127	#7F7F7F
Red	0, 100, 100	255, 0, 0	#FF0000
Green	120, 100, 100	0, 255, 0	#00FF00
Blue	240, 100, 100	0, 0, 255	#0000FF

● Color Tolerances

Color tolerances can be applied to allow for slight color variations. Color tolerances can only be set with HSB values.

A color tolerance value is distributed equally above and below the color value to which it applies. For example, if the color hue value is 50 and the hue tolerance value is 20, the filter will accept colors within a range of hue values from 30 to 70.

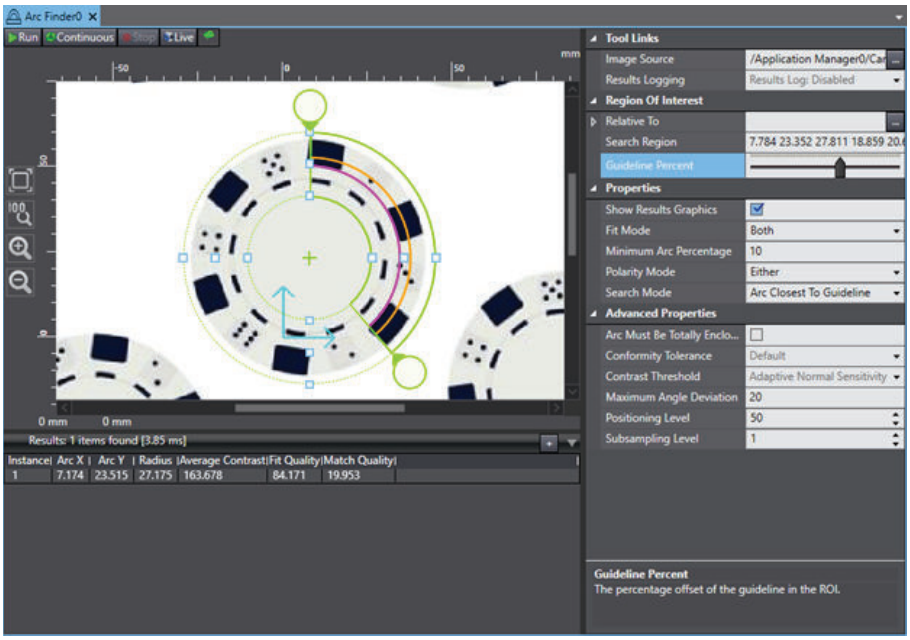
8-8-5 Finder Tools

Finder tools are used to identify objects and create detection points for location guidance.
The following Finder tools are described in this section.

- Refer to *Arc Finder* on page 8-199 for more information.
- Refer to *Blob Analyzer* on page 8-202 for more information.
- Refer to *Gripper Clearance* on page 8-208 for more information.
- Refer to *Labeling* on page 8-211 for more information.
- Refer to *Line Finder* on page 8-217 for more information.
- Refer to *Locator* on page 8-219 for more information.
- Refer to *Locator 3D* on page 8-232 for more information.
- Refer to *Locator Model* on page 8-233 for more information.
- Refer to *Shape Search 3* on page 8-239 for more information.
- Refer to *Shape Search 3 Model* on page 8-244 for more information.

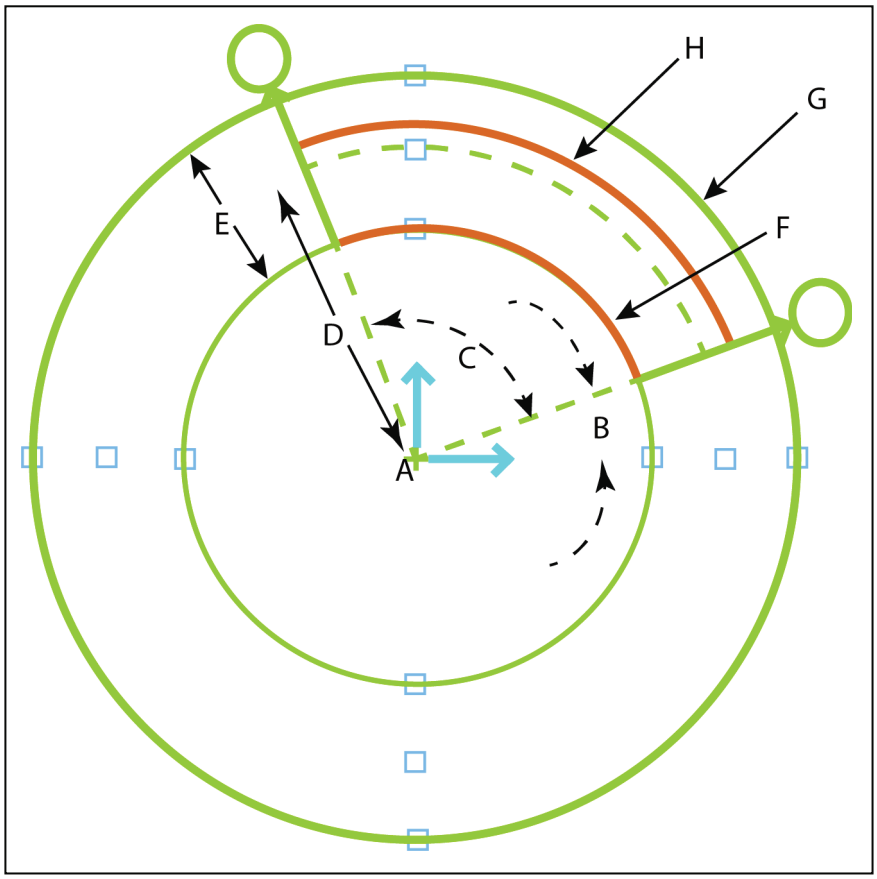
Arc Finder

This 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.
Arc Finder is most commonly used to return information about round edges or refine the orientation of a located object. For example, the tool in the following figure identifies the arc created by trapezoidal features on the chip. In this way, it can be used to locate circular patterns or shapes within an object. If the arc to be located should only be in a certain position, the guideline position and properties such as Search Mode can be adjusted to specify the desired range of the location.



To create an Arc Finder tool, right-click **Vision Tools** in the Multiview Explorer and select **Add Finder** and then **Arc Finder**. An Arc Finder tool will be added to the Vision Tools list.

The following figure identifies the specific segments of the Arc Finder tool, shown in the Editor.



Indicator	Description
A	Center
B	Rotation
C	Opening
D	Radius
E	Region of Interest Thickness
F	Inner Annular Bound
G	Outer Annular Bound
H	Guideline

● Arc Finder Configuration Items

Use the table below to understand the Arc Finder tool configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.

Group	Item	Description
Region of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Search Region	Defines the location and size of the region (X, Y, radius, thickness, mid-angle position, arc angle degrees).
	Guideline Percent	Adjusts the location of the orange guideline in the region of interest. Move the slider left to position the guideline closer to the center and to the right to position it farther away.
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Fit Mode	Select how the tool will calculate and return a valid arc from hypotheses. • Both: calculates the arc center and arc radius and uses both values to return the most accurate results. • Center: calculates the arc center. • Radius: calculates the arc radius.
	Minimum Arc Percentage	Specifies the minimum percentage of arc contours that need to be matched for a hypothesis to be considered valid.
	Polarity Mode	Select the polarity deviation required for a hypothesis to be considered valid. • Dark to Light: only detect arcs that are darker towards the center. • Light to Dark: only detect arcs that are lighter towards the center. • Either: accept any polarity as long as it is consistent. • Do Not Care: does not take polarity into account.
	Search Mode	Select the method used to generate and select a hypothesis. • Best Arc: chooses the best quality arc but will increase processing time. • Arc Closest To Guideline: chooses the arc closest to the guideline. • Arc Closest To Inside: chooses the arc closest to the inner circle. • Arc Closest To Outside: chooses the arc closest to the outer arc of the region of interest.
Advanced Properties	Arc Must Be Totally Enclosed	Specifies if the detected arc can exist outside of the region of interest. When enabled, the start and end points of the arc must be located on the angle boundary lines. Otherwise, the arc can enter or exit the region at any point.
	Conformity Tolerance	Set the maximum local deviation between the expected arc contours and the arc contours actually detected in the input image.
	Contrast Threshold	Defines the minimum contrast needed for an edge to be detected in the input image. This threshold is expressed in terms of a step in gray level values. Higher values reduce noise but also reduce the number of contours detected.
	Maximum Angle Deviation	Defines the maximum angle deviation between the calculated arc's tangents and the edge contours.
	Positioning Level	Set the configurable effort level of the instance positioning process. A value of 10 will provide coarser positioning and lower execution time. Conversely, a value of 100 will provide high accuracy positioning of arc entities. The setting range is 10 to 100.
	Subsampling Level	Set the subsampling level used to detect edges that are used by the tool to generate hypotheses. High values provide a coarser search and lower execution time than lower values. The setting range is 1 to 8.

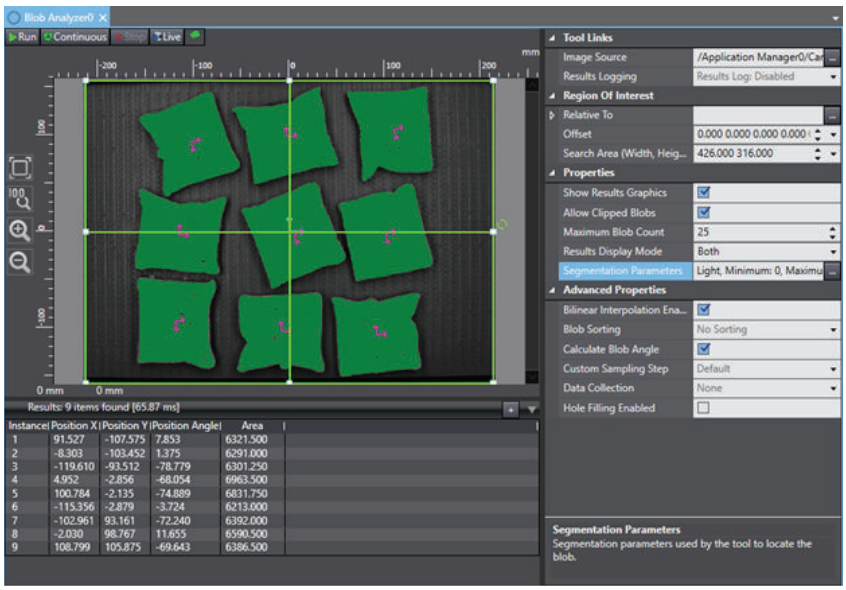
● Arc Finder Results

Use the table below to understand the results of the Arc Finder tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Arc X	X-coordinate of the arc center point.
Arc Y	Y-coordinate of the arc center point.
Radius	Radius of the arc, measured from the center determined by Arc X and Arc Y.
Opening	Angle (in degrees) measured between the two arc endpoints.
Rotation	Rotation of the region of interest, measured from the positive X-axis.
Average Contrast	Average contrast between light and dark pixels on either side of the arc, expressed in gray level values.
Fit Quality	Normalized average error between the calculated arc contours and the actual contours detected in the input image. Fit quality ranges from 0 to 100 where the best quality is 100. A value of 100 means that the average error is 0. A value of 0 means that the average matched error is equal to the Conformity Tolerance property.
Match Quality	Amount of matched arc contours for the selected instance expressed as a percentage. Match quality ranges from 0 to 100 where the best quality is 100. A value of 100 means that 100% of the arc contours were successfully matched to the actual contours detected in the input area.

Blob Analyzer

This tool uses pixel information within the region of interest to apply image segmentation algorithms for blob detection. A blob is any region within a gray scale image with a range of gray level values that differs from the adjoining areas of the region. The Blob Analyzer tool is primarily used to find irregularly-shaped objects.



To create a Blob Analyzer tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Finder** and then **Blob Analyzer**. A Blob Analyzer tool will be added to the Vision Tools list.

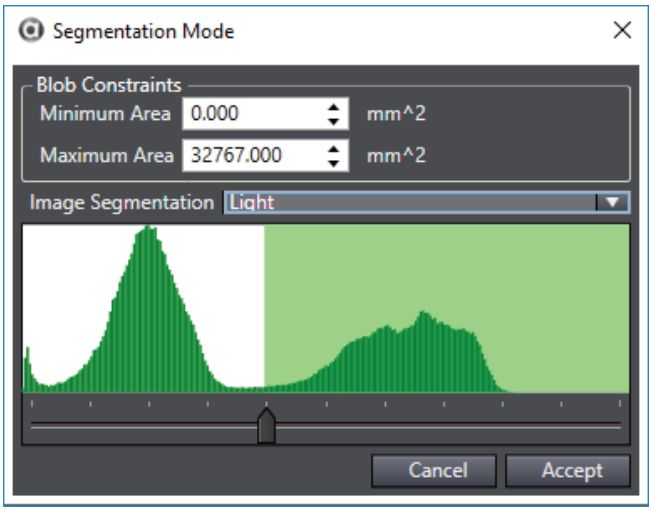
● Blob Analyzer Configuration Items

Use the table below to understand the Blob Analyzer tool configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the center coordinates of the region.
	Search Area (Width, Height)	Defines the size of the region.
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Allow Clipped Blobs	Enables the inclusion of blobs that have been cut off by the edge of the region of interest.
	Maximum Blob Count	Set the maximum number of blobs that the tool is able to return.
	Results Display Mode	Defines how results are rendered in the image display. • Marker: shows only the centroid marker of the blob. • Blob Image: shows the blobs highlighted with green. • Both: displays the centroid marker and the blobs.
	Segmentation Parameters	Properties used by the tool to locate the blob. Refer to *Blob Analyzer Segmentation Mode Editor* on page 8-203 for more information.
Advanced Properties	Bilinear Interpolation Enabled	Enables using bilinear interpolation to sample the input image. By default, this is enabled because it ensures subpixel accuracy.
	Blob Sorting	Select the order in which the blobs are processed and output. Most sorting options use the values in a specific result column. This is disabled by default.
	Calculate Blob Angle	Enables the calculation of each blob angle. This is enabled by default. The angle is calculated by collecting four additionally properties if they are not already calculated: Inertia Minimum, Inertia Maximum, Elongation, and Position Angle.
	Custom Sampling Angle	Defines the sampling step used in calculation. This is set to 1 by default. Enable this to adjust the setting.
	Data Collection	Enables the calculation and collection of additional results. Refer to *Blob Analyzer Data Collection* on page 8-205 for more information.
	Hole Filling Enabled	Enables all background pixels within the perimeter of a given blob become included in the blob. Note that this effects the tool window but not the main Vision Window.

● Blob Analyzer Segmentation Mode Editor

The Segmentation Mode editor is accessed by clicking the ellipsis next to the *Segmentation Parameters* property. It controls the parameters that dictate which pixels are selected as blobs.

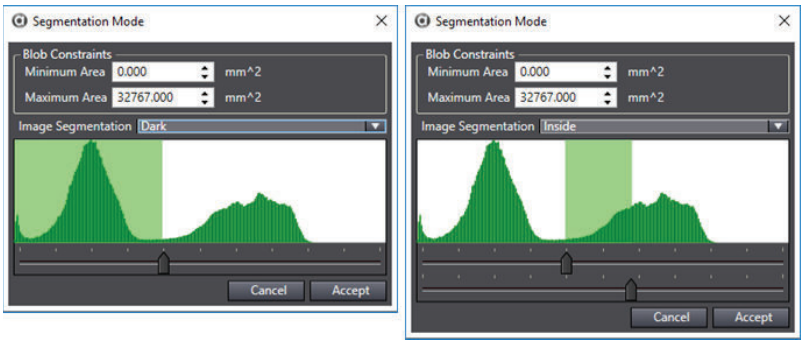


Blob Constraints determine the minimum and maximum area of a blob in calibrated units. This is useful for filtering out individual or small groups of pixels that are returned as their own blobs, but are not desired result instances.

Image Segmentation defines the method used to locate blobs. The following options can be selected.

- Light: Creates blobs from all pixels brighter than the set gray level boundaries.
- Dark: Creates blobs from all pixels darker than the set gray level boundaries.
- Inside: Creates blobs from all pixels inside the set gray level boundaries.
- Outside: Creates blobs from all pixels outside the set gray level boundaries.
- Dynamic Light: Creates blobs from the pixels on the brighter side of a percentage marker set by the user.
- Dynamic Dark: Creates blobs from the pixels on the darker side of a percentage marker set by the user.
- Dynamic Inside: Creates blobs from the pixels between two percentage bounds set by the user.
- Dynamic Outside: Creates blobs from the pixels outside of two percentage bounds set by the user.
- HSL Inside: Creates blobs from all the pixels that fall within an HSL tolerance.
- HSL Outside: Creates blobs from all the pixels that fall outside an HSL tolerance.

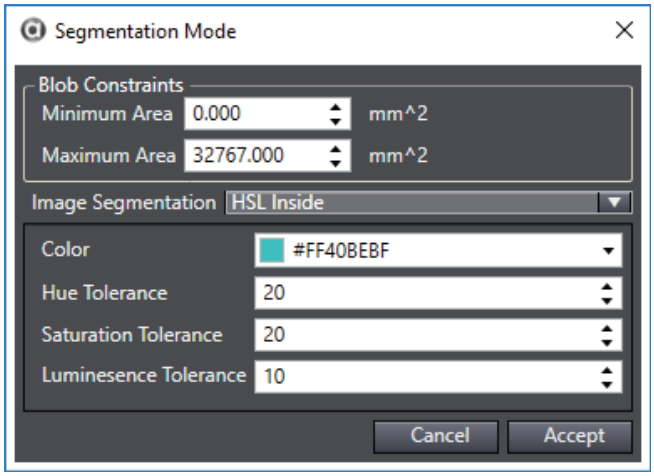
Image Segmentation settings are made using the slider(s) shown at the bottom of the editor (except for HSL Inside/Outside). The green area indicates which pixels in the histogram will be included in the blobs and the white area indicates which ones will not be included as shown in the following figure.



Additional Information

When a Dynamic option is selected in Image Segmentation, the display will continue to show the green and white overlay, but the calculation will not take the histogram into account.

HSL Inside and HSL Outside use a different viewer in the editor since they are based on HSL color instead of brightness. Therefore, the histogram has no impact. Instead, the editor appears as shown below and the histogram is replaced by the following properties. Refer to Color Spaces on page 1 for more information about color definition.



- Color: Defines the nominal color for which Blob Analyzer will search. Clicking the arrow will open an additional menu where the specific color can be defined.
- Hue Tolerance: Maximum difference between the nominal and pixel hue values for the pixel to be included in a blob.
- Saturation Tolerance: Maximum difference between the nominal and pixel saturation values for the pixel to be included in a blob.
- Luminescence Tolerance: Maximum difference between the nominal and pixel luminescence values for the pixel to be included in a blob.

Blob Analyzer Data Collection

To increase processing speed, Blob Analyzer provides adjustments to identify which specific results to calculate and collect. Only by selecting the necessary options in Data Collection can those results be properly calculated and displayed. For details on the results themselves, refer to the *Blob Analyzer Results* on page 8-206. The list below details the options. Any number of options can be selected.

- Chain Code Results: Refers to the sequence of direction codes that describes the boundary of a blob. Unlike the other options in this list, selecting Chain Code Results does not affect any results columns. It can be disabled for most applications.
- Extrinsic Inertial Results: Returns moments of inertia results. A moment of inertia measures 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 camera coordinate system.
- Gray Level Results: Returns information about the gray level distribution within the blob.
- Intrinsic Box Results: Returns information about the intrinsic bounding box, which is a bounding box that has been rotated to fit the edges of the blob as tightly as possible.
- Perimeter Results: Returns data regarding the perimeter.
- Topological Results: Returns the Hole Count result.

● Blob Analyzer Results

Use the table below to understand the standard results of the Blob Analyzer tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Position X	X-coordinate of the blob center of mass.
Position Y	Y-coordinate of the blob center of mass.
Position Angle	Calculated angle of the blob origin with respect to the X-axis. Only available when Calculate Blob Angle is enabled.
Area	Area of the blob in units defined by the workspace.
Elongation	The degree of dispersion of all pixels belonging to the blob around its center of mass. This is calculated as the square root of the ratio of the moment of inertia about the minor axis (Inertia Maximum) to the moment of inertia about the major axis (Inertia Minimum). Only available when Calculate Blob Angle is enabled.
Bounding Box Center X	X-coordinate of the center of the bounding box with respect to the camera coordinate system. Only available when Calculate Blob Angle is enabled.
Bounding Box Center Y	Y-coordinate of the center of the bounding box with respect to the camera coordinate system. Only available when Calculate Blob Angle is enabled.
Bounding Box Height	Height of the bounding box with respect to the Y-axis of the coordinate system. Only available when the angle is calculated.
Bounding Box Width	Width of the bounding box with respect to the X-axis of the coordinate system. Only available when the angle is calculated.
Bounding Box Left	X-coordinate of the left side of the bounding box with respect to the camera coordinate system. Only available when the angle is calculated.
Bounding Box Right	X-coordinate of the right side of the bounding box with respect to the camera coordinate system. Only available when the angle is calculated.
Bounding Box Top	Y-coordinate of the top side of the bounding box with respect to the camera coordinate system. Only available when the angle is calculated.
Bounding Box Bottom	Y-coordinate of the bottom side of the bounding box with respect to the camera coordinate system. Only available when the angle is calculated.
Bounding Box Rotation	Rotation of the bounding box with respect to the X-axis of the camera coordinate system.
Extent Left	Distance along the X-axis between the blob center of mass and the left side of the bounding box. Only available when Calculate Blob Angle is enabled.
Extent Right	Distance along the X-axis between the blob center of mass and the right side of the bounding box. Only available when Calculate Blob Angle is enabled.
Extent Top	Distance along the Y-axis between the blob center of mass and the top side of the bounding box. Only available when Calculate Blob Angle is enabled.
Extent Bottom	Distance along the Y-axis between the blob center of mass and the bottom side of the bounding box. Only available when Calculate Blob Angle is enabled.
Inertia Minimum	Moment of inertia about the major axis, which corresponds to the lowest moment of inertia. Only available when Calculate Blob Angle is enabled.
Inertia Maximum	Moment of inertia about the minor axis, which corresponds to the highest moment of inertia. Only available when Calculate Blob Angle is enabled.

Use the table below to understand the extrinsic inertial results of the Blob Analyzer tool.

Item	Description
Inertia X Axis	Moment of inertia about the X-axis of the camera coordinate system.
Inertia Y Axis	Moment of inertia about the Y-axis of the camera coordinate system.

Use the table below to understand the gray level results of the Blob Analyzer tool.

Item	Description
Gray Level Mean	Average gray level of the pixels belonging to the blob.
Gray Level Range	Calculated difference between the highest and lowest gray levels found in the blob.
Gray Std Dev	Standard deviation of gray levels for the pixels in the blob.
Gray Level Minimum	Lowest gray level found in the blob.
Gray Level Maximum	Highest gray level found in the blob.

Use the table below to understand the intrinsic box results of the Blob Analyzer tool.

Item	Description
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 Left	Leftmost coordinate of the bounding box with respect to the X-axis (major axis) or the principal axes.
Intrinsic Bounding Box Right	Rightmost coordinate of the bounding box with respect to the X-axis (major axis) or the principal axes.
Intrinsic Bounding Box Top	Topmost coordinate of the bounding box with respect to the Y-axis (minor axis) of the principal axes.
Intrinsic Bounding Box Bottom	Bottommost coordinate of the bounding box with respect to the Y-axis (minor axis) of the principal axes.
Intrinsic Bounding Box Rotation	Rotation of the intrinsic bounding box corresponding to the counterclockwise angle between the X-axis of the bounding box (major axis) and the X-axis of the camera coordinate system. Only available when Calculate Blob Angle is enabled.
Intrinsic Extent Left	Distance along the major axis between the blob center of mass and the left side of the intrinsic bounding box.
Intrinsic Extent Right	Distance along the major axis between the blob center of mass and the right side of the intrinsic bounding box.
Intrinsic Extent Top	Distance along the minor axis between the blob center of mass and the top side of the intrinsic bounding box.
Intrinsic Extent Bottom	Distance along the minor axis between the blob center of mass and the bottom side of the intrinsic bounding box.

Use the table below to understand the perimeter results of the Blob Analyzer tool.

Item	Description
Convex Perimeter	Perimeter calculated based on projections made at four different orientations: 0°, 45°, 90°, and 180°. The average diameter calculated from this projections is multiplied by pi to obtain these results.

Item	Description
Raw Parameter	Sum of the pixel edge lengths on the contour of the blob. This result is sensitive of to the blob's orientation with respect to the pixel grid, so results may vary greatly. Unless blobs are non-convex, Convex Perimeter results provide greater accuracy.
Roundness	Degree of similarity between the blob and a circle on a scale of 0 to 1, where 1 is a perfectly circular blob.

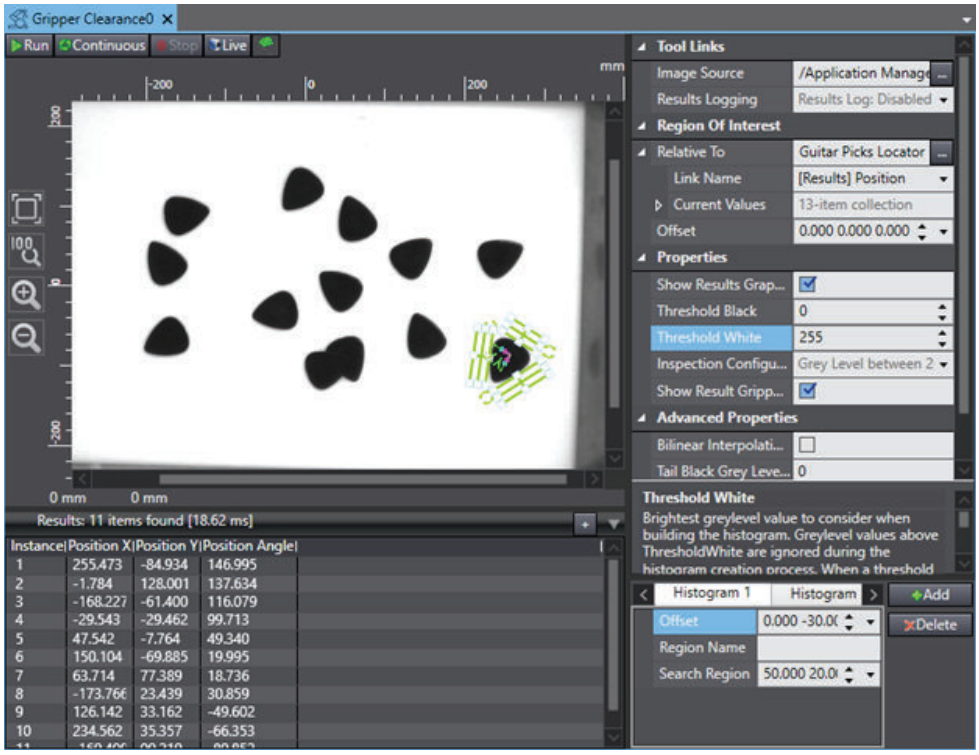
Use the table below to understand the topological results of the Blob Analyzer tool.

Item	Description
Hole Count	Number of holes found in the blob.

Gripper Clearance

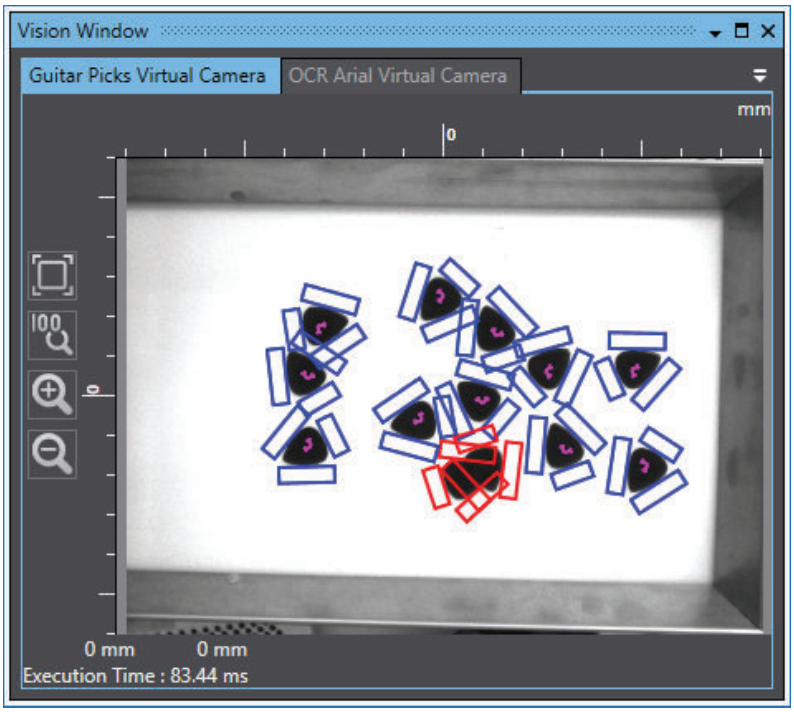
This tool uses histogram analysis to determine which parts can be picked without interference. It is configured as a series of rectangular histograms positioned around a part. The histograms are often set relative to a finder tool, such as Locator or Shape Search 3, so that they are positioned according to part locations.

Use the Gripper Clearance properties to define parameters that determine if the area around a part has clearance necessary for the gripper. These parameters are applied to the histograms to filter the parts. Instances passed through the filter can be picked by the gripper.



Clearance itself is determined by the Inspection Configuration property. This is configured by selecting one of the parameter options and experimentally determining an appropriate range. Note that a trial-and-error approach may be required to determine the correct values of the thresholds. If the results of the histogram fall within the defined range, then the area of that histogram region is assumed to be clear. All histograms of an instance must pass for the instance to be considered valid.

Passing instances are indicated with purple frames and blue histograms while failing instances are indicated with red histograms, as shown in the following figure.



To create a Gripper Clearance tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Finder** and then **Gripper Clearance**. A Gripper Clearance tool will be added to the Vision Tools list.

● Gripper Clearance Configuration Items

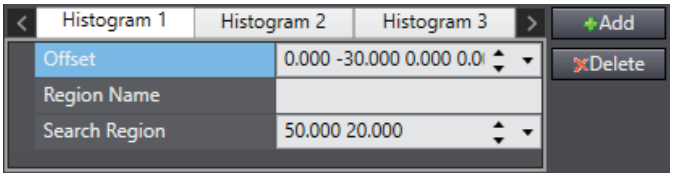
Use the table below to understand the Gripper Clearance tool configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the coordinates of the origin point referenced by the histograms.

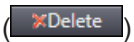
Group	Item	Description
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Threshold Black	Defines the darkest pixels the histogram considers when running. Any pixels with gray level lower than this value will be ignored while the histogram is building.
	Threshold White	Defines the brightest pixels the histogram considers when running. Any pixels with gray level above higher than this value will be ignored while the histogram is building.
	Inspection Configuration	Set the parameter and range that determines if a histogram is clear. • Gray Level: Maximum and minimum allowable gray level values for pixels within the region. These are set with a range of 0 to 255. • Variance: Maximum and minimum allowable variance of the gray level values within the region. These values must be greater than 0. • Histogram Pixel Count: Maximum and minimum allowable number of pixels within the region after applying tails and thresholds. These values must be greater than 0. • Histogram Pixel Percent: Maximum and minimum allowable percentage of pixels that are detected by the histogram after applying tails and thresholds. These values must be between 0 to 100.
	Show Result Image Histogram Regions	Specifies if the histogram regions are drawn in the ACE Vision Window. Show Results Graphics must be enabled for this to work.
Advanced Properties	Bilinear Interpolation Enabled	Enables using bilinear interpolation to sample the input image. By default, this is enabled because it ensures subpixel accuracy.
	Tail Black Gray Level Value	Percentage of pixels to ignore at the dark end of the gray level distribution. This is calculated after the pixels affected by the Threshold Black property have been removed.
	Tail White Gray Level Value	Percentage of pixels to ignore at the light end of the gray level distribution. This is calculated after the pixels affected by the Threshold White property have been removed.

● Gripper Clearance Histogram Pane

The histograms are measured using the Histogram Pane located beneath the properties area.



The **Add** button (

) and **Delete** button (

) are used to create and remove histograms from the tool. The Histogram Pane displays the properties for the currently selected histogram tab. The properties are described in the following table.

Item	Description
Offset	Defines the center coordinates of the histogram region with respect to the reference point defined by Offset in the main tool properties.
Region Name	User-defined name of the histogram. This is displayed in the lower left corner of the histogram region in the Vision Window.
Search Region	Defines the size (width, height) of the histogram region.

● Gripper Clearance Results

Gripper Clearance returns each instance that has passed the all histogram ranges. For these instances, the histogram analysis is available in additional columns. These are not shown by default and must be added using the Results Column Editor. The numbers of the histograms are variable and are denoted in the table below as <number> instead of actual values.



Additional Information

All results are calculated after applying tails and thresholds.

Use the table below to understand the results of the Gripper Clearance tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Position X	X-coordinate of the instance origin.
Position Y	Y-coordinate of the instance origin.
Position Angle	Angle of the instance origin.
Gray Level Mean <number>	Average gray level within the region of this histogram.
Histogram Pixel Count <number>	Number of pixels within the region of this histogram.
Histogram Pixel Percent <number>	Percentage of pixels within the region of this histogram.
Variance <number>	Variance of the gray level values of the pixels within the region of this histogram.

Labeling

Similar to the Labeling tool, this filter isolates masses of pixels that fall within a certain color range and meet different extraction conditions. Right-click in the image to set a single color or right-clicking and dragging to establish a region for a color range. Alternatively, use the Color Region section in the Properties to manually enter colors.
Once one or more color thresholds have been established, the tool will filter everything out of the image except for the pixels that fall within the ranges. Additional extraction conditions can be set to further limit the identified regions. Unlike the Labeling tool, the regions are not returned as data. Instead, Labeling Filter results in a new image.

● Labeling Configuration Items

Use the table below to understand the Labeling tool configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.

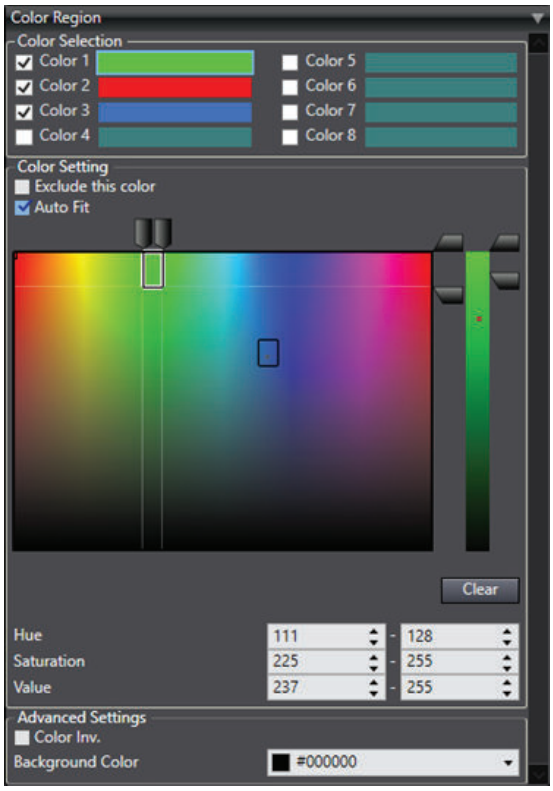
Group	Item	Description
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Rectangles Ellipses Circumferences Polygons WideArcs	Sets of regions of interest of a particular shape. The field will show how many regions exist (1-item collection,2-item collection, etc.). Click the plus sign in the field to create additional regions and click the arrow to see all regions.
	<region number>	Identifies the region of the specified shape. It will appear as [0], [1], [...] within the shape category. Click the minus sign in the field to remove the region and click the arrow to see its properties.
	Name	Available in all region types. Sets the user-defined name of the region.
	Overlap	Available in all region types. Select the behavior of the region. OR causes the region to be included in the search area and NOT excludes it from the search area. The behavior of specific regions is further defined in Z-Order. Refer to *8-8-3 Region of Interest* on page 8-194 for more information.
	Z-Order	Available in all region types. Sets the order for overlapping regions of interest. Higher numbers will be in front and resolved first. Refer to *Tools with Multiple Regions of Interest* on page 8-195 for more information.
	Offset	Available in all region types except for Polygons. Defines the coordinates of the region's center.
	Search Region (Width, Height)	Sets the height and width of the rectangular region. Only available in Rectangles.
	Radius X/Y	Defines the distance from the center to the exterior along the X- and Y-axes, respectively. Only available in Ellipses.
	Radius	Only available in Ellipses. Defines the distance from the center to the exterior along the region X- and Y-axes, respectively.

Group	Item	Description
Region of Interest	Thickness	Defines the distance between the two visible rings. Only available in Circumferences and WideArcs.
	Start/End Angle	Defines the start and end angle of the Wide Arc bounds. Angles are measured in degrees counterclockwise from the positive X-axis. The arc is created clockwise starting from the Start Angle and ending at the End Angle. Only available in WideArcs.
	Vertexes	Sets of vertexes in a polygon region of interest. The field will show how many vertexes exist (3-item collection, 4-item collection, etc). Click the plus line in the field to create additional regions and click the arrow to see all regions. Only available in Polygons.
	<vertex number>	Identifies a particular vertex in the region. It will appear as [0], [1], [2], and etc. The field defines the X/Y coordinates of the vertex. Click the minus sign to remove the vertex, as long as there are not fewer than three. Only available in Polygons.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once upon execution.
Properties	(Label Condition) Outside Condition	Defines the behavior of the area outside the search area. When enabled, the entirety of the area outside the regions of interest will be returned as the extracted color, connecting all detected masses that touch the edges of the regions.
	(Label Condition) Sort Condition	Select the order in which the returned masses are processed and output. The order will be descending (largest to smallest) by default; checking the Ascending box will sort them smallest to largest.
	(Label Condition) Neighborhood Mode	Specifies the connection conditions for labeling. Select 4 Neighborhood to process contiguous parts up, down, left, and right of the target pixel as the same label. Select 8 Neighborhood to add oblique directions.
	Extract Condition 1/2/3	Defines the type and limitation of conditions that constrain the returned masses
	(Extract Condition 1/2/3) Kind	Available under Extract Condition 1/2/3. Select the property that will define the constraints on the returned masses.
	(Extract Condition 1/2/3) Min/Max	Available under Extract Condition 1/2/3. Defines the minimum and maximum values that are acceptable for returned masses.
	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Hole Plug Color	Defines which color will fill in detected holes in the masses. This is disabled by default.

Group	Item	Description
Advanced Properties	Additional Data Set	Allows additional data values to be added to the output. Select the types of results based on what is needed. • Rectangle Coordinate Results: Upper left, lower right coordinates when enclosing a label with a circumscribed rectangle. • Cont Length Results: Length of the entire circumference of the label. • Circularity Results: Roundness of the label. • Elliptic Results: Long axis length, minor axis length, and angle when label is elliptically approximated. • Rotated Rectangle Results: Long side length, short side length, and angle of rotating short form circumscribing the label. • Inner Circle Results: Center coordinates and radius of the circle inscribed on the label. • Outer Circle Results: Center coordinates and radius of circumscribing circle. • Holes Number Results: Number of holes detected in the label.
	Image Type	Set the color scheme of the image to display. All Colors shows all extracted colors, Binary outputs all extracted colors as white and everything else as black, and the rest of the options only extract the masses that match one defined color.
	Max Result Display Count	Set the number of results to display. The tool will not output more instances than this value.

● Color Region Pane

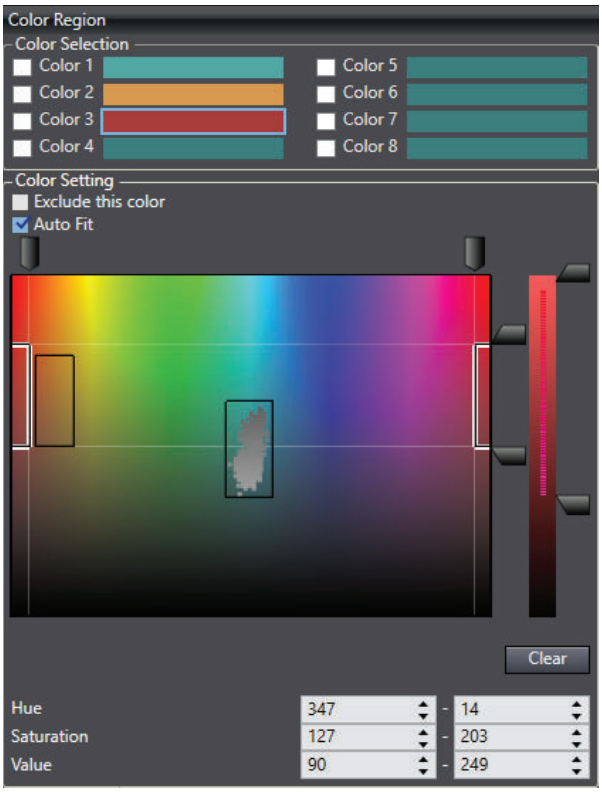
This pane shows the color ranges to be extracted from the image. This is imperative when using a color image. The color range(s) can be chosen by specifying the minimum and maximum for hue, saturation, and value. Conversely, the color range(s) can be selected in the image. Right-click on a pixel to select a specific color or right-click and drag to select a range. The detected colors will be displayed in the Color Setting region and the user can define the limits based on these data points.



The Color Region Pane is split into the following two sections.

1. Color Selection.
   This shows the colors that have been selected. Up to eight different colors can be extracted each time the tool is run. The user can determine which colors are extracted from among the selected ones by checking or unchecking the boxes next to them. For example, deselecting Colors 2 and 3 in Figure 1-2will cause only the green regions to be extracted.
2. Color Setting.
   This highlights the regions on the color spectrum that are used by the selected colors. They are also defined by the Hue, Saturation, and Value fields down below. The highlighted regions can be modified by dragging them in the color spectrum or adjusting the values in the fields down below.

The selected color range is shown as a rectangle with a black and white border. Other ranges defined will have a black border. The colors of the pixels in the most recently selected range will appear as gray markers on the rectangular field and as red marks in the Value slider on the right.



Additional Information

The X-axis of the rectangular color region is ranged from 0 to 360, represented in the number fields below it. Because the Hue range is supposed to be circular, it is possible for a red color range to begin close to the right side of the region and end on the left side.

Checking the box next to *Exclude this* color prevents the color from being extracted. Checking the box next to *Auto fit* allows the right-click short cut to be used to define this color range. This is enabled by default.

● Advanced Settings

This provides two additional properties that affect the resulting image. *Background Color* defines the color that will be set everywhere that is not an extracted mass. *Color Inv.* returns all masses as the background color and everywhere else as a white field.

● Locator Results

Use the table below to understand the results of the Locator tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To. This will only be different if the Locator is set relative to another tool.
Model Index	Index of the model located for each instance. If only one model is used, this will be the same for every instance.
Model Name	Name of the model located for each instance. Each Model Name is identical to the associated Locator Model tool name.

Item	Description
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 where the best quality is 100. A value of 100 means that the average error is 0. A value of 0 means that the average matched error is equal to the Conformity Tolerance.
Match Quality	Amount of matched model contours for the selected object instance expressed as a percentage. Match quality ranges from 0 to 100 where the best quality is 100. 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	Measurement of the clear area surrounding the specified object instance. Clear quality ranges from 0 to 100 where the best quality is 100. A value of 100 means that the instance is completely free of obstacles. If Minimum Clearances is Disabled, this value is 100.
Position X	X-coordinate of the instance origin.
Position Y	Y-coordinate of the instance origin.
Angle	Angle of the instance.
Scale Factor	Relative size of the instance with respect to its associated model.
Symmetry	Index of the instance of which this instance is a symmetry. Output Symmetric Instances must be enabled.
Time	Time in milliseconds needed to recognize and locate the object instance.



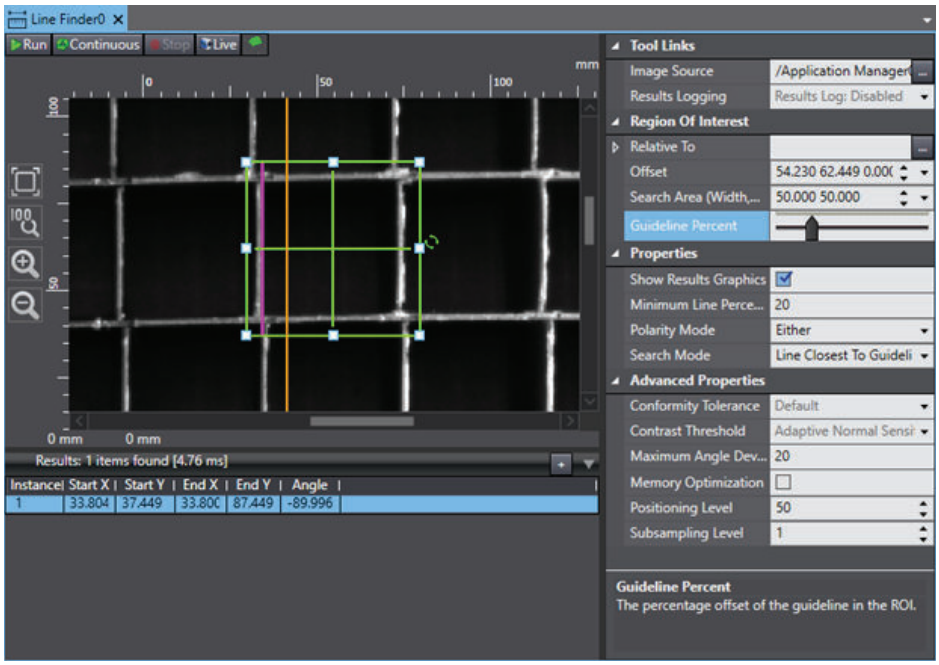
Additional Information

The result columns Fit Quality, Match Quality, and Clear Quality do not directly correlate with the Minimum Model Percentage property. This is because Minimum Model Percentage is compared to the initial coarse outline while these results are originated from the refined detail search.

Line Finder

This tool identifies linear features on objects and returns angle of inclination and the endpoint coordinates of the detected line.

Line Finder is most commonly used to return information about straight edges. For example, the tool in the following figure is used to locate line created by the left edge of the wire square. In this way, it can be used to locate and measure straight features within an object . Multiple detected lines can be used to calculate intersection points and refine a pick position based on part object geometries. If the line to be located should only be in a certain position, the guideline position and properties such as Search Mode can be used to decrease the detection range.



To create a Line Finder tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Finder** and then **Line Finder**. A Line Finder tool will be added to the Vision Tools list.

● Line Finder Results

Use the table below to understand the results of the Line Finder tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Start X	X-coordinate of the first endpoint.
Start Y	Y-coordinate of the first endpoint.
Center X	X-coordinate of the midpoint.
Center Y	Y-coordinate of the midpoint.
End X	X-coordinate of the second endpoint.
End Y	Y-coordinate of the second endpoint.
Angle	Angle (in degrees) of the detected line, measured from the positive X-axis of the camera frame.
Average Contrast	Average contrast between light and dark pixels on either side of the found arc, expressed in gray level values.
Fit Quality	Normalized average error between the calculated arc contours and the actual contours detected in the input image. Fit quality ranges from 0 to 100 where the best quality is 100. A value of 100 means that the average error is 0. A value of 0 means that the average matched error is equal to the Conformity Tolerance property.
Match Quality	Amount of matched arc contours for the selected instance expressed as a percentage. Match quality ranges from 0 to 100 where the best quality is 100. A value of 100 means that 100% of the arc contours were successfully matched to the actual contours detected in the input area.

Locator

This tool identifies objects in an image based on geometries defined in one or more Locator Models. Because of its speed, accuracy, and robustness, the Locator is the ideal frame provider for Robot Vision Manager inspection tools.

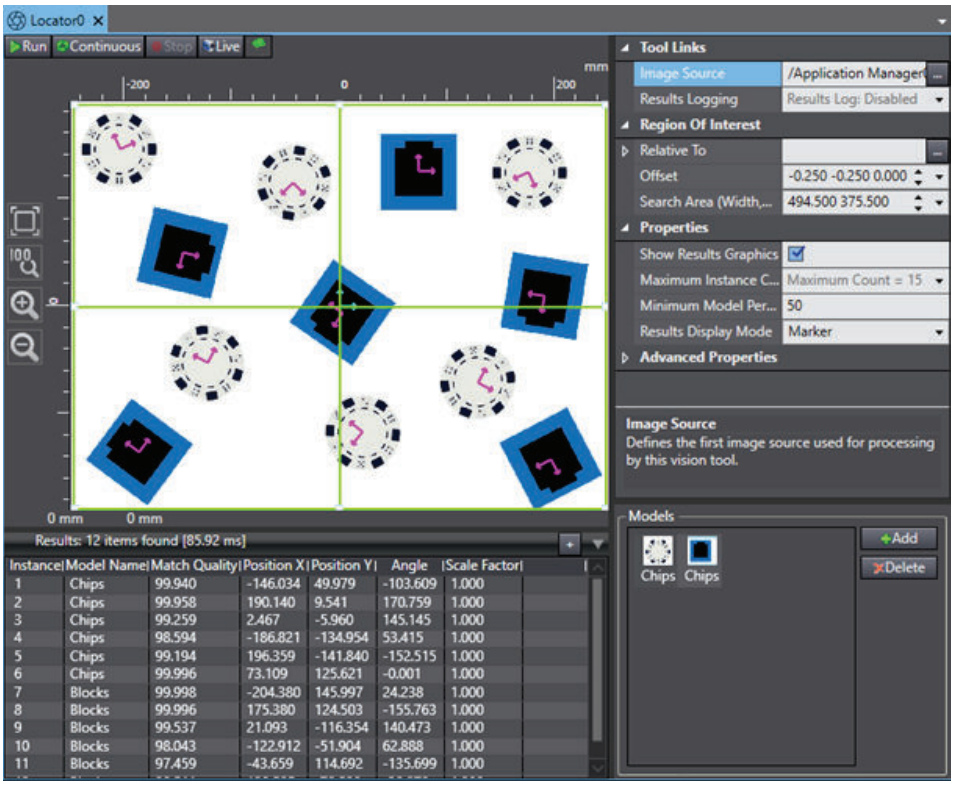
The Locator tool functions by detecting edges in the input images and then using them to generate a vectorized description of the image. The contours are generated on two coarseness levels: Outline and Detail. Outline is used to generate hypotheses of potential instances while Detail is used to confirm the hypotheses and refine the location. The detected contours are then compared to the model(s) to identify instances of the model(s) within the image.

A Locator can be set relative to other tools, such as another Locator or a Shape Search 3. This allows the Locator tool to be used to locate features, sub-features, or sub-parts on a parent object.

Additional Information

The Locator tool will not work until a Locator Model has been created. Refer to *Locator Model* on page 8-233 for more information.

Locator provides disambiguation between multiple similar models, but it typically has longer execution times than Shape Search 3. Shape Search 3 is often used for simple models while Locator is better when handling multiple models or situations where the model training process requires user control.



To create a Line Finder tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Finder** and then **Locator**. A Locator tool will be added to the Vision Tools list.

● Locator Configuration Items

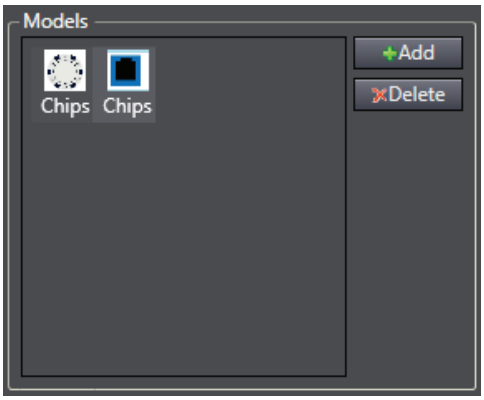
Use the table below to understand the Locator tool configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the center coordinates of the region.
	Search Area (Width, Height)	Defines the size of the region.
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Maximum Instance Count	Set the maximum number of instances to be detected in the image. All of the object instances matching the search constraints are output, up to a maximum of this value.
	Minimum Model Percentage	Specifies the connection conditions for labeling. Select 4 Neighborhood to process contiguous parts up, down, left, and right of the target pixel as the same label. Select 8 Neighborhood to add oblique directions.
	Results Display Mode	Defines how the results are rendered in the image display. Marker shows only the origin marker of the model, Model shows the outline of the detected model, and Marker and Model shows both.
Advanced Properties	Conformity Tolerance	Set the maximum local deviation between the expected model contours of an instance and the contours actually detected in the input image. Refer to *Conformity Tolerance* on page 8-227 for more information.
	Contour Detection	Set how the contour detection parameters are configured. Refer to *Contour Detection* on page 8-222 for more information.
	Contrast Polarity	Select the direction of polarity required for instance detection. Refer to *Contrast Polarity* on page 8-223 for more information.
	Contrast Threshold	Defines the minimum contrast needed for an edge to be detected in the input image. This threshold is expressed in terms of a step in gray level values. Refer to *Contrast Threshold* on page 8-223 for more information.
	Instance Ordering	Select the order in which the instances are processed and output. Refer to *Instance Ordering* on page 8-230 for more information.

Group	Item	Description
Advanced Properties	Minimum Clearance	Set the minimum percentage of the model bounding box area that must be free of obstacles to consider an object instance as valid. Refer to *Minimum Clearance* on page 8-231 for more information.
	Minimum Required Features	Set the minimum percentage of required features that must be recognized in order to consider the object instance as valid. *Minimum Required Features* on page 8-229 for more information.
	Model Disambiguation Enabled	Set to apply disambiguation to discriminate between similar models and between similar hypotheses of a single object. Refer to *Model Disambiguation Enabled* on page 8-229 for more information.
	Nominal Rotation	Set the required rotation range for valid instances. Refer to *Nominal Rotation* on page 8-227 for more information.
	Nominal Scale Factor	Set the required scale factor for an object instance to be recognized. Refer to *Nominal Scale Factor* on page 8-228 for more information.
	Output Symmetric Instances	Enable to output all symmetric poses of the object instances. If disabled, the tool will only output the best quality instance of a symmetric object. Refer to *Output Symmetric Instances* on page 8-231 for more information.
	Overlap Configuration	Enable to check if any instances overlap. These will be excluded from the results. Refer to *Overlap Configuration* on page 8-231 for more information.
	Percentage Of Points To Analyze	Set the percentage of points on a model contour that are actually used to locate instances. Refer to *Percentage of Points to Analyze* on page 8-229 for more information.
	Positioning Level	Set the level of positioning accuracy on a scale from 0 to 10. Refer to *Positioning Level* on page 8-228 for more information.
	Recognition Level	Set the level of recognition effort on a scale of 0 to 10. Refer to *Recognition Level* on page 8-228 for more information.
	Search Based On Outline Level Only	Set to cause the tool to only use the Outline Level contours of the model to detect instances. Refer to *Search Based on Outline Level Only* on page 8-224 for more information.
	Show Model Name	Set whether the model name will be displayed in the Vision Window. Refer to *Show Model Name* on page 8-230 for more information.
	Timeout	Enable to set the maximum time period (in milliseconds) that the tool is able to run.

● Locator Models Pane

The Models pane is used to edit the models used in this tool.



There are two ways to add models to the tool.

1. Click the **Add** Button and select the model from the defined Locator Model tools.
2. Click and drag the *Locator Model* object from the Multiview Explorer to the Models Pane. Models can be deleted from this pane by selecting them and clicking the **Delete** Button.

Additional Information

The order in which models are added defines the Model Index result value. Models cannot be reordered once they have been added to the Models Pane.

● Locator Advanced Properties - Edge Detection Parameters

The properties in this section modify the quality and quantity of contours that are generated from the input image.

■ Contour Detection

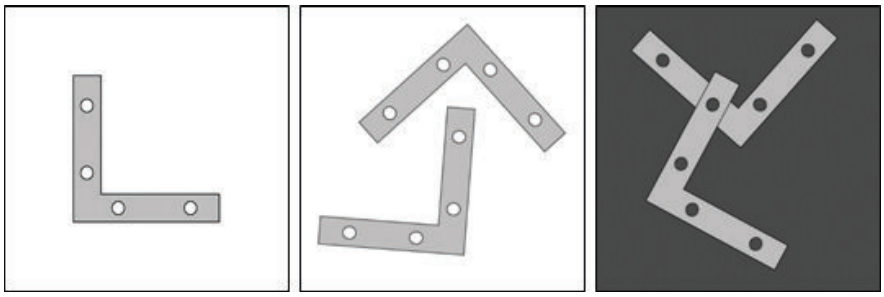
This property sets how the contour detection parameters are configured. For most applications, Contour Detection should be set to All Models, where the contour detection parameters are optimized by analyzing the parameters that were used to build all the currently active models. Custom contour detection should only be used when the default values do not work correctly. Setting this to Custom allows the user to specify *Outline Level*, *Detail Level*, and *Tracking Inertia* as described below.

- The Outline Level contours are used to rapidly identify potential instances of the object. Coarseness values range from 1 to 16 where 1 is full resolution and all other values are factors of 1. For example, at level 8, the resolution is 8 times lower than an image at full resolution. Lower values of Outline Level lead to higher execution times.
- The Detail Level contours are used to confirm recognition and refine the position of valid instances. For images that are not in perfect focus, better results will be obtained with a higher value of Detail Level. To obtain high-accuracy object location, use images with sharp edges and set Detail Level to the lowest coarseness value. Detail Level and Outline Level have the same range, but Detail Level must have a lower value. Lower values of Detail Level lead to higher execution times.
- Tracking Inertia defines the longest gap that can be closed to connect two edge elements when building the source contours. It is set on a scale of 0 to 1. Higher values can help close small gaps and connect contours that would otherwise be broken into smaller sections.

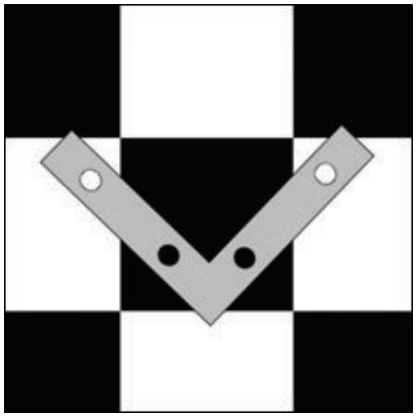
■ Contrast Polarity

This property defines the polarity change in gray level values between an object and its background, which can be dark to light, light to dark, or a combination of both. The reference polarity for an object is defined by its Model with respect to the polarity in the image on which the Model was created. It can be set to the following options.

- Normal searches only for instances that match the polarity of 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 (refer to the middle image in the following figure).
- Reverse searches only for instances that have opposite polarity from 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 (refer to the right image in the following figure).



- Normal And Reverse searches for both of the above. This will not include cases where polarity is reversed at various locations along the edges of an object.
- Do Not Care indicates that polarity should not be taken into account when searching for instances. This is useful when a model has multiple local polarity changes, such as in the checkerboard background of the figure below, where Do Not Care must be selected in order for the object to be detected.



■ Contrast Threshold

This property sets the minimum contrast needed for an edge to be detected in the input image. The threshold value is interpreted as the change in gray level value required to detect contours. Contrast Threshold can be set to Adaptive Low Sensitivity, *Adaptive Normal Sensitivity*, *Adaptive High Sensitivity*, or Fixed Value. Higher values reduce sensitivity to contrast, reducing noise and lowering the amount of low-contrast edges. Conversely, lower values increase sensitivity and add a greater amount of edges to the contours at the expense of adding more noise, which can generate false detections and/or increase execution time .

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, and can generally be used for most applications. The Fixed Value option allows the user to set the

desired value on a scale from 1 to 255, corresponding to the minimum step in gray level values required to detect contours. This is primarily used when there is little variance in lighting conditions.

■ Search Based on Outline Level Only

This property restricts the search to using only the Outline Level source contours to search, recognize, and position object instances. Detail Level contours are ignored completely. Enabling this can increase speed with possible loss of accuracy and detection of false instances.
An Outline-based search is mainly used for time critical applications that do not require a high-positioning accuracy or that only need to check for presence/absence of objects. To be effective, this type of search requires clean run time images that provide high-contrast contours with little or no noise or clutter.

■ Labeling Configuration Items

Use the table below to understand the Labeling tool configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Rectangles Ellipses Circumferences Polygons WideArcs	Sets of regions of interest of a particular shape. The field will show how many regions exist ("1-item collection","2-item collect', etc.). Click the plus sign in the field to create additional regions and click the arrow to see all regions.
	<region number>	Identifies the region of the specified shape. It will appear as [0], [1], [...] within the shape category. Click the minus sign in the field to remove the region and click the arrow to see its properties.
	Name	Available in all region types. Sets the user-defined name of the region.
	Overlap	Available in all region types. Select the behavior of the region. OR causes the region to be included in the search area and NOT excludes it from the search area. The behavior of specific regions is further defined in Z-Order. Refer to *8-8-3 Region of Interest* on page 8-194 for more information.

Group	Item	Description
Region of Interest	Z-Order	Available in all region types. Sets the order for overlapping regions of interest. Higher numbers will be in front and resolved first. Refer to *Tools with Multiple Regions of Interest* on page 8-195 for more information.
	Offset	Available in all region types except for Polygons. Defines the coordinates of the region's center.
	Search Region (Width, Height)	Sets the height and width of the rectangular region. Only available in Rectangles.
	Radius X/Y	Defines the distance from the center to the exterior along the X- and Y-axes, respectively. Only available in Ellipses.
	Radius	Only available in Ellipses. Defines the distance from the center to the exterior along the region X- and Y-axes, respectively.
	Thickness	Defines the distance between the two visible rings. Only available in Circumferences and WideArcs.
	Start/End Angle	Defines the start and end angle of the Wide Arc bounds. Angles are measured in degrees counterclockwise from the positive X-axis. The arc is created clockwise starting from the Start Angle and ending at the End Angle. Only available in WideArcs.
	Vertexes	Sets of vertexes in a polygon region of interest. The field will show how many vertexes exist ("3-item collection", "4-item collection", etc). Click the plus line in the field to create additional regions and click the arrow to see all regions. Only available in Polygons.
	<vertex number>	Identifies a particular vertex in the region. It will appear as [0], [1], [2], and etc. The field defines the X/Y coordinates of the vertex. Click the minus sign to remove the vertex, as long as there are not fewer than three. Only available in Polygons.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once upon execution.

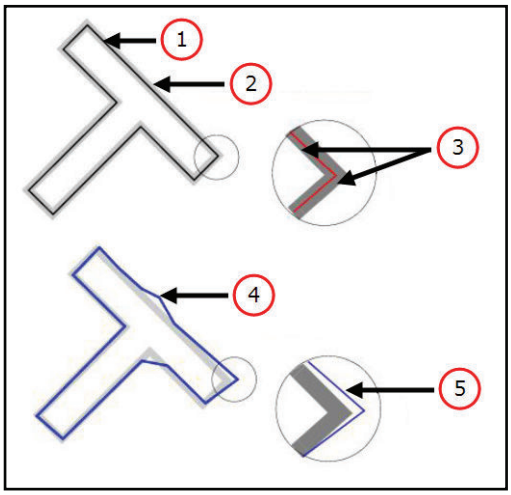
Group	Item	Description
Properties	(Label Condition) Outside Condition	Defines the behavior of the area outside the search area. When enabled, the entirety of the area outside the regions of interest will be returned as the extracted color, connecting all detected masses that touch the edges of the regions.
	(Label Condition) Sort Condition	Select the order in which the returned masses are processed and output. The order will be descending (largest to smallest) by default; checking the Ascending box will sort them smallest to largest.
	(Label Condition) Neighborhood Mode	Specifies the connection conditions for labeling. Select 4 Neighborhood to process contiguous parts up, down, left, and right of the target pixel as the same label. Select 8 Neighborhood to add oblique directions.
	Extract Condition 1/2/3	Defines the type and limitation of conditions that constrain the returned masses
	(Extract Condition 1/2/3) Kind	Available under Extract Condition 1/2/3. Select the property that will define the constraints on the returned masses.
	(Extract Condition 1/2/3) Min/Max	Available under Extract Condition 1/2/3. Defines the minimum and maximum values that are acceptable for returned masses.
	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Hole Plug Color	Defines which color will fill in detected holes in the masses. This is disabled by default.
Advanced Properties	Additional Data Set	Allows additional data values to be added to the output. Select the types of results based on what is needed. • Rectangle Coordinate Results: Upper left, lower right coordinates when enclosing a label with a circumscribed rectangle. • Cont Length Results: Length of the entire circumference of the label. • Circularity Results: Roundness of the label. • Elliptic Results: Long axis length, minor axis length, and angle when label is elliptically approximated. • Rotated Rectangle Results: Long side length, short side length, and angle of rotating short form circumscribing the label. • Inner Circle Results: Center coordinates and radius of the circle inscribed on the label. • Outer Circle Results: Center coordinates and radius of circumscribing circle. • Holes Number Results: Number of holes detected in the label.
	Image Type	Set the color scheme of the image to display. All Colors shows all extracted colors, Binary outputs all extracted colors as white and everything else as black, and the rest of the options only extract the masses that match one defined color.
	Max Result Display Count	Set the number of results to display. The tool will not output more instances than this value.

● Locator Advanced Properties - Search Parameters

The properties in this section are constraints that restrict the Locator's search process.

■ Conformity Tolerance

This property 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 instance hypothesis 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. See the following figure for an example.



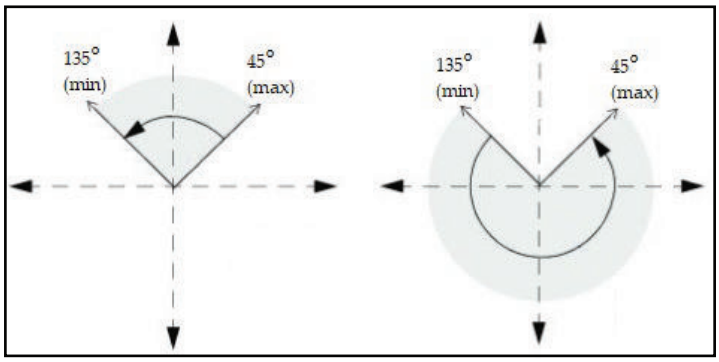
Item	Description
1	Model contour (red).
2	Conformity tolerance (gray).
3	Conformity tolerance (gray) value applies to both sides of model contour (red).
4	Contours of the found object (blue). Contours outside conformity tolerance are not valid.
5	Portion of object contour outside the conformity tolerance zone.

The Conformity Tolerance options are described below.

- Use Default: Enabling this causes the Locator tool to calculate a default value by analyzing the calibration, contour detection parameters, and search parameters. This box must be unchecked if either of the other options are going to be used.
- Range Enabled: Enables the use of the manually-set tolerance value. This is set by Tolerance, which defines the maximum difference in calibrated units by which a matched contour can deviate from either side of its expected position. It can be set on a range of 1 to 100.

■ Nominal Rotation

This property constrains the rotation range within which Locator can detect instances. Any possible instance must satisfy this property in order to be recognized as instances. By default, the range is set from -180 to 180 degrees. This can be changed depending on the needs of the application. The rotation range spans counterclockwise from the minimum to the maximum angle.



Conversely, Use Nominal can be enabled. This applies the value in the Nominal field to the Locator, which searches for instances within a tolerance of that angle. Any instances found within the tolerance will automatically be set to the angle in the Nominal field. If the actual angle of rotation is required, it is recommended to leave the Use Nominal box disabled and instead enter a small range, such as 89 to 91 degrees.

Additional Information

If the trained Locator Model has rotational symmetry, it is possible that Nominal Rotation will cause an instance to be detected that is not actually within the target angle range. In this case, raising the Minimum Model Percentage can be used to prevent the symmetric instance from being detected.

■ Nominal Scale Factor

This property sets the required scale factor for an object to be recognized. Similar to Nominal Rotation, it can constrain the range of scale factors for which the Locator will search by either setting a minimum and maximum value or a fixed nominal value. By default, the range is set from 0.9 to 1.1. Any possible instance must lie within this range in order to be output as an identified instance. Note that the scale factor parameter has one of the most significant impacts on search speed and using a large scale factor range can significantly slow down the process. The range should be configured to include only the scale factors that are expected from the application.

Conversely, Use Nominal can be enabled to make the Locator search only for the value in the Nominal range. However, 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 recommended to configure a narrow scale range instead of a nominal value. The Minimum, Maximum, and Nominal values can all be set within a range of 0.1 to 10.

■ Positioning Level

This property modifies the positioning accuracy. This can be set on a range from 0 to 10, although the default setting of 5 is the optimized and recommended setting for typical applications. Lower values provide coarser positioning with faster execution times while higher values provide high-accuracy positioning at the cost of speed.

Positioning Level does not have a large impact on execution speed. However, in applications where accuracy is not critical, the value can be decreased to lower the processing the time.

■ Recognition Level

This property slightly modifies the level of recognition effort. This can be set on a range from 0 to 10, but the default setting of 5 is the optimized and recommended setting for typical

applications. Lower values provides faster searches that may miss partially blocked instances while higher values more accurately recognize all objects in cluttered or noisy images at the cost of speed.
When adjusting the Recognition Level, it is important to repeatedly test the application to find the optimum speed at which the process will still find all necessary objects. If the recognition level is too low, some instances may be ignored, but if it is too high, the application may run too slowly.
Recognition Level does not affect positioning accuracy.

● Locator Advanced Properties - Model Parameters

The properties in this section constrain how the tool interacts with the model.

■ Minimum Required Features

This property defines the percentage of required features that need to be recognized for the Locator to accept a valid instance of an object. In most applications, a feature in the Locator Model will be set as Required if it needs to be present in every single instance, but this property allows some flexibility. Minimum Required Features is set as a percentage of all required features on a range from 0.1 to 100. Note that this parameter is expressed in terms of a percentage of the number of required features in a model and does not consider the amount of contour each required feature represents.
Refer to *Locator Model Feature Selection Pane* on page 8-235 for information on how to set a feature as required.

■ Model Disambiguation Enabled

This property determines if disambiguation is applied to the detected instances to resolve ambiguity between similar models by analyzing distinguishing features. This is enabled by default and should remain enabled in most applications. Disabling this will significantly reduce the time needed to learn or relearn models, but it prevents the Locator from differentiating between similar models. This should only be done in applications that require many different models.

■ Model Optimizer Enabled

This property determines if the models can be optimized interactively by building a model from multiple instances of an object. When enabled, the user can build an optimized model by creating an initial Locator Model and then running the Locator. Each new instance of the object found by the Locator is analyzed and compiled into the current optimized model. 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.
By default, this property is disabled and can remain disabled for most applications. It may be useful for applications where the objects have a variable shape.

■ Percentage of Points to Analyze

This property defines points used to create an optimization model. It is set as a percentage of the points on a model contour actually used to locate instances. For example, when it is set to the default value of 50%, one out of every two points is used. This can be set on a range from 0.1 to 100. Higher values can increase the accuracy of the optimized model but incur longer optimization time while lower values lower the accuracy while improving speed.

■ Show Model Name

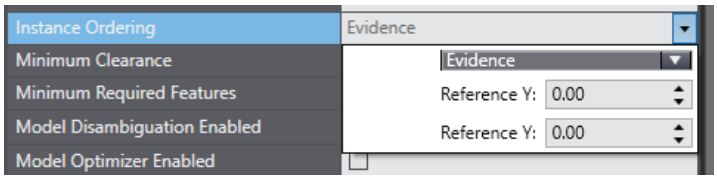
This property determines if the model name is displayed in the Vision Window. Enabling this will display the name of the respective Locator Model tool name(s) in the tool window and the Vision Window. This is only evaluated if Show Results Graphics is enabled.

● Locator Advanced Properties - Instance Output Constraints

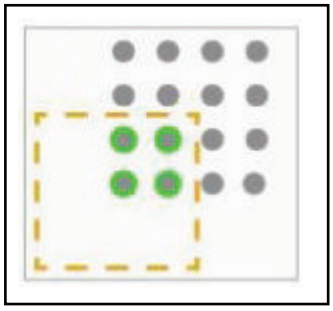
The properties in this section constrain how the tool outputs instances.

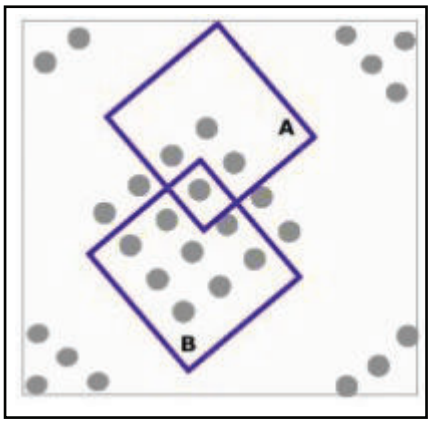
■ Instance Ordering

This property sets the order in which object instances are output. In general, it is adjusted by opening the dropdown menu and changing the method. Reference values may also be required.



- Evidence: The instances are ordered according to their hypothesis strength.
- Directional: Four of the options are Left to Right, Right to Left, Top to Bottom, and Bottom to Top. This refers to the position of the instance within the image and is useful for pick-and-place applications in which parts that are farther down a conveyor must be picked first.
- Quality: The instances are ordered according to their Match Quality. If two instances have the same Match Quality, then they are sorted by their Fit Quality. Note that this setting can significantly increase the search time because the 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 potential instances. The time for additional instances is then zero because the search process is already complete.
- Distance: Two of the options are Distance Image and Distance World. In both, the instances are ordered according to their proximity to a user-defined point in the camera coordinate system, as defined by the fields Reference X and Reference Y. In Distance Image, these fields are in terms of pixels while in Distance World they are expressed in calibrated length units.
- Shading Consistency: The instances are ordered 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 can assist in discriminating between similar hypotheses. This is a requirement for color processing of models and also often used for ball grid array applications, as illustrated in the figures below.





■ Minimum Clearance

This property sets the minimum percentage of the model bounding box area that must be free of other instances to consider an object instance as valid. It is disabled by default. When enabled, the Locator tool scans the bounding boxes of all instances for obstacles, such as other instances. If the amount of obstacle-free space is less than the bounding box percentage listed in the Minimum Clearance field, then the instance is not returned.
Enabling this property may significantly increase the search time. It is primarily intended for pick-and-place applications to confirm that objects have the necessary clearance to be picked. Minimum Clearance also activates the computation of the Clear Quality result for each instance.

■ Output Symmetric Instances

This property defines how the Locator will handle symmetric (or nearly symmetric) objects. It is disabled by default, causing the search process to output only the best quality instance of a symmetric object. If enabled, all possible symmetries of a symmetric object will be output. This can significantly increase execution time when there are many possible symmetries of an object, such as when the object is circular.

■ 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 images that have already been added to the queue of instances to process.
If *Disable Overlap Check* is selected, all overlap checking is disabled. When this option is selected, the remaining Overlap Configuration items are not available.
If a part is located in multiple images, the robot will attempt to pick at the same belt-relative position multiple times. If this occurs when Overlap Configuration is not disabled, consider increasing the Overlap distance.
If the newly-located part is within the specified Overlap Distance of a previously located part (accounting for belt travel), it is assumed to be the same part and will not be added as a duplicate new instance.
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.

● Locator Results

Use the table below to understand the results of the Locator tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To. This will only be different if the Locator is set relative to another tool.
Model Index	Index of the model located for each instance. If only one model is used, this will be the same for every instance.
Model Name	Name of the model located for each instance. Each Model Name is identical to the associated Locator Model tool name.
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 where the best quality is 100. A value of 100 means that the average error is 0. A value of 0 means that the average matched error is equal to the Conformity Tolerance.
Match Quality	Amount of matched model contours for the selected object instance expressed as a percentage. Match quality ranges from 0 to 100 where the best quality is 100. 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	Measurement of the clear area surrounding the specified object instance. Clear quality ranges from 0 to 100 where the best quality is 100. A value of 100 means that the instance is completely free of obstacles. If Minimum Clearances is Disabled, this value is 100.
Position X	X-coordinate of the instance origin.
Position Y	Y-coordinate of the instance origin.
Angle	Angle of the instance.
Scale Factor	Relative size of the instance with respect to its associated model.
Symmetry	Index of the instance of which this instance is a symmetry. Output Symmetric Instances must be enabled.
Time	Time in milliseconds needed to recognize and locate the object instance.

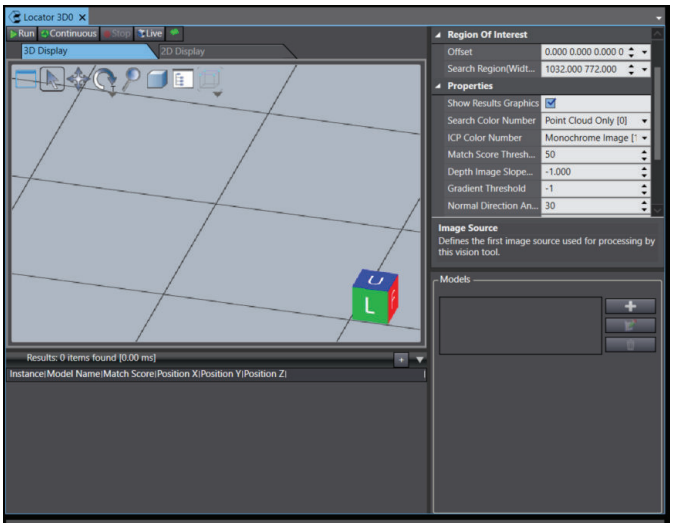


Additional Information

The result columns Fit Quality, Match Quality, and Clear Quality do not directly correlate with the Minimum Model Percentage property. This is because Minimum Model Percentage is compared to the initial coarse outline while these results are originated from the refined detail search.

Locator 3D

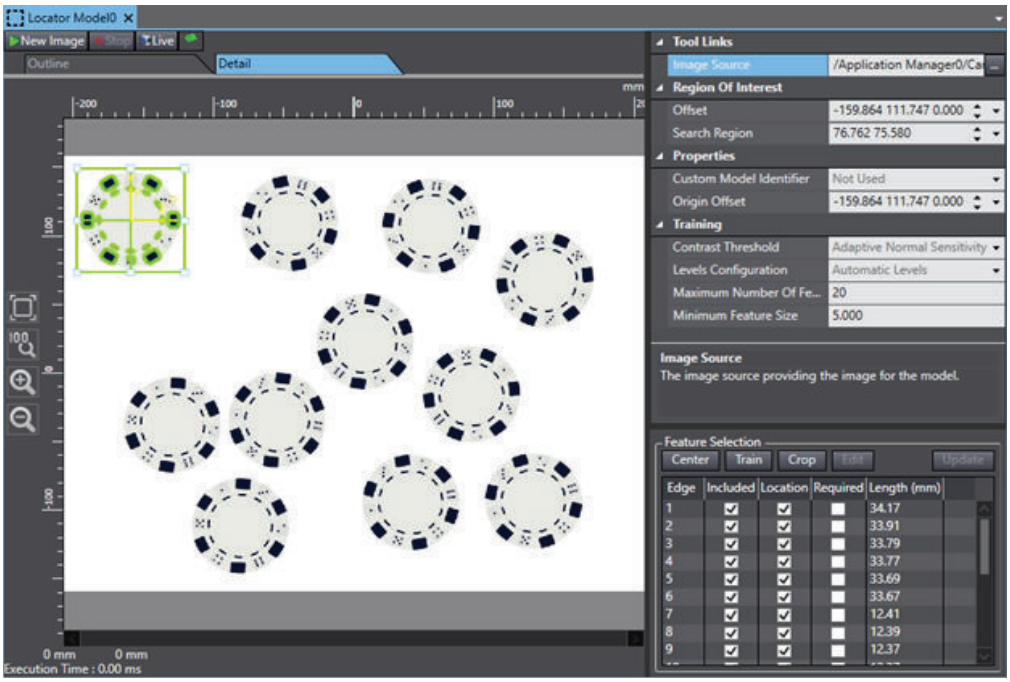
This tool is used with 3D cameras to identify features on an object in three dimensional space for location purposes. Double-click **Locator 3D** in the Multiview Explorer to open the panel.



Add the model and unit by clicking the **Add** Button. Use the *Region of Interest* and *Properties* to specify the object.

Locator Model

This model tool describes the geometry of an object to be found by the Locator tool.



To create a Locator Model, right-click **Vision Tools** in the Multiview Explorer, select **Add Finder** and then **Locator Model**. A Locator Model will be added to the Vision Tools list.

Additional Information

The Virtual Camera should be calibrated before any Locator Models are created. If it is not calibrated, any applied calibration will cause the model geometries to scale incorrectly.

● Locator Model Configuration Items

Use the table below to understand the Locator Model configuration items.

Group	Item	Description
Tools Links	Image Source	Defines the image source used for processing by this vision tool.
Region of Interest	Offset	Defines the center coordinates of the region.
	Search Region	Defines the size of the region.
Properties	Custom Model Identifier	Enable to identify this model with a user-defined identifier for a Robot Vision Manager Application.
	Origin Offset	Defines the model origin. This will be referenced as the Position point of this model any time it is referenced by another tool.
Training	Contrast Threshold	Set the minimum contrast needed for an edge to be detected in the input image and used for arc computation. This threshold is expressed in terms of a step in gray level value.
	Levels Configuration	Specifies the levels used for contour detection. In most cases, Automatic Levels can be enabled. The Outline value cannot be lower than the Detail value. For each level, coarseness values range from 1 to 16, where 1 has the best resolution. Every other value is a factor of the best resolution. For example, if one of these is set to 8, then the resolution is 8 times lower than an image at full resolution (level 1).
	Maximum Number of Features	Set the maximum number of features the model will detect.
	Minimum Feature Size	Set the minimum length of a feature (in millimeters) required for it to be selected. This has no impact on feature detection, only feature selection.

● Locator Model Region of Interest

The region of interest defines where Locator Model will look for the contours defined by contrast changes. Any contours located outside the region will not be detected or considered to be part of the model. Therefore, it is best to set the region as close to the part as possible so it can be identified without cropping any part edges.

The Locator Model region of interest can be set by dragging the corners of the green box in the display window or by modifying the numbers in the Offset and Search Region properties. In the property fields, the numbers can either be entered manually or changed using the up / down arrows for each value. Note that only the property fields can be used to rotate the region.

● Locator Model Origin

The model origin, depicted as a yellow frame, defines the position and orientation of the pick location. It can be set in the following ways.

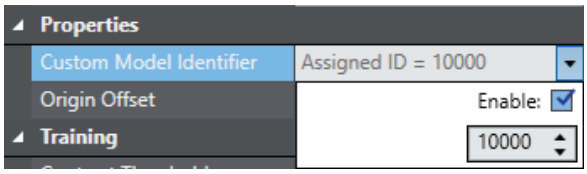
- Manually drag the yellow origin indicator to the desired location. Adjust the orientation by clicking and dragging the rotation symbol. This is the best method for irregularly-shaped parts with off-center masses. The coordinates in the property fields will automatically adjust to the new position.
- Manually enter a desired location in the Origin Offset field. This is typically used to make small adjustments on the origin position.
- Click the **Center** Button in the *Feature Selection* pane without a feature selected. The origin will be centered in the model's region of interest. This may not be optimal in irregularly-shaped parts

and depends entirely on the size of the region of interest. This does not affect the origin orientation. If a feature is selected when *Center* is clicked, the origin will be centered on the feature.

- Select a feature in the *Feature Selection* pane and click the **Center** Button. The origin will be centered on that feature. This is useful when a particular feature is going to be used as the pick point. This does not affect the origin orientation.
- Cropping the model will automatically center the origin in the region of interest if the origin is outside of the region. This does not affect the origin orientation.

● Locator Model Custom Model Identifier

The Custom Model Identifier can be any integer from 0 to 10000. It is defined in the property's dropdown menu, as shown in the following figure.

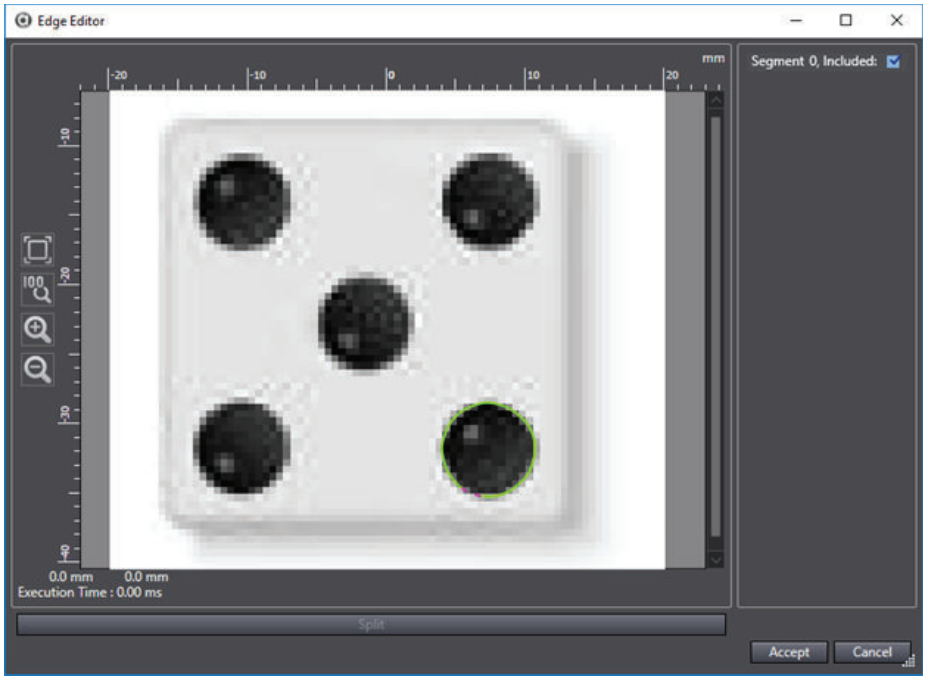


Additional Information

The **Enable** 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.

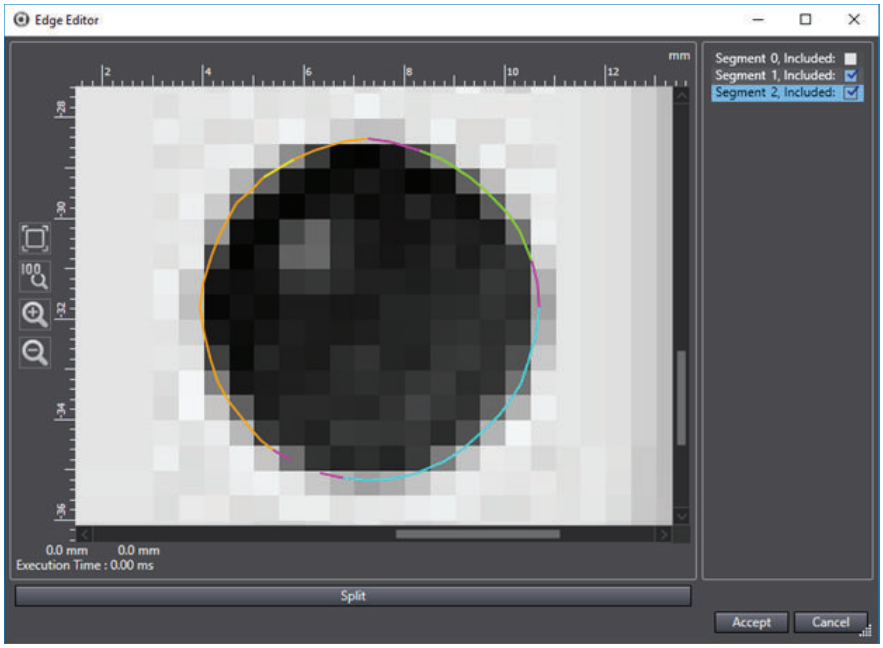
● Locator Model Feature Selection Pane

The Edge Editor allows the user to split a feature into multiple segments and choose which segments to include or exclude from a model. This can be useful when improper lighting results in a feature including both part object and shadow outline geometries.



The main part of this editor is the Vision Window, which is controlled in the same way as the tool Vision Window. The only feature that appears in this window is the one currently being edited. The right side of the editor shows the segments of the feature and the check boxes that control if they are included.

Features can be split into multiple segments by clicking somewhere on the feature and then clicking the **Split** Button below the image. This adds a new segment on the right side, to visually determine which segments are included in the overall feature. Only segments with checked boxes are included in the feature.

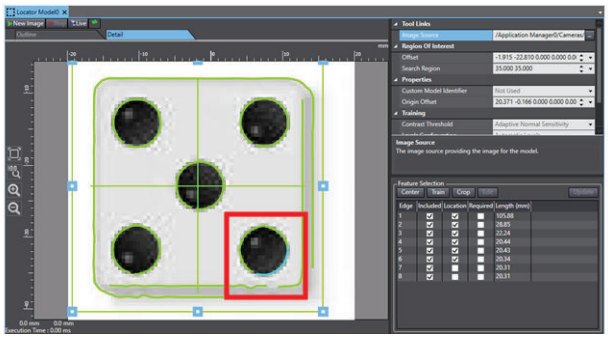


The feature is color-coded in this Vision Window with the following designations.

- Orange: Currently selected segment.
- Purple: Ends of segments. A purple line that connects to lines at both ends represents a split between two segments. If it only connects at one end, it is an endpoint of the feature.
- Green: Segment is included in the overall feature.
- Light blue: Segment is excluded from the overall feature.

Additional Information

Accepting the changes here will be reflected in the tool Vision Window. See the bottom-right die circle in the figure below. Making any changes to the Included property of the feature in the *Feature Selection* pane will revert all changes made in the Edge Editor.



Locator Model Results

Locator Model has no data or image results. Its only function is to create a model to be used in the Locator tool. Refer to *Locator* on page 8-219 for more information.

Configuration Process

The general steps required to configure a Locator Model are listed below.

1. Create a new Locator Model.
2. Select the correct Image Source and click the **New Image** Button to load an image
3. Define the Maximum Number Of Features and Minimum Feature Size properties based on the geometries of the part to be located.
4. Set the region of interest around the part.
5. Click the **Train** Button to train the model.
6. Edit feature selection as needed in the Feature Selection pane. Use the Edge Editor when necessary
7. Define model origin and crop the image around the model.

Training a Model

The region of interest defines where the Locator Model will look for contours identified by contrast changes. Any contours outside the region will not be detected or considered to be part of the model. Therefore, it is best to set the region as close as possible to the part to be identified without omitting any edges of the part.

Once the region has been defined, the model needs to be trained by clicking the **Train** Button. This is also required whenever the region of interest is adjusted or a new image is loaded. Use this to decide which features to include using the Feature Selection pane. A Locator tool will search for features included in the model when detecting instances. Locator searches by comparing a potential match with the Minimum Model Percentage property of that tool, which defines a percentage that an instance needs to match to be considered a valid instance. This can be further controlled by marking some features as Required in this pane. If a Locator detects a potential instance that meets the Minimum Model Percentage but lacks the required features, the detected candidate will not be returned as a result.

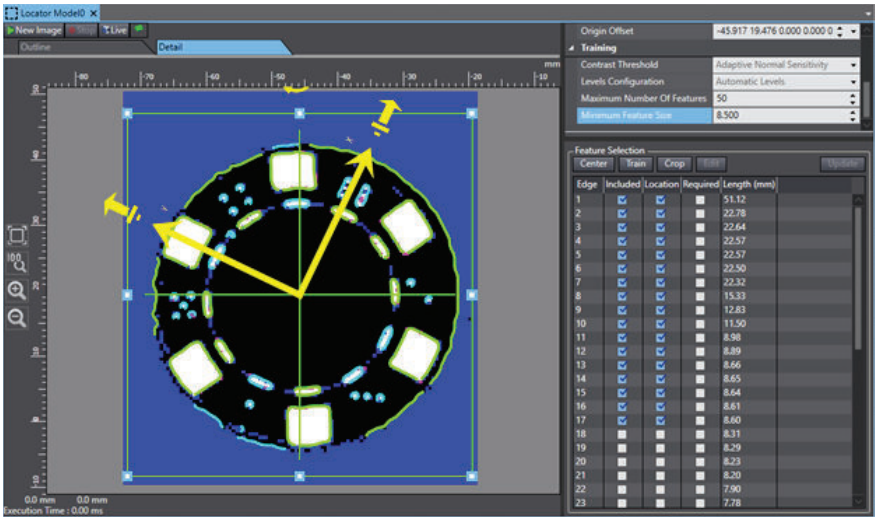
Whenever any change is made in the Feature Selection pane, click the **Update** Button for the changes to take effect.

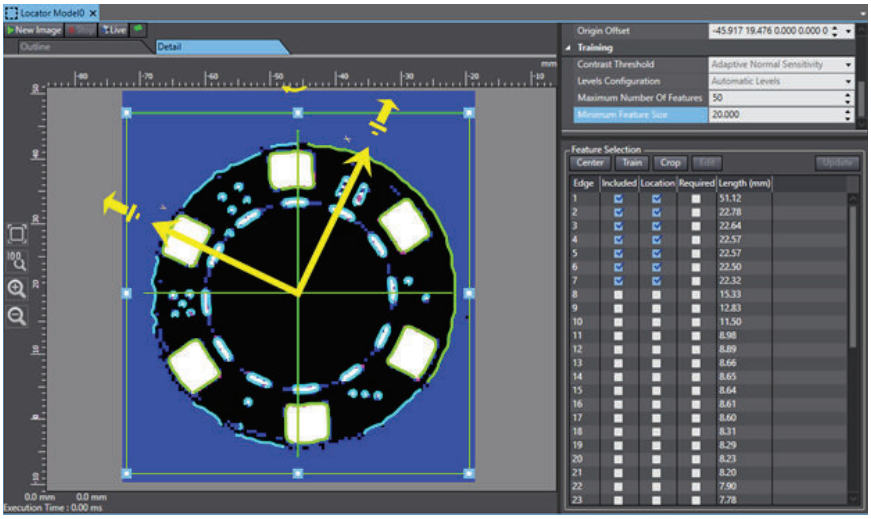
Refer to *Locator Model Feature Selection Pane* on page 8-235 for a detailed description of the feature selection process. Refer to *Locator* on page 8-219 for more information on instance detection.

The training process is also controlled by the parameters in the Training section. Contrast Threshold and Levels Configuration can usually be set to automatic parameters, but the other two need to be defined.

Maximum Number Of Features defines the maximum number of edge contours that will be detected to potentially be included in the model. Locator Model will always return all detected edges if it is less than or equal to this value. However, if there are more edges within the region of interest, the largest ones will be returned until the number of edges is equal to this value. For example, if Maximum Number Of Features is set to 20 and the tool detects 30 edges, only the 20 largest will be shown in the Feature Selection pane.

Conversely, Minimum Feature Size has no effect on which edges are returned. Instead, it defines which features are initially included in the model itself. When the Train button is clicked or a new image is loaded, Locator Model will automatically include all features in the model that have length greater than or equal to this value.
All other features within the Maximum Number Of Features constraint will be returned and shown in the Feature Selection pane, but they will not automatically be included. The two figures below show how this property is utilized during training. In the first figure below, all the features included in the second figure below are still returned, but only some of them are actually included in the model.





Additional Information

If a single contour of the part is broken into multiple features, it may be necessary to adjust lighting or the Contrast Threshold and Levels Configuration.

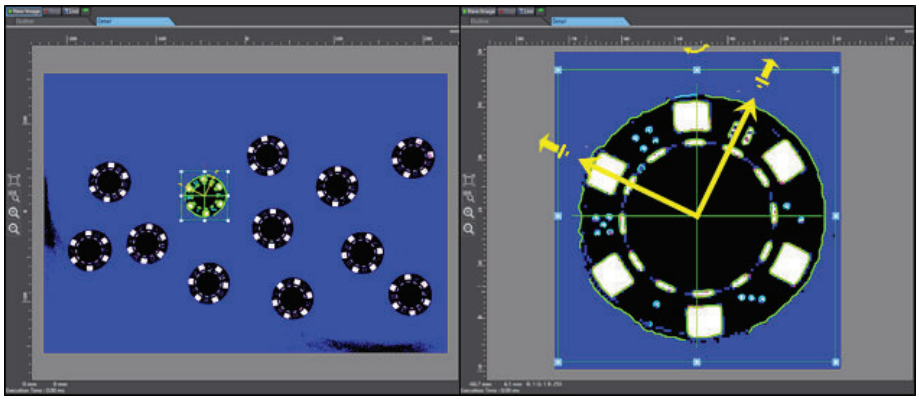
Once the model has been fully trained, the model origin must be set. This point is depicted as a yellow frame and defines the position and orientation of the pick location. When the model is used in a Locator tool, this location will be used as the point of reference to define instance positions. The origin can be set in five ways as described below.

- Manually drag the yellow origin indicator to the desired location. Adjust the orientation by clicking and dragging the rotation symbol or the extendable arrows. This is the best method for irregularly-shaped parts with off-center masses. The coordinates in the property fields will automatically adjust to the new position.
- Manually enter a desired location in the Origin Offset field. This is typically used to make small adjustments of the origin.
- Click the **Center** Button in the Feature Selection pane without a feature selected. The origin will be centered in the region of interest. This may not be optimal for irregularly-shaped parts and depends entirely on the size of the region of interest. This does not affect the origin orientation. If a feature is selected when the Center button is clicked, the origin will be centered on the feature.
- Select a feature in the *Feature Selection* pane and click the **Center** Button. The origin will be centered on that feature. This is useful when a particular feature is going to be used as the pick point. This does not affect the origin orientation.

In many situations the origin can be centered on one feature and the roll angle can be set by aligning an extendable arrow with another part geometry that may or may not be included in the model.
Cropping the model will automatically center the origin in the region of interest if the origin is located outside of the region after cropping. This does not affect the origin orientation.
After testing that the Locator Model and Locator sufficiently locate the object, it is recommended to crop the image using the Crop button. The following figure shows a model in the original image on the left and the cropped version on the right.

Additional Information

Cropping Locator Model Images can reduce project file size.

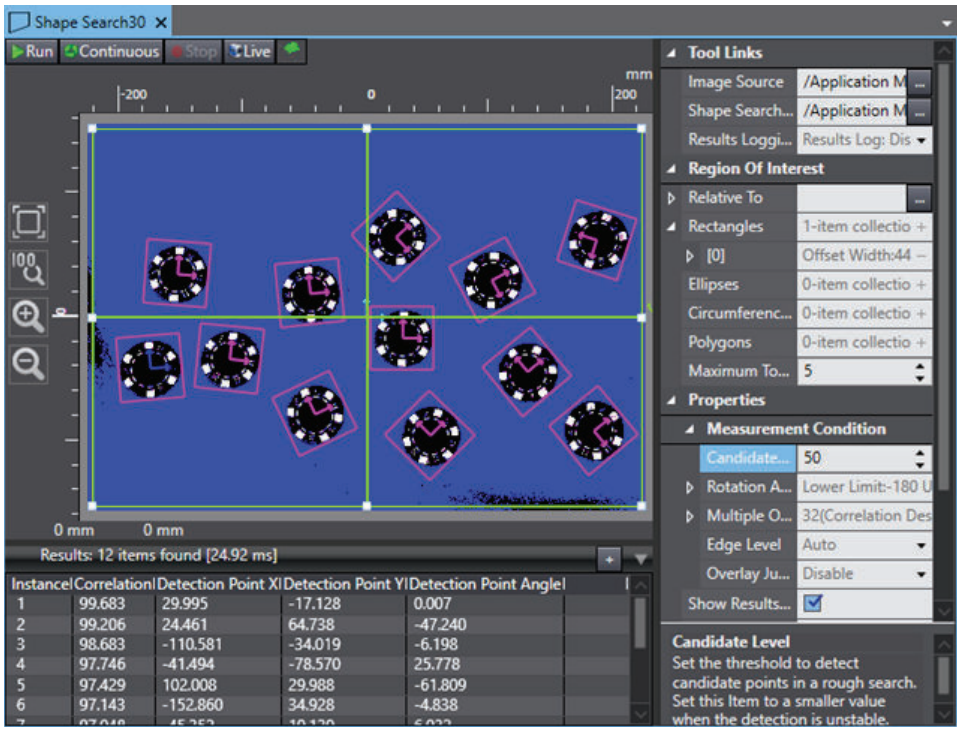


Shape Search 3

This tool identifies objects in an image based on geometries defined in a Shape Search 3 Model. Because the model requires a small degree of training, Shape Search 3 can output instances and correlation values based on similarity, measurement target position, and orientation.
Unlike other search methods where color and texture information are used to detect objects, Shape Search 3 uses edge information as features. This enables highly robust and fast detection despite environmental variations including shading, reflections, lighting, shape deformations, pose, and noise.

Additional Information

- Shape Search 3 will not work until a Shape Search 3 Model has been created. Refer to *Shape Search 3 Model* on page 8-244 for more information.
- Shape Search 3 typically provides shorter execution times than Locator, but it cannot provide disambiguation between multiple similar models.
- Shape Search 3 is useful for simple models while Locator is better for handling multiple models or situations where the model training process requires user control.



To create a Shape Search 3 tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Finder** and then **Shape Search 3**. A Shape Search 3 tool will be added to the Vision Tools list.

● Shape Search 3 Configuration Items

Use the table below to understand the Shape Search 3 configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Shape Search 3 Model	Select the Shape Search 3 Model that will be searched for in the image.
	Results Logging	Enables logging of tool results to a file.

Group	Item	Description
Region Of Interest	Relative To	
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Rectangles Ellipses Circumferences Polygons	Sets of regions of interest of a particular shape. The field will show how many regions exist ("1-item collection", "2-item collect, etc.). Click the plus sign in the field to create additional regions and click the arrow to see all regions.
	<region number>	Identifies the region of the specified shape. It will appear as [0], [1], [...] within the shape category. Click the minus sign in the field to remove the region and click the arrow to see its properties.
	Name	Available in all region types. Sets the user-defined name of the region.
	Overlap	Available in all region types. Select the behavior of the region. OR causes the region to be included in the search area and NOT excludes it from the search area. The behavior of specific regions is further defined in Z-Order. Refer to *Tools with Multiple Regions of Interest* on page 8-195 for more information.
	Z-Order	Available in all region types. Sets the order for overlapping regions of interest. Higher numbers will be in front and resolved first. Refer to *Tools with Multiple Regions of Interest* on page 8-195 for more information.
	Offset	Available in all region types except for Polygons. Defines the coordinates of the region's center.
	Search Region (Width, Height)	Only available in Rectangles. Sets the height and width of the rectangular region.
	Radius X/Y	Only available in Ellipses. Defines the distance from the center to the exterior along the X- and Y-axes in the camera coordinate system respectively.
	Radius	Only available in Circumferences. Defines the distance from the center to the unseen center ring.
	Thickness	Only available in Circumferences. Defines the distance between the two visible rings.
	Vertexes	Only available in Polygons. Sets of vertexes in a polygon region of interest. The field will show how many vertexes exist ("3-item collection", "4-item collection, etc.). Click the plus line in the field to create additional regions and click the arrow to see all regions.
	<vertex number>	Only available in Polygons. Identifies a particular vertex in the region. It will appear as [0], [1], [2], [...] and etc. The field defines the X/Y coordinates of the vertex. Click the minus sign to remove the vertex, as long as there are not fewer than three.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once.

Group	Item	Description
Properties	Measurement Condition - Candidate Level	Set the percentage of match required for the tool to detect instances. Any instance that has a Correlation Results value lower than this value will not be recognized.
	Rotation Angle Range	Set the angle range within which the tool will detect candidates.
	Correlation Range	Set the range of correlation values within which the tool will return results. This operates as min-max properties.
	Multiple Output	Set the maximum number of output instances and the order in which they are sorted.
	Count	Sub-property of Multiple Output. Enter the maximum number of instances.
	Sort Condition	Sub-property of Multiple Output. Select the necessary order of instance sorting.
	Edge Level	Enable automatic edge level adjustment or enter a value to set it manually.
	Overlay Judgment	Prevent the tool from detecting overlapped instances. Increased value will result in overlapped instances to be removed.
	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Show Corresponding Model	Highlights the edges of detected models in the tool window.
	Show Edge Image	Shows only the detected edges of each model. All edge pixels become white and all others are set to black.
Advanced Properties	Acceptable Distortion Level	Select the degree of influence of correlation values when model edge has small uneven patterns. To avoid reduction of correlation value, set this item to High.
	Complex Background	Stabilizes measurement result when there are many edges in the model background.

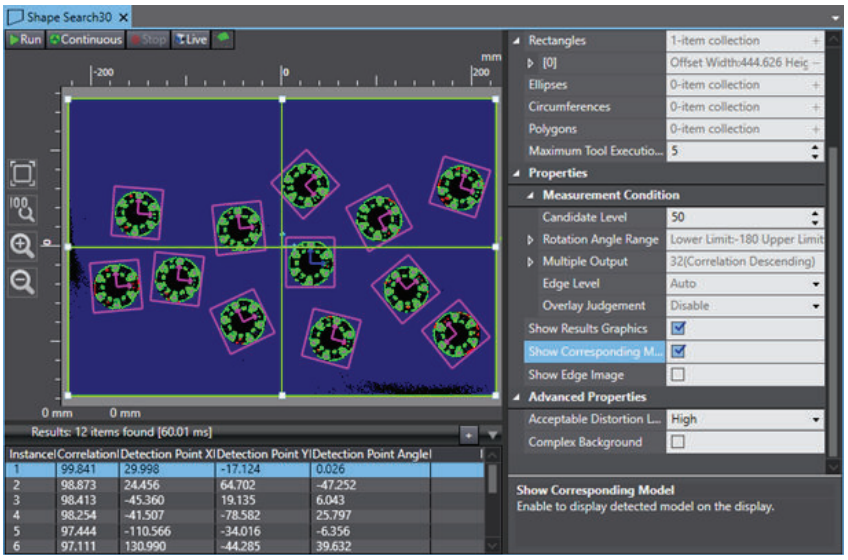
● Shape Search 3 - Detecting Models

Shape Search 3 must reference a Shape Search 3 Model in the Shape Search 3 Model property. This is done by clicking the ellipsis next to the field and selecting an appropriate tool. Shape Search 3 will then compare the model to the image and search within the region(s) of interest for contours that match the model.

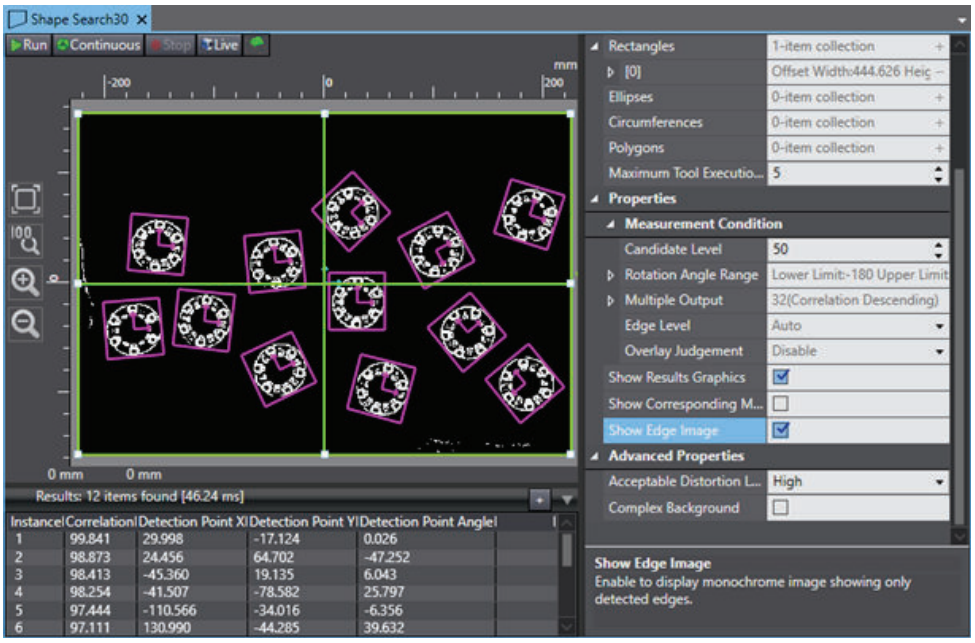
● Shape Search 3 - Image View

Shape Search 3 always displays the model region(s) of interest and origin in purple around the detected instances. It also allows different ways to view the detected instances. These are controlled by the Show Corresponding Model and Show Edge Image properties. If both of these are disabled, only the border of the model region(s) of interest will be displayed, as shown in Figure 1-1above. Enabling these provides a different view.

Show Corresponding Model shows all of the edges drawn in the Shape Search 3 Model in green. It also applies a darkened mask to everything except the pixels within the model edges (refer to the following figure).



Show Edge Image changes everything to black except for the edges detected in the image, which become white (refer to the following figure).



Additional Information

These views are overlaid if both are selected.

● Shape Search 3 Results

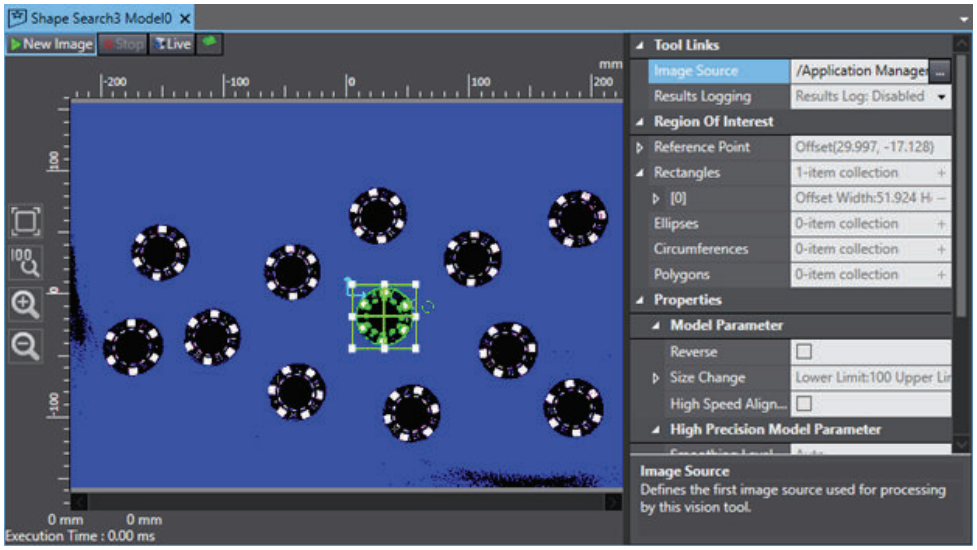
Use the table below to understand the results of the Shape Search 3.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To. This will only be different if Shape Search 3 is set relative to another tool.
Correlation	Percentage the instance correlates with the model. Any instance will be omitted if this value is lower than the defined Candidate Level property.
Detection Point X	X-coordinate of the instance origin.
Detection Point Y	Y-coordinate of the instance origin.

Item	Description
Detection Point Angle	Angle of the instance origin.

Shape Search 3 Model

This model describes the geometry of an object to be found by the Shape Search 3 tool. Shape Search 3 Model is designed to detect specific edges in an image and register them.
Before running the tool, the region of interest must be verified in the correct location so the model can be properly trained.



To create a Shape Search 3 Model, right-click **Vision Tools** in the Multiview Explorer, select **Add Finder** and then **Shape Search 3 Model**. A Shape Search 3 Model will be added to the Vision Tools list.

● Shape Search 3 Model Configuration Items

Use the table below to understand the Shape Search 3 Model configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Reference Point	Define the model origin. This will be referenced as the instance center in the Shape Search 3 tool.
	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.

Group	Item	Description
Region of Interest	Rectangles Ellipses Circumferences Polygons WideArcs	Sets of regions of interest of a particular shape. The field will show how many regions exist ("1-item collection", "2-item collect", etc). Click the plus sign in the field to create additional regions and click the arrow to see all regions.
	<region number>	Identifies the region of a particular shape. It will appear as [0], [1], [...] within each region. Click the minus sign in the field to remove the region and click the arrow to see its properties.
	Name	Available in all region types. Sets the user-defined name of the region.
	Overlap	Available in all region types. Select the behavior of the region. OR causes the region to be included in the search area and NOT excludes it from the search area. The behavior of specific regions is further defined in Z-Order. Refer to *Tools with Multiple Regions of Interest* on page 8-195 for more information.
	Z-Order	Available in all region types. Sets the order for overlapping regions of interest. Higher numbers will be in front and resolved first. Refer to *Tools with Multiple Regions of Interest* on page 8-195 for more information.
	Offset	Available in all region types except for Polygons. Defines the coordinates of the region's center.
	Search Region (Width, Height)	Sets the height and width of the rectangular region. Only available in Rectangles.
	Radius X/Y	Defines the distance from the center to the exterior along the X- and Y-axes, respectively. Only available in Ellipses.
	Radius	Defines the distance from the center to the unseen center ring. Only available in Circumferences and WideArcs.
	Thickness	Defines the distance between the two visible rings. Only available in Circumferences and WideArcs.
	Start/End Angle	Defines the start and end angle of the Wide Arc bounds. Angles are measured in degrees counterclockwise from the positive X-axis. The arc is created clockwise starting from the Start Angle and ending at the End Angle. Only available in WideArcs.
	Vertexes	Sets of vertexes in a polygon region of interest. The field will show how many vertexes exist ("3-item collection", "4-item collection', etc). Click the plus line in the field to create additional regions and click the arrow to see all regions. Only available in Polygons.
	<vertex number>	Identifies a particular vertex in the region. It will appear as [0], [1], [2], [...]. The field defines the X/Y coordinates of the vertex. Click the minus sign to remove the vertex, as long as there are not fewer than three. Only available in Polygons.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once.

Group	Item	Description
Properties	Model Parameter - Reverse	Enables detection of objects whose dark areas and bright areas are fluctuating due to glossiness or shine.
	Model Parameter - Size Change	Set the upper and lower limit of model size fluctuation.
	Model Parameter - High Speed Alignment Mode	Enables detection of simple shapes such as alignment marks. Disable this when detecting complex patterns or shapes with thin elements or lines.
	High Precision Model Parameter - Smoothing Level	Set the level of smoothing for the High Precision Edge Model. Enabling the Auto box causes the tool to automatically set it as necessary. Disabling it allows the user to adjust it manually. Higher values remove more details from the high precision model.
	Edge Settings - Mask Size	Select the neighborhood of pixels to use to detect model edges. Higher selections will help detection when brightness varies among pixels.
	Edge Settings - Edge Level	Set the lower limit of edge level for an edge to be recognized. Edges with a higher edge level than this value will be included in the model. Higher settings will result in fewer edges.
	Edge Settings - Noise Removal Level	Set the upper limit of noise level to eliminate. Noise with a level below this value will be eliminated. Higher numbers will lead to more features being removed from the model.
	Edge Settings - Show Edge Model	Specifies if the Edge Model is drawn in the Vision Window.
	Edge Settings - Show High Precision Edge Model	Specifies if the High Precision Edge Model is drawn in the Vision Window.

● Shape Search 3 Model - Training a Model

Shape Search 3 Model automatically runs whenever any change is made to the parameters or region(s) of interest. Because of this, the training process only requires positioning the reference point and the region(s) of interest in the necessary location(s). As many regions can be added as necessary to create a custom model shape. The Overlap can further define a custom model.
When the trained model is used in Shape Search 3, the position result for each detected instance will be returned as the position of the model Reference Point property. Therefore, before continuing, the location of the reference point needs to be verified.
Shape Search 3 Model has fewer controlling parameters than Locator Model since the focus of Shape Search 3 is speed. However, several properties provide control over this tool. In particular, the parameters in the Edge Settings section are useful to adjust the model. Mask Size helps balance brightness issues, Edge Level sets the necessary deviation of edges in order to be included, and Noise Removal Level removes unnecessary features.

● Shape Search 3 Results

Use the table below to understand the results of the Shape Search 3.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To. This will only be different if Shape Search 3 is set relative to another tool.
Correlation	Percentage the instance correlates with the model. Any instance will be omitted if this value is lower than the defined Candidate Level property.
Detection Point X	X-coordinate of the instance origin.

Item	Description
Detection Pint Y	Y-coordinate of the instance origin.
Detection Point Angle	Angle of the instance origin.

8-8-6 Inspection Tools

Inspection tools are typically used for part inspection purposes.
The following Inspection tools are described in this section.

- Refer to *Arc Caliper* on page 8-247 for more information.
- Refer to *Arc Edge Locator* on page 8-247 for more information.
- Refer to *Caliper* on page 8-252 for more information.
- Refer to *Color Data* on page 8-256 for more information.
- Refer to *Edge Locator* on page 8-259 for more information.
- Refer to *Feeder Histogram* on page 8-263 for more information.
- Refer to *Image Histogram* on page 8-265 for more information.
- Refer to *Inspection* on page 8-268 for more information.
- Refer to *Precise Defect Path* on page 8-272 for more information.
- Refer to *Precise Defect Region* on page 8-276 for more information.

Arc Caliper

This tool identifies and measures the gap between one or more edge pairs of arc-shaped objects. Using pixel gray-level values within regions of interest, Arc Caliper is able to build projections needed for edge detection. Edges can be displayed in a radial or annular position. After detecting potential edges, the tool determines which edge pairs are valid by applying the user-defined constraints configured for each one. The valid pairs are then scored and measured.

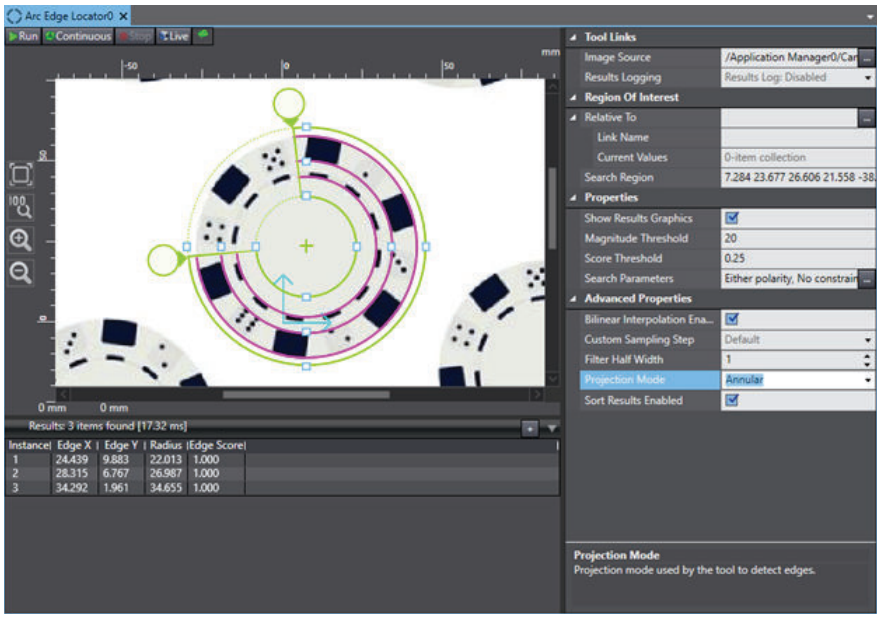
Arc Edge Locator

This tool identifies and measures the position of one or more edges on a circular object. Using pixel gray-level values within regions of interest, Arc Edge Locator is able to build projections needed for edge detection. Edges can be displayed in a radial or annular position. After detecting potential edges, the tool determines which edges are valid by applying the user-defined constraints on the edge candidates. The valid edges are then scored and measured.



Additional Information

While Arc Edge Locator can determine the position of one or more edges, it cannot measure the length of lines detected in the region of interest. To measure arcs and lines on an object, use the Arc Finder or Line Finder tools. Refer to *Arc Finder* on page 8-199 and *Line Finder* on page 8-217 for more information.



To create an Arc Edge Locator tool, right-click **Vision Tools** in the Multiview Explorer, select **AddInspection** and then **Arc Edge Locator**. An Arc Edge Locator tool will be added to the Vision Tools list.

● Arc Finder Configuration Items

Use the table below to understand the Arc Finder tool configuration items.

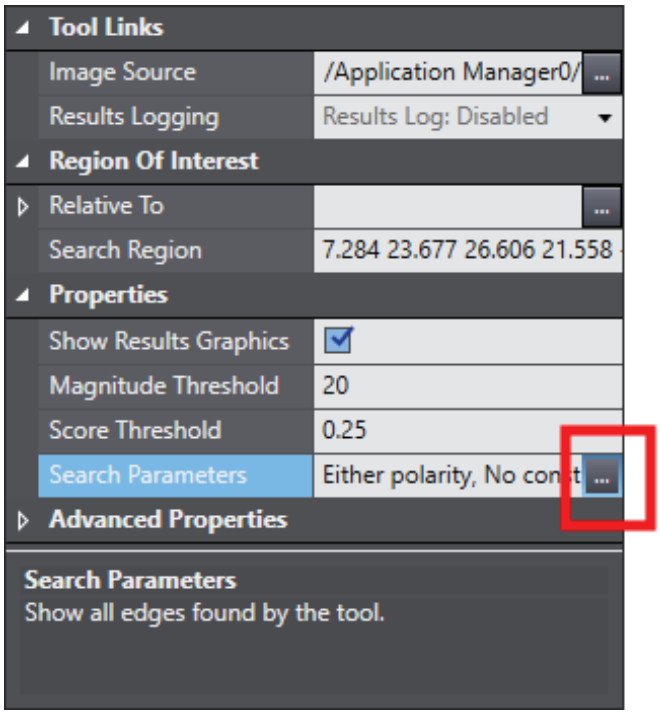
Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Search Region	Defines the location and size of the region (X, Y, radius, thickness, mid-angle position, arc angle degrees).
	Guideline Percent	Adjusts the location of the orange guideline in the region of interest. Move the slider left to position the guideline closer to the center and to the right to position it farther away.

Group	Item	Description
Region of Interest	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Fit Mode	Select how the tool will calculate and return a valid arc from hypotheses. • Both: calculates the arc center and arc radius and uses both values to return the most accurate results. • Center: calculates the arc center. • Radius: calculates the arc radius.
	Minimum Arc Percentage	Specifies the minimum percentage of arc contours that need to be matched for a hypothesis to be considered valid.
	Polarity Mode	Select the polarity deviation required for a hypothesis to be considered valid. • Dark to Light: only detect arcs that are darker towards the center. • Light to Dark: only detect arcs that are lighter towards the center. • Either: accept any polarity as long as it is consistent. • Do Not Care: does not take polarity into account.
	Search Mode	Select the method used to generate and select a hypothesis. • Best Arc: chooses the best quality arc but will increase processing time. • Arc Closest To Guideline: chooses the arc closest to the guideline. • Arc Closest To Inside: chooses the arc closest to the inner circle. • Arc Closest To Outside: chooses the arc closest to the outer arc of the region of interest.

Group	Item	Description
Advanced Properties	Arc Must Be Totally Enclosed	Specifies if the detected arc can exist outside of the region of interest. When enabled, the start and end points of the arc must be located on the angle boundary lines. Otherwise, the arc can enter or exit the region at any point.
	Conformity Tolerance	Set the maximum local deviation between the expected arc contours and the arc contours actually detected in the input image.
	Contrast Threshold	Defines the minimum contrast needed for an edge to be detected in the input image. This threshold is expressed in terms of a step in gray level values. Higher values reduce noise but also reduce the number of contours detected.
	Maximum Angle Deviation	Defines the maximum angle deviation between the calculated arc's tangents and the edge contours.
	Positioning Level	Set the configurable effort level of the instance positioning process. A value of 10 will provide coarser positioning and lower execution time. Conversely, a value of 100 will provide high accuracy positioning of arc entities. The setting range is 10 to 100.
	Subsampling Level	Set the subsampling level used to detect edges that are used by the tool to generate hypotheses. High values provide a coarser search and lower execution time than lower values. The setting range is 1 to 8.

● Arc Edge Locator - Edge Constraint Editor

Clicking the ellipsis next to Search Parameters, as shown in the following figure, opens the Edge Constraint Editor. This is used to set specific constraints on the detected edges.



The Editor has two different sections named Edges and Region Image as described below.

■ Edges

This section defines the constraints on the detected edges. Constraints affect all detected edges with the following properties.

- Polarity: defines the gray level deviation for which the tool searches. This is performed with respect to the clockwise direction of the region of interest. For example, in the following figure, Either is selected, and dark and light areas can be seen on different sides of multiple edges.
- Constraints: enables constraining the edge by both Position and Magnitude. A slider bar appears below the image in both cases. The edge must be between the two position sliders and its magnitude must be higher than defined in the magnitude slider (refer to the following figure for an example).
- Score Threshold: defines the minimum score (quality) an edge must have to be considered valid. The value is set between 0 and 1.

■ Region Image

The image beneath the edge constraints shows only the pixels within the region of interest in the main image window. The image is altered from the region of interest so that it appears rectangular instead of circular. Regardless of the region's orientation in the main image, this will always be shown with the region's positive X-axis oriented to the right.

■ Arc Edge Locator Results

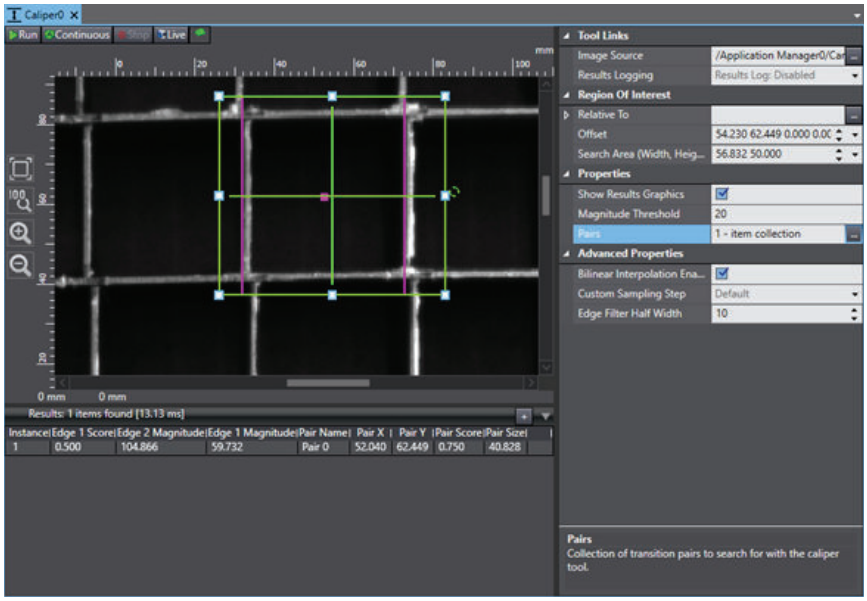
Use the table below to understand the results of the Arc Edge Locator tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Edge X	X-coordinate of the edge midpoint.
Edge Y	Y-coordinate of the edge midpoint.

Item	Description
Radius	Radius of the edge. This is only properly measured when Projection Mode is set to Annular.
Edge Score	Calculated score (quality) of the edge, computed according to the constraints set by the Edge Constraints properties.
Magnitude	Measurement of how well the region of interest arc or radii matches the found arc or radii. This will be returned as negative if the found arc is a reflection of the region of interest arc.
Magnitude Score	Score between 0 and 1 calculated according to the Magnitude Constraint property.
Position Score	Score between 0 and 1 calculated according to the Position Constraint property.
Projection Magnitude	Measurement of the deviation between the gray level of the projection pixels and the pixels surrounding it. This is returned on a range between -255 and 255. Positive and negative peaks in the value indicate potential edges. Sharp peaks indicate strong, well-defined edges while dull peaks may indicate noise or poorly-defined edges.
Projection Magnitude	Average gray 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.

Caliper

This tool identifies and measures the distance between one or more parallel edge pairs in an image. It uses pixel gray-level values within the region of interest to build projections needed for edge detection. After the potential edges are detected, Caliper determines which edge pairs are valid by applying constraints that are manually configured for each edge pair.



To create a Caliper tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Inspection** and then **Caliper**. A Caliper tool will be added to the Vision Tools list.

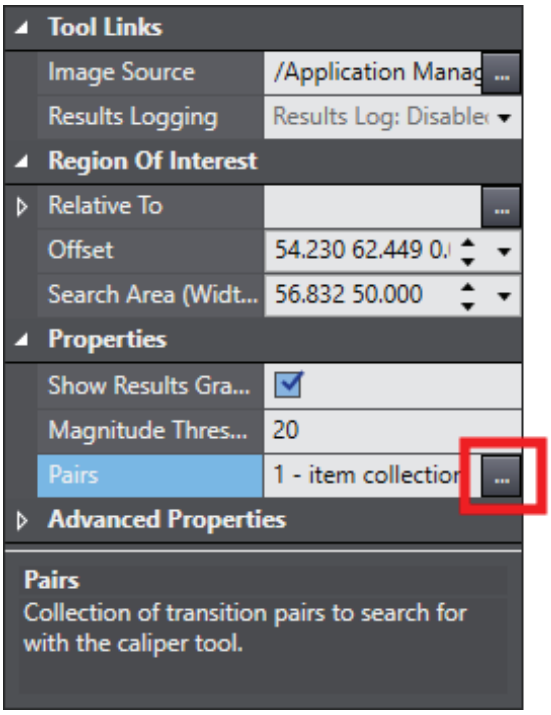
● Caliper Configuration Items

Use the table below to understand the Caliper configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the center coordinates of the region.
	Search Area (Width, Height)	Defines the size of the region.
	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Magnitude Threshold	Specifies the minimum gray level deviation required to detect an edge. A subpixel peak detection algorithm is applied on the region of every minimum or maximum of the curve that exceeds this threshold in order to locate edges.
	Pairs	Collection of transition pairs to search for with the caliper tool. Click the ellipsis to make detailed changes. Refer to *Caliper - Edge Constraint Editor* on page 8-253 for more information.
Advanced Properties	Bilinear Interpolation Enabled	Enables using bilinear interpolation to sample the input image. By default, this is enabled because it ensures subpixel accuracy.
	Custom Sampling Step	Defines the sampling step used in calculation. This is set to 1 by default. Enable this to adjust the setting. Higher values decrease execution time and sensitivity and can improve processing in high-resolution images.
	Edge Filter Half Width	Half width of the convolution filter used to compute the edge magnitude curve, from which actual edges are detected. The filter approximates the first derivative of the projection curve. The half width of the filter should be set in order to match the width of the edge in the projection curve (the extent of the gray scale transition, expressed in number of pixels).

● Caliper - Edge Constraint Editor

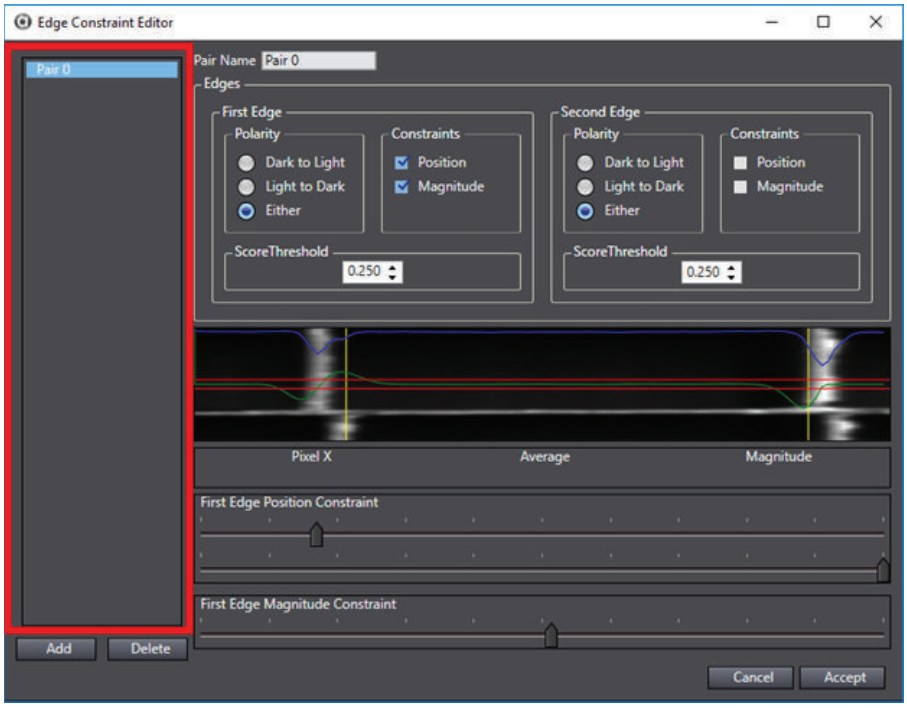
Clicking the ellipsis next to Pairs opens the Edge Constraint Editor. Use this to set specific constraints on the detected pair(s).



The editor has several different sections that are described below.

■ Pairs Pane

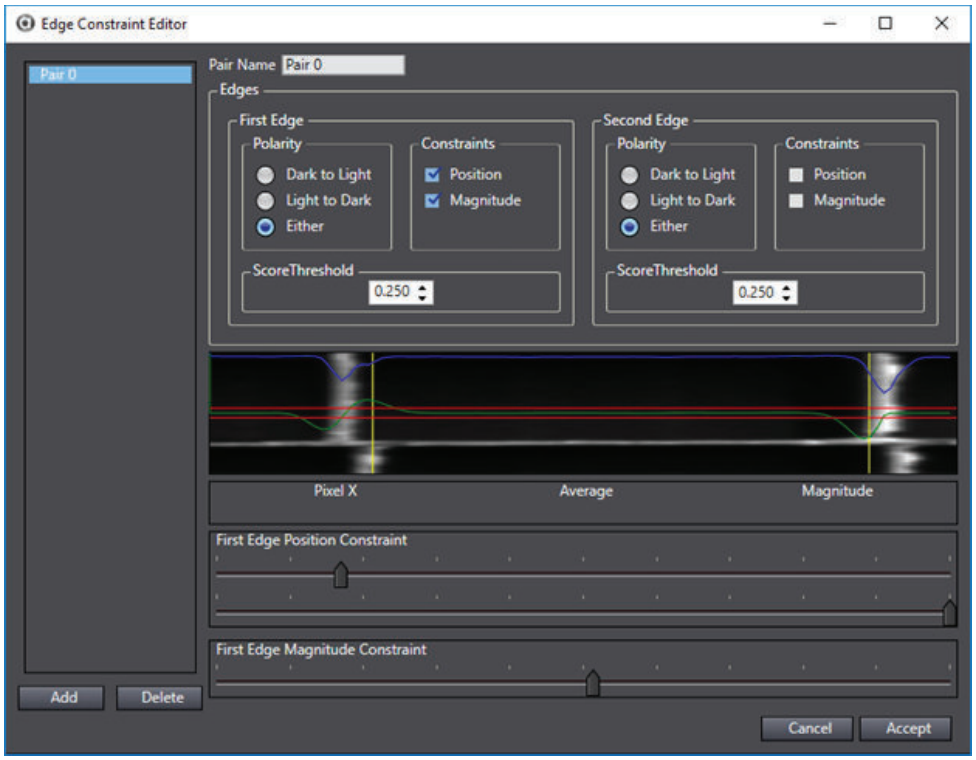
The area highlighted in the figure below lists the pairs of edges to be detected. By default, Caliper will try to detect only one. The Add and Delete buttons at the bottom can change the amount of pairs detected when the tool is run. The name of each pair is adjusted by selecting the pair in this pane and then changing the Pair Name field.



■ Edges

This section defines the constraints on the detected edges within a pair. Each edge is adjusted individually with the following properties.

- Polarity: defines the gray level deviation for which the tool searches. This is performed with respect to moving from left to right across the region (shown in an image in this editor). For example, in the following figure, *Either* is selected, and dark and light areas can be seen on different sides of the two edges.
- Constraints: enables constraining the edge by both Position and Magnitude. A slider bar appears below the image in both cases. The edge must be between the two position sliders and its magnitude must be higher than defined in the magnitude slider (refer to the following figure for an example).
- Score Threshold: defines the minimum score (quality) an edge must have to pass. This value is set between 0 and 1.



■ Region Image

The image beneath the edge constraints shows only the pixels within the region of interest in the main image window. Regardless of the region's orientation in the main image, this will always be shown with the region's positive X-axis oriented to the right.

● Caliper Results

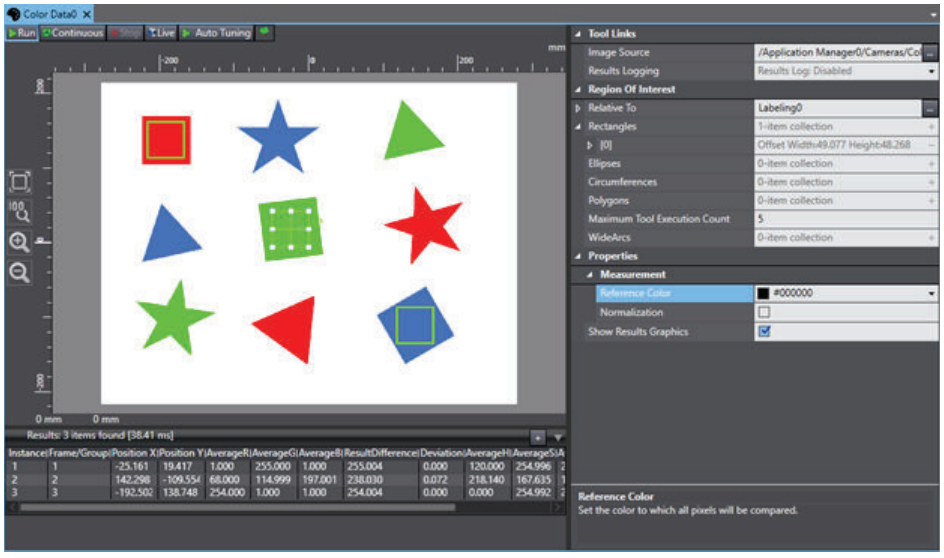
Use the table below to understand the results of the Caliper tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Pair Name	Name of the edge pair.
Pair X	X-coordinate of the center point of the caliper measure at the midpoint of the edge pair.
Pair Y	Y-coordinate of the center point of the caliper measure at the midpoint of the edge pair.
Pair Score	Score of the resultant pair, which is equal to the mean score of the two edges in the pair.

Item	Description
Pair Size	Distance between the midpoints of the pair edges.
Edge 1 X	X-coordinate of the midpoint of the first edge.
Edge 1 Y	Y-coordinate of the midpoint of the first edge.
Edge 1 Magnitude	Magnitude of the first edge.
Edge 1 Magnitude Score	Magnitude score of the first edge.
Edge 1 Position Score	Position score of the first edge.
Edge 1 Score	Score of the first edge, computed according to the constraints set by the Edge Constraints Properties.
Edge 2 X	X-coordinate of the midpoint of the second edge.
Edge 2 Y	Y-coordinate of the midpoint of the second edge.
Edge 2 Magnitude	Magnitude of the second edge.
Edge 2 Magnitude Score	Magnitude score of the second edge.
Edge 2 Position Score	Position score of the second edge.
Edge 2 Score	Score of the second edge, computed according to the constraints set by the Edge Constraints Properties.

Color Data

This tool finds the average color within a region and performs statistical analysis using the deviation from a user-defined reference color and color variation of the measurement range. It is primarily used to obtain data that will be analyzed by an Inspection tool.



To create a Color Data tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Inspection** and then **Color Data**. A Color Data tool will be added to the Vision Tools list.

● Color Data Configuration Items

Use the table below to understand the Color Data configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Rectangles Ellipses Circumferences Polygons WideArcs	Sets of regions of interest of a particular shape. The field will show how many regions exist ("1-item collection", "2-item collect", etc). Click the plus sign in the field to create additional regions and click the arrow to see all regions.
	<region number>	Identifies the region of a particular shape. It will appear as [0], [1], [...] within each region type. Click the minus sign in the field to remove the region and click the arrow to see its properties.
	Name	Available in all region types. Sets the user-defined name of the region.
	Overlap	Available in all region types. Select the behavior of the region. OR causes the region to be included in the search area and NOT excludes it from the search area. The behavior of specific regions is further defined in Z-Order. Refer to *Tools with Multiple Regions of Interest* on page 8-195 for more information.
	Z-Order	Available in all region types. Sets the order for overlapping regions of interest. Higher numbers will be in front and resolved first. Refer to *Tools with Multiple Regions of Interest* on page 8-195 for more information.
	Offset	Available in all region types except for Polygons. Defines the coordinates of the region center.
	Search Region (Width, Height)	Sets the height and width of the rectangular region. Only available in Rectangles.
	Radius X/Y	Defines the distance from the center to the exterior along the X- and Y-axes, respectively. Only available in Ellipses.
	Radius	Defines the distance from the center to the unseen center ring. Only available in Circumferences and WideArcs.
	Thickness	Defines the distance between the two visible rings. Only available in Circumferences and WideArcs.
	Start/End Angle	Defines the start and end angle of the Wide Arc bounds. Angles are measured in degrees counterclockwise from the positive X-axis. The arc is created clockwise starting from the Start Angle and ending at the End Angle. Only available in WideArcs.
	Vertexes	Sets of vertexes in a polygon region of interest. The field will show how many vertexes exist ("3-item collection", "4-item collection", etc). Click the plus line in the field to create additional regions and click the arrow to see all regions. Only available in Polygons.
	<vertex number>	Identifies a particular vertex in the region. It will appear as [0], [1], [2], [...]. The field defines the X/Y coordinates of the vertex. Click the minus sign to remove the vertex, as long as there will not be fewer than three. Only available in Polygons.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once.

Group	Item	Description
Properties (Measurement)	Reference Color	Defines the color that will be compared to all pixels in the search area. The Auto Tuning button sets this property based on what is currently in the region of interest.
	Normalization	Specify whether to normalize the brightness in calculating the color difference. When enabled, the result is not affected by the total brightness and only the color tone can be detected.
	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.

● Color Data - Measuring Color Data

The colors in a region are measured against a reference color, which can be set in two ways. If there is a specific color against which the pixels need to be measured, it can be chosen manually using the Reference Color property. Alternatively, in color images, a particular color can be highlighted within the search area. Click the **Auto Tuning** Button to automatically set this as the Reference Color. All results will then be measured against it.

The ResultDifference result illustrates the difference between the measured color range and the Reference Color by using the following formula.

=√((AverageR-Reference R)2+(AverageG-Reference G)2+(AverageB-Reference B)2)

This tool is particularly useful when used in conjunction with Inspection tool. When the Mode Of Operation for an Inspection Filter is set to Color Data, the Inspection tool can determine whether Color Data tool results fall within a specified range, allowing instances to be categorized by hue. Refer to Inspection on page 1 for more information.



Additional Information

Color Data accuracy is dependent on the camera white balance settings, which may need to be modified outside of the ACE software.

● Color Data Results

Use the table below to understand the results of the Color Data tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Position X	X-coordinate of the reference point. If Color Data is set relative to another tool, this will be the X-coordinate of the associated instance. Otherwise, it will be the origin of the image field of view.
Position Y	Y-coordinate of the reference point. If Color Data is set relative to another tool, this will be the Y-coordinate of the associated instance. Otherwise, it will be the origin of the image field of view.
AverageR	Average R (Red) value of color within the search area.
AverageG	Average G (Green) value of color within the search area.
AverageB	Average B (Blue) value of color within the search area.
ResultDifference	Color difference between the average color in the measurement area and the Reference Color.
Deviation	Color deviation in the measurement region.
AverageH	Average H (Hue) value within the search area.
AverageS	Average S (Saturation) value within the search area.
AverageV	Average V (Value) value within the search area.

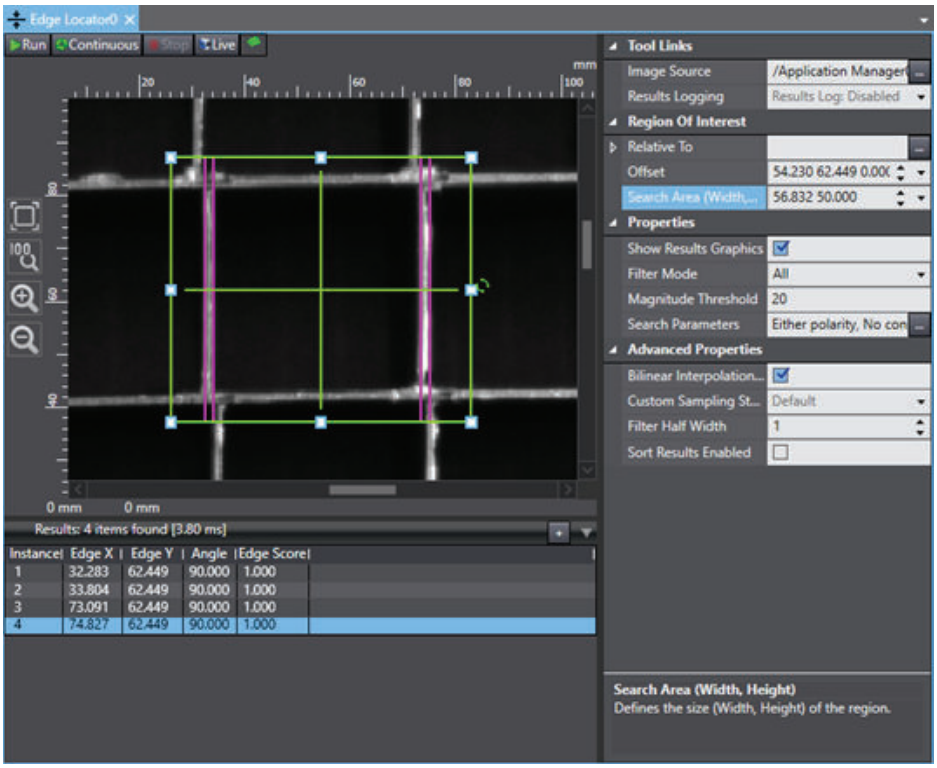
Item	Description
MonAve	Only calculated in gray scale images. Average gray level value within the search area.
MonDev	Only calculated in gray scale images. Gray level deviation in the measurement region.

Edge Locator

This tool identifies and measures the position of one or more straight edges on an object. Using pixel gray-level values within regions of interest, Edge Locator is able to build projections needed for edge detection. After detecting potential edges, the tool determines which edges are valid by applying the user-defined constraints on the edge candidates. The valid edges are then measured to determine score (quality), length, and position.

Additional Information

While Edge Locator can determine the position of one or more edges, it cannot measure the length of lines detected in the region of interest. To extrapolate and measure a line on an object, use the Line Finder tool. Refer to *Line Finder* on page 8-217 for more information.



To create an Edge Locator tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Inspection** and then **Edge Locator**. An Edge Locator tool will be added to the Vision Tools list.

● Edge Locator Configuration Items

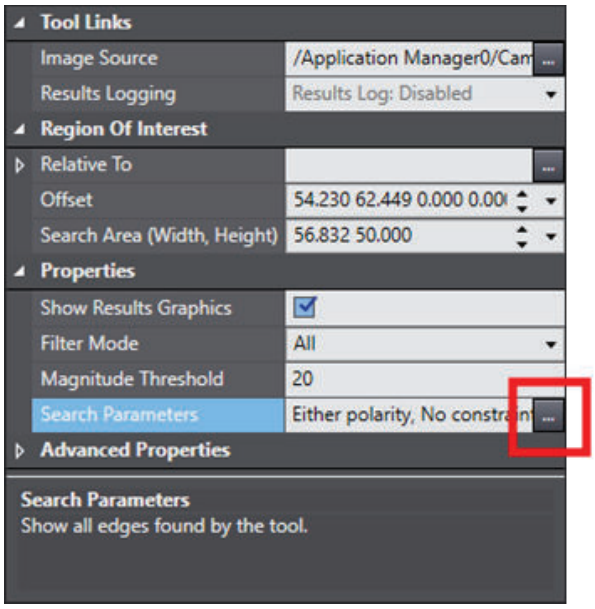
Use the table below to understand the Edge Locator configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.

Group	Item	Description
Region of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the center coordinates of the region.
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Filter Mode	Select the edge results to be returned. Measured in the positive direction of the X-axis of the region of interest • **All** returns all detected results. • **First Edge** returns the first edge detected. • **Last Edge** returns the last edge detected. • **Middle** returns the line that is halfway between the first and last edges.
	Magnitude Threshold	Specifies the minimum gray level deviation required to detect an edge. A sub-pixel peak detection algorithm is applied on the region of every minimum or maximum of the curve that exceeds this threshold in order to locate edges.
	Search Parameters	Sets the constraints and thresholds calculated when determining the validity of the edge. Click the ellipsis to make detailed changes. Refer to *Edge Locator - Edge Constraint Editor* on page 8-260 for more information.
Advanced Properties	Bilinear Interpolation Enabled	Enables using bilinear interpolation to sample the input image. By default, this is enabled because it ensures subpixel accuracy.
	Custom Sampling Set	Defines the sampling step used in calculation. This is set to 1 by default. Enable this to adjust the setting.
	Filter Half Width	Half width of the convolution filter used to compute the edge magnitude curve, from which actual edges are detected. The filter approximates the first derivative of the projection curve. The half width of the filter should be set in order to match the width of the edge in the projection curve (the extent of the gray scale transition, expressed in number of pixels).
	Sort Results Enabled	Enables the results to be sorted.

● Edge Locator - Edge Constraint Editor

Clicking the ellipsis next to Search Parameters opens the Edge Constraint Editor. Use this to set specific constraints on the detected edges.

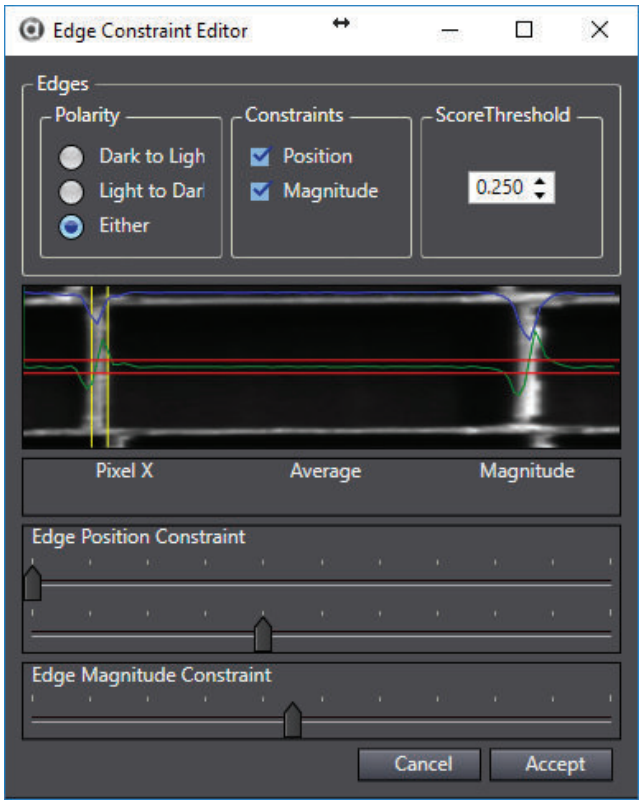


The Editor has two different sections named Edges and Region Image as described below.

■ Edges

This section defines the constraints on the detected edges. Constraints affect all detected edges with the following properties.

- Polarity: defines the gray level deviation for which the tool searches. This is performed with respect to the clockwise direction of the region of interest. For example, in the following figure, *Either* is selected, and dark and light areas can be seen on different sides of multiple edges.
- Constraints: enables constraining the edge by both Position and Magnitude. A slider bar appears below the image in both cases. The edge must be between the two position sliders and its magnitude must be higher than defined in the magnitude slider (refer to the following figure for an example).
- Score Threshold: defines the minimum score (quality) an edge must have to be considered valid. The value is set between 0 and 1.



■ Region Image

The image beneath the edge constraints shows only the pixels within the region of interest in the main image window. The image is altered from the region of interest so that it appears rectangular instead of circular. Regardless of the region's orientation in the main image, this will always be shown with the region's positive X-axis oriented to the right.

● Edge Locator Results

Use the table below to understand the results of the Edge Locator tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Edge X	X-coordinate of the edge midpoint.
Edge Y	Y-coordinate of the edge midpoint.
Angle	Angle of the edge segment with respect to the image X-axis.
Edge Score	Calculated score (quality) of the edge, computed according to the constraints set by the Edge Constraints properties.
Magnitude	Peak value of the edge in the magnitude curve. Negative values indicate a light-to-dark transition while positive values indicate the opposite.
Magnitude Score	Score between 0 and 1 calculated according to the Magnitude Constraint property.
Position Score	Score between 0 and 1 calculated according to the Position Constraint property.
Projection Magnitude	Measurement of the deviation between the gray level of the projection pixels and the pixels surrounding it. This is returned on a range between -255 and 255. Positive and negative peaks in the value indicate potential edges. Sharp peaks indicate strong, well-defined edges while dull peaks may indicate noise or poorly-defined edges.

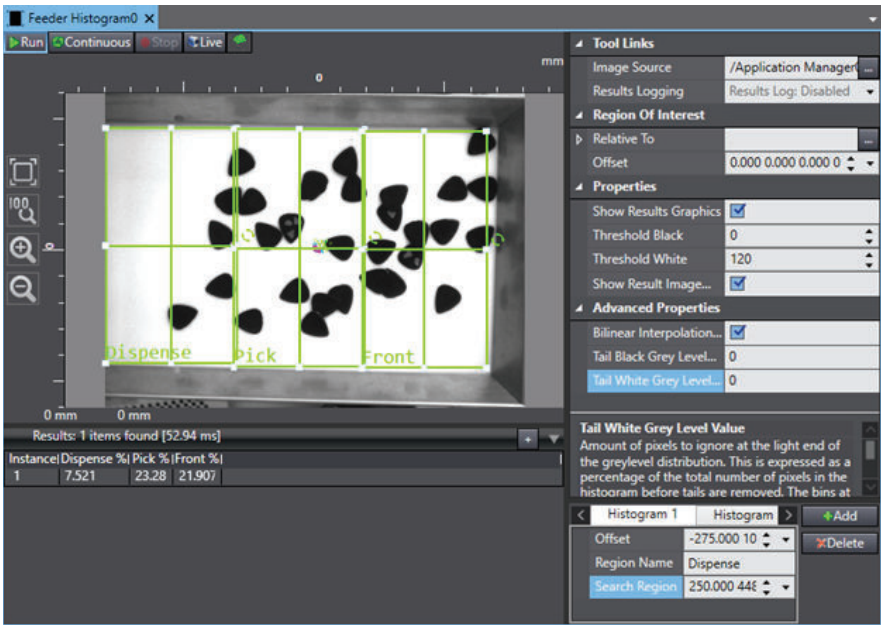
Item	Description
Projection Average	Average gray level value for all projection paths within the physical area bounded by the region of interest. This minimizes variations in pixel values caused by non-edge features or noise.

Feeder Histogram

This tool is used to calculated product density in user-defined regions of interest. It is designed to function in conjunction with an AnyFeeder to identify the density of products within regions associated with the dispense, pick, and front zones. Refer to *8-6-1 AnyFeeder Object* on page 8-82 for more information.

Additional Information

Feeder Histogram does not support color images.



To create a Feeder Histogram tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Finder** and then **Feeder Histogram**. The Feeder Histogram tool will be added to the Vision Tools list.

● Feeder Histogram Configuration Items

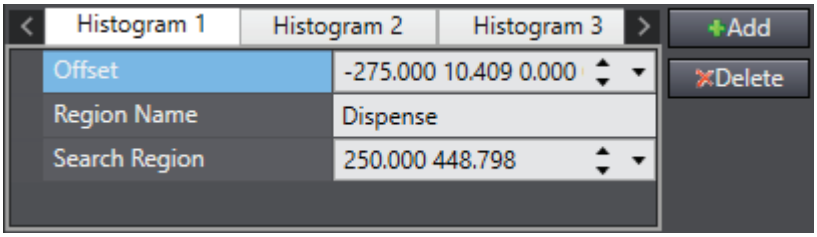
Use the table below to understand the Feeder Histogram configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the relative tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the coordinates of the reference point that the histograms reference.

Group	Item	Description
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Threshold Black	Defines the darkest pixels the histogram considers when running. Any pixels with gray level lower than this value will be ignored while the histogram is building.
	Threshold White	Defines the brightest pixels the histogram considers when running. Any pixels with gray level above higher than this value will be ignored while the histogram is building.
	Show Result Image Histogram Regions	Specifies if the histogram regions are drawn in the ACE Vision Window. Show Results Graphics must be enabled for this to work.
Advanced Properties	Bilinear Interpolation Enabled	Enables using bilinear interpolation to sample the input image. By default, this is enabled because it ensures subpixel accuracy.
	Tail Black Gray Level Value	Percentage of pixels to ignore at the dark end of the gray level distribution. This is calculated after the pixels affected by the Threshold Black property have been removed.
	Tail White Gray Level Value	Percentage of pixels to ignore at the light end of the gray level distribution. This is calculated after the pixels affected by the Threshold White property have been removed.

● Feeder Histogram Pane

Feeder Histogram is designed to create a series of histograms to measure product density in multiple ranges in the image. These histograms are organized using the Histogram Pane positioned beneath the properties, as shown in the figure below.



The **Add** and **Delete** buttons are used to create or remove histograms from the tool. Typically, 2 to 4 zones will be configured depending on the complexity of feeding logic required for the application. The pane also shows properties that apply only to the selected histogram. They are shown in the table below.

Item	Description
Offset	Defines the center coordinates of the histogram region with respect to the reference point defined by Offset in the main tool properties.
Region Name	User-defined name of the histogram. This is displayed in the lower left corner of the histogram region in the region.
Search Region	Defines the size (width, height) of the histogram region.

The product density is calculated for each histogram region using the following formula:
Product Density = Histogram Pixel Count / Image Pixel Count
where Histogram Pixel Count is the number of pixels that fall within the defined thresholds and Image Pixel Count is the number of total pixels in the histogram region.

● Feeder Histogram Results

A Feeder Histogram tool will return one result containing the product densities for each histogram. These are not shown by default and must be added using the Results Column Editor. The numbers of the histograms are variable and are denoted here as <number> instead of actual values.

Additional Information

Results are calculated after applying tails and thresholds.

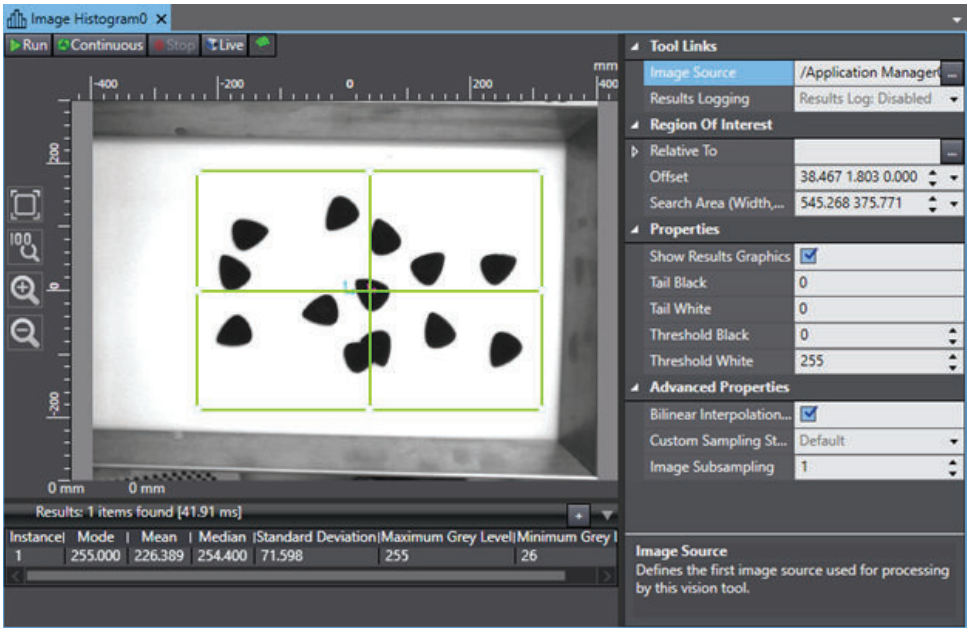
Items	Description
Instance	Index of the result instance.
Histogram <number> %	Product density percentage of the denoted histogram.

Image Histogram

This tool computes image statistics for all the pixels contained in the region of interest. This is used primarily for identifying product or clutter density, verifying the camera or lighting adjustment during setup of an application, and providing input for the Inspection tool.

Additional Information

Image Histogram only functions for gray scale images.



To create an Image Histogram tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Inspection** and then **Image Histogram**. An Image Histogram tool will be added to the Vision Tools list.

● Image Histogram Configuration Items

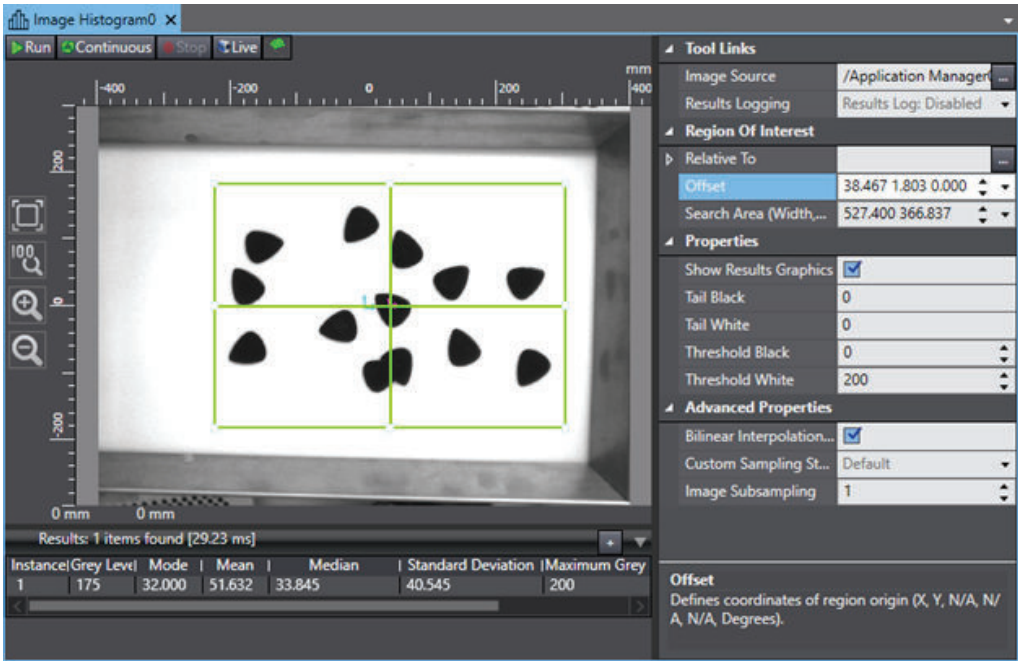
Use the table below to understand the Image Histogram configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the coordinates of the reference point that the histograms reference.
	Search Area (Width, Height)	Defines the size of the region.
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Tail Black	Percentage of pixels to ignore at the dark end of the gray level distribution. This is calculated after the pixels affected by the Threshold Black property have been removed.
	Tail White	Percentage of pixels to ignore at the light end of the gray level distribution. This is calculated after the pixels affected by the Threshold White property have been removed.
	Threshold Black	Defines the darkest pixels the histogram considers when running. Any pixels with gray level lower than this value will be ignored while the histogram is building.
	Threshold White	Defines the brightest pixels the histogram considers when running. Any pixels with gray level above higher than this value will be ignored while the histogram is building.
	Show Result Image Histogram Regions	Specifies if the histogram regions are drawn in the ACE Vision Window. Show Results Graphics must be enabled for this to work.
Advanced Properties	Bilinear Interpolation Enabled	Enables using bilinear interpolation to sample the input image. By default, this is enabled because it ensures subpixel accuracy.
	Custom Sampling Step	Defines the sampling step used in calculation. This is set to 1 by default. Enable this to adjust the setting. Higher values result in higher processing time.
	Image Subsampling	Factor used to subsample the image in the region of interest. The value defines the length (in pixels) of square neighborhoods that will be subsampled. For example, if it is set to 1, there is no subsampling. If it is set to 2, the image is subsampled in 2x2 tiles.

● Image Histogram - Configuration

Image Histogram is designed to return a range of statistics about the pixels within a region of interest. The final calculation ignores pixels that have been excluded by thresholds or tails.
These properties provide a way for the user to ignore pixels in the histogram. Threshold Black and Threshold White set a maximum and minimum gray level to consider when building the histogram. Tail Black and Tail White remove a percentage of pixels from the ends of the spectrum within the thresholds. In this way, the user is able to eliminate results from areas that are not required by the application.

For example, the tool shown in the figure below can be manipulated to only return results about the pixels within the guitar picks by setting the Threshold White property to 200, as shown in the figure below. This removes all pixels with a gray level value higher than 200 from analysis.



● Image Histogram Results

Use the table below to understand the results of the Image Histogram tool.

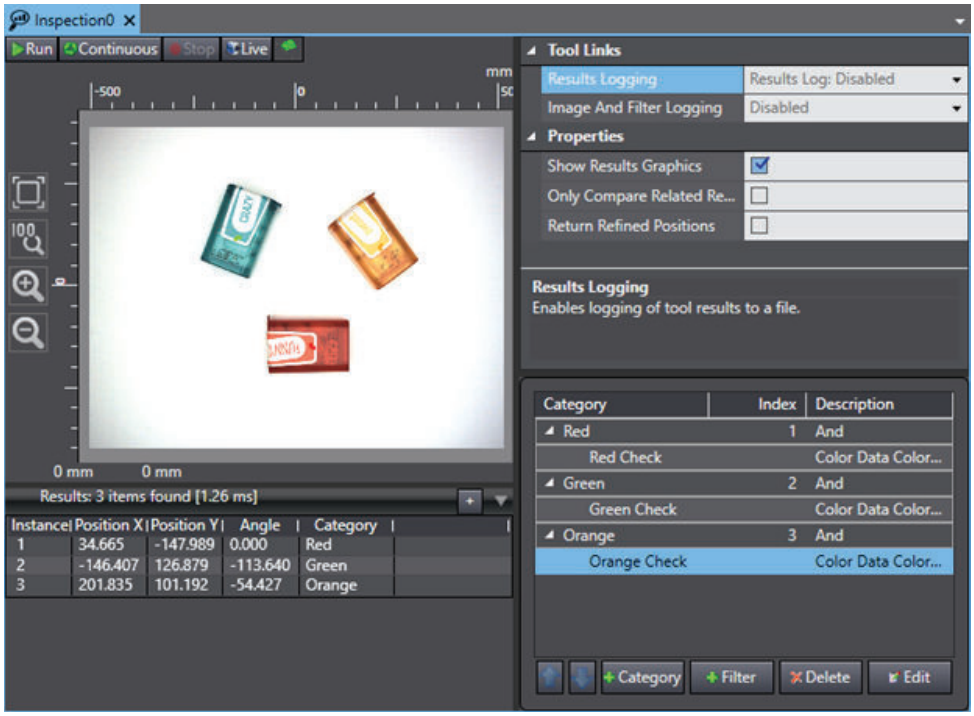
Additional Information

All results are calculated after applying white and black tails and thresholds.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Mode	Gray level value that appears in the largest number of pixels compared to other gray level values.
Mode Pixel Count	Number of pixels that correspond to the Mode gray level value.
Mean	Average gray level distribution of the pixels in the region of interest.
Median	Median of the gray level distribution of the pixels in the region of interest
Standard Deviation	Standard deviation of the gray level distribution of the pixels in the region of interest.
Variance	Variance of the gray level distribution of the pixels in the region of interest.
Minimum Gray Level	Minimum gray level value in the region of interest.
Maximum Gray Level	Maximum gray level value in the region of interest.
Gray Level Range	Difference between the Minimum Gray Level and Maximum Gray Level values.
Histogram Pixel Count	Number of pixels in the region of interest.
Image Width	Width (in pixels) of the region of interest.
Image Height	Height (in pixels) of the region of interest.
Tail Black Gray Level	Darkest gray level value that remains in the histogram after the tail is removed.
Tail White Gray Level	Brightest gray level value that remains in the histogram after the tail is removed.

Inspection

This tool organizes instances based on the results of other tools and inspection filters. Custom categories and filters can be created to identify information about data or images from another tool. In this way, returned instances from another vision tool can be sorted into groups.



To create an Inspection tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Inspection** and then **Inspection**. An Inspection tool will be added to the Vision Tools list.

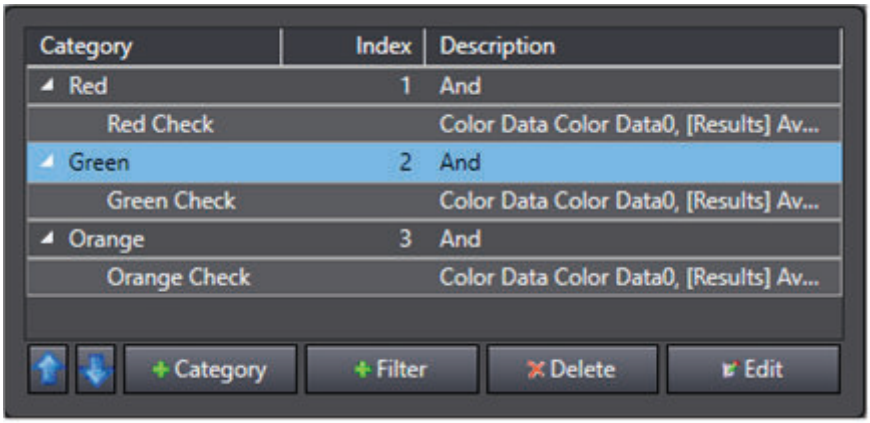
● Inspection Configuration Items

Use the table below to understand the Inspection configuration items.

Group	Item	Description
Tool Links	Results Logging	
	Image And Filter Logging	Enables logging of inspection data to a file. • Log all data: records all numerical data. • Log pictures for inspections that fail: saves images for any failed inspections. • All inspections must fail: records data only if all inspections in the tool have failed. The File Mode dropdown defines how often files are recorded.
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Only Compare Related Results	Specifies which results are compared. Enable to compare only those result values that are connected by instances. When disabled, all results are compared.
	Return Refined Positions	Specifies if instance positions will be adjusted by subsequent tools. When enabled, the input position is determined by the tool directly before this one in the sequence. When disabled, the original position is input.

● Inspection Filter Pane

The filter pane is found below the Properties viewer on the right side of the tool editor. This shows the current categories and filters used by the tool.



Item	Description
Up/Down Arrows	Move selected categories up or down to determine the order in which they will be used to sort instances. Moving categories retains the order of filters within that category. Filters cannot be moved outside of their category.
Category	Add a new category at the bottom of the list.
Filter	Add a new filter at the bottom of the selected category list.
Delete	Delete the selected filter or category. Deleting a category deletes all filters within it.
Edit	Edit the selected filter or category. This can also be done by double-clicking a filter or category.

● Inspection Categories

The purpose of categories is to group instances using defined filters. For each instance, categories are evaluated starting with the category listed at the top of the Filter pane. An instance is put into the first category for which it passes the required number of filters as defined by the operator.
Each category is assigned a name and an operator. The operator (AND or OR) is applied to all filters within that category to evaluate the instances. The operator associated with a category is displayed in the Description field of the Configuration section. If the category has an AND operator, an instance must pass all filters in that category to qualify. Conversely, if the category has an OR operator, an instance needs to pass only one of the filters to qualify.
The name and operator of a category can be changed by editing the category in the filter pane.

■ Inspection Category Order

Because instances are put into the first category for which they qualify, the order of the category in the Filter pane is important. The Up/Down arrow buttons are used to adjust the category order.
If the selected category is at the top or bottom of the list, the Up or Down Arrow button will be disabled, respectively. These cannot be used to move filters between categories.



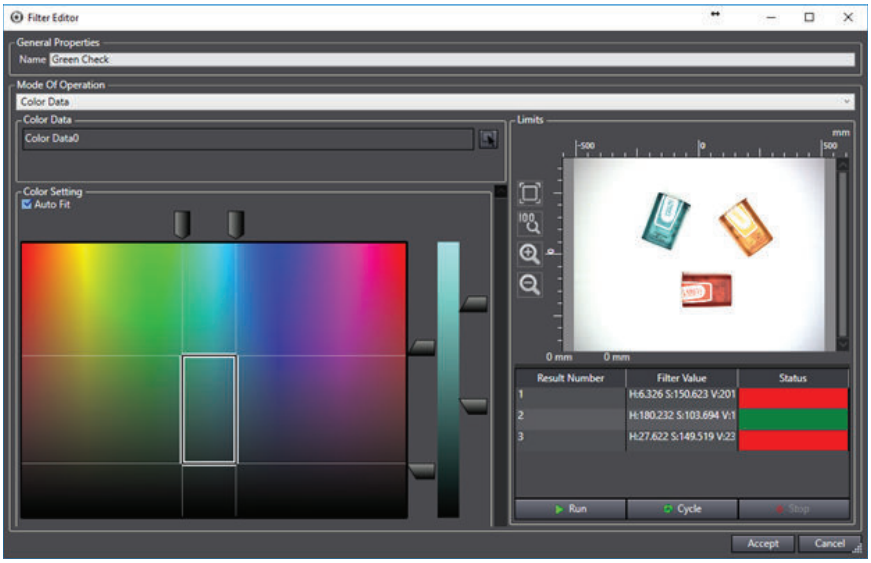
Additional Information

The location results associated with each category are exposed to Robot Vision Manager in a separate frame index for each category. The frame number is equal to the category's position; the first is associated with Frame 1, the second with Frame 2, and the pattern continues.

● Filters

Filters define the actual inspection or comparison to be performed. Each filter has a name and belongs to a category. A filter cannot be added if an existing category is not selected (the Add Filter button will be disabled until then). The inspection performed by a filter is displayed in the Description field of the Filter pane.

Editing a filter opens the Filter Editor. The filter setting items are described below.

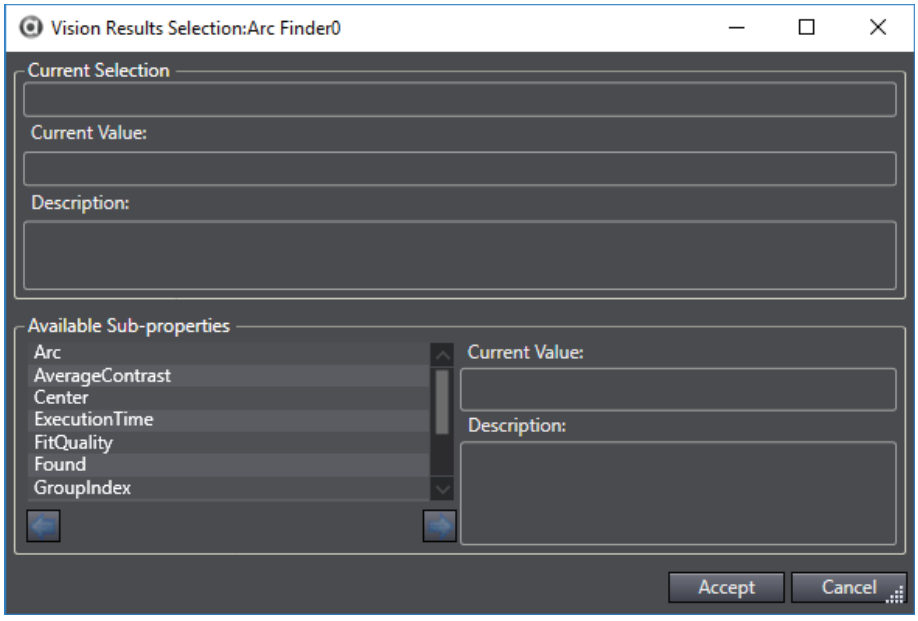


Item	Description
Name	An identifying name for this filter.
Mode Of Operation	Defines the type of evaluation this filter performs. The options are: • Measure Distance Between Two Points • Measure The Shortest Distance Between A Point and a Line • Measure The Angle Between Two Lines • Test The Value Of A Vision Result Variable • Vision Tool Transform Results • Number of Results • Number of Characters • OCR Quality • String • Color Data
Vision Tools	The appropriate vision tools and the necessary properties. This varies depending on the Mode Of Operation.
Limits	The bounds of this filter. If the results of an inspection fall between these values, the instance passes this filter. Like the Vision Tools, this also varies with the Mode Of Operation, but it is a Minimum and Maximum Value in most cases.
Results	Sample results from evaluating this filter on the current image. This can be used to show whether certain instances pass or fail, allowing the user to tune the limits. In most cases, there are three result columns in this editor: • Result Number: The instance number. • Filter Value: The evaluated property from the vision tool. • Status: Shows if the instance has satisfied this filter. This field will be green if the instance passes and red if it fails.

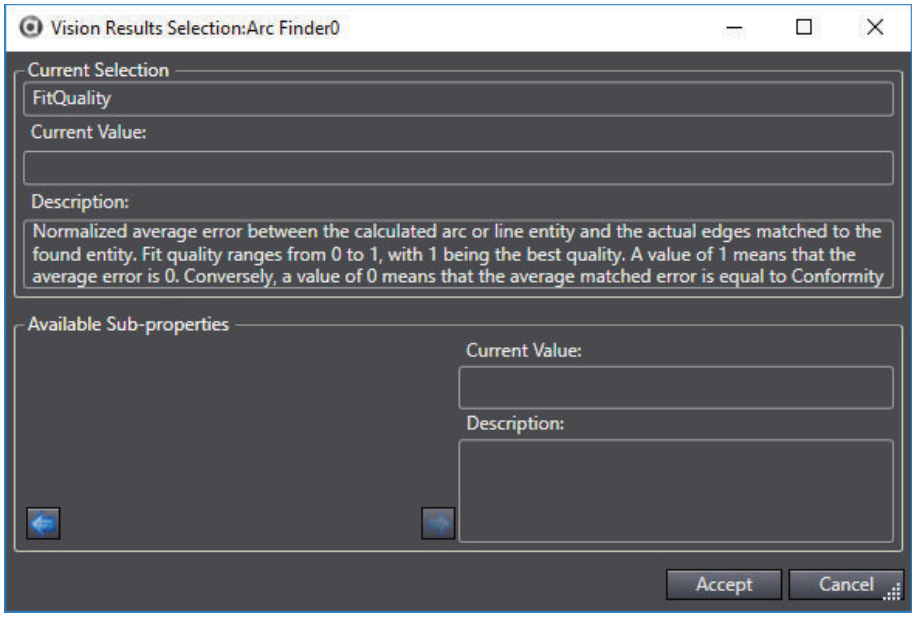
Item	Description
Control Buttons	• **Run**: Run the tool once. • **Cycle**: Run the tool continuously. • **Stop**: Stop the tool or live operation.

● Inspection Vision Results Selection

If the Mode Of Operations is set to Test The Value Of A Vision Result Variable, an additional Result from another Vision Tool will need to be selected. This requires its own window, as described below.



Highlighting any of the sub-properties will display its description. When the appropriate sub-property has been highlighted, click the **Right Arrow** Button to select and test it.



The fields above the available sub-properties pane will be filled. Click the **Accept** Button to confirm the selection and close the window.

Additional Information

Some properties have sub-properties that can also be selected. In these cases, the left arrow is used to step back to the previous level.

● Inspection Results

Use the table below to understand the results of the Inspection tool.

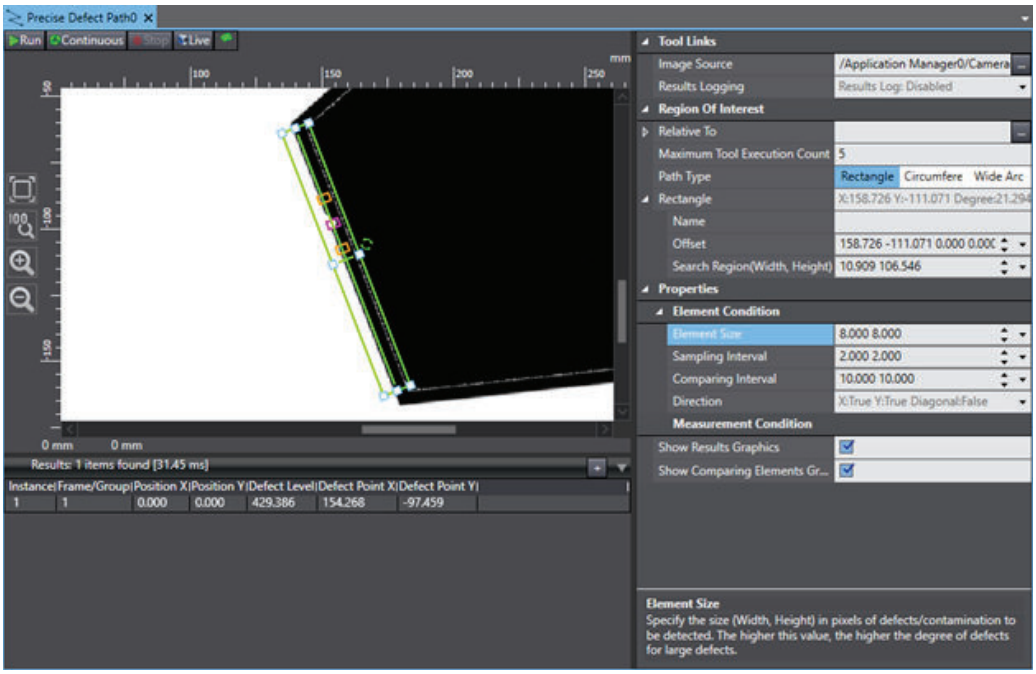
Additional Information

The Results columns refer to existing filters and thus change depending on the name of the filter. For these columns in the following table, these instances will include <filter> instead of specific names.

Item	Description
Instance	Index of the result instance, starting with 1.
<filter> Pass Status	Shows whether or not the instance passes a filter by displaying *True* if it passes and *False* if it does not.
<filter> Measurement	The measured value of the evaluated result from filter.
Category	The category for which this instance qualifies. If this instance does not qualify for any category, this will display *Unassigned*.
Position X	X-coordinate of the reference point. If Inspection is set relative to another tool, this will be the X-coordinate of the associated instance. Otherwise, it will be the origin of the image field of view.
Position Y	Y-coordination of the reference point. If Inspection is set relative to another tool, this will be the Y-coordinate of the associated instance. Otherwise, it will be the origin of the image field of view.
Angle	Angle of the located instance.

Precise Defect Path

This tool performs differential processing on the image to detect defects and contamination on the edges of plain measurement objects with high precision. This is done by using elements of varying size and comparison intervals. By changing these parameters, fine customization of speed and precision is possible. Precise Defect Path is primarily used for identifying 1-dimensional defects or variations and can be utilized for part edge inspection, as shown in the following figure.



To create a Precise Defect Path tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Inspection** and then **Precise Defect Path**. A Precise Defect Path tool will be added to the Vision Tools list.

● Precise Defect Path Configuration Items

Use the table below to understand the Precise Defect Path configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Enables logging of tool results to a file.	Enables logging of tool results to a file.

Group	Item	Description
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once.
	Path Type	Defines the shape and detection method of the region of interest. The region changes shape depending on this property.
	Rectangle Circumference Wide Arc	Changes to match the selection in Path Type. Shows the properties of the region of interest. Click the arrow next to it to see all properties in detail.
	Name	Available for all Path Type selections. Sets the user-defined name of the region.
	Offset	Available for all Path Type selections. Defines the coordinates of the region's center.
	Search Region (Width, Height)	Sets the height and width of the rectangular region. Only available in Rectangles.
	Radius	Defines the distance from the center to the unseen center ring. Only available when Path Type is set to Circumference or Wide Arc.
	Thickness	Defines the distance between the two visible rings. Only available when Path Type is set to Circumference or Wide Arc.
	Start/End Angle	Defines the start and end angle of the Wide Arc bounds. Angles are measured in degrees counterclockwise from the positive X-axis. The arc is created clockwise starting from the Start Angle and ending at the End Angle. Only available when Path Type is set to Wide Arc.

Group	Item	Description
Properties	(Element Condition) Element Size	Set the width and height in pixels of defects / contamination to be detected. Higher values increase the degree of defects for larger defects.
	(Element Condition) Sampling Interval	Set the interval in pixels for creating elements. Smaller values lead to more precise defect detection, but slow the processing speed.
	(Element Condition) Comparing Interval	Set the number of neighboring elements used to compare the calculated degree of defect. The number is determined by multiplying the Comparing Interval and Sampling Interval values.
	(Element Condition) Direction	Select the direction(s) used to detect defects. The processing time will be shorter if fewer directions are checked. • X (circumferential): Detects defects along the X-axis if Path Type is set to Rectangle and along the circumference if Path Type is set to **Circumference** or **Wide Arc**. • Y (radial):Detects defects along the Y-axis if Path Type is set to Rectangle and along the radii if Path Type is set to **Circumference** or **Wide Arc**. • Diagonal: Detects defects in diagonal directions.
	(Measurement Condition) Show Result Graphics	Specifies if the graphics are drawn in the Vision Window.
	(Measurement Condition) Show Comparing Elements Graphics	Specifies if the comparison elements are drawn in the Vision Window. Elements are automatically created during measurement, but they are only shown if this is enabled.

● Precise Defect Path - Defect Detection

This tool functions by creating elements in the search area and comparing them to determine which elements deviate the most from their neighboring elements. The elements to be compared are determined by all parameters in the Element Condition section. The size and spacing of the created elements is also determined by the properties. This tool focuses primarily on detecting defects along a path, making it useful for locating scratches or dents in a uniform surface or edge, as shown in the table above.

When run, Precise Defect Path will return the position of the largest detected defect, shown in the Vision Window. Only one result will be returned per input instance. For edge inspections, multiple Precise Defect Path tools may be linked to an Inspection tool or used within a Custom Vision Tool with logic to filter results as needed.

● Precise Defect Region Results

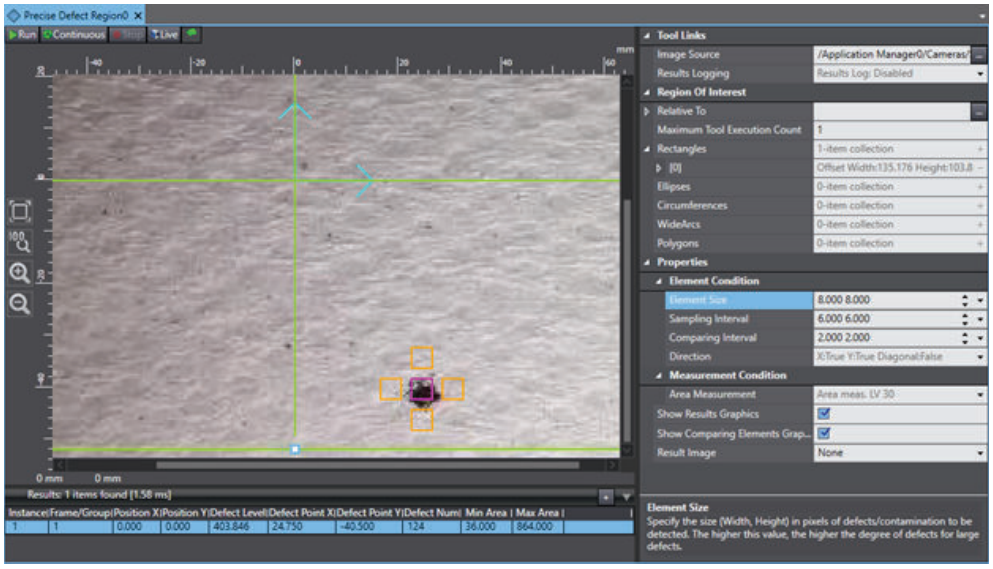
Use the table below to understand the results of the Precise Defect Region tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Position X	X-coordinate of the search area origin.
Position Y	Y-coordinate of the search area origin.
Defect Level	Maximum degree of defect in the search area.
Defect Point X	X-coordinate of the element of the largest detected defect.

Item	Description
Defect Point Y	Y-coordinate of the element of the largest detected defect.
Defect Num	Number of defects found in the search area.
Min Area	Pixel area of the smallest defect in the search area.
Max Area	Pixel area of the largest defect in the search area.

Precise Defect Region

This tool performs differential processing on the image to detect defects and contamination within plain measurement objects with high precision. This is done by using elements of varying size and comparison intervals. By changing these parameters, fine customization of speed and precision is possible. Precise Defect Region is primarily used for identifying 2-dimensional defects or variations and can be utilized for inspection of a part area.



To create a Precise Defect Region tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Inspection** and then **Precise Defect Region**. A Precise Defect Region tool will be added to the Vision Tools list.

● Precise Defect Region Configuration Items

Use the table below to understand the Precise Defect Region configuration items.

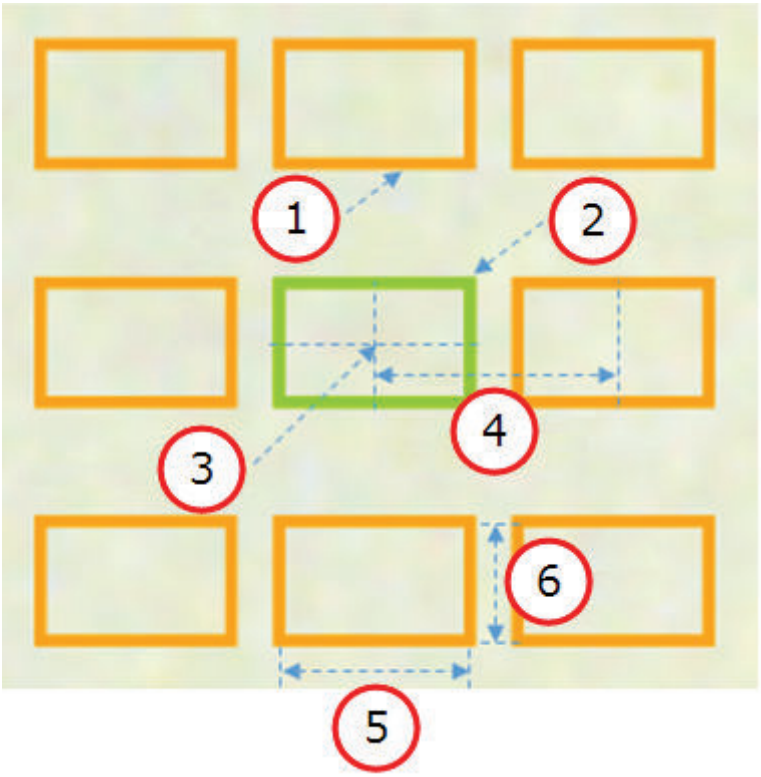
Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.

Group	Item	Description
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Rectangle Ellipse Circumferences Polygons WideArcs	Sets of regions of interest of a particular shape. The field will show how many regions exist ("1-item collection", "2-item collection", etc.). Click the plus sign in the field to create additional regions and click the arrow to see all regions.
	<region number>	Identifies the region of the specified shape. It will appear as [0], [1], [...] within the shape category. Click the minus sign in the field to remove the region and click the arrow to see its properties.
	Name	Available in all region types. Sets the user-defined name of the region.
	Overlap	Available in all region types. Select the behavior of the region. OR causes the region to be included in the search area and NOT excludes it from the search area. The behavior of specific regions is further defined in Z-Order. Refer to *Tools with Multiple Regions of Interest* on page 8-195for more information.
	Z-Order	Available in all region types. Sets the order for overlapping regions of interest. Higher numbers will be in front and resolved first. Refer to *Tools with Multiple Regions of Interest* on page 8-195 for more information.
	Offset	Available in all region types except for Polygons. Defines the coordinates of the region center.
	Search Region (Width, Height)	Sets the height and width of the rectangular region. Only available in Rectangles.
	Radius X/Y	Defines the distance from the center to the exterior along the X- and Y-axes, respectively. Only available in Ellipses.

Group	Item	Description
Region of Interest	Radius	Defines the distance from the center to the unseen center ring. Only available in Circumferences and WideArcs.
	Thickness	Defines the distance between the two visible rings. Only available in Circumferences and WideArcs.
	Start/End Angle	Defines the start and end angle of the Wide Arc bounds. Angles are measured in degrees counterclockwise from the positive X-axis. The arc is created clockwise starting from the Start Angle and ending at the End Angle. Only available in WideArcs.
	Vertexes	Sets of vertexes in a polygon region of interest. The field will show how many vertexes exist ("3-item collection", "4-item collection, etc.). Click the plus line in the field to create additional regions and click the arrow to see all regions. Only available in Polygons.
	<vertex number>	Identifies a particular vertex in the region. It will appear as [0], [1], [2], [...]. The field defines the X/Y coordinates of the vertex. Click the minus sign to remove the vertex, as long as there are not fewer than three. Only available in Polygons.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once.
Properties	(Element Condition) Element Size	Set the width and height in pixels of defects / contamination to be detected. Higher values increase the degree of defects for larger defects.
	(Element Condition) Sampling Interval	Set the interval in pixels for creating elements. Smaller values lead to more precise defect detection, but slow the processing speed.
	(Element Condition) Comparing Interval	Set the number of neighboring elements used to compare the calculated degree of defect. The number is determined by multiplying the Comparing Interval and Sampling Interval values.
	(Element Condition) Direction	Set the direction for detecting defects. The processing time will be shorter if fewer directions are checked.
	(Measurement Condition) Area Measurement	Enables measurement of the size of defects, dividing the high defect detection regions into groups and outputting the surface and center of gravity coordinates of the group with the largest area.
	(Measurement Condition) Show Result Graphics	Specifies if the graphics are drawn in the Vision Window.
	(Measurement Condition) Show Comparing Elements Graphics	Specifies if the comparison elements are drawn in the Vision Window. Elements are automatically created during measurement, but they are only shown if this is enabled.
	(Measurement Condition) Result Image	Sets the display condition of the resulting image. Select None to display the original image, Area Image to display the image in area measurement mode, and Profile Image to show areas with high defect levels in bright colors. Note that Area Measurement must be enabled for Area image to work properly.

● Precise Defect Region - Defect Detection

This tool functions by creating elements in the search area and comparing them to determine which elements deviate the most from their neighboring elements. The elements to be compared are determined by all parameters in the Element Condition section. The size and spacing of the created elements is also determined by the properties. To see an example of how the various parameters affect detection, refer to the figure below.



Item	Description
1	Comparing Element
2	Result Element
3	Defect Point
4	Comparing Interval x Sampling Interval
5	Element Size X
6	Element Size Y

When run, Precise Defect Region will return the position of the largest detected defect, shown in the Vision Window. Only one result will be returned per input instance. For example, if Precise Defect Region is set relative to another tool, one result will be returned for each instance created by that tool.

If Area Measurement is enabled, the result will include the number of defects on that part in the Defect Num result. Otherwise, Defect Num will be 0.

The tool will display a grid of elements at the location of the largest detected variation. This allows the user to gain information on the size and proximity of features being detected as defects.

● Precise Defect Region Results

Use the table below to understand the results of the Precise Defect Region tool.

Item	Description
Instance	Index of the result instance.

Item	Description
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To.
Position X	X-coordinate of the search area origin.
Position Y	Y-coordinate of the search area origin.
Defect Level	Maximum degree of defect in the search area.
Defect Point X	X-coordinate of the element of the largest detected defect.
Defect Point Y	Y-coordinate of the element of the largest detected defect.
Defect Num	Number of defects found in the search area.
Min Area	Pixel area of the smallest defect in the search area.
Max Area	Pixel area of the largest defect in the search area.

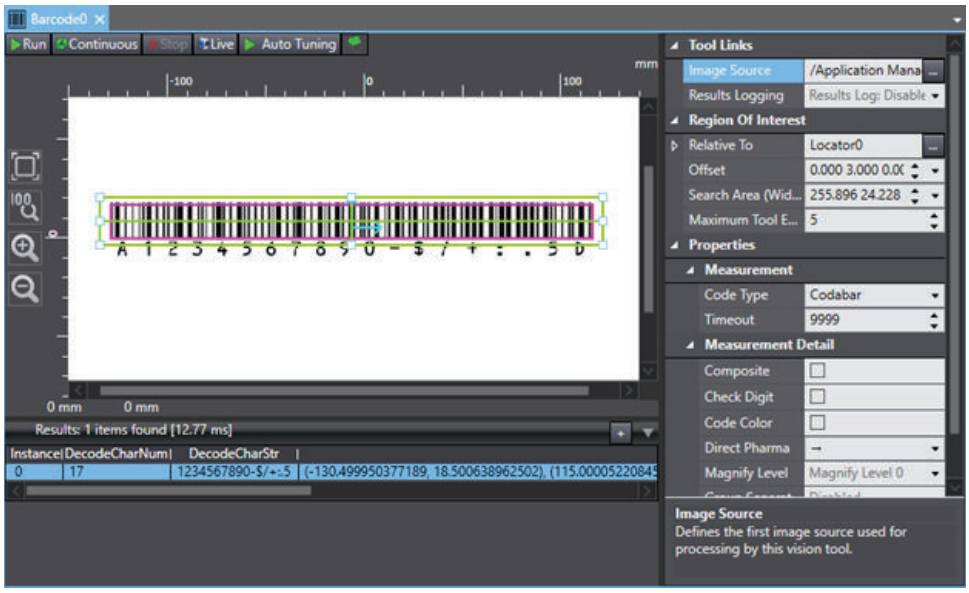
8-8-7 Reader Tools

Reader tools are typically used for reading data from located objects and features.
The following Reader tools are described in this section.

- Refer to *Barcode* on page 8-280 for more information.
- Refer to *Data Matrix* on page 8-283 for more information.
- Refer to *OCR* on page 8-285 for more information.
- Refer to *OCR Dictionary* on page 8-291 for more information.
- Refer to *QR Code* on page 8-292 for more information.

Barcode

This tool reads a barcode in a region of interest and returns text string data.



To create a Barcode tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Reader** and then **Barcode**. A Barcode tool will be added to the Vision Tools list.

● Barcode Configuration Items

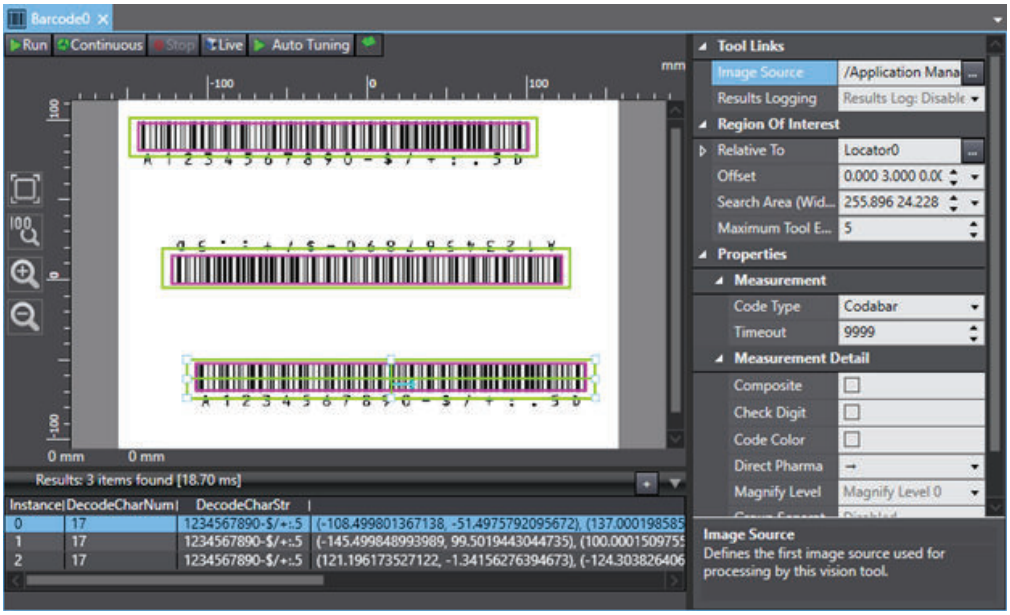
Use the table below to understand the Barcode configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the center coordinates of the region.
	Search Area (Width, Height)	Defines the size of the region.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once.
Properties	(Measurement) Code Type	Select the appropriate code type from among the following options: • JAN/EAN/UPC • Code39 • Codabar • ITF • Code93 • Code128/GS1-128 • GS1 DataBar • Pharmacode
	(Measurement) Timeout	Set the time in milliseconds that can elapse before the tool execution is aborted. Any instances recorded before this time period is reached will be returned.
	(Measurement Detail) Composite	Enables the detection and reading composite code, defined as a combination of 1D and 2D barcode.
	(Measurement Detail) Check Digit	Select whether to perform a check using the check digit. When this is enabled, the check digit is not included in the returned character string.
	(Measurement Detail) Code Color	Enables the tool to read white code on a black background. This is disabled by default since standard barcode color is black on a white background.
	(Measurement Detail) Direct Pharma	Select the direction in which to read the code. By default, this set from left to right.
	(Measurement Detail) Magnify Level	Set the image magnify level to be applied when reading code. Increasing this value allows the tool to better read high-resolution images.
	(Measurement Detail) Group Separator Replacement String	Defines the character to be set between different strings in various code types. Enabling this allows the user to better control the string output.
	Show Result Graphics	Specifies if the graphics are drawn in the Vision Window.

● Barcode Reading

The Barcode tool is typically configured relative to a Finder tool such as Locator or Shape Search 3 to locate and orient the region of interest on one barcode at a time. To configure, use an image with only one barcode present and configure the region of interest around the code with sufficient white

space. Once the region has been set, the **Auto Tuning** Button can be used to attempt to automatically configure many of the properties in the property grid, particularly Code Type, as shown in the previous figure. If Auto Tuning does not work properly, a small adjustment of the region of interest may resolve this. Note that the orientation of the region relative to the barcode must be close to 0°. After the parameters have been set, the Barcode tool can be run like any other vision tool. Clicking the **Run** Button executes the tool and returns a text string for each region of interest created using the Relative To property, as shown in the figure below.



● Barcode Results

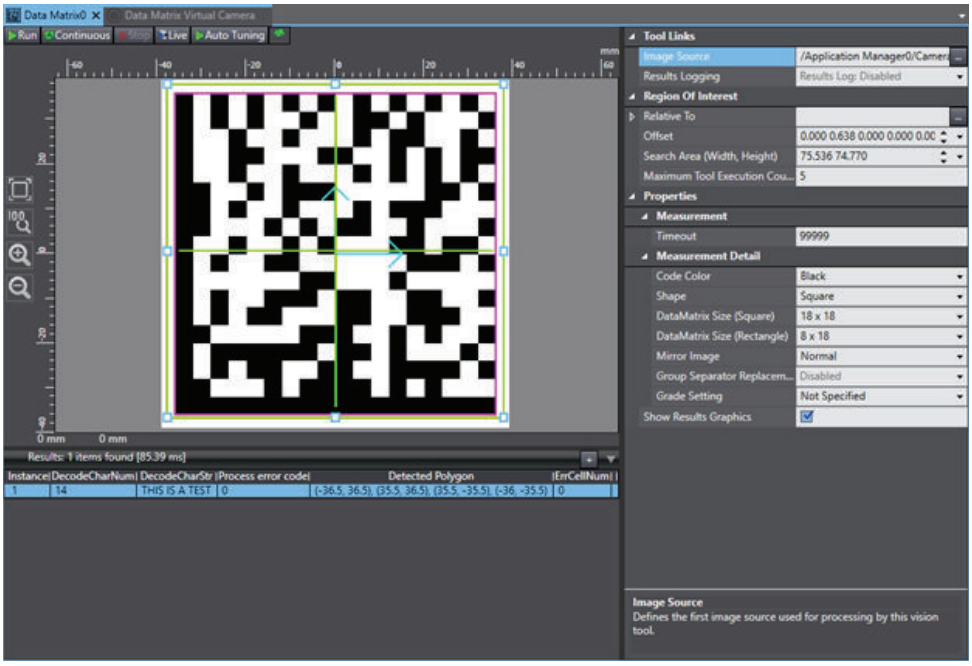
Use the table below to understand the results of the Barcode tool.

Item	Description
Instance	Index of the result instance.
DecodeCharNum	Number of decoded characters.
DecodeCharStr	String of decoded characters.
Detected Polygon	Vertices of the rectangle around the barcode.
ProcErrorCode	ProcErrorCode
1D Decode	1D decode, data range of 0 to 4.
1D Symbol Contrast	1D symbol contrast, data range of 0 to 4.
1D Minimum Reflection Ratio	1D minimum reflection ratio, data range of 0 to 4.
1D Minimum Edge Contrast	1D minimum edge contrast, data range of 0 to 4.
1D Modulation	1D modulation, data range of 0 to 4.
1D Defect	1D defect, data range of 0 to 4.
1D Decodability	1D decodability, data range of 0 to 4.
1D Additional	1D additional, data range of 0 to 4.
1D One Line Overall	1D one line overall.
PDF Decode	PDF decode, data range of 0 to 4.
PDF Symbol Contrast	PDF symbol contrast, data range of 0 to 4.
PDF Minimum Reflection Ratio	PDF minimum reflection ratio, data range of 0 to 4.
PDF Minimum Edge Contrast	PDF minimum contrast edge, data range of 0 to 4.
PDF Modulation	PDF modulation, data range of 0 to 4.
PDF Defect	PDF defect, data range of 0 to 4.

Item	Description
PDF Decodability	PDF decodability, data range of 0 to 4.
PDF Additional	PDF additional, data range of 0 to 4.
PDF Line Overall	PDF overall quality, data range of 0 to 4.
PDF Code Word	Code word yield, data range of 0 to 4.

Data Matrix

This tool reads a data matrix in a region of interest and returns text string data.



To create a Data Matrix tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Reader** and then **Data Matrix**. A Data Matrix tool will be added to the Vision Tools list.

● Data Matrix Configuration Items

Use the table below to understand the Data Matrix configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the center coordinates of the region.
	Search Area (Width, Height)	Defines the size of the region.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once.

Group	Item	Description
Properties	(Measurement) Timeout	Set the time in milliseconds that can elapse before the tool execution is aborted. Any instances recorded before this time period is reached will be returned.
	(Measurement Detail) Code Color	Select the color of the code. Select Auto to automatically detect the color, Black to look for black code on a white background, and White to look for white code on a black background.
	(Measurement) Shape	Select the shape of the code. Select Auto tool to automatically detect the shape, Square to look for a square matrix, and Rectangle to look for a rectangular matrix.
	(Measurement) Data Matrix Size (Square)	Only measured if Shape is set to Square or Auto and a square data matrix is detected. Select the size of the code in cells.
	(Measurement Detail) Data Matrix Size (Rectangle)	Only measured if Shape is set to Rectangle or Auto and a rectangular data matrix is detected. Select the size of the code in cells.
	(Measurement) Italic Robust	Enables the detection of italic characters.
	(Measurement Detail) Mirror Image	Specify the direction in which the matrix will be read. Select **Auto** for the tool to automatically detect the direction, Normal for the matrix to be read normally, or Mirror for the matrix to be read in reverse.
	(Measurement Detail) Group Separator Replacement String	Defines the character to be set between different strings in various code types. Enabling this allows the user to better control the string output.
	(Measurement Detail) Grade Setting	Select the method of print quality evaluation. • Not Specified: no specific print quality. • ISO/IEC 15415: evaluate with ISO/IEC 15415 standard. • ISO/IEC TR 29158: evaluate with ISO/IEC TR 29158 standard.
	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.

● Data Matrix Reading

The Data Matrix tool is typically configured relative to a Finder tool such as Locator or Shape Search 3 to decode all matrices in an image. To configure, use an image with only one matrix present and configure the region of interest around the code with sufficient white space. Once the region has been set, the Auto Tuning button can be used to attempt to automatically configure many of the properties in the property grid. If Auto Tuning does not work properly, increasing white space around the matrix or improving its alignment with the center of the matrix may resolve this. After the parameters have been set, the Data Matrix tool can be run like any other vision tool. Clicking the **Run** Button executes the tool and returns a text string for each region of interest created using the Relative To property.

● Data Matrix Results

Use the table below to understand the results of the Data Matrix tool.

Item	Description
Instance	Index of the result instance.
DecodeCharNum	Number of decoded characters.

Item	Description
DecodeCharStr	String of decoded characters.
Detected Polygon	Coordinates of the vertices that form a rectangle around the matrix.
Process error code	Process error code.
ErrCellNum	ErrCellNum
Grade ISO15415 Integrated Quality	Integrated Quality.
Grade ISO15415 Decodability	Decode, range of 0 to 4.
Grade ISO15415 Symbol Contrast	Contrast, range of 0 to 4.
Grade ISO15415 Modulation	Modulation, range of 0 to 4.
Grade ISO15415 Fixed Pattern Damage	Fixed pattern damage, range of 0 to 4.
Grade ISO15415 Axial Nonuniformity	Axis nonuniformity, range of 0 to 4.
Grade ISO15415 Grid Nonuniformity	Grid nonuniformity, range of 0 to 4.
Grade ISO15415 Unused Error Correction	Correction of error not used, range of 0 to 4.
Grade ISO15415 Reflectance Margin	Reflectance, range of 0 to 4.
Grade ISO15415 Print Scale	Print expansion/contraction, range of 0 to 4.
Grade ISO15415 Print Scale X	Print expansion/contraction X, range of 0 to 4.
Grade ISO15415 Print Scale Y	Print expansion/contraction Y, range of 0 to 4.
Value ISO15415 Symbol Contrast	Symbol contrast, range of 0 to 4.
Value ISO15415 Axial Nonuniformity	Axial nonuniformity, range of 0 to 4.
Value ISO15415 Grid Nonuniformity	Grid nonuniformity, range of 0 to 4.
Value ISO15415 Unused error Correction	Correction of error not used, range of 0 to 4.
Value ISO15415 Print Scale	Print expansion/contraction, range of 0 to 4.
Value ISO15415 Print Scale X	Print expansion/contraction X, range of 0 to 4.
Value ISO15415 Print Scale Y	Print expansion/contraction Y, range of 0 to 4.
Grade ISOTR29158 Integrated Quality	Integrated Quality, range of 0 to 4.
Grade ISOTR29158 Cell Contrast	Contrast, range of 0 to 4.
Grade ISOTR29158 Self Modulation	Modulation, range of 0 to 4.
Grade ISOTR29158 Fixed Pattern Damage	Fixed pattern damage, range of 0 to 4.
Grade ISOTR29158 Axial Nonuniformity	Axis nonuniformity, range of 0 to 4.
Grade ISOTR29158 Grid Nonuniformity	Grid nonuniformity, range of 0 to 4.
Grade ISOTR29158 Unused Error Correction	Correction of error not used, range of 0 to 4.
Grade ISOTR29158 Print Scale	Print expansion/contraction, range of 0 to 4.
Grade ISOTR29158 Print Scale X	Print expansion/contraction X, range of 0 to 4.
Grade ISOTR29158 Print Scale Y	Print expansion/contraction Y, range of 0 to 4.
Value ISOTR29158 Cell Contrast	Contrast, range of 0 to 4.
Value ISOTR29158 Axial Uniformity	Axis nonuniformity, range of 0 to 4.
Value ISOTR29158 Grid Uniformity	Grid nonuniformity, range of 0 to 4.
Value ISOTR29158 Unused Error Correction	Correction of error not used, range of 0 to 4.
Value ISOTR29158 Print Scale	Print expansion/contraction, range of 0 to 4.
Value ISOTR29158 Print Scale X	Print expansion/contraction X, range of 0 to 4.
Value ISOTR29158 Print Scale Y	Print expansion/contraction Y, range of 0 to 4.

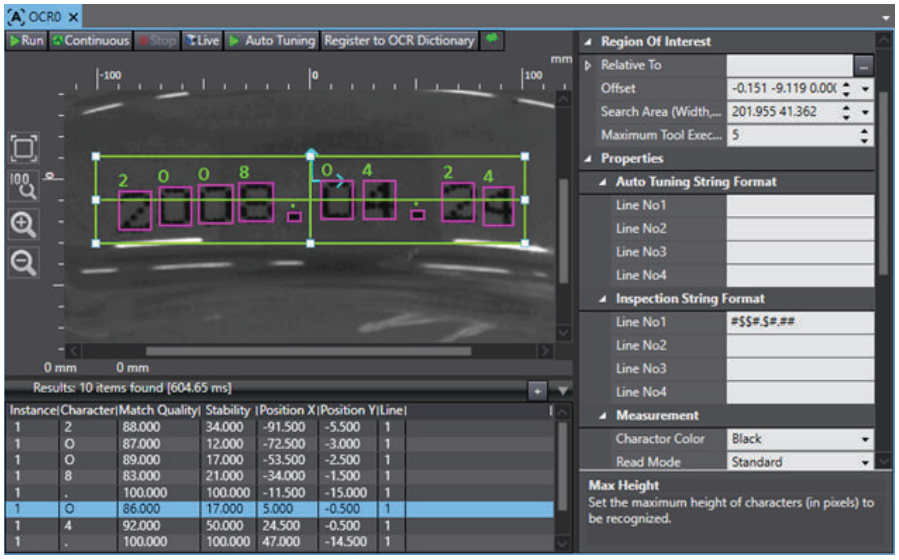
OCR

This tool detects text characters in images and compares them to an internal font property to output character strings. A custom user dictionary can also be prepared to recognize characters in special fonts using the OCR Dictionary tool. Refer to *OCR Dictionary* on page 8-291 for more information.

OCR provides a higher level of recognition stability than character inspection when reading closely spaced characters, curved text strings, and other deviational characters. It can also reference internal dictionary data, but a specialty dictionary only needs to be created if abnormal fonts are used.

Additional Information

Typical OCR applications use strings of number and capital letters. If the application involves detection of lower case letters, a user-defined OCR Dictionary is required.



To create an OCR tool, right-click Vision Tools in the Multiview Explorer, select **Add Reader** and then **OCR**. An OCR tool will be added to the Vision Tools list.

● OCR Configuration Items

Use the table below to understand the OCR configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	OCR Dictionary	Defines which OCR Dictionary tool is used for reference and registering characters.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the center coordinates of the region.
	Search Area (Width, Height)	Defines the size of the region.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once.

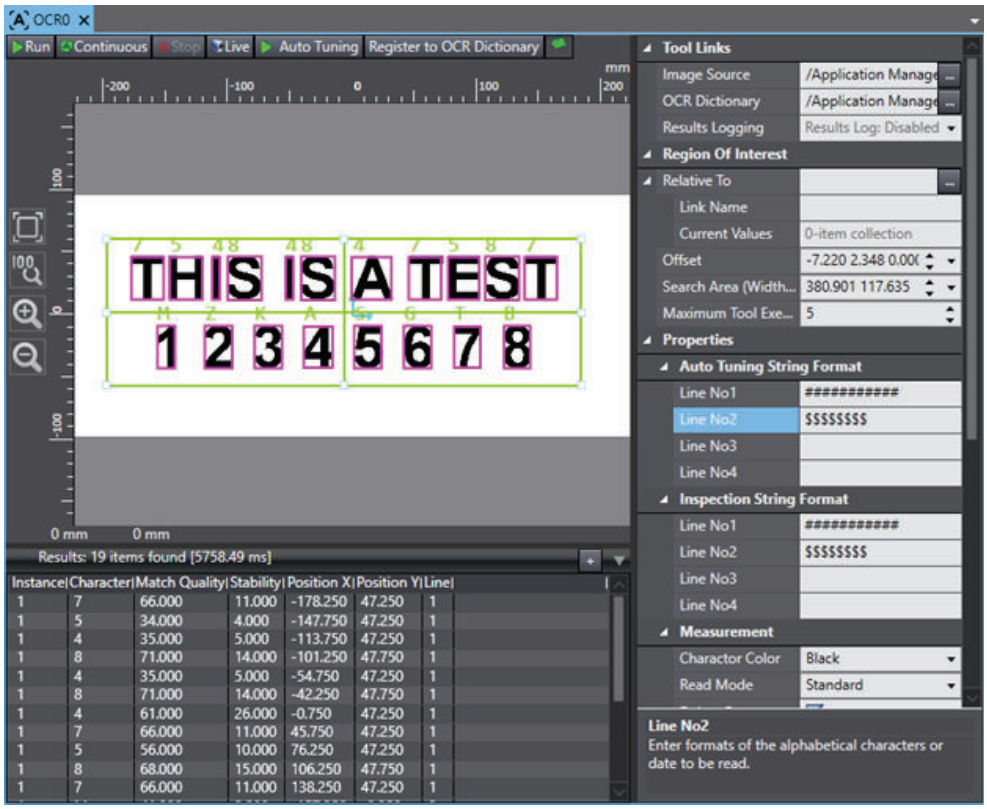
Group	Item	Description
Properties	(Auto Tuning String Format) Line No1/2/3/4	Enter the specific characters that are expected to be read in the region of interest. This defines what the tool will look for during the Auto Tuning process. If this field is left blank, it will not be populated. The generic characters can also be entered here. Refer to the Inspection String Format property for details.
	(Inspection String Format) Line No1/2/3/4	Enter the character types that are expected to be read in the region of interest. Enter $ for a character, # for a number, and a symbol for that specific symbol. For example, if a slash is expected, / would be entered. If Auto Tuning String Format is populated, this will change to match it. Otherwise, it will be based on detected. If one of the characters in this property is changed and then the tool is run, the results will be based on this property.
	(Measurement) Character Color	Set the color of characters to be recognized.
	(Measurement) Read Mode	Set the reading method used. Standard can be selected for most applications. If there is a large gap between characters, select **Fast**.
	(Measurement Detail) Delete Frame	Enables the removal of black pixels at the border of the measurement region as noise.
	(Measurement) Italic Robust	Enables the detection of italic characters.
	(Measurement Detail) Rotation Robust	Enables detection of characters that are rotated relative to each other. The rotation range is ±15°.
	(Measurement Detail) Font	Set the font of the characters to be recognized. Select **Common** for most font types and **Dot** when the font is comprised of a series of dots.

Group	Item	Description
Properties	(Measurement Detail) Thick Threshold	Set the thickness of the characters to be recognized on a scale of 0-128. Higher values will improve reading of broken or fine characters.
	(Measurement Detail) Dot Pitch X/Y	Set the dot pitch of characters to be recognized. When Font is set to Common, these define the horizontal (X) and vertical (Y) widths of character breaks. When Font is set to Dot, these define the horizontal (X) and vertical (Y) dot intervals.
	(Measurement Detail) Max Width	Defines the maximum width (in pixels) of characters to be recognized. This is disabled by default.
	(Measurement Detail) Max Height	Set the maximum height (in pixels) of characters to be detected.
	(Measurement Detail) Min Height (%)	Set the minimum height of characters to be recognized as a percentage of the Max Height property. Characters below this threshold are recognized as symbols.
	(Measurement Detail) Max Aspect Ratio	Set the maximum aspect ratio (calculated as height divided by width) of characters to be recognized. Characters with an aspect ratio greater than this value are recognized as special characters or symbols.
	(Measurement Detail) Hyphen Threshold	Set an upper and lower limit as a percentage of the closest neighboring character.
	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Results Display	Select the type of result marker to be displayed. Recognition Markers displays only markers related to recognized characters and All Markers displays markers for all characters.
	Cut out image display	Enables display to show all characters as cutouts. Recognized characters will be isolated on a white background. The gray region displayed in the cutout image display is the region bounded by the Dot Pitch X and Dot Pitch Y parameters.

● OCR - Acquiring Text Strings

Click the **Auto Tuning** Button and ACE will attempt to automatically detect the types of characters in the region. These will be displayed in the Inspection String Format section, as shown the previous figure. If the tool is not correctly tuning to the text, it can be manipulated using this section and Auto Tuning String Format.

For example, the tool will try to find the best numerical matches for all the letters in Line 1 and the best letter matches for all the numbers in Line 2 if the corresponding lines in the Auto Tuning String Format section are set to the number symbol # and the character symbol $, respectively. In this way, characters that are read inaccurately can be corrected to their proper form. The specific expected characters can also be entered to further improve the results.

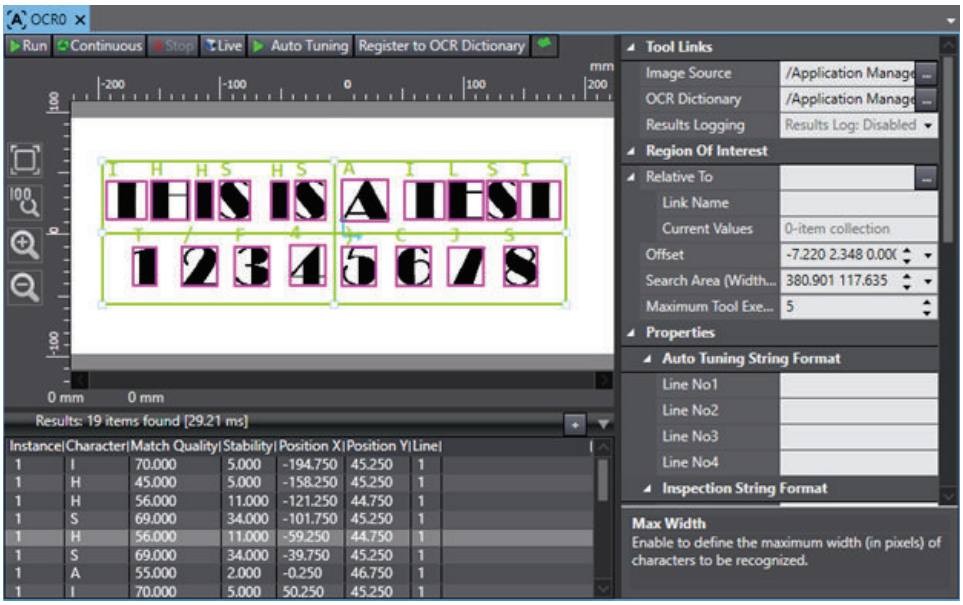


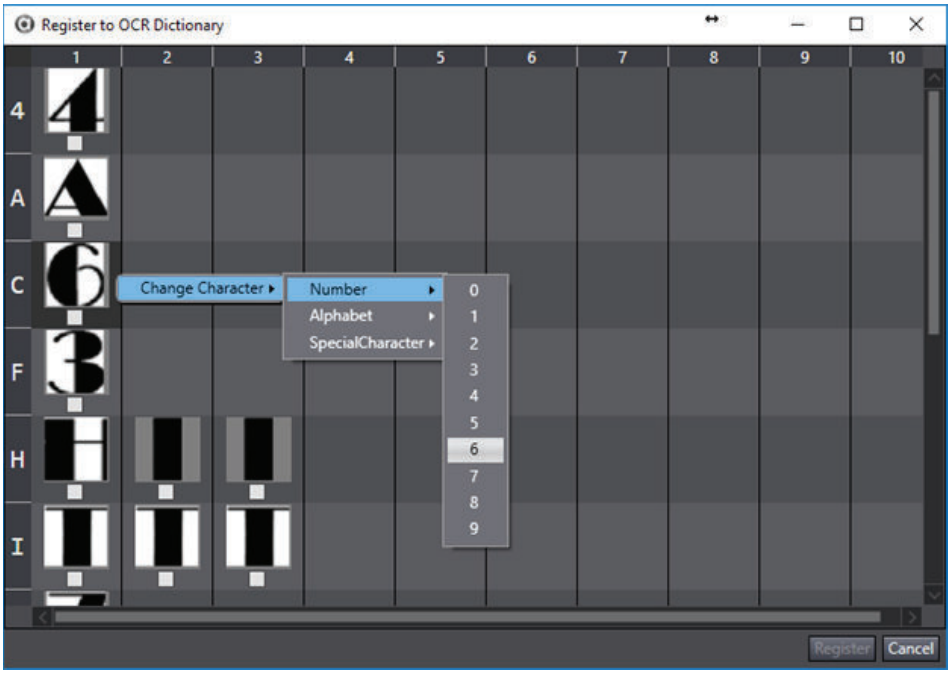
● OCR - Registering to a Dictionary

With some fonts and applications, OCR incorrectly identifies a character or fails to distinguish one character from another. When this is the case, it is recommended to register these characters to an OCR Dictionary tool. OCR tools that reference an OCR Dictionary will compare detected characters to those saved in the dictionary before returning a text string.

To register detected characters to a dictionary, the OCR Dictionary property must be linked to an OCR Dictionary Tool. Clicking the **Register to OCR Dictionary** Button will then allow the user to select what characters to save. Refer to the following figures for more information.

The registered characters do not need to match the appropriate character upon registration. For example, in the figures below, the character in C is actually a 6. That can be moved now (as shown) or later in the OCR Dictionary tool. Refer to *OCR Dictionary* on page 8-291 for more information.





● OCR Results

Use the table below to understand the results of the OCR tool.

Item	Description
Instance	Index of the result instance, starting with 1.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To. This will only be different if the Locator is set relative to another tool.
Character	Detected character identification.
Match Quality	Percentage that the detected character matches the recorded character in the internal dictionary or OCR Dictionary tool.
Stability	Measurement of how likely the character is to be the result that was identified.
Position X	X-coordinate of the detected character with respect to the image origin.
Position Y	Y-coordinate of the detected character with respect to the image origin.
Line	Character's line of text within the region of interest.

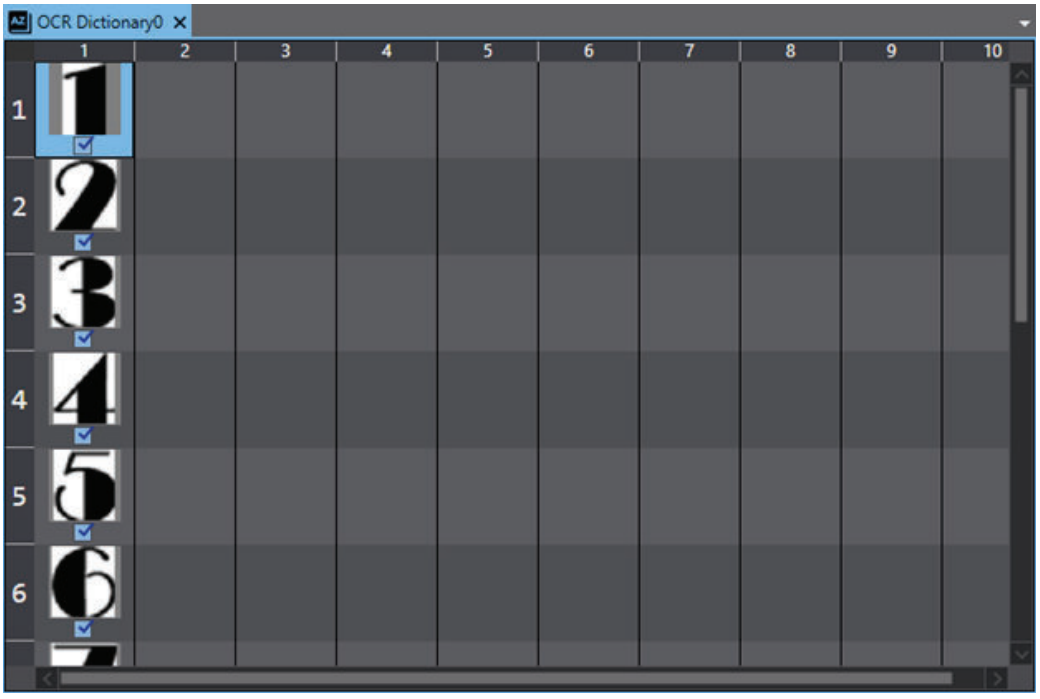
OCR Dictionary

This tool is used only as a reference for OCR tools and can only be configured for and populated by an OCR tool. It is used to save characters that the OCR identifies incorrectly or fails to identify. An OCR tool will first compare detected characters to an internal dictionary data and then to a referenced OCR Dictionary. Internal dictionary data cannot be modified. Refer to *OCR* on page 8-285 for more information.

Because OCR Dictionary is only a reference, it has no properties or results. It only contains data about registered characters.

Additional Information

The following figure shows the OCR Dictionary with some registered characters. The OCR Dictionary will appear blank if no characters have been registered.



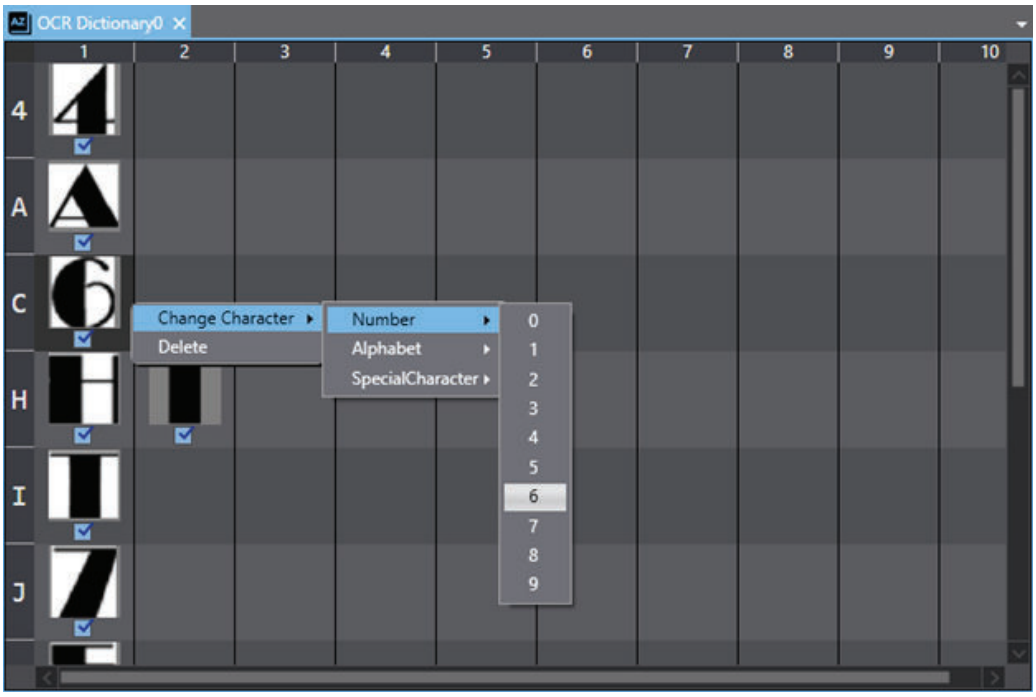
To create an OCR Dictionary tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Reader** and then **OCR Dictionary**. An OCR Dictionary tool will be added to the Vision Tools list.

● OCR Dictionary Configuration

Up to ten dictionary entries can be registered for each character. If more than ten characters are registered, the tool will only save the first ten and remove the others.

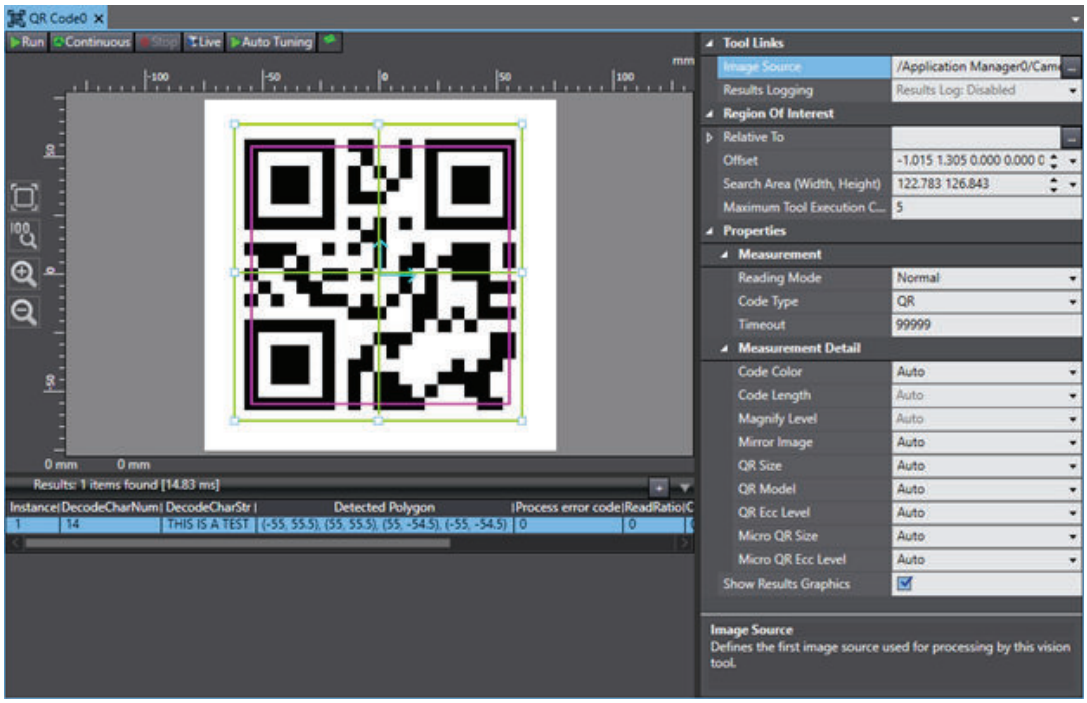
The characters can be changed or deleted at any time by right-clicking on the figure. Selecting **Delete** will remove the entry from the dictionary. Selecting **Change Character** will allow the entry to be saved to any other character.

If the character to which the entry would be changed already contains ten entries, the entry will not move. The following figure shows the OCR Dictionary after all necessary changes have been made. The following figure shows one such change in progress.



QR Code

This tool reads QR Codes and Micro QR Codes in the image and returns text string data.



To create a QR Code tool, right-click **Vision Tools** in the Multiview Explorer, click **Add Reader** and then click **QR Code**. The tool will be added to the Vision Tools list.

● QR Code Configuration Items

Use the table below to understand the QR Code configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region Of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the center coordinates of the region.
	Search Area (Width, Height)	Defines the size of the region.
	Maximum Tool Execution Count	Maximum number of tool executions for Relative To performance. If this tool is not set relative to another tool, the tool will only run once.

Group	Item	Description
Properties	(Measurement) Reading Mode	Set the method of code reading. Select **Normal** for standard applications and DPM to read 2D code where direct parts marking (DPM) is applied.
	(Measurement) Code Type	Select the type of code from either QR or MicroQR.
	(Measurement) Timeout	Set the time in milliseconds that can elapse before the tool execution is aborted. Any instances recorded before this time period is reached will be returned.
	(Measurement Detail) Code Color	Select the color of the code. Select **Auto** to automatically detect the color, **Black** to look for black code on a white background, and **White** to look for white code on a black background.
	(Measurement Detail) Code Length	Specify the code length (in characters). Auto is enabled by default, causing the tool to automatically detect the number of characters. Disable this to set an exact number.
	(Measurement Detail) Magnify Level	Set the reduction ratio for images when reading code. This is automatically determined by the teaching process. Disable **Auto** to set it manually.
	(Measurement Detail) Mirror Image	Select the direction in which the tool will read the code. Select **Auto** for the tool to automatically detect the direction during the teaching process, **Normal** for the code to be read normally, or Mirror for the code to be read in reverse.
	(Measurement Detail) QR Size	Only applicable if Code Type is set to QR. Select the size of the QR code (in cells). Select **Auto** for the size to be detected automatically.
	(Measurement Detail) QR Model	Only applicable if Code Type is set to QR. Set the QR code model. Select **Auto** for the model to be detected automatically.
	(Measurement Detail) QR Ecc Level	Select the code error correction (ECC) level. Select **Auto** to automatically adjust it as necessary.
	(Measurement Detail) Micro QR Size	Only applicable if Code Type is set to MicroQR. Select the size of the Micro QR code (in cells).
	(Measurement Detail) Micro QR Ecc Level	Only applicable if Code Type is set to MicroQR. Select the code error correction (ECC) level. Select **Auto** to automatically adjust it as necessary.
	Show Result Graphics	Specifies if the graphics are drawn in the Vision Window.

● QR Code Reading

The QR Code tool is typically configured relative to a Finder tool such as Locator or Shape Search 3 to decode all codes in an image. To configure, use an image with only one QR Code present and configure the region of interest around it with sufficient white space.

Once the region has been set, the Auto Tuning button can be used to attempt to automatically configure many of the properties in the property grid. To expedite this, most of the properties are set to Auto by default so they automatically detect the necessary information. If Auto Tuning does not work properly, increasing white space around the code or improving its alignment with the center of the code may resolve this.

Auto Tuning will not function if the code around which the region of interest is aligned does not match the Code Type property. For example, if Code Type is set to MicroQR and the code in the

image is a standard QR code, the data will not be output. Unlike most of the other properties, Code Type cannot be set automatically, and must be verified that it is accurate before running the tool. After the parameters have been set, the QR Code can be run like any other vision tool. Clicking the **Run** Button executes the tool and returns a text string for each region of interest created using the Relative To property.

● QR Code Results

Use the table below to understand the results of the QR Code tool.

Item	Description
Instance	Index of the result instance.
DecodeCharNum	Number of decoded characters.
DecodeCharStr	String of decoded characters.
Detected Polygon	Coordinates of the vertices that form a rectangle around the code.
Process error code	Process error code
ReadRatio	Effective codeword ratio.
Contrast	Sharpness of an image on a scale of 0 to 100.
Focus	Number of false cell detections in the finder pattern, timing pattern, and data region.
ErrCellNum	ErrCellNum
Position	X/Y coordinates of the center of the detected QR code.
Angle	Angle of the detected QR code.

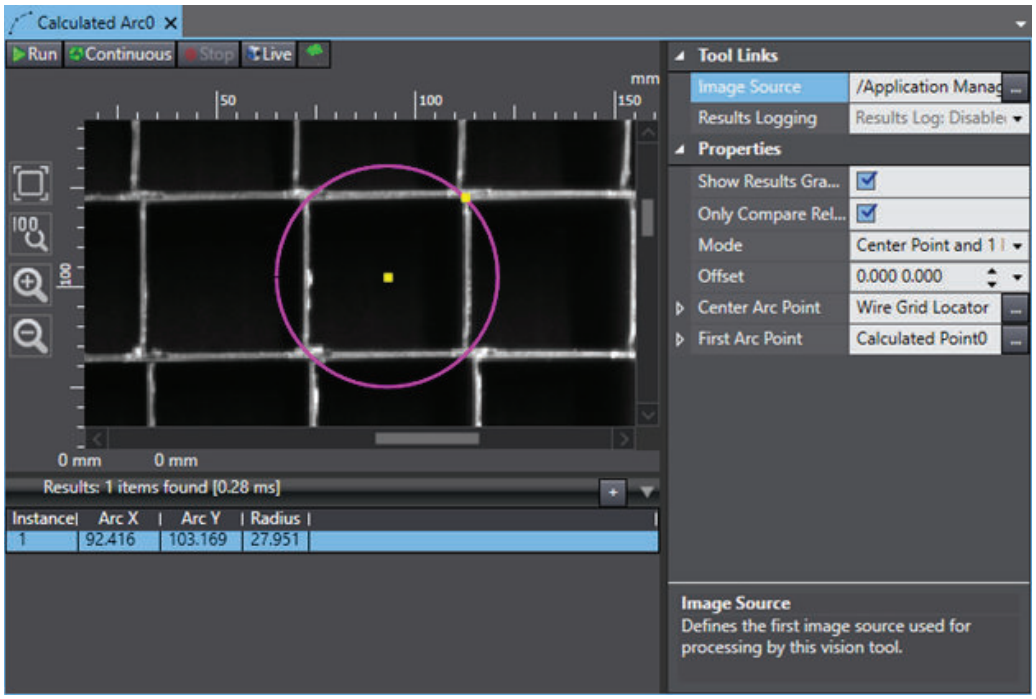
8-8-8 Calculation Tools

Calculation tools are used for calculating or refining detection points. The following Calculation tools are described in this section.

- Refer to *Calculated Arc* on page 8-295 for more information.
- Refer to *Calculated Frame* on page 8-298 for more information.
- Refer to *Calculation Line* on page 8-300 for more information.
- Refer to *Calculated Point* on page 8-302 for more information.

Calculated Arc

This tool is used to create a graphical circle enclosing an arc based on referenced elements. The circle can be used for tasks such as better defining a circular part or creating a clearance histogram. It can also be used to identify the center of a part with corners, as shown in the following figure.



To create a Calculated Arc tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Calculation** and then **Calculated Arc**. A Calculated Arc tool will be added to the Vision Tools list.

● Calculated Arc Configuration Items

Use the table below to understand the Calculated Arc configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.

Group	Item	Description
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Only Compare Related Results	Specifies which results are compared. Enable to compare only those result values that are connected by instances. When disabled, all results are compared.
	Mode	Select the method used to calculate the frame. Refer to *Calculated Arc - Mode Property* on page 8-297 for more information.
	Offset	Defines the arc offset from the calculated location
	First Arc Point	Select the tool that contains the first, second, or third point on the desired arc.
	Second Arc Point	Select the tool that contains the appropriate point on the desired arc. Only available when Mode is set to Three Points On The Arc.
	Third Arc Point	Select the tool that contains the appropriate point on the desired arc. Only available when Mode is set to Three Points On The Arc.
	Center Arc Point	Select the tool that contains the center point for the arc. Only available when Mode is set to Center Point and 1 Point on Arc
	Link Name	Select the property in the defined Relative To tool that will provide the input values. Available in all Point properties.
	Current Values	Additional property to all point properties. Collection of current input values. Available in all Point properties.

● Calculated Arc - Mode Property

This property defines the way the arc is calculated. Depending on the selection, different tools and properties will need to be referenced. For example, the Center Point and 1 Point on Arc mode as shown in Figure 1-1uses the properties Center Arc Point and First Arc Point to find the circumscribed circle of a grid square. The links to use for those properties are defined by clicking the ellipsis next to each property and then selecting the correct source. Only tools that yield the possible result will be shown in the box.

The possible mode types are listed below.

- Three Points On The Arc: Requires First Arc Point, Second Arc Point, and Third Arc Point. The arc is created across the three points and the center is calculated based on these.
- Center Point and 1 Point on Arc: Requires Center Arc Point and First Arc Point. The arc is created based on the center and calculated radius.



Additional Information

All calculations are adjusted based on Offset.

● Calculated Arc Results

Use the table below to understand the results of the Calculated Arc tool.

Item	Description
Instance	Index of the result instance.
Frame/ Group	Index of the related result. It is associated with the tool that this tool is set Relative To. This will only be different if the tool is set relative to another tool
Arc X	X-coordinate of the arc center.

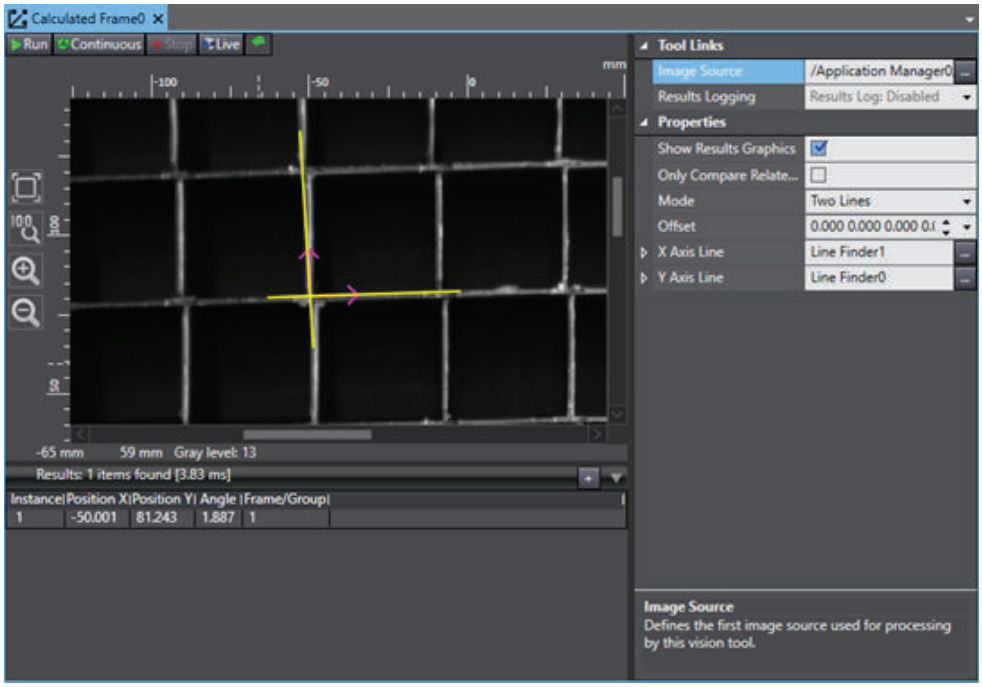
Item	Description
Arc Y	Y-coordinate of the arc center.
Radius	Calculated arc radius.
Opening	Opening of the arc in degrees. Since this tool always generates circles, this is invariably returned as 360.
Rotation	Rotation of the arc. This is always returned as 0.
Thickness	Thickness of the arc. This is always returned as 0.

Calculated Frame

This tool is used to create a vision frame from referenced elements. Frames allow placement of vision tools on objects that are not always in the same location or orientation. When a new vision tool is created, it can be specified to be relative to a vision frame. If the object that defines the vision frame moves, so will the frame and the tools that are relative to that frame.

Additional Information

A fixed frame can also be created using this tool.



To create a Calculated Frame tool, right-click **Vision Tools** in the Multiview Explorer, click **Add Calculation** and then click **Calculated Frame**. A Calculated Frame tool will be added to the Vision Tools list.

● Calculated Frame Configuration Items

Use the table below to understand the Calculated Frame configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.

Group	Item	Description
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Only Compare Related Results	Specifies which results are compared. Enable to compare only those result values that are connected by instances. When disabled, all results are compared.
	Mode	Select the method used to calculate the frame. Refer to *Calculated Frame - Mode Property* on page 8-299 for more information.
	Offset	Defines the frame offset from the calculated location. If Mode is set to Fixed, this refers to the offset from the image origin.
	X Axis Line	Select the tool that contains the appropriate line. Only available when Mode is set to Two Lines or Origin Point Following Line Angle.
	Y Axis Line	Select the tool that contains the appropriate line. Only available when Mode is set to Two Lines.
	Origin Point	Select the tool that contains the appropriate point. Only available when Mode is set to Two Points, One Point, or Origin Point Following Line Angle.
	Positive X Point	Select the tool that contains the appropriate point. Only available when Mode is set to Two Points.
	Origin Transform	Select the tool that contains the appropriate point. Only available when Mode is set to Frame Relative.
	Link Name	Select the property in the defined Relative To tool that will provide the input values. Available in all Line, Point, and Transform properties.
	Current Values	Collection of current input values. Available in all Line, Point, and Transform properties.

● Calculated Frame - Mode Property

This property defines the way the frame is calculated. Depending on the selection, different tools and properties will need to be referenced. For example, the Two Lines mode as shown in the figure above uses the properties X Axis Line and Y Axis Line. The links to use for those properties are selected by clicking the ellipsis next to each property and then selecting the correct source. Only tools that yield the possible result will be shown in the box. Points are generated from most tools, but lines are only generated by Calculated Line and Line Finder tools.

The possible mode types are listed below.

- Two Lines: Requires X Axis Line and Y Axis Line. The frame is positioned at the intersection and oriented so that the X-axis is collinear with X Axis Line.
- Two Points: Requires Origin Point and Positive X Point. The frame is positioned at Origin Point and oriented so that the X-axis points towards Positive X Point.
- Fixed: No required references. The frame is positioned according to the Offset property.
- One Point: Requires Origin Point. The frame is positioned and oriented to match Origin Point.
- Frame Relative: Requires Origin Transform. The frame is positioned and oriented to match Origin Transform.
- Origin Point Following Line Angle: Requires Origin Point and X Axis Line. The frame is positioned at Origin Point and oriented so that the X-axis is parallel to X Axis Line.

Additional Information

All calculations are adjusted based on Offset unless otherwise specified.

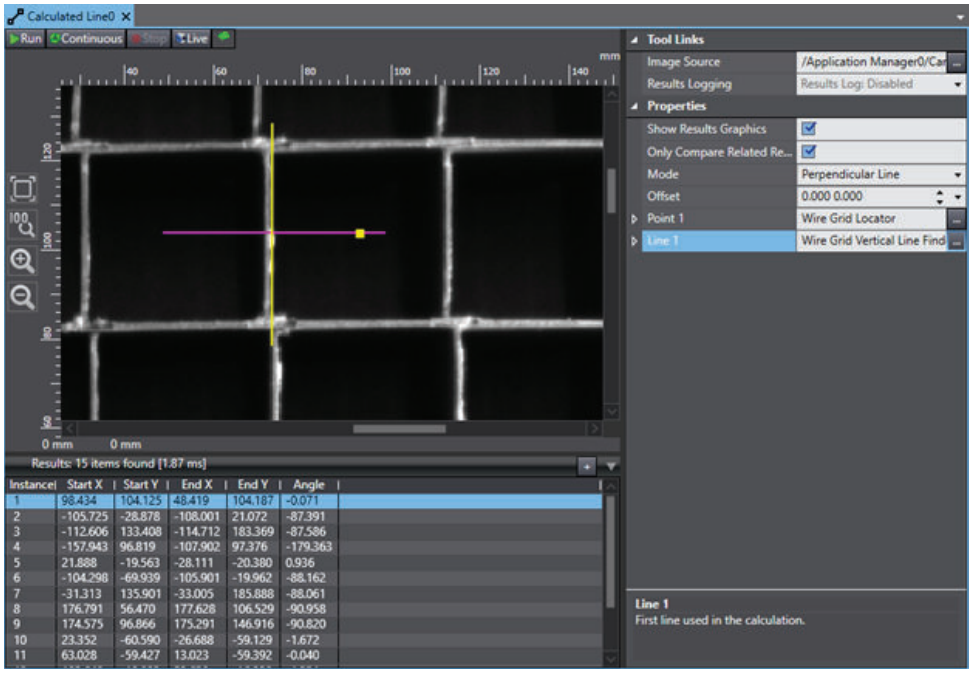
● Calculated Frame Results

Use the table below to understand the results of the Calculated Frame tool

Item	Description
Instance	Index of the result instance.
Frame/ Group	Index of the related result. It is associated with the tool that this tool is set Relative To. This will only be different if the tool is set relative to another tool
Position X	X-coordinate of the instance origin.
Position Y	Y-coordinate of the instance origin.
Angle	Angle of the instance with respect to the X-axis of the camera coordinate system.

Calculation Line

This tool is used to create lines based on referenced elements These lines are primarily used to create other graphical features, such as a Calculated Point or Calculated Frame.



To create a Calculated Line tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Calculation** and then **Calculated Line**. A Calculated Line tool will be added to the Vision Tools list.

● Calculated Line Configuration Items

Use the table below to understand the Calculated Line configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.

Group	Item	Description
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Only Compare Related Results	Specifies which results are compared. Enable to compare only those result values that are connected by instances. When disabled, all results are compared.
	Mode	Select the method used to calculate the line. Refer to Calculated Line - Mode Property on page 1 for more information.
	Offset	Defines the line offset from the calculated location.
	Point 1	Select the tool that contains the first endpoint of the line.
	Point 2	Select the tool that contains the second endpoint of the line. Only available when Mode is set to Two Points.
	Line 1	Select the tool that contains the line perpendicular to the calculated line. Only available when Mode is set to Perpendicular Line.
	Link Name	Select the property in the defined Relative To tool that will provide the input values. Available in all Line, Point, and Transform properties.
	Current Values	Collection of current input values. Available in all Line, Point, and Transform properties.

● Calculated Line - Mode Property

This property defines the way the line is calculated. Depending on the selection, different tools and properties will need to be referenced. For example, the Perpendicular Line mode as shown in Figure 1-1uses the properties Point 1 and Line 1. The links to use for those properties are selected by clicking the ellipsis next to each property and then selecting the correct source. Only tools that yield the possible result will be shown in the box. Points are generated from most tools, but lines are only generated by Calculated Line and Line Finder tools.

The possible mode types are listed below.

- Two Points: Requires Point 1 and Point 2. The line is positioned between the two points.
- Perpendicular Line: Requires Point 1 and Line 1. The line is positioned with an endpoint at Point 1 and passes through Line 1 so that the two lines are perpendicular.



Additional Information

All calculations are adjusted based on Offset.

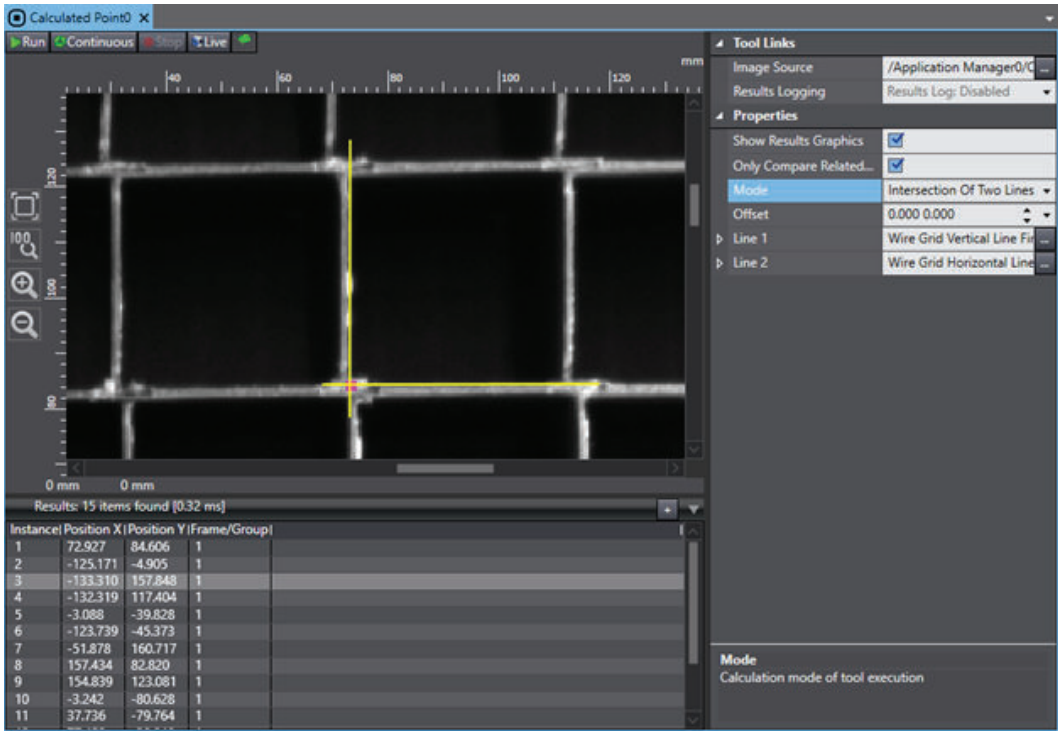
● Calculation Line Results

Use the table below to understand the results of the Calculation Line tool.

Item	Description
Instance	Index of the result instance.
Frame/Group	Index of the related result. It is associated with the tool that this tool is set Relative To. This will only be different if the tool is set relative to another tool.
Start X	X-coordinate of the first endpoint.
Start Y	Y-coordinate of the first endpoint.
Center X	X-coordinate of the midpoint.
Center Y	Y-coordinate of the midpoint.
End X	X-coordinate of the second endpoint.
End Y	Y-coordinate of the second endpoint.
Angle	Calculated angle of the line with respect to the X-axis of the camera coordinate system.

Calculated Point

This tool is used to create points based on referenced elements. These points can be used to create other graphical features, such as Calculated Arc or Calculated Line, or to act as a reference point from which other measurements can be made.



To create a Calculated Point tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Calculation** and then **Calculated Point**. A Calculated Point tool will be added to the Vision Tools list.

● Calculated Point Configuration Items

All calculations are adjusted based on Offset unless otherwise specified.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.

Group	Item	Description
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Only Compare Related Results	Specifies which results are compared. Enable to compare only those result values that are connected by instances. When disabled, all results are compared.
	Mode	Select the method used to calculate the point. Refer to *Calculated Point - Mode Property* on page 8-304.
	Offset	Defines the point's offset from the calculated location.
	Point 1	Select the tool that contains the first reference point. Only available when Mode is set to Midpoint, Point On A Line Closest To A Point, or Point On An Arc Closest To A Point.
	Point 2	Select the tool that contains the second reference point. Only available when Mode is set to Midpoint.
	Line 1	Select the tool that contains the first reference line. Only available when Mode is set to Point On A Line Closest To A Point, Line – Arc Intersection, or Intersection Of Two Lines.
	Line 2	Select the tool that contains the second reference line. Only available when Mode is set to Intersection Of Two Lines.
	First Arc	Select the tool that contains the first reference arc. Only available when Mode is set to Point On An Arc Closest To A Point, Line – Arc Intersection, or Intersection of Two Arcs.
	Second Arc	Select the tool that contains the second reference arc. Only available when Mode is set to Intersection of Two Arcs.
	Link Name	Select the property in the defined Relative To tool that will provide the input values. Available in all Line, Point, and Transform properties.
	Current Values	Collection of current input values. Available in all Line, Point, and Transform properties.

● Calculated Point - Mode Property

This property defines the way the point is calculated. Depending on the selection, different tools and properties will need to be referenced. For example, the Intersection Of Two Lines mode uses the properties Line 1 and Line 2. The links to use for those properties are selected by clicking the ellipsis next to each property and then selecting the correct source. Only tools that yield the correct type of result will be shown in the box. Points are generated by most tools, but lines and arcs are only generated by Calculated Arc/Line and Arc/Line Finder tools.

The possible mode types are listed below.

- Midpoint: Requires Point 1 and Point 2. The point is positioned halfway between the two points.
- Point On A Line Closest To A Point: Requires Point 1 and Line 1. The point is positioned on the line as close to the reference point as possible.
- Point On An Arc Closest To A Point: Requires Point 1 and First Arc. The point is positioned on the arc as close to the reference point as possible.
- Fixed: No required references. The frame is positioned according to the Offset property.
- Line – Arc Intersection: Requires First Arc and Line 1. The point is the positioned at the intersection of the line and the arc.
- Intersection Of Two Lines: Requires Line 1 and Line 2. The point is positioned at the intersection of the two lines.
- Intersection of Two Arcs: Requires First Arc and Second Arc. The point is positioned at the intersection of the two points.



Additional Information

All calculations are adjusted based on Offset unless otherwise specified.

● Calculated Point Results

Use the table below to understand the results of the Calculated Point tool.

Item	Description
Instance	Index of the result instance.
Frame/ Group	Index of the related result. It is associated with the tool that this tool is set Relative To. This will only be different if the tool is set relative to another tool
Position X	X-coordinate of the point location.
Position Y	Y-coordinate of the point location.

8-8-9 Image Process Tools

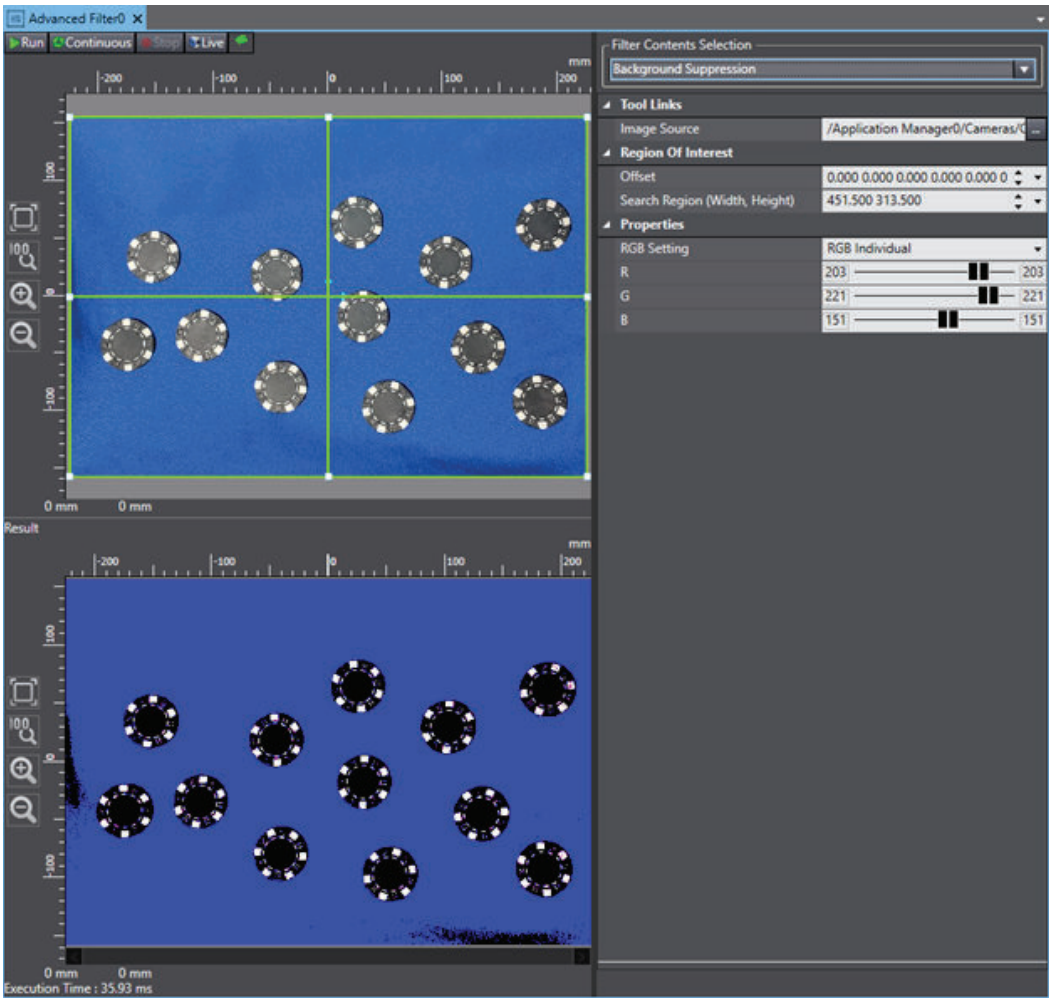
Image Process tools are used to manipulate image data. The output image of an Image Process tool can be used as an Image Source by another tool.

The following Image Process tools are described in this section.

- Refer to *Advanced Filter* on page 8-305 for more information.
- Refer to *Color Matching* on page 8-312 for more information.
- Refer to *Image Processing* on page 8-315 for more information.
- Refer to *Image Sampling* on page 8-318 for more information.
- Refer to *Position Compensation* on page 8-321 for more information

Advanced Filter

This tool applies a filter or operation to the input image to better facilitate processing by other tools, including additional Advanced Filters. It can be used to perform tasks such as Background Suppression, Color Gray Filter, Erosion/Dilation, and edge extraction. In this way, it can prepare the image to be used by less versatile tools. For example, the Background Suppression filter shown in the following figure can be used by a Locator or Shape Search 3 tool to more easily detect the poker chips.



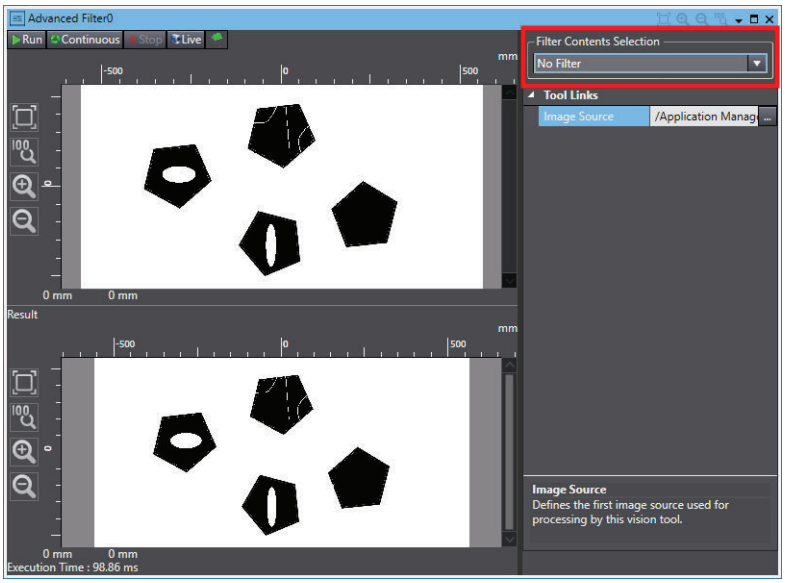
To create an Advanced Filter tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Image Process** and then click **Advanced Filter**. An Advanced Filter tool will be added to the Vision Tools list.

Additional Information

The Advanced Filter is also available through the Recipe Manager.

● Advanced Filter - Filter Contents Selection

The following filters can be applied by selecting the appropriate option in Filter Contents Selection.



■ No Filter

This is the default setting and makes no change to the input image. This setting is typically used to temporarily disable Advanced Filter during testing and creation of the application. It has no practical use during run time.

■ Smoothing Weak/ Strong

This filter blends each pixel value with the others in its neighborhood to blur the image, a process used to reduce detail and emphasize deviations. This is used to blur an image to emphasize contrast. In particular, it can be used to reduce deviations between individual pixels to make consistent deviations more distinct, thus improving edge detection. Smoothing Weak and Smoothing Strong filters function by using a Gaussian blur to distribute color / gray level within neighborhoods of pixels. Smoothing Strong does this to a higher degree than Smoothing Weak.

■ Dilation

This filter changes the value of each pixel to the brightest (highest gray level value) within its neighborhood. This is used to artificially raise the brightness of an image. Dilation determines the highest gray level value within a kernel and then changes all pixels within that kernel to match it.

Additional Information

Since large sections of pixels are changing each time the tool is run, the quality of the image decreases with each iteration.

■ Erosion

This filter changes the value of each pixel to the darkest (lowest gray level value) within its neighborhood. This is used to artificially lower the brightness of an image. Erosion determines the lowest gray level value within a kernel and then changes all pixels within that kernel to match it.

Similar to Dilation, the quality of the image decreases with each iteration.

■ Median

This filter changes the value of each pixel to the median gray level value of the pixels within its neighborhood. This is used to reduce details and defects while maintaining the overall shapes and edges within an image. Median determines the median values within a neighborhood and enhances those over the other pixels.
Similar to Dilation and Erosion, the quality of the image decreases with each iteration.

■ Edge Extraction

This filter changes all pixels to black except for those on detected edges. If there is a strong deviation between pixels, then those pixels are highlighted. Otherwise, they are changed to black to emphasize the edges.

■ Vertical Edge Extraction

This filter performs Edge Extraction only with edges detected from deviations that would result in vertical edges.

■ Horizontal Edge Extraction

This filter performs Edge Extraction only with edges detected from deviations that would result in horizontal edges.

■ Edge Enhance

This filter blends pixel values along detected deviations to increase edge visibility. This emphasizes the deviation between touching dark and light regions. It is mainly used in the case of blurry images to clearly show the offset between light and dark regions.

■ Color Gray Filter

This filter returns a gray scale image by setting a standard white color. The image can be process based on RGB or HSV colors.
When RGB is selected, the option selected in the RGB Filter property will define the governing color scheme of the filter. For example, if Red is selected in the RGB Filter property, the gray level value in the resulting pixels will be equal to the red value of the original pixels. This works similarly for HSV except that everything outside of the defined tolerances will become black in the resultant image. Refer to *8-8-4 Color Spaces* on page 8-197 for more information.

■ Background Suppression

This filter sets a threshold for gray level or individual / collective color level to filter backgrounds. It is designed to filter out the background of images in order to emphasize the targets in the image. This supports both gray scale and color images. In both cases, a defined range of color or gray level values will be passed through the filter.
All pixels with color levels or gray level below the range will be converted to black and all pixels above the range will be converted to white (or Red, Green, or Blue, if RGB Setting is set to RGB Individual).

■ Labeling

Similar to the Labeling tool, this filter isolates masses of pixels that fall within a certain color range and meet different extraction conditions. Set a single color by right-clicking in the image

or set a color range by right-clicking and dragging to establish a region. Alternatively, manually enter the colors using the Color Region section in the Properties.
Once one or more color thresholds have been established, the tool will filter everything out of the image except for the pixels that fall within the ranges. Additional extraction conditions can be set to further limit the identified regions. Unlike the Labeling tool, the regions are not returned as data. Instead, Labeling Filter results in a new image.

■ Image Operation/ 2 Image Operation

The Image Operation filter uses mathematical or bit operations to alter the value of pixels using a constant operand value.
The 2 Image Operation filter uses mathematical or bit operations to alter the value of pixels by using a second image as an operand.
In a 2 Image Operation filter, the images must be the same size and type (monochrome or color) and the operand value is the gray level or color value of corresponding pixels. The operation options are listed below.

- Arithmetic Operation: Perform a mathematical calculation between the two values.
  - Add: Add the operand value to the pixel gray level to a maximum of 255.
  - Subtraction: Subtract the operand value from the pixel gray level to a minimum of 0.
  - Subtraction (Absolute): Subtract the operand value from the pixel gray level with no minimum. The resulting gray level will be the absolute value of the operation result.
  - Multiplication: Multiply the operand value by the pixel gray level to a minimum of 0 and a maximum of 255.
  - Multiplication (Normalization): Performs a multiplication operation and then normalizes for brightness.
  - Average: Only available for 2 Image Operations. Returns the average of the corresponding pixel values.
  - Maximum: Only available for 2 Image Operations. Returns the maximum of the corresponding pixel values.
  - Minimum: Only available for 2 Image Operations. Returns the minimum of the corresponding pixel values.
- Bit Operation: Perform a logical test on the image.
  - NOT: Reverses the RGB or gray level polarity of the image, regardless of the operand value.
  - AND: Performs an AND operation on the pixels by comparing the binary digits of the two values. It is generally used to compare two images. It can also be used in masking.
  - OR: Performs an OR operation on the pixels by comparing the binary digits of the two values. It is generally used to merge two images together.
  - XOR: Performs an XOR operation on the pixels by comparing the binary digits of the two values. It is generally used to create a binary image (black and white).
  - NAND: Performs a NAND operation on the two pixels by comparing the binary digits of the two values. Like AND, this is used to compare images, but it also essentially returns negatives of what the AND result would be.
  - NOR: Performs a NOR operation on the pixels by comparing the binary digits of the two values. It is generally used to merge two images together and return the negatives.
  - XNOR: Performs an XNOR operation on the pixels by comparing the binary digits of the two values. It is generally used to create a negative binary (black and white) image.

- Bit Shift: Shift the values of the binary digits to the left or right. The bits on the digit that is shifted away from (first digit in a right shift and last digit in a left shift) becomes 0. This is only available in a single image operation. This is used when Arithmetic Multiplication is too computationally taxing.
- Change Pixel Value: Assign a fixed value to all pixels that fall within a certain gray level range. Change Pixel Value defines the resulting gray level and the Bounds define the range to be changed or retained, depending on the Change Pixel Mode.

● Advanced Filter Configuration Items

The properties of Advanced Filter change depending on what type of filter is selected in Filter Contents Selection. The table below shows all properties.
The Filter Contents Selection column defines those filter(s) for which the property is available. If it is blank, the property is available for all or most of them.

Group	Filter Contents Selection	Item	Description
Tool Links		Image Source	Defines the image source used for processing by this vision tool.
	2 Image Operation	Image Source2	Defines the second image source used as the operand for 2 Image Operations.
Region Of Interest		Offset	Defines the center coordinates of the region.
		Search Region (Width, Height)	Defines the size of the region.
Properties		Iteration	Defines the amount of iterations the filter will run each time this tool is initiated.
		Kernel Size	Sets the size of pixel neighborhoods used in the processing of this tool.
	Color Gray Filter	Filter Kind	Selects whether the image will be filtered in RGB or HSV.
	Color Gray Filter	RGB Filter	Determines what RGB value(s) will be considered to be the gray level value in the resulting image. Only available when Filter Kind is set to RGB.
	Color Gray Filter	Gain (Red/Green/Blue)	These set the scale from 0-1 of how the gray level value is calculated. Only available when Filter Kind is set to RGB and RGB Filter is set to custom.
	Color Gray Filter	Standard Hue	Defines the nominal hue color that the filter will return on a 360-degree circle. Red is 0, green is 120, and blue is 240. Only available when Filter Kind is set to HSV.

Group	Filter Contents Selection	Item	Description
Properties	Color Gray Filter	Hue Range	Sets the tolerance for the Hue value within which the gray level value will be returned. Only available when Filter Kind is set to HSV.
	Color Gray Filter	Color Chroma	Sets the bounds for the filter's saturation range.
	Background Suppression	Gray	Sets the gray level threshold to be retained in the results. Only available with gray scale images.
	Background Suppression	RGB Setting	RGB Common adjusts R, G, and B all together and RGB Individual adjusts them separately. Only available with color images. Select the method of color threshold creation.
	Background Suppression	RGB Common	Sets the threshold of color to be retained in the results. Only available with color images and when RGB Setting is set to RGB Common.
	Background Suppression	R/G/B	Sets the threshold of each individual color setting to be retained in the results. Only available with color images and when RGB Setting is set to RGB Individual.
	Labeling Filter	Hole Plug Color	Sets the color that will fill all holes in the detected masses.
	Labeling Filter	Outside Trimming	When enabled, everything outside of the region of interest will be returned as the extracted color.
	Labeling Filter	Label Select	Allows the selection of a single extracted mass. The sort condition and number define which one will be selected.
	Labeling Filter	Neighborhood Mode	Defines the ways pixels will be connected in order to qualify as a single mass. 4 Neighborhood will only allow pixels to connect to others that are up, down, left, or right, while 8 Neighborhood will also connect diagonal directions.
	Labeling Filter	Extract Condition	Defines the conditions for this tool.
	Labeling Filter	1/2/3	Will consider when extracting masses from the image. Conditions can be set by the type (kind) and minimum and maximum values.

Group	Filter Contents Selection	Item	Description
Properties	Labeling Filter	Extract Condition Setting	Defines how the combined extraction settings will behave. Select **And** to return only those masses that satisfy all conditions and Or to return masses that satisfy any condition.
	Labeling Filter	Binary Image	Returns a monochrome image.
	Image Operation/2 Image Operation	Operation Type	Defines the type of operation that will be performed on the image(s).
	Image Operation/2 Image Operation	Arithmetic Mode	Sets the type of operation to be performed on the image(s). Only available when Operation Type is set to *Arithmetic Operation.* . Refer to *Image Operation/ 2 Image Operation* on page 8-308 for more information.
	Image Operation	Arithmetic Value	Sets the operand value for the arithmetic operation. Only available when Operation Type is set to *Arithmetic Operation.*
	Image Operation/2 Image Operation	Operation Mode	Sets the type of operation to be performed on the image(s). Only available when Operation Type is set to Bit Operation. Refer to *Image Operation/ 2 Image Operation* on page 8-308 for more information.
	Image Operation	Operation Value	Sets the operand value for the bit operation. Only available when Operation Type is set to **Bit Operation.**
	Image Operation	Bit Shift Mode	Sets the type of operation to be performed on the image. Only available when Operation Type is set to **Bit Shift.** Refer to *Image Operation/ 2 Image Operation* on page 8-308 for more information.
	Image Operation	Bit Shift Value	Sets the operand value for the bit shift. Only available when Operation Type is set to **Bit Shift.**
	Image Operation	Change Pixel Mode	Selects whether the pixels outside or inside the bounds will be changed. Only available when Operation Type is set to **Change Pixel Value.**
	Image Operation	Change Pixel Value	Sets the gray level value to which pixels will be changed. Only available when Operation Type is set to **Change Pixel Value**.
	Image Operation	Bounds	Sets the bounds that define what gray level values will be changed or retained. Only available when Operation Type is set to **Change Pixel Value**.

● Configuring the Advanced Filter

The Advanced Filter tool can execute several types of functions on an image, so it is important that the appropriate choice is selected from the *Filter Contents Selection* menu before any other changes are made. The Properties will adjust to match the type of filter.

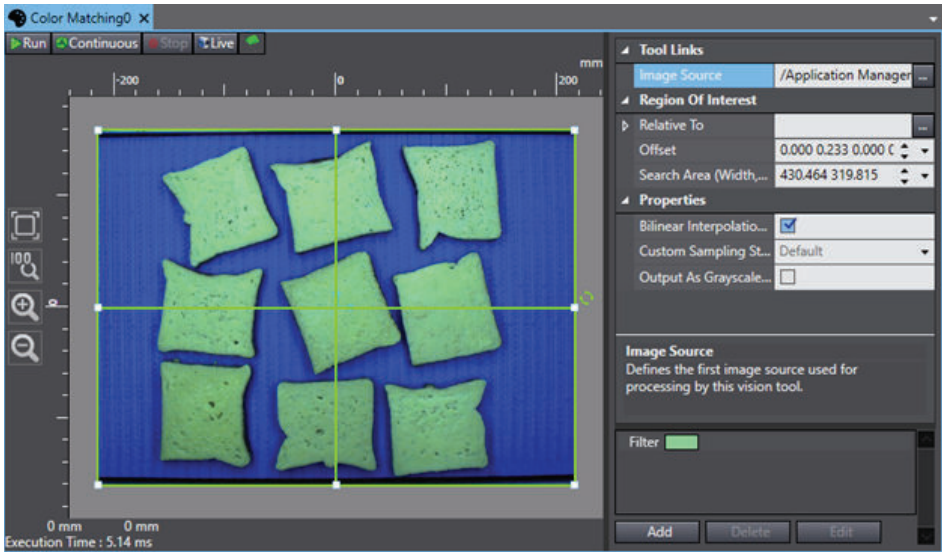
There is no generic way to configure this type of tool because it is designed to perform a multitude of different operations. However, in general, the tool is operated by first adjusting the region of interest to the appropriate location and size. The exception to this is Color Gray Filter, which does not have a region of interest. Any filters or operations performed will take place only within the established region.
Most filter types require the Iteration and Kernel Size item. The Iteration value can be increased for the tool to run multiple times whenever executed. This leads to higher processing time, but it can be useful to remove detail from an image. Conversely, Kernel Size affects the size of the pixel neighborhoods. Larger selections lead to fewer calculations and faster processing time. Modifying these will impact performance and can be used to yield optimum images.
Several filters use both RGB and HSV color schemes. Refer to *8-8-4 Color Spaces* on page 8-197 for more information.

● Advanced Filter Results

Advanced Filter returns a modified image that can be used by other vision tools. To do this, set Advanced Filter as the Image Source property of the subsequent tool. The resultant image can be viewed in the Result section of the object editor.

Color Matching

This tool searches and analyzes images to find areas of color that match user-defined filters. It is typically used to analyze an area on an object for the purpose of verifying if the object meets defined color criteria. It can also be used to filter unnecessary colors from the image before use in later tools.
The Color Matching tool applies defined filters to the region of interest for this analysis. Any number of filters can be added.



To create a Color Matching tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Image Process** and then **Color Matching**. A Color Matching tool will be added to the Vision Tools list.

● Color Matching Configuration Items

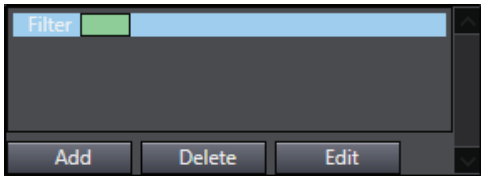
Use the table below to understand the Color Matching configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.

Group	Item	Description
Region Of Interest	Relative to	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the defined Relative To tool that will provide the input values.
	Current Values	Collection of current input values.
	Search Area (Width, Height)	Defines the size of the region.
Properties	Bilinear Interpolation Enabled	Enables bilinear interpolation to sample the input image. By default, this is enabled because it ensures subpixel accuracy.
	Custom Sampling Set	Defines the sampling step used in calculation. This is set to 1 by default. Enable this to adjust the setting.
	Output As Gray scale Image	Specifies the color scheme of the output image. When enabled, the resultant image is converted to gray scale after color filters are applied.

● Color Matching Filters Pane

The filters required for Color Matching are organized using the pane located below the properties. Color Matching will output the colors that are included in any filter listed in this pane. All other pixels in the image will be output as black.

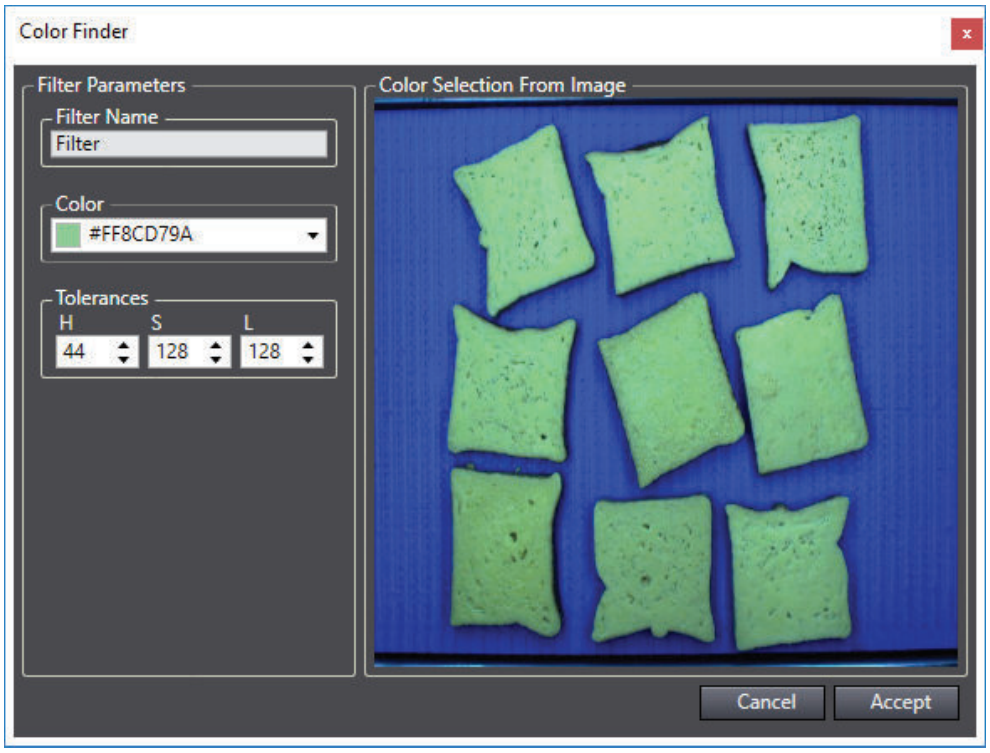


Any number of filters can be added to the pane. They are managed using the three buttons described below.

- Add: Creates a new filter and adds it at the end of the current list of filters.
- **Delete**: Removes the currently selected filter. This is only available if a filter is highlighted.
- **Edit**: Opens the Color Finder editing window. This is only available if a filter is highlighted.

● Color Matching - Color Finder Editor

The Color Finder editor allows modification of specific filters.
The right side of the editor shows the input image. The left side is where the Filter Parameters are adjusted as described below.



■ Filter Name

The name of the filter as it appears in the Filters Pane. This is modified by typing in the text box and causes no change to the operation of the tool. It is used to label filters and is useful when there are multiple filters.

■ Color

Defines the color to be searched for by the filter and included in the resulting image. This is defined in two different ways:

1. Right-click and drag in the image on the right to return the average color from the resulting rectangular region. For example, the Color in Figure 1-2was created by clicking and dragging within one of the green-tinted bread slices of the corresponding image.
2. Click the arrow next to the Color field and choose a specific color from the color wheel window.

■ Tolerances

Specifies the ranges of allowable color from the nominal one defined by the Color parameter. Any color that has a Hue (H), Saturation (S), or Luminescence (L) value that has a smaller difference from the nominal than the values defined in the H, S, and L fields is included in the resultant image. These are initially set to 25 by default, but using the right-click method to select a color will automatically change these so that the entire region defined in the image is output.

● Color Matching Results

Color Matching returns a modified image that can be used by other vision tools. This image will only include the colors defined in the filters. All other pixels will be returned as black. To use this image with another tool, set Color Matching as the Image Source property of the subsequent tool. The resultant image can be viewed in the Vision Window as shown in the following figure.

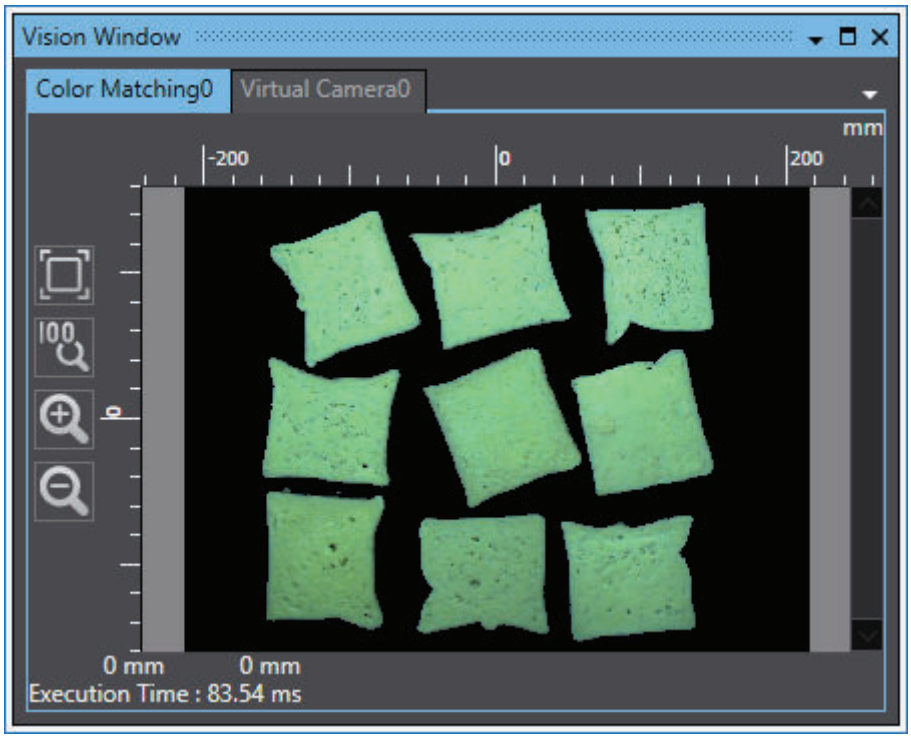
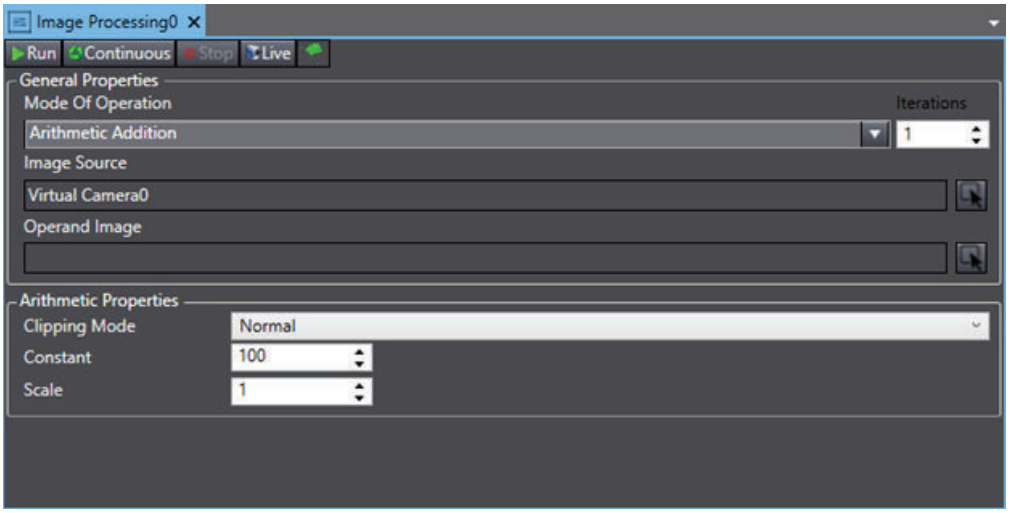


Image Processing

This tool applies a filter to gray scale images to better facilitate processing by other tools, including additional Image Processing tools.

Additional Information

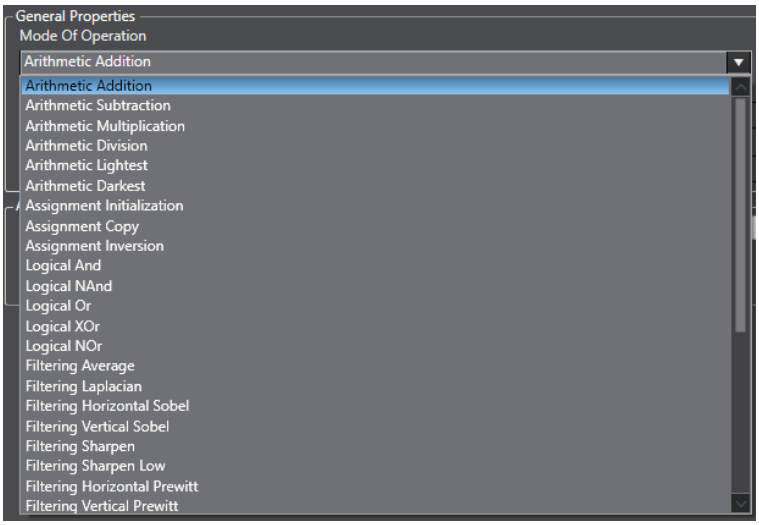
Advanced Filter tool provides more capabilities and may offer faster execution times than the Image Processing tool. Refer to Advanced Filter on page 1 for more information.



To create an Image Processing tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Image Process** and then **Image Processing**. An Image Processing tool will be added to the Vision Tools list.

● Image Processing - Mode of Operation

The following filters can be applied by selecting the appropriate option in Mode Of Operation.



Mode of Operation	Description
Arithmetic Addition	Add the operand value to the pixel gray level to a maximum of 255.
Arithmetic Subtraction	Subtract the operand value from the pixel gray level to a minimum of 0.
Arithmetic Multiplication	Multiply the operand value by the pixel gray level to a minimum of 0 and a maximum of 255.
Arithmetic Division	Divide the operand value by the pixel gray level to a minimum of 0 and a maximum of 255.
Arithmetic Lightest	Returns the maximum of the pixel value and operand value.
Arithmetic Darkest	Returns the minimum of the pixel value and operand value.
Assignment Initialization	Assigns a defined gray-level value to all pixels in the image.
Assignment Copy	Copies the input value of each pixel to the corresponding output pixels. Virtually no change is made to the image.
Assignment Inversion	Inverts the gray level value of each pixel and outputs it.
Logical And	Performs an AND operation on the pixels by comparing the binary digits of the two values.
Logical NAnd	Performs a NAND operation on the two pixels by comparing the binary digits of the two values.
Logical Or	Performs an OR operation on the pixels by comparing the binary digits of the two values.
Logical XOr	Performs an XOR operation on the pixels by comparing the binary digits of the two values.
Logical Nor	Performs an OR operation on the pixels by comparing the binary digits of the two values.
Filtering Average	Changes the color of each pixel to the average gray level value of the pixels within its neighborhood.
Filtering Laplacian	Increases or decreases the gray level values of light and dark pixels at high contrast areas, respectively, to enhance edges. This filter is extremely sensitive to noise and should only be performed after the image has been blurred or smoothed.
Filtering Horizontal Sobel	Uses the Sobel operator to brighten pixels at horizontal edges and darken the others. This has slightly better noise filtering than Filtering Horizontal Prewitt.

Mode of Operation	Description
Filtering Vertical Sobel	Uses the Sobel operator to brighten pixels at vertical edges and darken the others. This has slightly better noise filtering than Filtering Vertical Prewitt.
Filtering Sharpen	Increases or decreases the gray level values of light and dark pixels, respectively, to increase contrast.
Filtering Sharpen Low	Increases or decreases the gray level values of light and dark pixels, respectively, to increase contrast. This filter performs this to a lesser degree than Filtering Sharpen.
Filtering Horizontal Prewitt	Increases or decreases the gray level values of light and dark pixels, respectively, to increase contrast. This filter performs this to a lesser degree than Filtering Sharpen.
Filtering Vertical Prewitt	Uses the Prewitt operator to brighten pixels at horizontal edges and darken the others. This has slightly worse noise filtering than Filtering Horizontal Sobel.
Filtering Gaussian	Blurs the image to remove contrast between individual pixels, removing noise and enhancing overall contours.
Filtering High Pass	Increases or decreases the gray level values of light and dark pixels at high contrast areas, respectively, to enhance edges. This filter is extremely sensitive to noise and should only be performed after the image has been blurred or smoothed. It is functionally identical to Filtering Laplacian.
Filtering Median	Changes the color of each pixel to the median gray level value of the pixels within its neighborhood.
Morphological Dilate	Changes the color of each pixel to the brightest (highest gray level value) within its neighborhood.
Morphological Erode	Changes the color of each pixel to the darkest (lowest gray level value) within its neighborhood.
Morphological Close	Removes small dark particles and holes within an image.
Morphological Open	Removes peaks from an image, leaving only the image background.
Histogram Equalization	Enhances the input image by flattening the histogram of the input image.
Histogram Stretching	Increases the contrast in an image by applying a piecewise linear intensity transformation based on the image histogram.
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.
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.
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).

● Image Processing Configuration

The Image Processing tool is generally configured by selecting a Mode Of Operation and one or two input images. All of the available Modes can be performed with one image and some can be performed with two.

The properties in the bottom half of the tool editor will change depending on the selected Mode Of Operation, as shown in the following table. Any mode with a Constant property can be operated with two images. If an Operand Image is selected, the filter will use the values of the corresponding pixels in that image. Otherwise, it will default to the Constant property.

● Image Processing Configuration Items

The properties of Image Processing change depending on what type of filter is selected in Mode Of Operation. General properties are described below.

- Mode of Operation: Defines the type of operation applied to the image(s).
- Iterations: Defines the amount of iterations the filter will run
- each time this tool is initiated.
- Image Source: Select the source of the input image. This must be gray scale.
- Operand Image: Select the source of the operand image. This must be gray scale.

The table below describes all other properties specific to the Mode of Operation selection. The Mode Of Operation column defines the mode(s) for which the property is available.

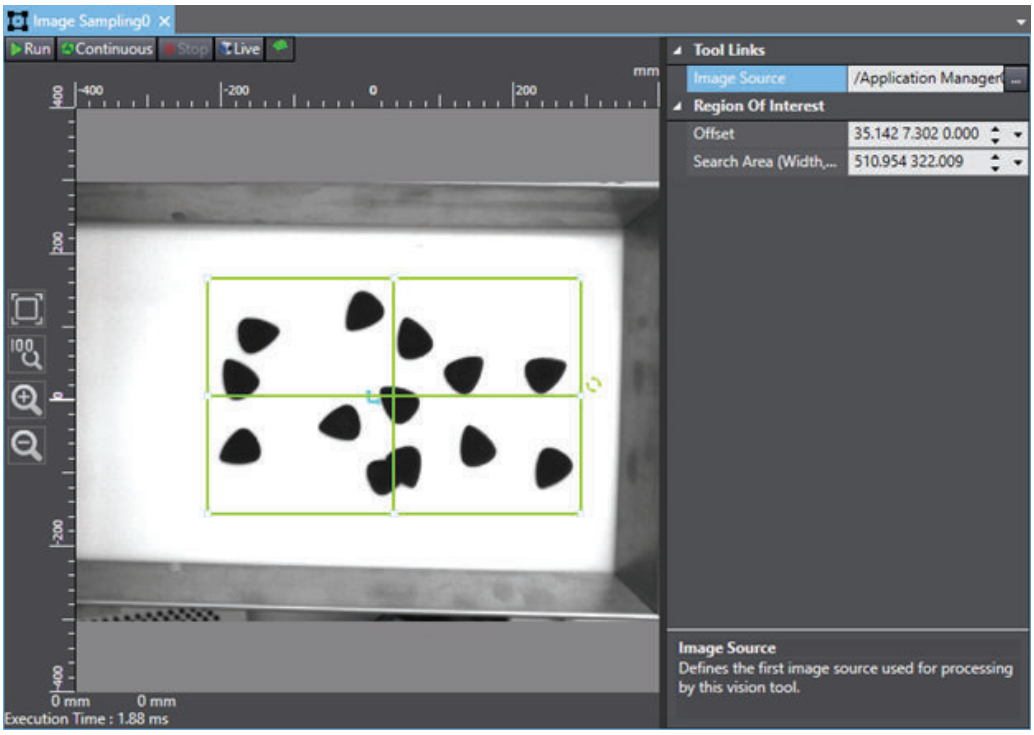
Mode of Operation	Properties	Description
Arithmetic Operations	Clipping Mode	Select the method by which the calculation handles resultant values below 0. Normal converts all of them to 0 while Absolute returns the absolute value of any result below 0.
	Constant	Value used in operation if there is no operand image.
	Scale	Defines the initial gray level scale of the image before processing. For example, if this value is equal to 0.5, all gray level values are halved before the operation is performed.
Assignment Initialization	Constant	Value used in operation if there is no operand image.
Logical Operations	Constant	Value used in operation if there is no operand image.
Filtering Operations	Clipping Mode	Select the method by which the calculation handles resultant values below 0. Normal converts all of them to 0 while Absolute returns the absolute value of any result below 0.
	Kernel Size	Defines the size of the square neighborhoods of pixels that the image is split into during filter operation.
	Scale	Defines the initial gray level scale of the image before processing. For example, if this value is equal to 0.5, all gray level values are halved before the operation is performed.
Morphological Operations	Neighborhood Size	Defines the size of the square neighborhoods of pixels that the image is split into during morphological operation. This property cannot be changed.
Histogram Operations	Threshold	Defines the gray level value threshold for the histogram. The gray level values of all pixels in the image change to either 0 or 255 depending on this value and the type of histogram (available for Histogram Light / Dart Threshold Mode of Operation only).
Transform Operations	Flags	Defines the method of calculation performed by each transform.

● Image Processing Results

There are no results in the tool editor. Image Processing outputs a modified version of the input image and it does not return any data. The modified image can be viewed in the Vision Window or by selecting **Image Processing** as another tool's Image Source.

Image Sampling

This tool is used to extract a rectangular section of an image and output it as a separate image



To create an Image Sampling tool, right-click **Vision Tools** in the Multiview Explorer, select **Add Image Process** and then **Image Sampling**. An Image Sampling tool will be added to the Vision Tools list.

● Image Processing Configuration Items

The properties of Image Processing change depending on what type of filter is selected in Mode Of Operation. General properties are described below.

- Mode of Operation: Defines the type of operation applied to the image(s).
- Iterations: Defines the amount of iterations the filter will run
- each time this tool is initiated.
- Image Source: Select the source of the input image. This must be gray scale.
- Operand Image: Select the source of the operand image. This must be gray scale.

The table below describes all other properties specific to the Mode of Operation selection. The Mode Of Operation column defines the mode(s) for which the property is available.

Mode of Operation	Properties	Description
Arithmetic Operations	Clipping Mode	Select the method by which the calculation handles resultant values below 0. Normal converts all of them to 0 while Absolute returns the absolute value of any result below 0.
	Constant	Value used in operation if there is no operand image.
	Scale	Defines the initial gray level scale of the image before processing. For example, if this value is equal to 0.5, all gray level values are halved before the operation is performed.
Assignment Initialization	Constant	Value used in operation if there is no operand image.
Logical Operations	Constant	Value used in operation if there is no operand image.

Mode of Operation	Properties	Description
Filtering Operations	Clipping Mode	Select the method by which the calculation handles resultant values below 0. Normal converts all of them to 0 while Absolute returns the absolute value of any result below 0.
	Kernel Size	Defines the size of the square neighborhoods of pixels that the image is split into during filter operation.
	Scale	Defines the initial gray level scale of the image before processing. For example, if this value is equal to 0.5, all gray level values are halved before the operation is performed.
Morphological Operations	Neighborhood Size	Defines the size of the square neighborhoods of pixels that the image is split into during morphological operation. This property cannot be changed.
Histogram Operations	Threshold	Defines the gray level value threshold for the histogram. The gray level values of all pixels in the image change to either 0 or 255 depending on this value and the type of histogram (available for Histogram Light / Dart Threshold Mode of Operation only).
Transform Operations	Flags	Defines the method of calculation performed by each transform.

● Image Sampling Configuration

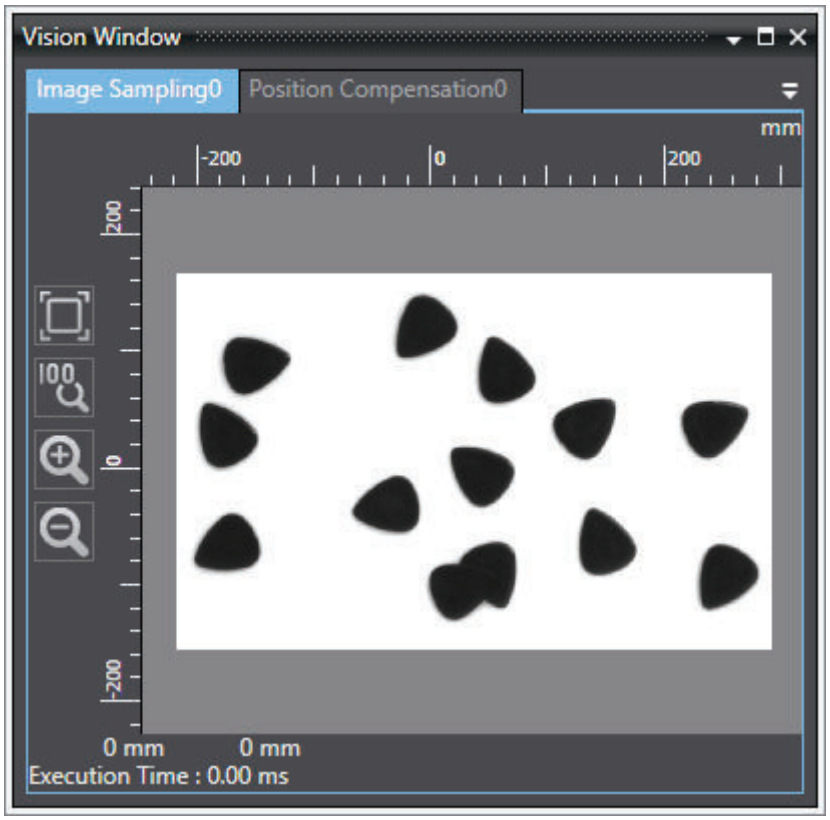
The Image Sampling tool is primarily used to limit image processing to a specific area of the vision field of view. This means that subsequent tools will only need to view the resultant region of this tool. For example, in Figure 1-1the image is cropped so that only the region with guitar chips is returned.

To do this, the region of interest can be defined in the target location. Everything outside of it will be removed from the resultant image, which can be used as an image source in other tools.



Additional Information

The resultant image does not automatically center the new image. The new image will be returned with respect to the original image frame. This can be seen in the figure below where the right extent and left extent are not centered about zero.

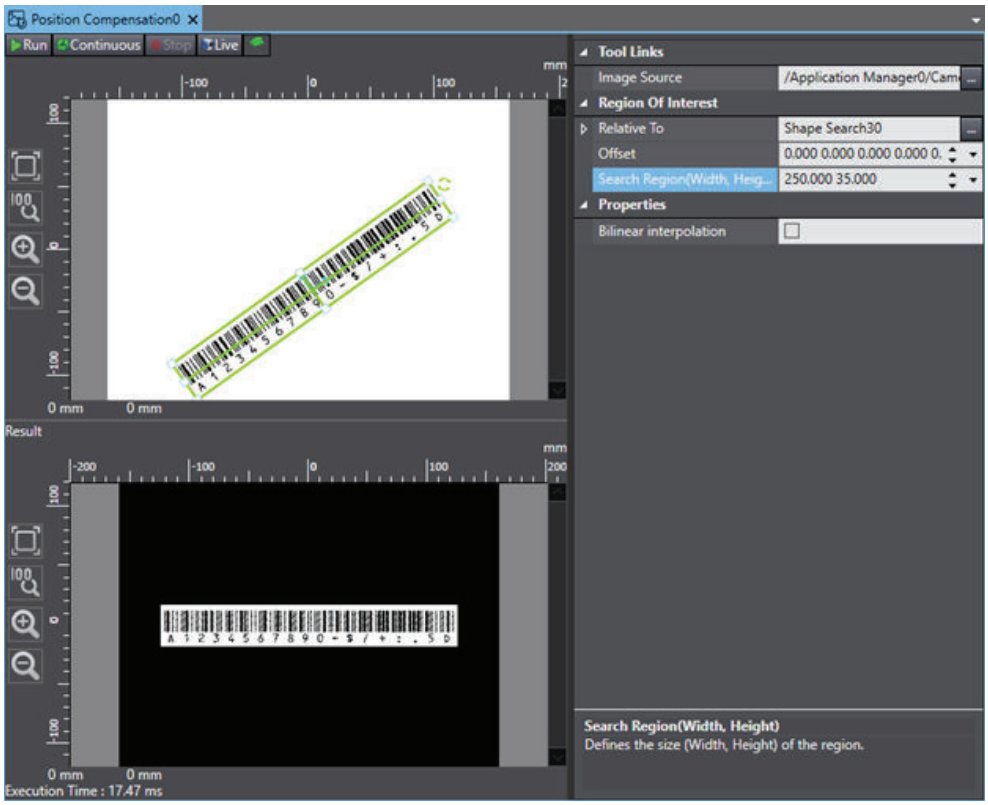


● Image Sampling Results

There are no results in the tool editor. Image Sampling returns a cropped image that can be used by other vision tools. To do this, set Image Sampling as the Image Source property of the subsequent tool. The resultant image can be viewed in the Vision Window, as shown in the figure above.

Position Compensation

This tool is used to execute a transformation on a region of an image and orient it into a specific orientation. This contrasts with the Relative To function of most tools, which transforms the tool itself to an instance of an object without changing image orientation. Position Compensation instead orients an image so that processing only needs to occur in one location.
While Relative To is more convenient for processing during run time, Position Compensation can be used to make an operation more user-friendly while configuring the application. For example, Relative To can move any reader tool into an orientation and read the character string, but Position Compensation can be used during configuration so the character strings are always in a readable orientation.



To create a Position Compensation tool, right-click **Vision Tools** in the Multiview Explorer, select A**dd Image Process** and then **Position Compensation.** The Position Compensation tool will be added to the Vision Tools list.

There are no results in the tool editor. Position Compensation returns a modified image that can be used by other vision tools. To do this, set Position Compensation as the Image Source property of the subsequent tool. The resultant image can be viewed in the Result section of the object editor. Refer to previous figure for an example.

● Position Compensation Configuration Items

Use the table below to understand the Position Compensation configuration items.

Group	Item	Description
Tools Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Region of Interest	Relative To	The tool relative to which this tool executes. The output values of the selected tool are the input values of this one.
	Link Name	Select the property in the relative tool that will provide the input values.
	Current Values	Collection of current input values.
	Offset	Defines the center coordinates of the region.
	Search Area (Width, Height)	Defines the size of the region.
Properties	Bilinear Interpolation	Enables using bilinear interpolation to sample the input image. By default, this is enabled because it ensures subpixel accuracy.

● Position Compensation Configuration

The Position Compensation tool is primarily used to focus an image on only a specific area of the vision field of view. This is useful during configuration to allow the user to develop an application while images are in an ideal orientation. For example, the barcode in the previous figure is rotated and centered so that a Barcode tool will only need to focus on the center region of the image.
To do this, Position Compensation can be set in a known orientation to guarantee that the instance is centered and returned in a target orientation. All pixels outside of the region of interest will be returned as black. This may cause interference with reader tools if there is not sufficient white space around the object to be read.

8-8-10 Custom Tools

Use custom tools to specify a program that will be called when the tool is executed. The following Custom Tools are described in this section.

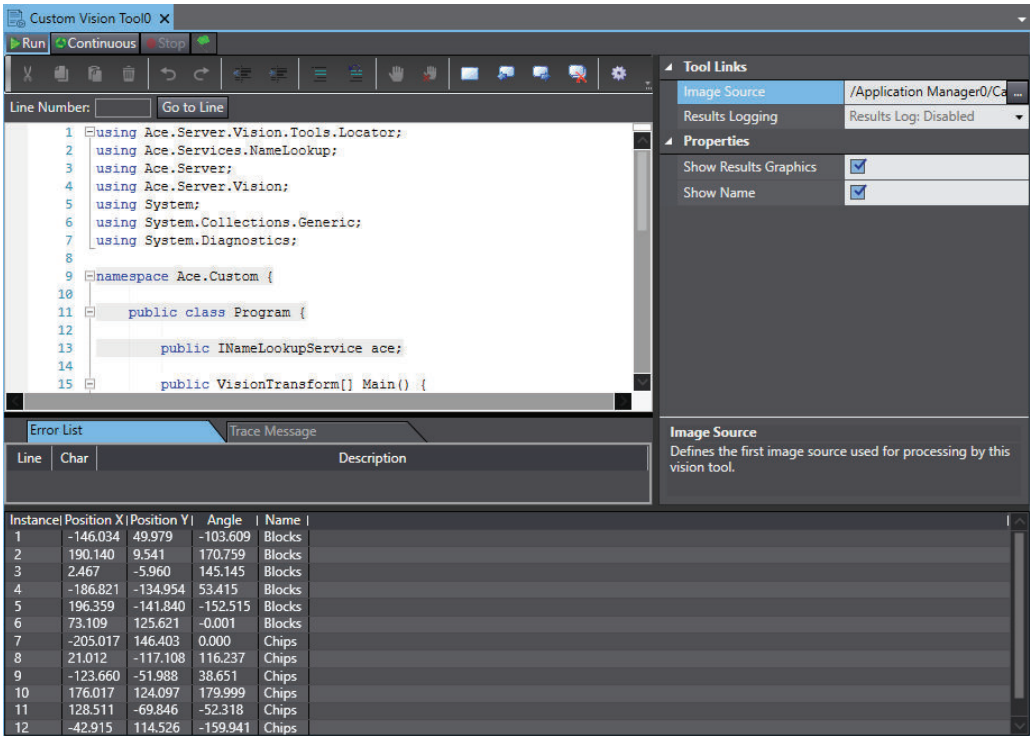
- Refer to *Custom Vision Tool* on page 8-323 below for more information.
- Refer to *FH Tool* on page 8-327 for more information.
- Refer to *Remote Vision Tool* on page 8-327 for more information.

Custom Vision Tool

This tool is a C# program that can be referenced and executed as a vision tool. Other vision tools and objects can be referenced and used within the program. At the end of the program, a collection of defined vision transform results will be returned.

Additional Information

A basic understanding of C# is required to use the Custom Vision Tool.



To create a Custom Vision tool, right-click **Vision Tools** in the Multiview Explorer, click **Add Custom** and then click **Custom Vision Tool**. A Custom Vision tool will be added to the Vision Tools list.
Use the table below to understand the results of the Custom Vision tool.

Item	Description
Instance	Index of the result instance, starting with 1.
Position X	X-coordinate of the instance. This is returned using the x entry of the VisionTransform list.
Position Y	Y-coordinate of the instance. This is returned using the y entry of the VisionTransform list.
Angle	Angle of the instance. This is returned using the roll entry of the VisionTransform list.
Name	Name of the instance. This is returned using a VisionTransformResult.Name method.

● Custom Vision Tool Configuration Items

Use the table below to understand the Custom Vision tool configuration items.

Group	Item	Description
Tool Links	Image Source	Defines the image source used for processing by this vision tool.
	Results Logging	Enables logging of tool results to a file.
Properties	Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Show Name	Specifies if the name associated with each instance is written in the Vision Window.

● Storing part information for later use.

Any information can be stored in the tag property or directly into a controller variable structure to be used later in a process.

● Simulating application to improve emulation accuracy

In many situations, the Random Instances feature does not provide adequate control over vision result generation for effectively emulating the application. A Custom Vision Tool can be used to program the logic of vision result generation to be as simple or complex as necessary.

● Creating Custom Vision Processing Logic

This can include the use of tools configured ahead of time or modifying tool parameters before executing them during run time. A Custom Vision Tool has access to vision tool parameters, frames, and results to allow for complete flexibility in selecting which results are accessed and returned.

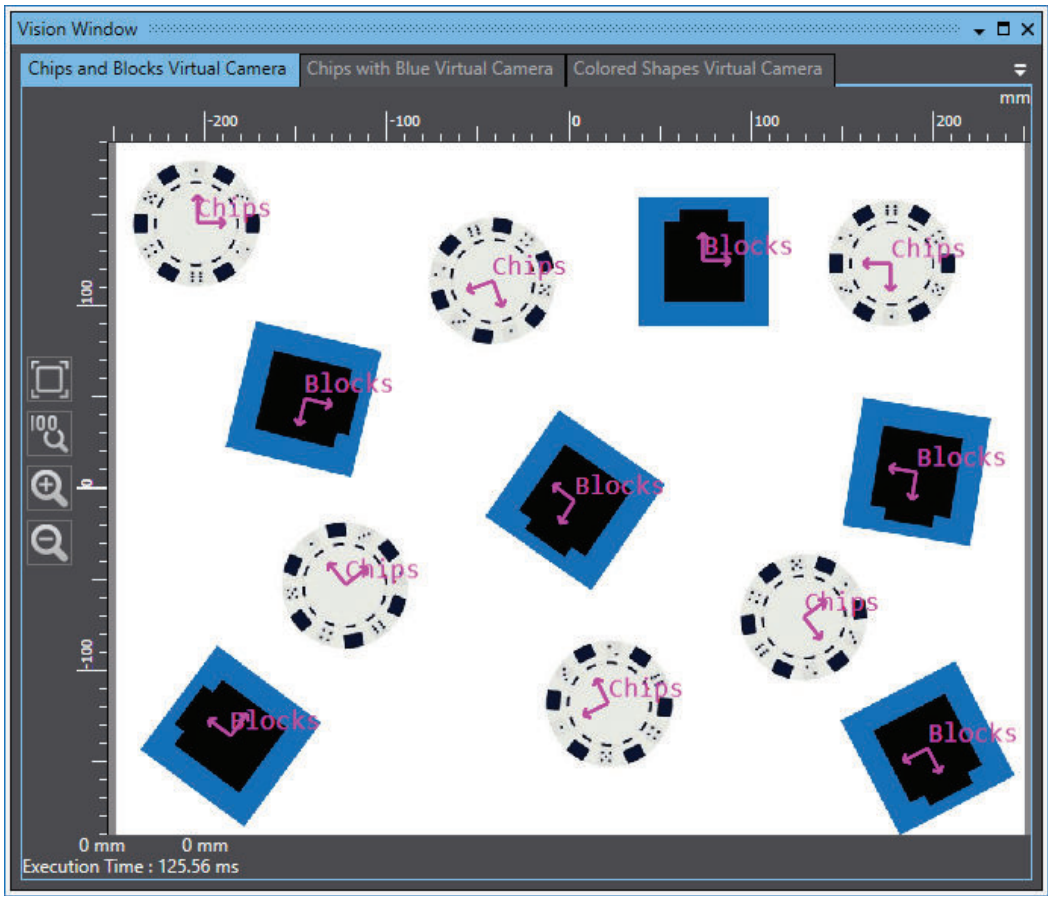
● Custom Vision Tools

Custom Vision Tools can be used to do almost anything with other vision tools in the workspace. For example, the tool can be used to return instances from a Locator. The instances are drawn in the Vision Window as shown in the figure below.
Like other tools, Custom Vision Tool must have a camera or tool selected in the Image Source to operate.

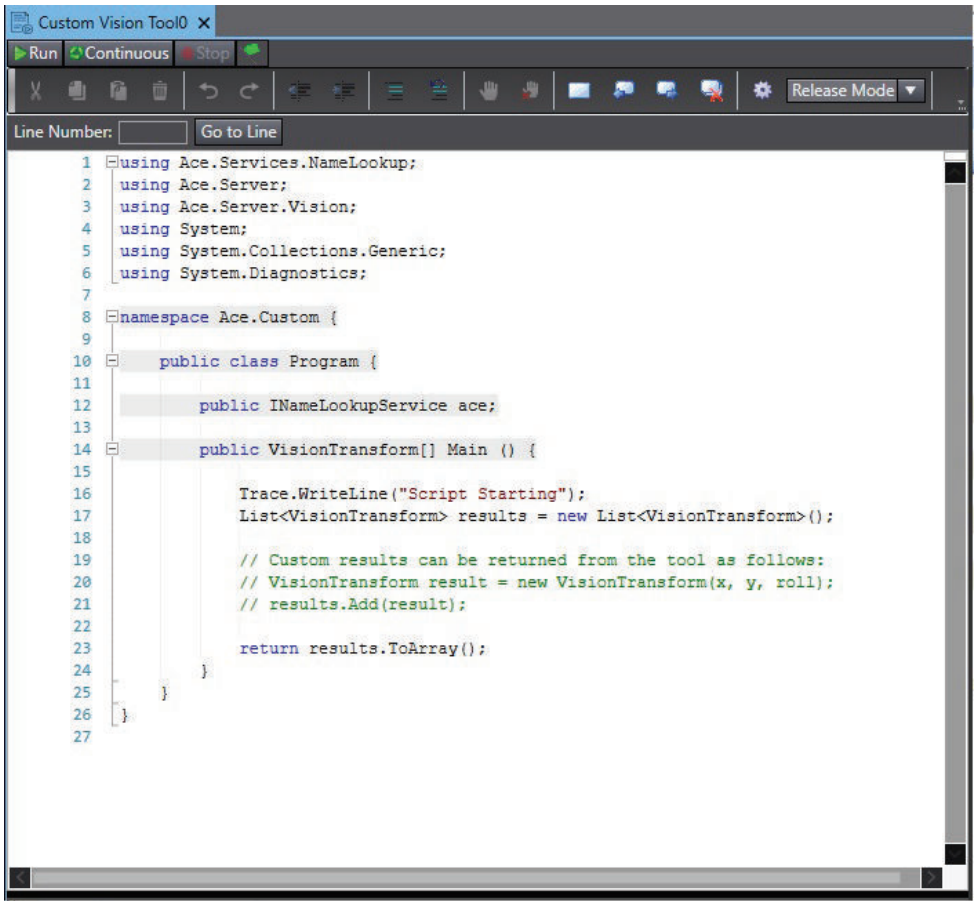


Additional Information

Every Custom Vision Tool should end with the line, return results.ToArray();. This outputs the results to the Results section of the tool editor.



When a Custom Vision Tool is used, the following script is created.



The script is comprised of several major elements:

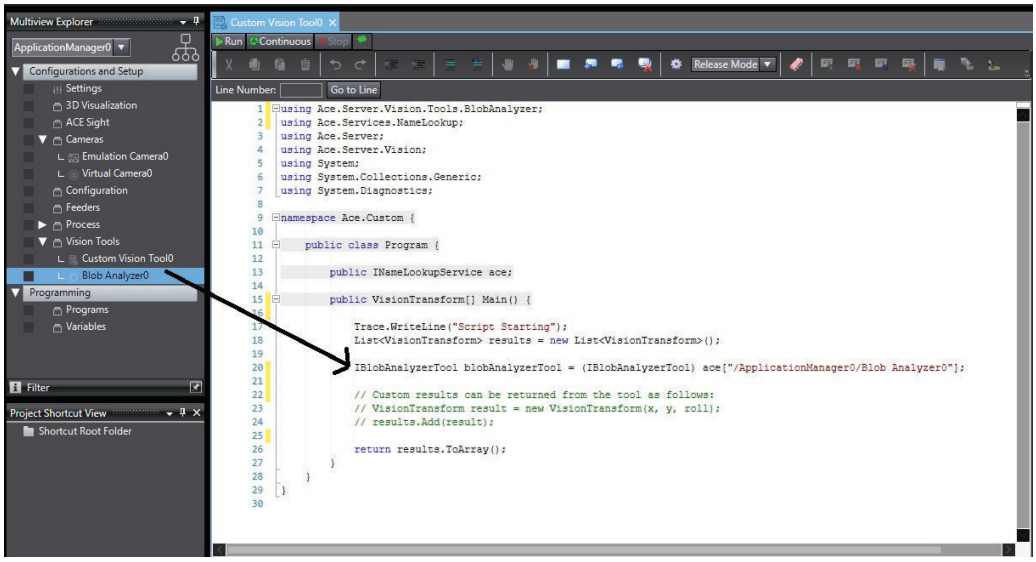
- C# class definition
- INameLookupService field
- Main method entry point
- Return values from the script

The script contains many lines that cannot be modified and are considered protected. These are designated by the light gray background. Notice that the namespace, class name, INameLookupService field name, and Main method lines are protected and cannot be changed.

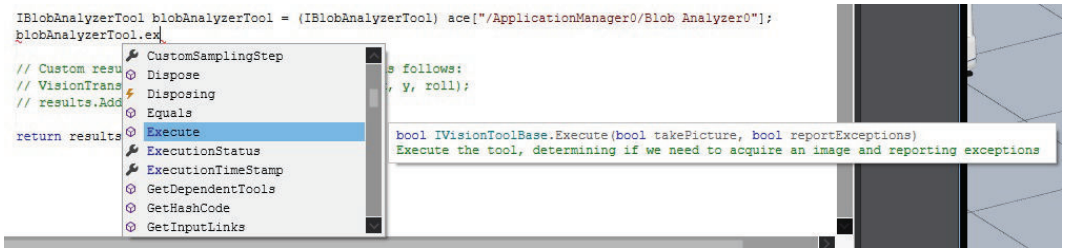
When the custom vision tool is invoked, the text in the script is dynamically compiled in memory and the Main method is called. By default, the script must return an array of VisionTransform objects. These define the coordinates of any objects located by the tool. The default custom vision tool simply returns an empty array of transformations. It is up to the user to define the logic of the script and to return an array of transformations.

● Referencing Other Objects and Tools

It is often useful to reference other vision tools or objects when running a script. Reference other objects by dragging and dropping vision tools from the Multiview Explorer to the body of the script as such:



Any object can be dragged and dropped from the workspace explorer to the body of a script. When an object is dropped, the script automatically creates a reference to the object through the INameLookupService field defined in the scrip. Once an object has been inserted into a script, access the properties and methods of that object as such:



Here is a sample of a custom vision tool that executes a blob tool and returns the results of the blob tool:

```
public VisionTransform[] Main() {

    Trace.WriteLine("Script Starting");
    List<VisionTransform> results = new List<VisionTransform>();

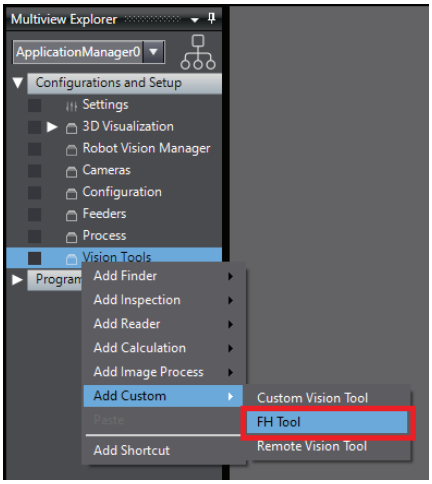
    IBlobAnalyzerTool blobAnalyzerTool = (IBlobAnalyzerTool) ace["/ApplicationManager0/Blob Analyzer0"];
    blobAnalyzerTool.Execute(false);

    // Add all blob results but rotate by 45 degrees
    foreach(var blobResult in blobAnalyzerTool.Results){
        VisionTransform newPosition = blobResult .Position * new VisionTransform(0, 0, 45);
        results.Add(newPosition);
    }

    return results.ToArray();
}
}
```

FH Tool

This tool enables configuration and calibration of an FH-series 2D vision system connected to ACE. The FH-series vision system must have a scene configured before the FH Tool can be used in ACE. After the FH-series vision system is pre-configured, add the FH Tool, set the configuration items, and then calibrate the camera.



Refer to the following manuals for more information.

- *Automation Control Environment (ACE) Version 4 Camera Configuration User's Manual (Cat. No. I679)*
- *FH Series Vision System Hardware Setup Manual (Cat. No. Z366)*

Remote Vision Tool

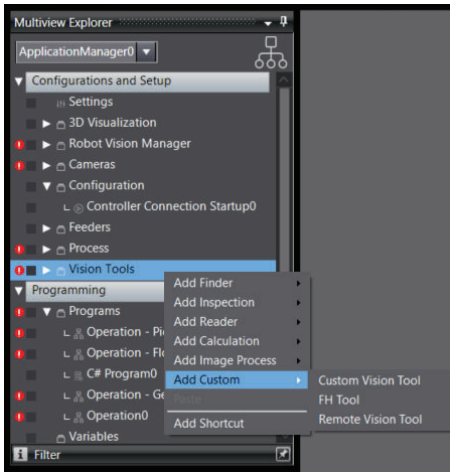
Some applications require Process Managers to operate several different processes for multiple robots and cameras. However, these applications sometimes need to operate on a single PC, where the PC hardware can limit available processing power. For example, if a PC is operating with multiple robot processes and several different vision tools, the operation may produce a lag in response and motion. In this case, it may be useful to offload vision operations to another computer to improve application speed and performance.

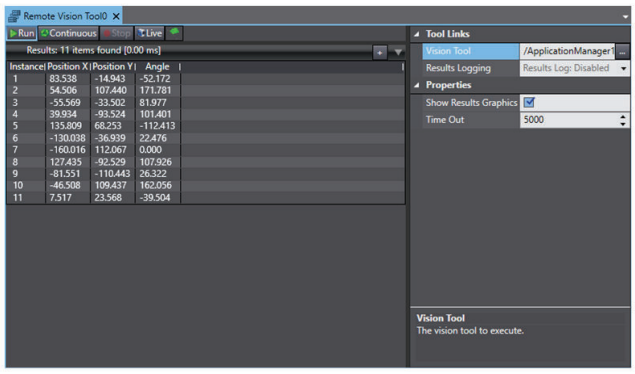
This transfer can be accomplished by creating a Remote Vision Tool (RVT), which references another vision tool in a separate Application Manager and returns the results. To distribute vision operations, create and configure another Application Manager and then move an existing Vision Tool to it using the Move To option. Then, an RVT on the original Application Manager can be configured to reference it and return its VisionTransform results. In this way, the Vision Tool processing requirements are offloaded to a second PC.

To create a Remote Vision Tool, right-click on **Vision Tools** in the Multiview Explorer and select: **Add Custom** and then **Remote Vision Tool**, as shown in the following figure.

Additional Information

This is not necessary for eV+ applications using Robot Vision Manager objects because eV+ can directly request results from multiple Application Managers.





● Remote Vision Tool Configuration

Item		Description
Tools Links	Vision Tool	Selects the vision tool that the Remote Vision Tool will execute and from which it will acquire results.
	Results Logging	Enables logging of tool results to a file.
Properties	Show Results Graphics	Specifies if the graphics are drawn in the Vision Window.
	Time Out	Set the maximum time period (in milliseconds) that the tool is able to run. This includes both execution of tool and acquisition of results.

RVTs will likely utilize a camera that is connected to a separate computer. Therefore, all hardware connections should be verified before configuration. The camera calibrations should also be performed. This can be done either remotely or on the server computer. For more information, refer to Remote Application Manager.

While the function of RVT depends entirely on the Vision Tool to which it is linked, the configuration is largely the same. The user should first create and configure a separate Application Manager to act as a vision server and create the necessary vision tools there. Only then should the correct Vision Tool be selected in the properties. Linking an RVT to a Vision Tool that already has errors will cause the RVT to fail.



Additional Information

Each RVT can link to only one Vision Tool, but RVTs in the same Application Manager can link to Vision Tools in multiple other Application Managers.

● Runtime

During runtime, RVT will continuously update to show the latest values of its results, as would any other Vision Tool. However, RVT does not return the images or graphics from the remote vision server. Therefore, it is recommended to perform any image processing or editing in the Application Manager containing the linked Vision Tool.

● Results

Object	Definition
Instance	Index of the result instance, starting with 1.
Position X	X-coordinate of the origin of the returned instance.
Position Y	Y-coordinate of the origin of the returned instance.
Angle	Angle of the returned instance.
Name	Identifies Name of returned vision instance.

● Tool Compatibility

RVT is not able to connect to every type of Vision Tool. It is primarily designed to work with Finder Tools so it can return the locations of instances. RVT supports the following Vision Tools:

- Blob Analyzer
- Labeling
- Locator
- Shape Search 3
- Custom Vision Tool

In the first four Vision Tools listed, the RVT obtains its results by returning the values for each instance that match the Results columns shown, refer to Remote Vision Tools Editor. Any other Vision Tool can be supported through a Custom Vision Tool in the Application Manager, acting as a vision server. RVT supports VisionTransform object type to give proper values to Position X, Position Y, and Angle.

9

Troubleshooting

This section describes the different messages that are available for troubleshooting purposes

9-1 Event Log 9-2
9-1-1 Accessing the Event Log 9-2
9-1-2 Display Messages by Event Type 9-3
9-1-3 Sorting Messages 9-3
9-1-4 Selecting and Copying Messages 9-3

9-2 Clearing the Event Log 9-4

9-3 Error Log 9-5
9-3-1 Accessing the Error Log 9-5

9-4 Device Node Event Log 9-7
9-4-1 Accessing and Clearing the Device Node Event Log 9-7
9-4-2 Device Node Configuration Conflicts 9-8

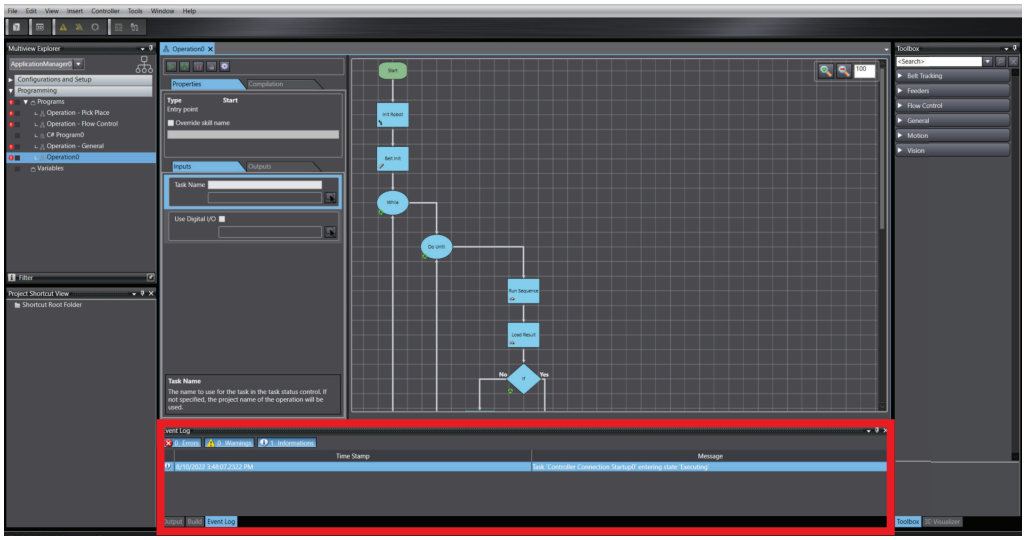
9-5 Diagnostic Summary 9-9
9-5-1 Saving a Core Dump File 9-9

9-1 Event Log

The Event Log tool displays event messages that have been logged since system startup. The Event Log collects the time stamp and details about the particular event message that can be used by service personnel to troubleshoot problems.

Additional Information

When reporting a problem to field service, it is helpful to include the Event Log information in any communication.



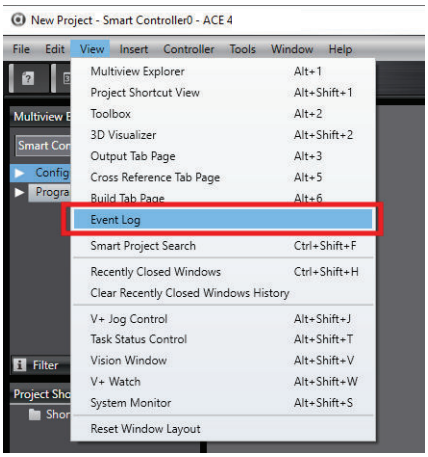
The events are sorted by clicking the **Type, Time Stamp**, or **Message** column headings. Each click will toggle the sort between ascending and descending order. The events are filtered by clicking the event type buttons in the Event Log area.

Additional Information

In addition to the Event Log, most events are also written to the Windows Application Log. Therefore, if the Event Log is no longer available (for example, it was cleared it or ACE software has been shut down), the ACE events can be viewed in the Windows Application Log file.

9-1-1 Accessing the Event Log

To access the Event Log, select **View** from the menu bar and then click **Event Log**. The Event Log will appear in the Multifunction Tab area.



9-1-2 Display Messages by Event Type

Click the message type buttons to display messages by type. Click the **Error** Button (Error) to display all error message types. Click the **Warning** Button (Warning) to display all warning message types. Click the **Information** Button (Information) to display all information message types.

9-1-3 Sorting Messages

Click the **Time Stamp** column heading to sort messages by the event occurrence time. Click the **Message** column heading to sort messages by the message.

9-1-4 Selecting and Copying Messages

Right-click any message to display a pop-up. Choose **Copy** or press Ctrl+C to copy a selected message to the clipboard. To select all messages, choose **Select All** or press Ctrl+A. All messages will be selected.

9-2 Clearing the Event Log

Right-click a message and choose Clear All. All messages will be cleared from the Event Log.



Additional Information

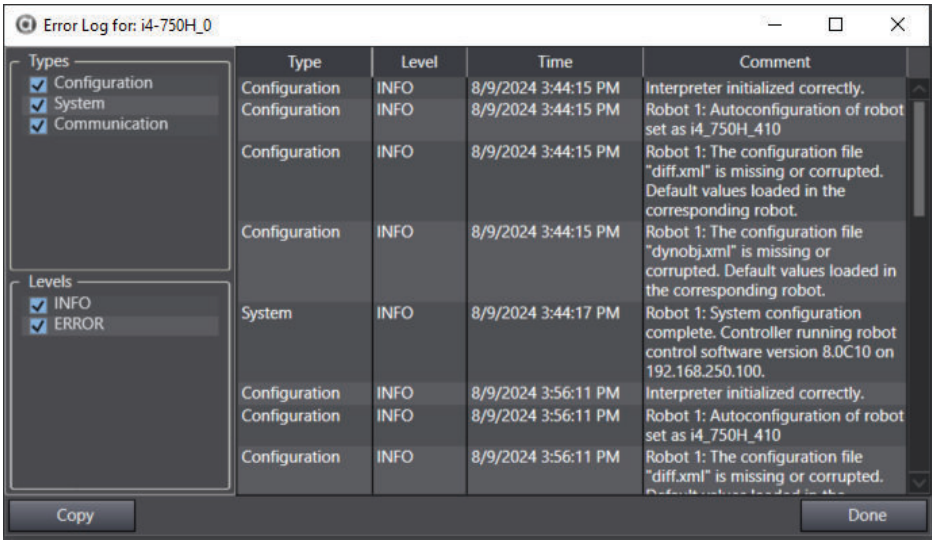
Many items logged to the Event Log are also logged to the Windows Application Event Log. Use the Windows Application Event Log to retrieve past ACE events as an alternative.

9-3 Error Log

The Error Log displays a history of robot controller events since the controller was powered on for the first time. The Error Log stores the V+ event type, level, time stamp, and comment for each entry. Use the Error Log to see a history of robot controller events related to configuration, fieldbus, V+, and System changes. The Error Log can also be used for troubleshooting purposes.
The information contained in the Error Log is stored in the robot controller at D:\ADEPT\CUSTOM \LOG.XML. Refer to *5-9 Task Status Control* on page 5-40 for more information.

Additional Information

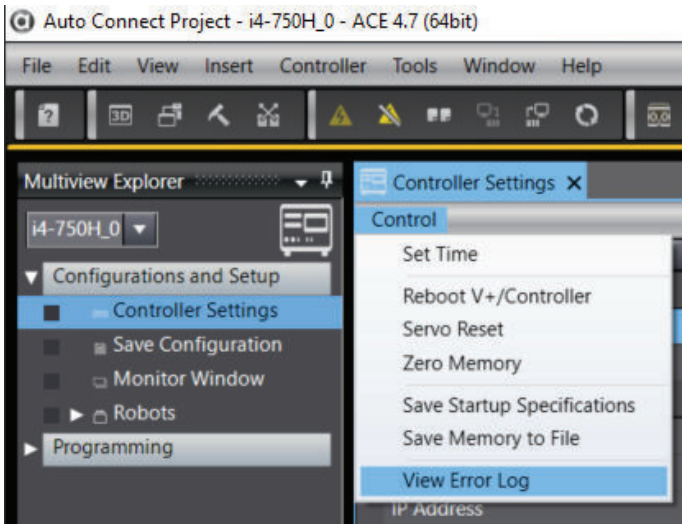
- Some errors (such as servo errors) may require accessing the device node event log for additional information. Refer to *9-4 Device Node Event Log* on page 9-7 for more information.
- The Error Log is available when connected to a robot controller. Accessing the Error Log is not possible in Emulation Mode.



Use the *Types* and *Levels* selections to filter events. The *Copy* button will copy all events in the list to the clipboard.

9-3-1 Accessing the Error Log

To access the Error Log, open the Controller Settings. Then, click the **Control** menu item and select **View Error Log**. The Error Log will be displayed.
Refer to *7-2 Controller Settings* on page 7-5 for more information.


Auto Connect Project - i4-750H_0 - ACE 4.7 (64bit)
File Edit View Insert Controller Tools Window Help
Multiview Explorer
i4-750H_0
Configurations and Setup
Controller Settings
Save Configuration
Monitor Window
Robots
Programming
Controller Settings
Control
Set Time
Reboot V+/Controller
Servo Reset
Zero Memory
Save Startup Specifications
Save Memory to File
View Error Log
IP Address

9-4 Device Node Event Log

Each device node in the system contains an event log. The Device Node Event Log collects an ID number, description, time stamp, occurrence count, and additional information about the last event that can be used by service personnel to troubleshoot problems.

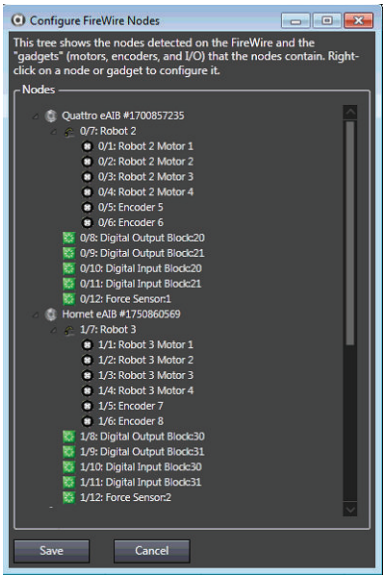
Additional Information

- ACE must be connected to a physical controller to access the Device Node Event Log.
- The Device Node Event Log is not available in Emulation Mode.

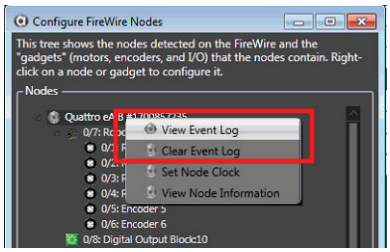
9-4-1 Accessing and Clearing the Device Node Event Log

To access or clear the Device Node Event Log, use the following procedure.

1. While online with a physical controller, access the Controller Settings area. Refer to *7-2 Controller Settings* on page 7-5 for more information.
2. Click the **Configure** Button to display the Configure Options Dialog Box.
3. Select **Configure Device Nodes** and then click the **Finish** Button. The Configure Device Nodes Dialog Box will be displayed.

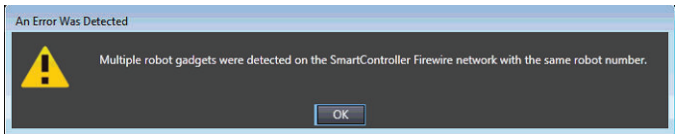


4. Right-click a device node and then select **View Event Log** to access the Device Node Event Log. Select **Clear Event Log** to clear the Device Node Event Log. After the selection is made, the procedure is complete.



9-4-2 Device Node Configuration Conflicts

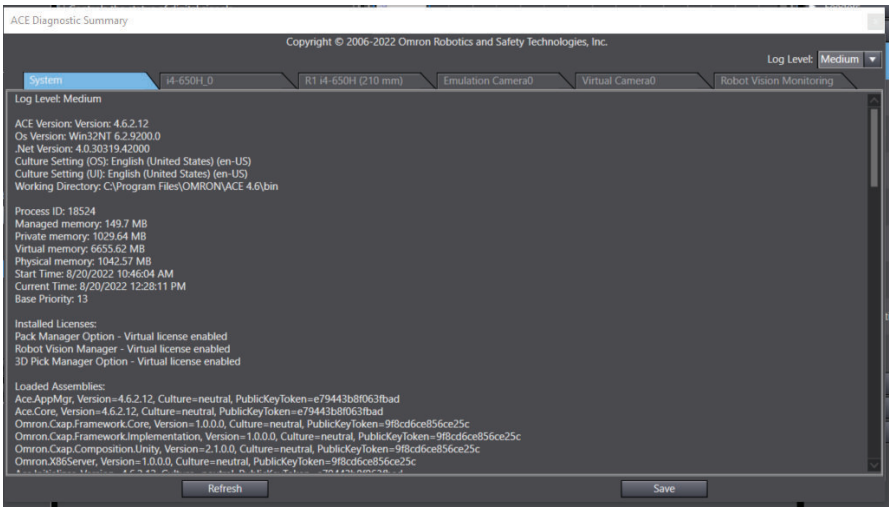
When connecting to a controller, ACE software will check for device node configuration conflicts. If a conflict is present, error messages will be displayed after the connection is established.



To correct device node configuration conflicts, access the Configure Device Nodes Dialog Box and adjust the configuration to resolve the issue described in the error message.

9-5 Diagnostic Summary

Use the Diagnostic Summary to view the project information, including objects, processes and license status. Click **Help** and then click **Diagnostic Summary** to open the tool. Click each tab to view the status of each function within the project. For complete information, set the Log Level to **High**. Use the arrow button or expand the window to view all available tabs. Click each tab to view the details for that project object. Correct any identified errors before proceeding with the project.

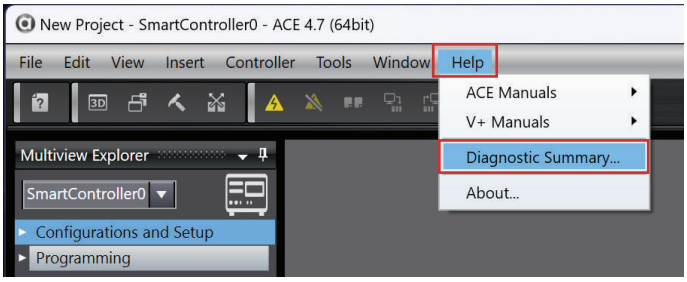


9-5-1 Saving a Core Dump File

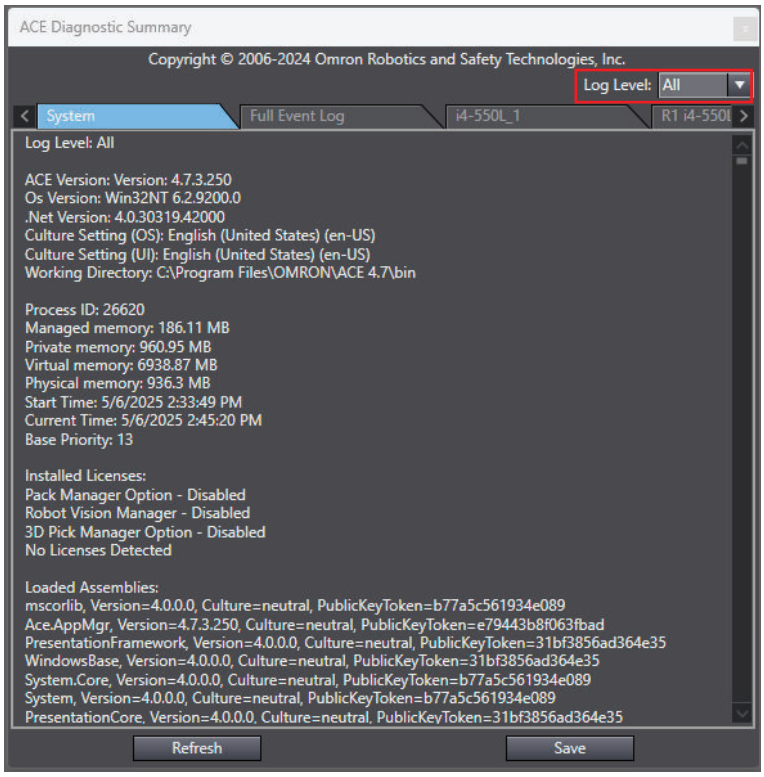
In the event of a critical robot controller firmware failure, a Core Dump file is generated on the controller SD card. You can save this file and share it with an OMRON Support representative when reporting the failure. Follow the procedure below to save a Core Dump file.

1 Power ON the robot and restart the ACE software.

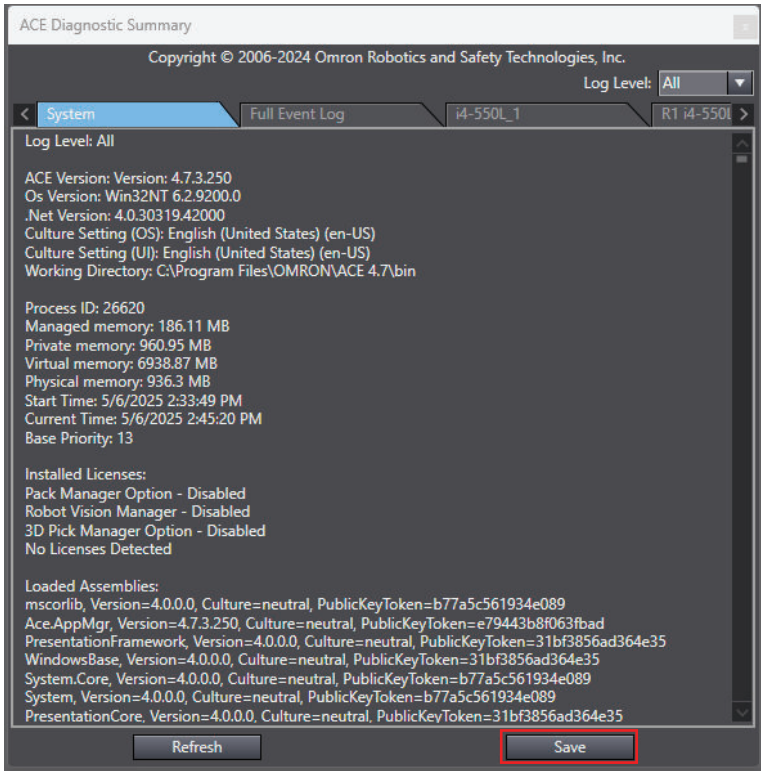
2 In the ACE interface, Click **Help** and then click **Diagnostic Summary** to open the tool.



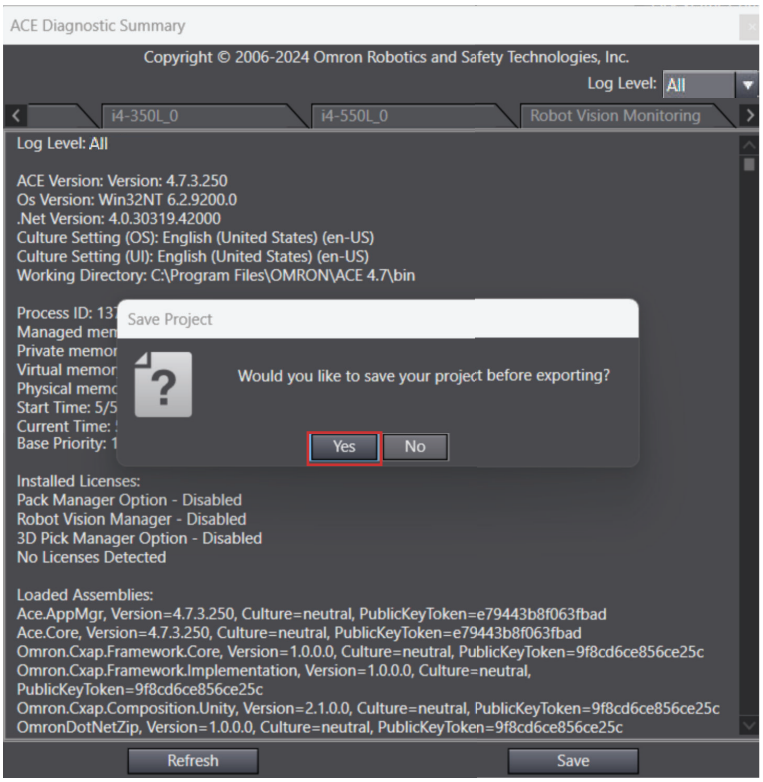
3 Set the Log Level to **High** or **All** for complete information.



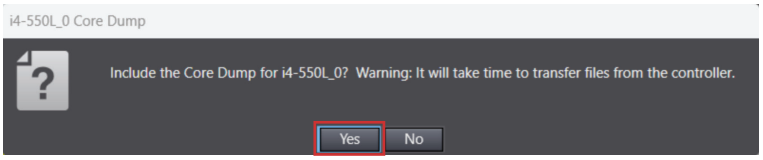
4 Click **Save**.



5 Click **Yes** if you get a prompt about saving the project.



6 Click **Yes** to include the Core Dump file.



7 Select a save location for the Core Dump file to complete the procedure.

A

Appendices

A-1 Latch Signal Test A-2
A-1-1 Latch Signal Test Procedure A-2
A-2 Uninterruptible Power Supply (UPS) Installation A-3
A-2-1 UPS Software Required A-3
A-2-2 AC-AC Hardware Configuration A-3
A-2-3 DC-DC Hardware Configuration A-7
A-3 Commissioning Physical Hardware A-13
A-4 Version Control Introduction A-14
A-4-1 Git Configuration A-14
A-4-2 Multiple User Configuration A-15

A-1 Latch Signal Test

Use the following procedure to test a latch signal. The following settings and configurations must be complete before the procedure is followed.

- Sensor communication and power supply is present.
- Network settings are complete.
- The object must be present and configured in ACE project.
- The position latch signal connections must be present

A-1-1 Latch Signal Test Procedure

Use the following procedure to test a latch signal.

1 Access the Monitor Window. Refer to *7-4 Monitor Window* on page 7-37 for more information.

2 Clear the latch FIFO buffer using the program keyword CLEAR.LATCHES.
For belts, use do@1 clear.latches(-n) where n is the belt object number.
For robots, use do@1 clear.latches(+n) where n is the robot number.

3 Ensure the latch FIFO buffer is empty by entering listr latched(-n) or listr latched(+n), where n is the same as in step 2. This should return 0 if the FIFO buffer is clear. If not repeat step 2 and 3, or confirm the latch signal was wired and configured properly.

4 Trigger the sensor by clicking the **Run** Button in the object. Enter listr latched(-n) or listr latched(+n) respectively to return the most recent latch signal from the FIFO buffer. For the example circuit and configuration described above on Encoder Channel 0, listr latched(-1) would return 1001. If this occurs, the test procedure is successful and complete.

A-2 Uninterruptible Power Supply (UPS) Installation

Use the following procedure to install and configure an uninterruptible power supply to use with ACE.

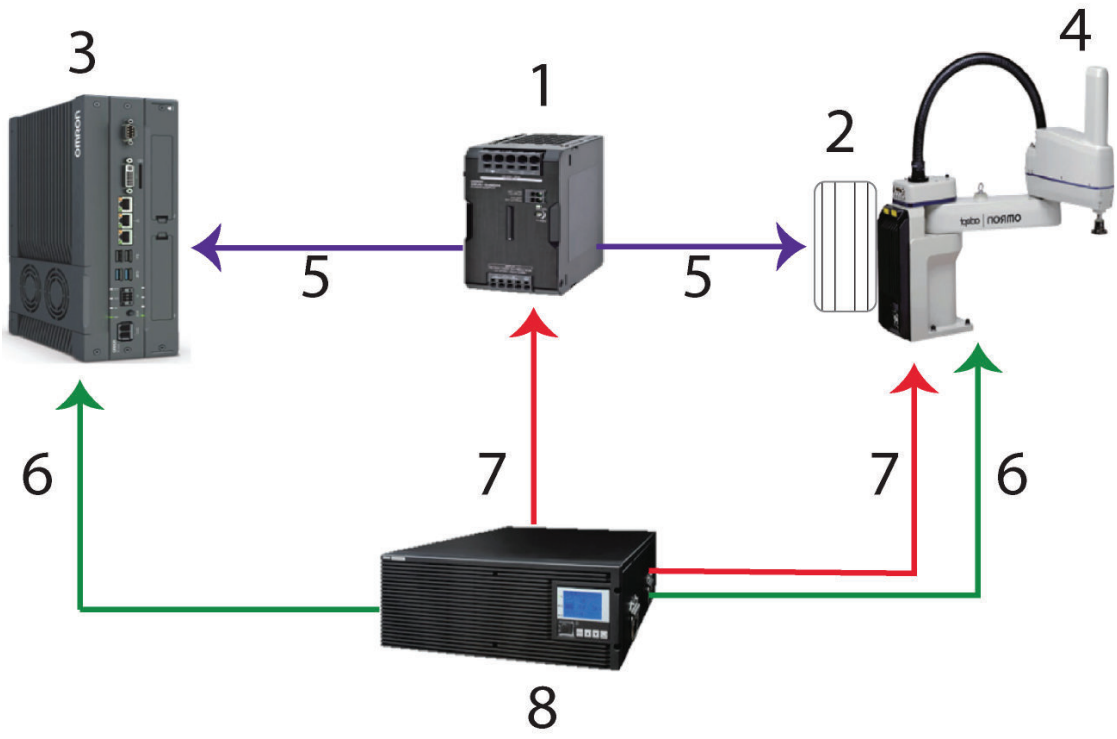
A-2-1 UPS Software Required

- PowerAct Pro (Master Unit)
- PowerAct Pro (Slave Unit)

The PowerAct Pro software is included with the AC UPS. The Slave Unit needs to be installed on the computer running the ACE software.

A-2-2 AC-AC Hardware Configuration

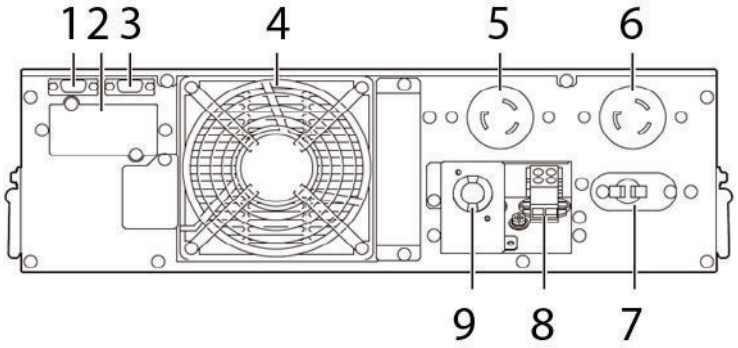
The basic hardware needed to use the UPS within a robot installation is: Omron AC UPS BU5002RWL, Omron Web Card SC20G2, Omron S8VKS DC Power Supply, Omron NY Industrial PC and an Industrial Robot.



Item	Description
1	Switching Power Supply 20A/480W S8VKS48024
2	Amplified Power Control Logic
3	Omron Industrial PC

Item	Description
4	Industrial Robot
5	24VDC
6	Ethernet Cable
7	240VAC
8	AC UPS BU5002RWL with SC20G2 Network Card Installed

The BU5002RWL back panel has the power and network interface connections. The following figure identifies the ports, and the table identifies the back panel features and purpose for each port.



Item	Description
1	Contact signal port
2	Optional slot
3	RS-232C Port
4	Cooling fan
5	200-240VAC output, L6-20P socket
6	200-240VAC output, L6-20P socket
7	AC input overcurrent protection switch
8	200-240VAC output terminal
9	AC input cable

Equipment Connection Procedure

Use the following steps to make the equipment connections for the full system.

1 Install the SC20G2 into the UPS.

2 Connect the Industrial PC to the S8VK power supply outlet.

3 Connect the robot Industrial PC to the SC20G2 with Ethernet cable.

4 Connect the robot to the UPS 240VAC output, port 5.

5 Connect the S8VK power supply to the UPS 240VAC output, port 6.

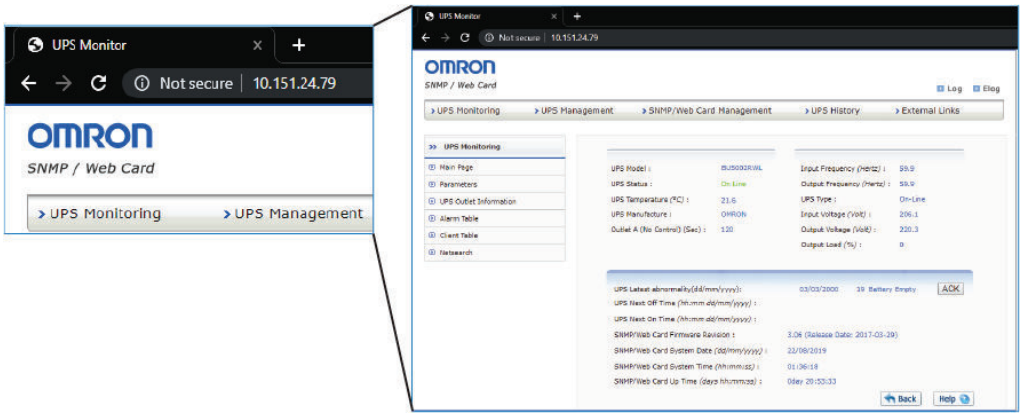
6 Connect the UPS to a 240VAC power supply/source.

AC-AC UPS Software Configuration

When the SC20G2 is installed in the UPS it operates as the **Master Unit** in the PowerAct Pro connection. The connected devices operate as Slave Units to the UPS. The Slave Units are then configured to initialize .exe or .bat files on the Industrial PC. When these files are executes the UPS is able to initiate a safe stop to the program upon low battery.
Use the following steps to set up PowerAct Pro:

1 Install PowerAct Pro onto the Industrial PC

2 Connect to the AC UPS with the IP, 10.151.24.79, in the browser, opening the UPS Monitor.

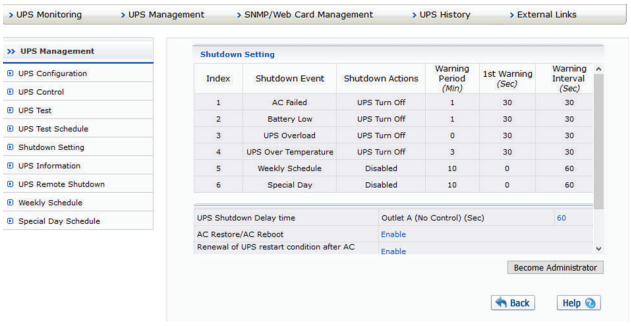


3 Select **UPS Management**, along the top bar.

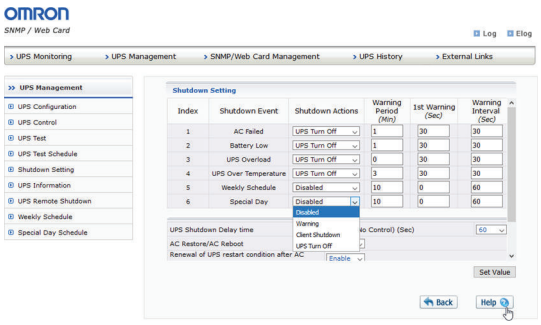
4 Select **Shutdown Setting** from the options in the left side of the window.

5 Select the **Become Administrator** Button.

6 When prompted, the login is OMRON and the password is admin. Then click **OK** to open the Shutdown Settings window.



The Shutdown Event menu enables customized shut-down parameters for the UPS. The UPS reaction is then customized to different events, including Warning delays and UPS Shutdown delays.



UPS Shutdown Sequence (Typical)

When a UPS is connected and configured, a UPS event can trigger the following sequence to execute a controlled shutdown of the system.

1. The UPS communicates to the PC running ACE software that a UPS event occurred.
2. An event listener on the PC running ACE detects the UPS event.
3. A script on the PC running ACE is triggered that makes the SignalUPSEvent.exe program run.
4. The SignalUPSEvent.exe activates the items set in the ACE Software UPS settings such as saving the project file disabling high power on connected controllers.
5. The PC running the ACE software shuts down in a controlled manner after a designated amount of time.

S8BA-Series UPS Configuration

The S8BA-Series UPS must be configured and connected to the PC running the ACE software. Refer to the Automatic Shutdown Software for Omron Uninterruptible Power Supply (UPS) PowerAttendant Lite for Windows Users Manual (K1L-D-18004B) for more information.
The PowerAttendant Lite (PAL) software provides configuration and management of the S8BA-Series UPS and must be installed on the PC running ACE software. Use the PowerAttendant Lite software to configure the operation of the S8BA-series UPS and configure settings such as power failure delay time, shutdown start delay time, script execution time, UPS shutdown delay time, and operating system termination mode.

PowerAttendant Lite Configuration with ACE Notification

Use the following procedure to configure PowerAttendant Lite and set the notification for use with ACE software.

1. Make all wiring and communication connections between the S8BA-Series UPS and the PC running the ACE software. Refer to the *Automatic Shutdown Software for Omron*

Uninterruptible Power Supply (UPS) PowerAttendant Lite for Windows Users Manual (K1L-D-18004B)l for more information.

2 Install PowerAttendant Lite on the PC running the ACE software.

3 Place the provided NotifyAce.usc script in the C:\Users\Public\Public Documents\OMRON\PAL directory. The NotifyACE.usc file is located in C:\Program Files\OMRON\ACE 4.6\UPS.

4 Open the *PowerAttendant Lite* application installed on the PC running the ACE software and click **Shutdown Parameter** in the left-side panel

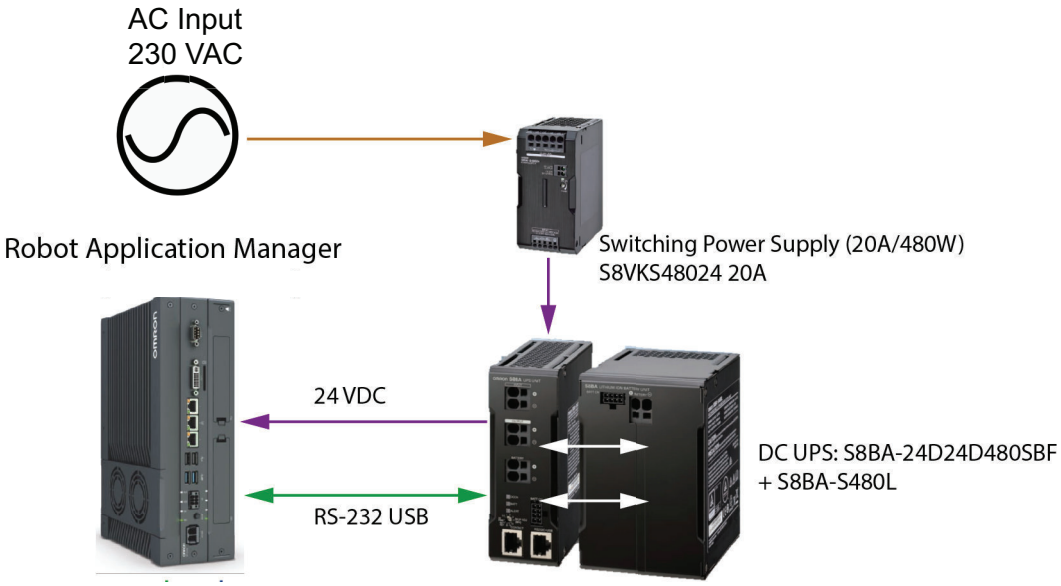
5 Set the shutdown parameters for the specific application. Example parameter values are provided below. These parameters should be adjusted for specific application requirements. After these steps are finished, the procedure is complete. Close PowerAttendant Lite.

1) Power Failure Delay Time: 30 seconds
2) Shutdown Start Delay Time: 0 seconds
3) Script Execution Time: 60 seconds
4) UPS Shutdown Delay Time: 180 seconds
5) OS Termination Mode: Shutdown
6) Click **Submit** in the Shutdown Parameter Setting Window.

6 Confirm the ACE UPS response is set. Refer to *Enable ACE Software UPS Response* on page 5-62 for more information.

A-2-3 DC-DC Hardware Configuration

The basic hardware needed to use the DC-DC UPS is shown in the following figure. The Omron components are the S8VK-S48024 20A, S8BA-24D24D480SBF and S8BA-S480L



Precautions for Correct Use

- The UPS will not maintain high power (230 VAC) to the robot or conveyor belts.
- Some user intervention may be required to restore normal operation after an unexpected power interruption.
- Test and verify the setup before an unexpected power loss event occurs.

DC-DC UPS Software Configuration

Once the units are connected, following the figure in the *A-2-3 DC-DC Hardware Configuration* on page A-7, the Power Attendant Lite (PAL application) must be installed and configured. Once installed, write a script to trigger the operation and configure it.
Use the following steps to set up the Power Attendant Lite.

1 Install Power Attendant Lite to the Robot Application Manager

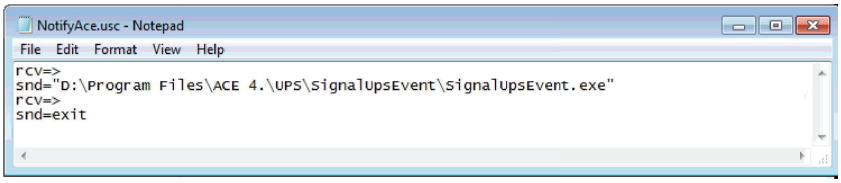
2 Launch Power Attendant Lite and Login: Username = Admin, password = omron

3 Set the parameters as follows:

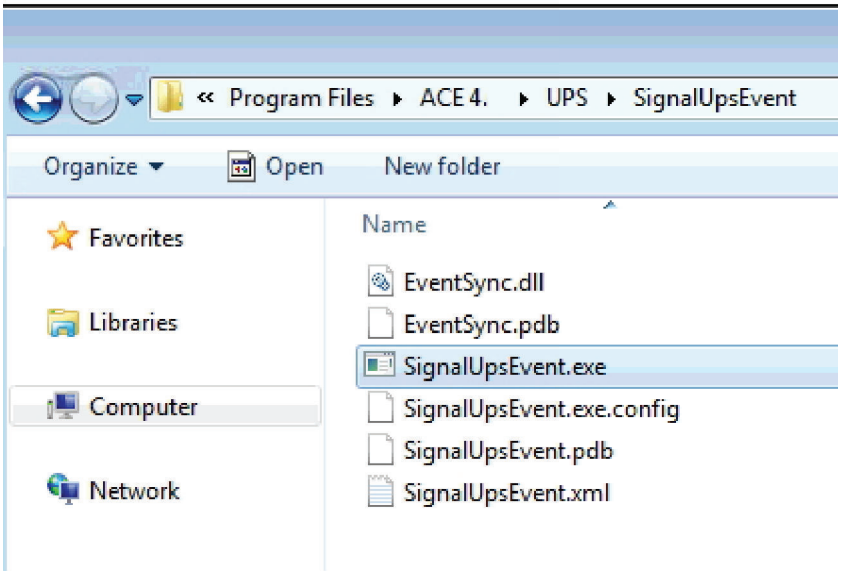
- Power Failure Delay Time： 30 seconds
- Shutdown Start Delay Time： 0 seconds
- Script Execution Time： 60 seconds
- UPS shut down delay time： 180 seconds
- Using script： Choose the script that is registered in *Script setting*
- OS Termination Mode： Shutdown

DC-DC UPS Script File

Open Notepad and create the script file similar to the following example, NotifyAce.usc, and save with a .usc extension in the UPS folder of the ACE 4. installation directory.
In this example, D:\Program Files\ACE4.\UPS

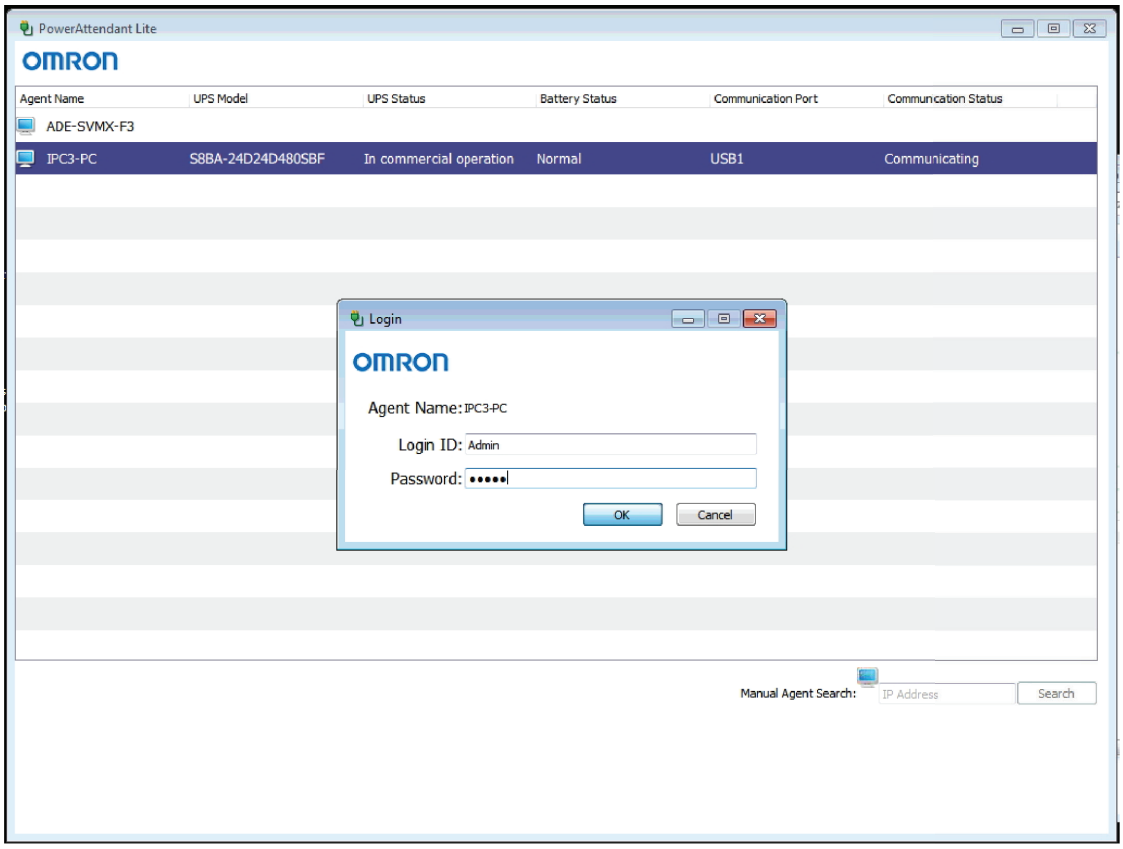


The files in the SignalUpsEvent directory are shown in the following figure.

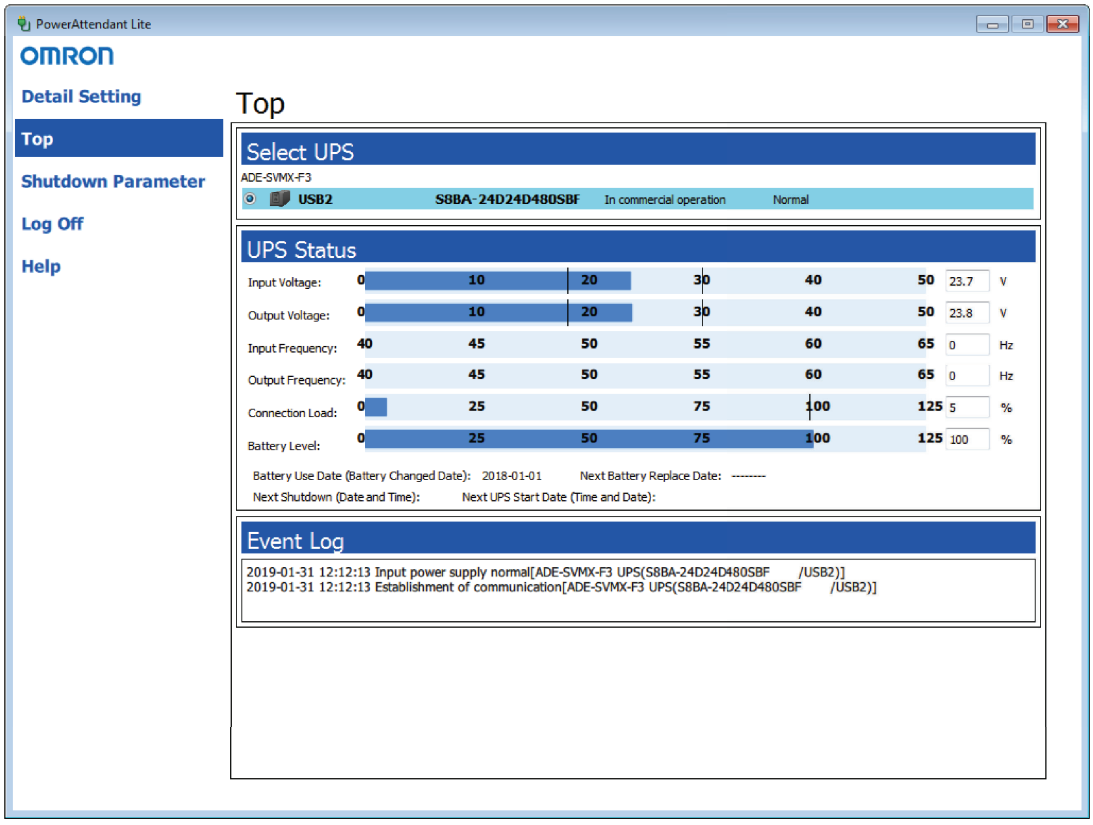


DC-DC Set Up Power Attendant Lite

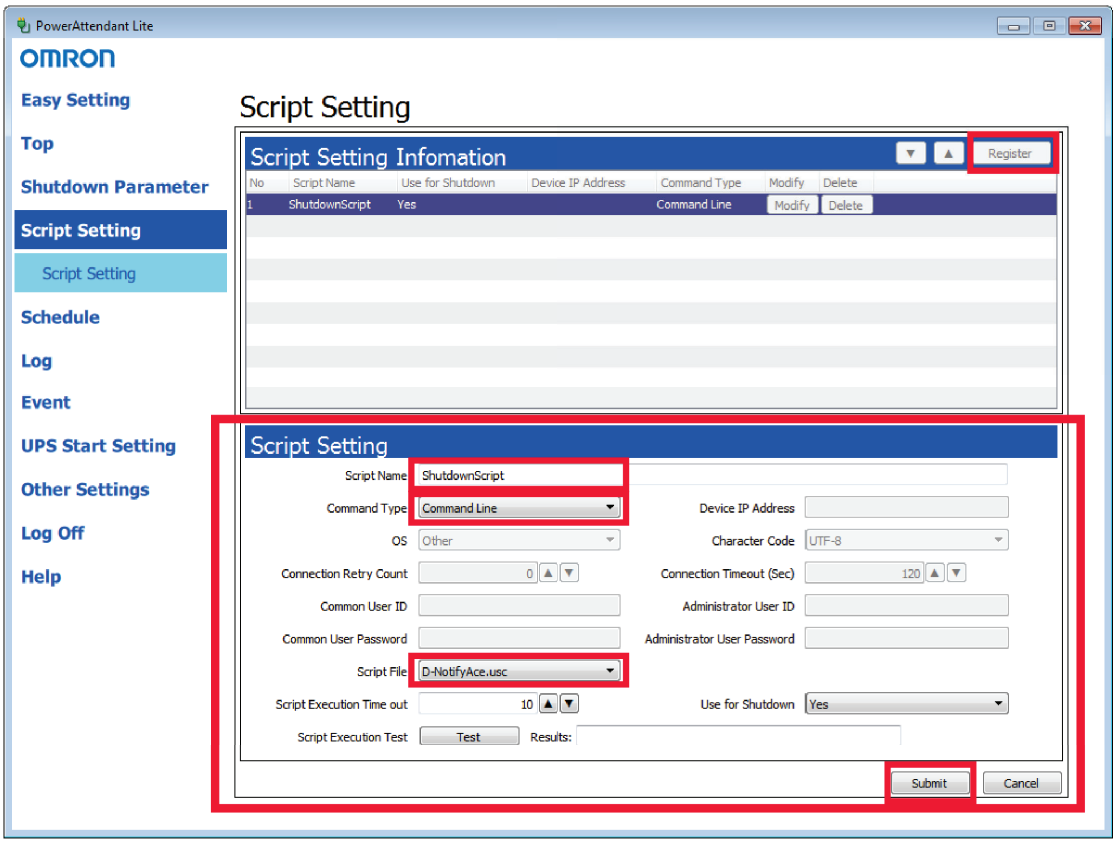
The following steps are used to enable the UPS and set the Power Attendant Lite parameters to trigger if there is a power loss.
Log onto the Power Attendant Lite and then double click the UPS. Log onto the UPS with User Name Admin and password Omron, as shown in the following figure.



Once logged on, the following screen opens.



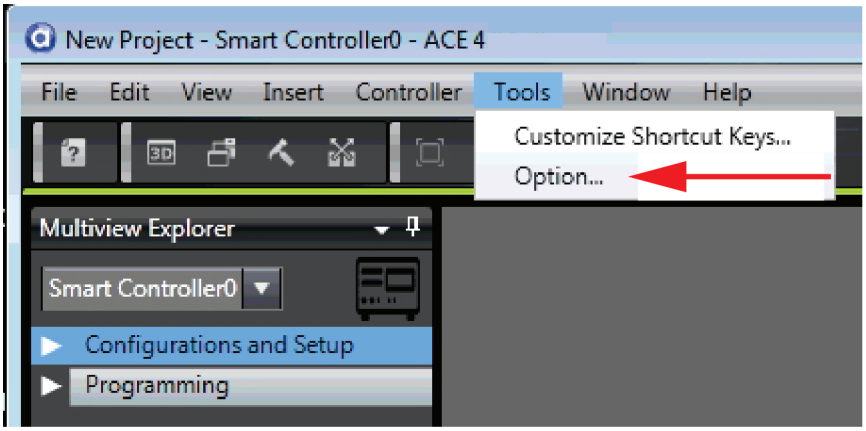
Click **Detail Setting**, on the left menu panel. Then click the **Register** Button in the top right of the panel.



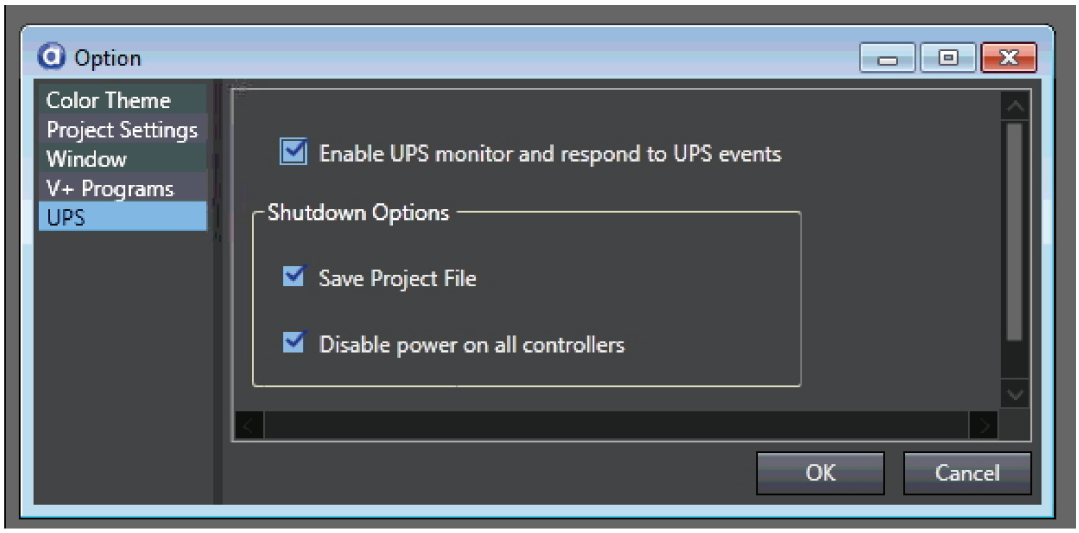
Next, enter the Script Name that was previously saved. Select Command Type: **Command. Line.** Select the script file and then click **Submit.**

DC-DC ACE Enable UPS Shutdown

ACE Multiview Explorer opened, confirm that Smart Controller is selected, then from the Toolbar, click **Tools** and then **Option**.



When the Option panel opens, select the **UPS** option.



Select the **UPS** option in the left-side panel.
Check the **Enable UPS monitor and respond to UPS events** check box.
Select the appropriate Shutdown options:

- Save Project File
- Disable power on all controllers

Click **OK** to close the panel and enable the changes.

A-3 Commissioning Physical Hardware

To transition from Emulation Mode to physical hardware during system commissioning, use the following steps.

1. Ensure all hardware is properly installed according to the reference documentation.
2. Connect to the controller and configure all necessary network and controller parameters. Test communication with all other devices such as cameras, PLCs, etc.



Additional Information

The project created in Emulation Mode cannot transfer its configuration to the physical controller.

3. Test any latch signals and latch configurations.
4. Open the project with Emulation Mode disabled.
5. Set the IP address in the Controller Settings to match the physical controller and verify communication with the controller.
6. Perform any necessary Virtual Camera calibrations.
7. Perform any necessary Robot-to-Belt calibrations.
8. Perform any necessary Robot-to-Sensor calibrations.
9. Train any Locator Models and Locator tools to locate parts.
10. Link any Part objects to Locator tools (rather than using the Custom Vision tool).
11. Teach all processes.
12. Teach any global V+ location variables as needed.
13. Test, debug, and troubleshoot the application.
14. Program system start-up functions as required for the application.



Additional Information

Calibrations are typically performed with calibration pointers and disks, while Teach Process wizards and teaching V+ locations should be performed with the gripper or end of arm tooling used for the process.

A-4 Version Control Introduction

The ACE project version control function (called version control function hereafter) controls the change record of an ACE project by recording the specific content that was changed, the author of the change, and when the change was made to the project.
When the version control function is used to control projects, teams can effectively manage and develop programs by multiple developers. Version control enables various control capabilities by combining the ACE projects with an open source software version control system (Git) that is commonly used in software development.
The version control function effectively solves problems such as the following, which occur in large-scale development of production machines.

- The complexity of source code management due to increase in the scale of program development.
- The increase in the workload of change management due to the increased variation of production machines.
- The increase in the complexity of source code management due to the increased opportunities of development by multiple developers.
- The version control function generates a change record of a project at any time. Use it to return to the desired project by tracing back the change record and comparing the present version with past projects. The version control function provides the capability to check the difference and merge the changes when applied to the master project.

A-4-1 Git Configuration

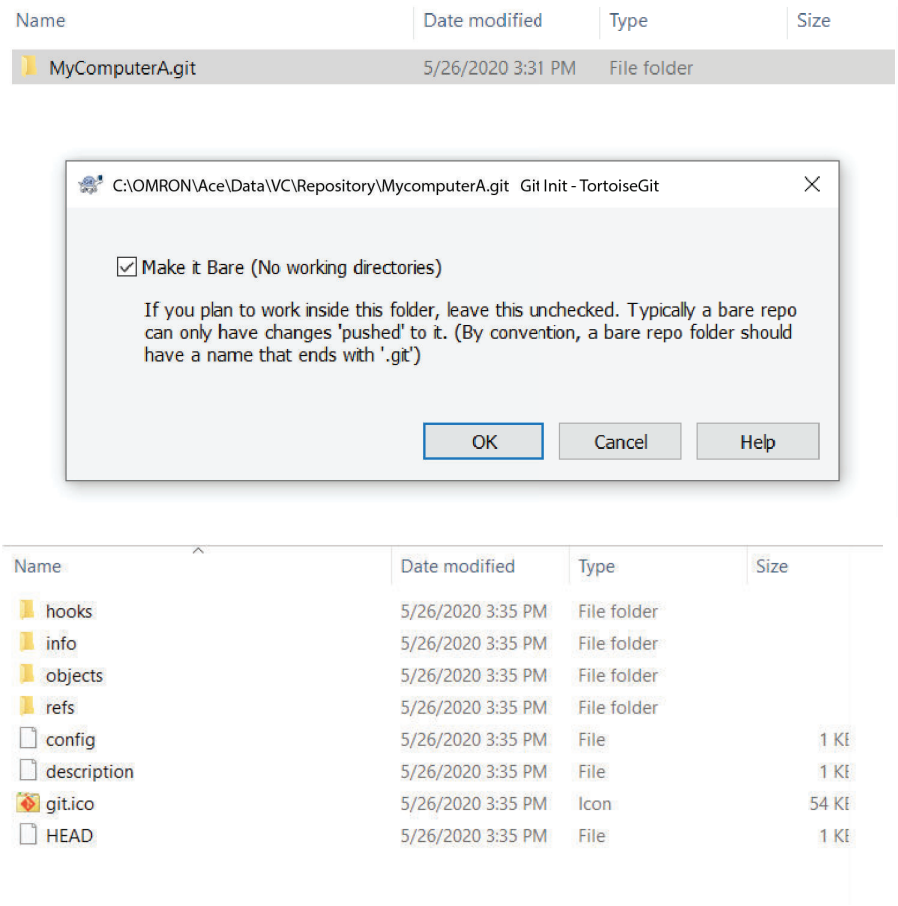
This section describes Git configuration. Refer to *2-1-2 ACE Installation Instructions* on page 2-2 for Git installation information.

Create Shared Folder

Open Windows Explorer and create a new folder. Create this folder in any location with any name. As an example, go to C: drive and create a folder Git, in C:\OMRON\Ace\Data\VC\Git.
Right-click the newly created folder and select **Properties** from the pop-up menu. Then, in the dialog box that is displayed, click the **Share** tab to perform the sharing settings. Configure the folder to allow full access from other users' computers on which the Multiview Explorer version control function is used. The folder can be shared with specific users that are selected or share with Everyone.

Create a Remote Repository

Under the recently created shared folder, C:\OMRON\Ace\Data\VC\Git, create a new folder.
According to the Git conventions, it is required that the new folder name is the remote repository name followed by ".git". For example, if the repository name is "MyComputerA" then name the folder is "MyComputerA.git". Then, right-click the created folder and select **Git Create repository here** from the pop-up menu.
A pop-up will display, check the box and click **OK**. The repository is created.



Additional Information

One ACE project is controlled per repository. Create a directory for each project to use the version control function.

A-4-2 Multiple User Configuration

The ACE Manager version control function works with Git which has distributed version control system. Git also has a mechanism to share the repository with multiple users. The configuration consists of local repositories registered in the computers of each user and the remote repository shared by multiple users. At a certain time for each user, the local repository and remote repository can be synchronized. This is done through either a push or pull synchronization.

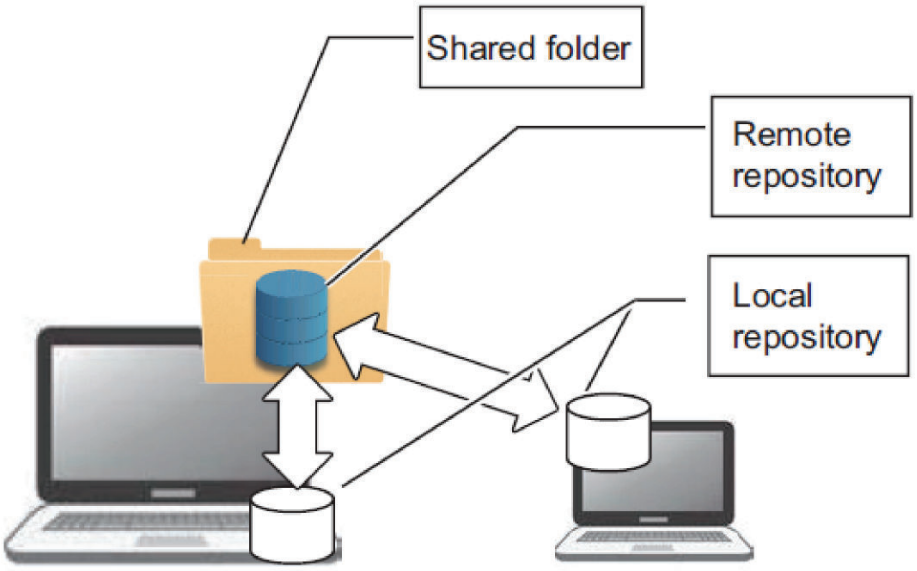
To share changes in the local repository with other users, perform a push operation to the remote repository. To apply changes made by other users to the local repository, perform a pull operation from the remote repository.

There are the following three practical configurations depending on the difference in how the remote repository is shared.

- Using a shared repository on the user's computer to share it as the remote repository
- A dedicated Git server to share it as the remote repository
- Utilizing a Git server service on the Internet to share the remote repository

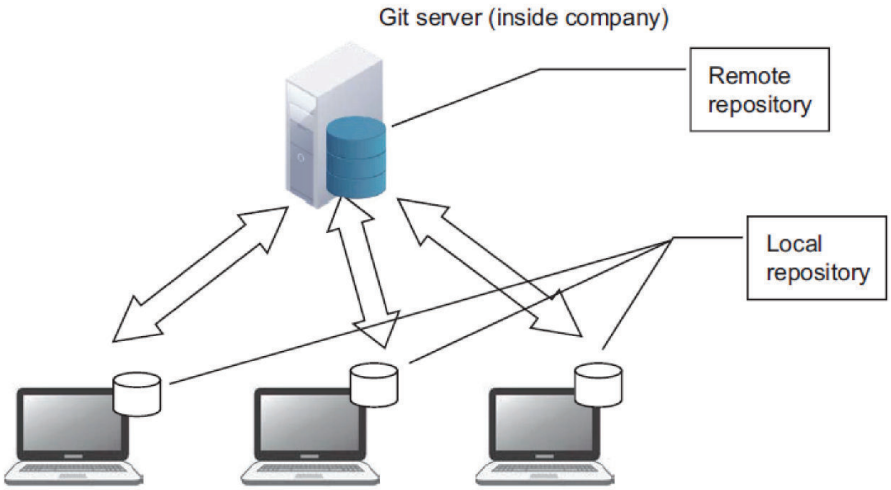
Shared Computer Repository

Creating a shared repository is the easiest way to build a remote repository. In this method, a Windows shared folder is used to publish a remote repository that is synchronized with the local repository, through the local network, so that it can be accessed from other computers.



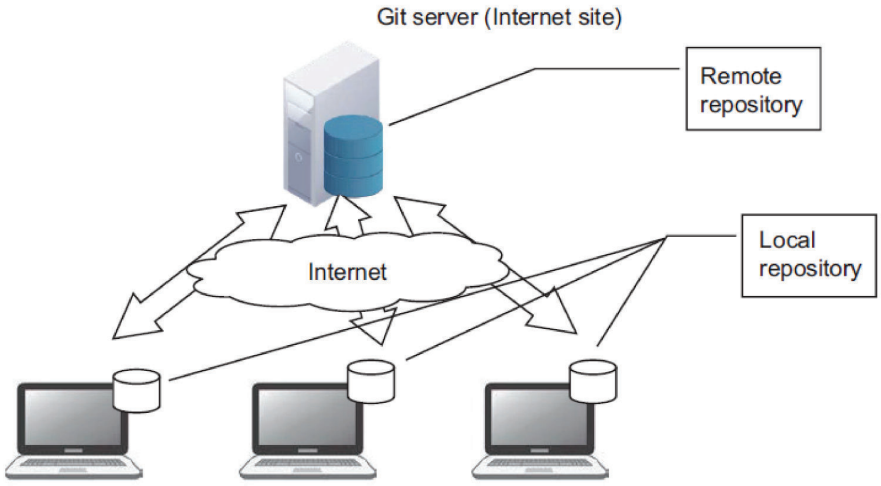
Dedicated Git Server

Build a dedicated Git server to share the remote repository within the company by using Git server software, Gitbucket (for Windows) or if the network uses Linux; GitLab, Gitbucket, Gitblit, Gogs. Although this configuration has costs for building and maintaining the server, it has an advantage in reducing the risk of data leakage because the system is closed within the company.



Internet Git Server Service

Internet based Git server services such as GitHub are also available. Although these are commercial, paid services that incur a cost, there are advantages that they require no server maintenance and allow development in parallel with external developers.



THIS DOCUMENT CONTAINS IMPORTANT SAFETY INSTRUCTIONS. SAVE THIS DOCUMENT.

OMRON Corporation Industrial Automation Company

Kyoto, JAPAN **Contact : www.ia.omron.com**

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.
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

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 ROBOTICS AND SAFETY TECHNOLOGIES, INC.
4225 Hacienda Drive, Pleasanton, CA 94588 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) 21-6023-0333 Fax: (86) 21-5037-2388

Authorized Distributor:

©OMRON Corporation 2019-2026 All Rights Reserved.
In the interest of product improvement, specifications are subject to change without notice.

Cat. No. I633-E-11 0626 (0419)

24000-000N