

Automation Software

## Sysmac Studio

### Robot Integrated System Building Function with IPC Application Controller Operation Manual

SYSMAC-SE2□□□

SYSMAC-SE200D-64



## 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

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- Microsoft, Windows, Excel, and Visual Basic are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- ODVA, CIP, CompoNet, DeviceNet, and EtherNet/IP are trademarks of ODVA.
- The SD and SDHC logos are trademarks of SD-3C, LLC. 
- NVIDIA, the NVIDIA logo, GeForce, and the GeForce logo are the trademarks or registered trademarks of NVIDIA Corporation in the USA and other countries.
- ATI™, Radeon™ is a trademark of Advanced Micro Devices, Inc..
- Celeron, Intel and Intel Core are trademarks of Intel Corporation in the U.S. and / or other countries.
- Git and the Git logo are either registered trademarks or trademarks of Software Freedom Conservancy, Inc., corporate home of the Git Project, in the United States and/or other countries.

Other 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

---

Thank you for purchasing the Sysmac Studio Automation Software.

The Sysmac Studio allows you to use a computer to program and set up Sysmac devices.

This manual describes the operating procedures of the Sysmac Studio mainly for NJ/NX/NY-series Controllers. Use this manual together with the user's manuals for the other devices that you use.

This manual contains information that is necessary to configure Robot Integrated System using Robot Integrated CPU Unit and IPC Application Controller. Please read this manual and make sure you understand the functionality and performance of the Sysmac Studio before you attempt to use it in a control system. Keep this manual in a safe place where it will be available for reference during operation.

## Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

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

Also, this manual is intended for the personnel, who understand the following contents.

- Personnel who understand the programming language specifications in international standard IEC 61131-3 or Japanese standard JIS B 3503, for programming.
- Personnel in charge of working with a robot and well knowing how to handle the robot.

## Applicable Products

This manual covers the following products.

- Sysmac Studio Standard Edition
- NJ-series Robot Integrated CPU Unit
- IPC Application Controller

Part of the specifications and restrictions for the products are given in other manuals. Refer to *Related Manuals* on page 15.

# Manual Structure



## Page Structure

The following page structure is used in this manual.

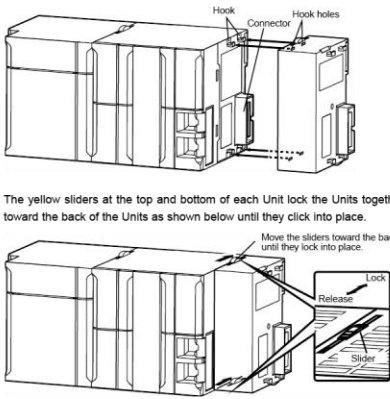
Level 2 heading — 4-3 Mounting Units

Level 3 heading — 4-3-1 Connecting Controller Components

A step in a procedure — 1 Join the Units so that the connectors fit exactly.  
Indicates a procedure.

Special information —  Precautions for Correct Use  
Icons indicate precautions, additional information, or reference information.  


Manual name — NJ-series CPU Unit Hardware User's Manual (W500) 4-9



The Units that make up an NJ-series Controller can be connected simply by pressing the Units together and locking the sliders by moving them toward the back of the Units. The End Cover is connected in the same way to the Unit on the far right side of the Controller.

1 Join the Units so that the connectors fit exactly.

2 The yellow sliders at the top and bottom of each Unit lock the Units together. Move the sliders toward the back of the Units as shown below until they click into place.

Precautions for Correct Use

The sliders on the tops and bottoms of the Power Supply Unit, CPU Unit, I/O Units, Special I/O Units, and CPU Bus Units must be completely locked (until they click into place) after connecting the adjacent Unit connectors.

This illustration is provided only as a sample. It may not literally appear in this manual.

## Icon

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 for Controllers and Units with different unit versions and for different versions of Support Software is given.

## **Precaution on Terminology**

In this manual, download refers to transferring data from the Sysmac Studio to the physical Controller and upload refers to transferring data from the physical Controller to the Sysmac Studio. For the Sysmac Studio, synchronization is used to both upload and download data. Here, synchronize means to automatically compare the data for the Sysmac Studio on the computer with the data in the physical Controller and transfer the data in the direction that is specified by the user.



# Sections in this Manual

---

<b>1</b>	<b>Features and Specifications</b>	<b>10</b>	<b>Other Functions</b>	<b>1</b>	<b>10</b>
<b>2</b>	<b>Software Setup</b>	<b>A</b>	<b>Appendices</b>	<b>2</b>	<b>A</b>
<b>3</b>	<b>Building the Robot Integrated System</b>			<b>3</b>	
<b>4</b>	<b>System Design</b>			<b>4</b>	
<b>5</b>	<b>IPC Application Controller Setup</b>			<b>5</b>	
<b>6</b>	<b>Online connection with the IPC Application Controller</b>			<b>6</b>	
<b>7</b>	<b>Debugging of the Robot Integrated System</b>			<b>7</b>	
<b>8</b>	<b>Application Sample Wizard</b>			<b>8</b>	
<b>9</b>	<b>Maintenance</b>			<b>9</b>	

# CONTENS

<b>Introduction</b> .....	<b>1</b>
Intended Audience .....	1
Applicable Products .....	1
<b>Manual Structure</b> .....	<b>2</b>
Page Structure.....	2
Icon.....	2
Precaution on Terminology .....	3
<b>Sections in this Manual</b> .....	<b>5</b>
<b>Terms and Conditions Agreement</b> .....	<b>10</b>
<b>Safety Precautions</b> .....	<b>11</b>
Definition of Precautionary Information .....	11
Symbols.....	11
<b>Regulations and Standards</b> .....	<b>13</b>
Software Licenses and Copyrights .....	13
<b>Versions</b> .....	<b>14</b>
<b>Related Manuals</b> .....	<b>15</b>
<b>Revision History</b> .....	<b>17</b>

<b>1 Features and Specifications</b> .....	<b>1-1</b>
<b>1-1 Introduction</b> .....	<b>1-2</b>
<b>1-2 Features</b> .....	<b>1-3</b>
<b>1-3 Robot Integrated System</b> .....	<b>1-4</b>
<b>1-4 Specifications</b> .....	<b>1-6</b>
1-4-1 Product Model Numbers.....	1-6
1-4-2 Supported Languages .....	1-6
1-4-3 Applicable Models .....	1-6
1-4-4 Applicable Computers .....	1-6
<b>2 Software Setup</b> .....	<b>2-1</b>
<b>2-1 Installing the Sysmac Studio</b> .....	<b>2-2</b>
<b>3 Building the Robot Integrated System</b> .....	<b>3-1</b>
<b>3-1 Types of Robot Integrated Systems</b> .....	<b>3-2</b>
<b>3-2 Coordinate Systems of Robot Integrated System</b> .....	<b>3-3</b>
3-2-1 Workspace Coordinate System.....	3-3
3-2-2 Robot – World Coordinate System .....	3-4
3-2-3 Robot - Joint Coordinate System .....	3-4
3-2-4 Robot - Tool Coordinate System .....	3-5
3-2-5 Belt Coordinate System.....	3-5
3-2-6 Camera Coordinate System .....	3-6
<b>3-3 Calibration of Robot Integrated System</b> .....	<b>3-8</b>
3-3-1 Type of Calibration .....	3-8
3-3-2 Robot-to-Belt Calibration .....	3-8
3-3-3 Robot-to-Camera Calibration .....	3-9
3-3-4 Robot-to-Latch Calibration .....	3-10
<b>3-4 Basic Robot Motion</b> .....	<b>3-12</b>
<b>3-5 Belts (Conveyors)</b> .....	<b>3-16</b>
<b>3-6 Flow of System Basic Building Controlled with IPC Application Controller</b> .....	<b>3-17</b>
<b>4 System Design</b> .....	<b>4-1</b>
<b>4-1 Starting and Exiting the Sysmac Studio</b> .....	<b>4-2</b>
4-1-1 Starting the Sysmac Studio .....	4-2
4-1-2 Exiting the Sysmac Studio.....	4-3
<b>4-2 Creating a Project</b> .....	<b>4-4</b>



4-2-1	Creating a New Project .....	4-4
4-2-2	Application Manager Device Configuration .....	4-6
4-2-3	Robot Integrated CPU Unit Configuration .....	4-6
<b>4-3</b>	<b>Saving the Project .....</b>	<b>4-8</b>
<b>4-4</b>	<b>Robot Settings .....</b>	<b>4-8</b>
4-4-1	Setting up IO EndEffectors .....	4-8
<b>4-5</b>	<b>Parts of the Window .....</b>	<b>4-11</b>
4-5-1	Application Window .....	4-11
4-5-2	Project Shortcut View .....	4-12
4-5-3	Edit Pane .....	4-12
4-5-4	Toolbox .....	4-12
4-5-5	Search and Replace Pane .....	4-12
4-5-6	Task Status Control Pane .....	4-12
4-5-7	3D Visualizer .....	4-12
4-5-8	V+ Jog Control Pane .....	4-12
4-5-9	Vision Window .....	4-12
4-5-10	V+ Watch Tab Page .....	4-12
4-5-11	System Monitor Pane .....	4-12
<b>4-6</b>	<b>Offline and Online Modes of the Sysmac Studio and How to Use them .....</b>	<b>4-13</b>
<b>5</b>	<b>IPC Application Controller Setup .....</b>	<b>5-1</b>
<b>5-1</b>	<b>Application Manager Overview .....</b>	<b>5-2</b>
5-1-1	Pack Manager .....	5-2
5-1-2	Robot Vision Manager .....	5-2
<b>5-2</b>	<b>Application Manager Settings .....</b>	<b>5-3</b>
<b>5-3</b>	<b>IPC Application Controller Connection Settings .....</b>	<b>5-4</b>
<b>5-4</b>	<b>3D Visualization .....</b>	<b>5-5</b>
5-4-1	Boxes and Cylinders .....	5-5
5-4-2	CAD Data .....	5-6
<b>5-5</b>	<b>Robot Vision Manager .....</b>	<b>5-9</b>
5-5-1	Robot Vision Manager Settings .....	5-9
5-5-2	Belt Latch Calibration .....	5-9
5-5-3	Belt Calibration .....	5-13
5-5-4	Camera Calibration .....	5-18
5-5-5	Gripper Offset Table .....	5-23
5-5-6	Vision Sequence .....	5-26
5-5-7	Overlap Tool .....	5-28
5-5-8	Communication Tool .....	5-29
<b>5-6</b>	<b>Camera .....</b>	<b>5-34</b>
5-6-1	Virtual Camera .....	5-34
5-6-2	Emulation Camera .....	5-38
5-6-3	Basler Camera .....	5-40
5-6-4	Sentech Camera .....	5-45
5-6-5	Custom Device .....	5-46
<b>5-7</b>	<b>Configuration .....</b>	<b>5-48</b>
5-7-1	Controller Connection Startup .....	5-48
5-7-2	Data Mapper .....	5-50
5-7-3	Note .....	5-52
5-7-4	OPC Container .....	5-53
5-7-5	Program System Startup .....	5-55
5-7-6	Recipe Manager .....	5-56
5-7-7	Recipe Manager Script .....	5-63
<b>5-8</b>	<b>Feeders .....</b>	<b>5-65</b>
5-8-1	AnyFeeder .....	5-65

5-8-2	IO Feeder .....	5-72
<b>5-9</b>	<b>Process .....</b>	<b>5-74</b>
5-9-1	Part Buffer .....	5-74
5-9-2	Part Target .....	5-77
5-9-3	Part .....	5-81
5-9-4	Belt .....	5-82
5-9-5	Process Manager .....	5-93
5-9-6	Allocation Script .....	5-130
5-9-7	Pallet .....	5-131
5-9-8	Vision Refinement Station .....	5-134
<b>5-10</b>	<b>Vision Tools .....</b>	<b>5-137</b>
5-10-1	Adding Vision Tools .....	5-138
5-10-2	Vision Tool Editor .....	5-139
5-10-3	Region of Interest .....	5-140
5-10-4	Color Spaces .....	5-142
<b>5-11</b>	<b>C# Program .....</b>	<b>5-144</b>
5-11-1	C# Program Editor .....	5-144
5-11-2	C# Variable Objects .....	5-147

## **6 Online connection with the IPC Application Controller..... 6-1**

<b>6-1</b>	<b>Data Arrangement of the Robot Integrated System Using the IPC Application Controller, and Device Relationship in the Sysmac Studio.....</b>	<b>6-2</b>
<b>6-2</b>	<b>Online connection .....</b>	<b>6-4</b>
<b>6-3</b>	<b>Project Synchronisation .....</b>	<b>6-5</b>

## **7 Debugging of the Robot Integrated System ..... 7-1**

<b>7-1</b>	<b>Introduction for Debugging of the Robot Integrated System Including the IPC Application Controller</b>	<b>7-2</b>
7-1-1	Mechanism and Procedure of Offline Debugging .....	7-2
<b>7-2</b>	<b>3D Visualizer .....</b>	<b>7-4</b>
7-2-1	3D Visualizer Basic Features .....	7-4
7-2-2	Creating a 3D Workspace .....	7-4
7-2-3	Description of Icons .....	7-4
7-2-4	Collision Detection Settings .....	7-16
<b>7-3</b>	<b>Event logs .....</b>	<b>7-17</b>
<b>7-4</b>	<b>V+ Jog Control .....</b>	<b>7-19</b>
7-4-1	Starting V+ Jog Control .....	7-19
7-4-2	V+ Jog Control Setting Items .....	7-20
<b>7-5</b>	<b>Task Status Control .....</b>	<b>7-25</b>
7-5-1	Starting Task Status Control .....	7-25
7-5-2	Displayed Items in Task Status Control .....	7-26
7-5-3	Robot Integrated CPU Unit Settings Items .....	7-27
7-5-4	Task Manager .....	7-27
7-5-5	IO Watcher .....	7-29
7-5-6	V+ File Browser .....	7-30
7-5-7	Virtual Front Panel .....	7-31
7-5-8	Application Manager Control .....	7-33
<b>7-6</b>	<b>Vision Window .....</b>	<b>7-34</b>
7-6-1	Vision Window Display .....	7-34
7-6-2	Vision Windows Functions .....	7-34
<b>7-7</b>	<b>Monitoring Variables .....</b>	<b>7-37</b>
7-7-1	Adding C# Variable Objects to the V+ Watch Window .....	7-37

## **8 Application Sample Wizard ..... 8-1**

<b>8-1</b>	<b>Overview of Application Sample Wizards.....</b>	<b>8-2</b>
<b>8-2</b>	<b>Application Sample Wizards .....</b>	<b>8-3</b>
8-2-1	Application Samples.....	8-3
8-2-2	Application Sample Wizard Flowchart.....	8-3
8-2-3	Startup Application Sample Wizard.....	8-5
<b>8-3</b>	<b>Pack Manager .....</b>	<b>8-21</b>
8-3-1	Pack Manager Settings .....	8-21
8-3-2	Pack Manager Offline Debugging .....	8-25
<b>8-4</b>	<b>Robot Vision Manager .....</b>	<b>8-26</b>
8-4-1	Robot Vision Manager Configuration.....	8-26
8-4-2	Offline Debugging of the Robot Vision Manager .....	8-27
<b>9</b>	<b>Maintenance .....</b>	<b>9-1</b>
<b>9-1</b>	<b>Troubleshooting .....</b>	<b>9-2</b>
9-1-1	Introduction for functions of troubleshooting and maintenance .....	9-2
<b>10</b>	<b>Other Functions .....</b>	<b>10-1</b>
<b>10-1</b>	<b>Printing.....</b>	<b>10-2</b>
10-1-1	Items You Can Print .....	10-2
<b>10-2</b>	<b>System Monitor .....</b>	<b>10-3</b>
<b>10-3</b>	<b>Hardware Diagnostics.....</b>	<b>10-6</b>
<b>10-4</b>	<b>Data Collection .....</b>	<b>10-8</b>
<b>10-5</b>	<b>Motor Tuning .....</b>	<b>10-10</b>
<b>10-6</b>	<b>Searching and Replacing .....</b>	<b>10-11</b>
10-6-1	Search and Replace Targets and Setting Items.....	10-11
<b>Appendices .....</b>		<b>1</b>
<b>A-1</b>	<b>Precautions on Use of Version Control .....</b>	<b>2</b>
A-3-1	Precautions Common to All Devices.....	2
A-3-2	Robot Integrated CPU Unit .....	2
A-3-3	Application Manager .....	2

# Terms and Conditions Agreement

---

## WARRANTY

---

- The warranty period for the Software is one year from the date of purchase, unless otherwise specifically agreed.
- If the User discovers defect of the Software (substantial non-conformity with the manual), and return it to OMRON within the above warranty period, OMRON will replace the Software without charge by offering media or download from OMRON's website. And if the User discovers defect of media which is attributable to OMRON and return it to OMRON within the above warranty period, OMRON will replace defective media without charge. If OMRON is unable to replace defective media or correct the Software, the liability of OMRON and the User's remedy shall be limited to the refund of the license fee paid to OMRON for the Software.

## LIMITATION OF LIABILITY

---

- THE ABOVE WARRANTY SHALL CONSTITUTE THE USER'S SOLE AND EXCLUSIVE REMEDIES AGAINST OMRON AND THERE ARE NO OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTY OF MERCHANTABILITY OR FITNESS FOR PARTICULAR PURPOSE. IN NO EVENT, OMRON WILL BE LIABLE FOR ANY LOST PROFITS OR OTHER INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF USE OF THE SOFTWARE.
- OMRON SHALL HAVE NO LIABILITY FOR DEFECT OF THE SOFTWARE BASED ON MODIFICATION OR ALTERNATION TO THE SOFTWARE BY THE USER OR ANY THIRD PARTY. OMRON SHALL NOT BE RESPONSIBLE AND/OR LIABLE FOR ANY LOSS, DAMAGE, OR EXPENSES DIRECTLY OR INDIRECTLY RESULTING FROM THE INFECTION OF OMRON PRODUCTS, ANY SOFTWARE INSTALLED THEREON OR ANY COMPUTER EQUIPMENT, COMPUTER PROGRAMS, NETWORKS, DATABASES OR OTHER PROPRIETARY MATERIAL CONNECTED THERETO BY DISTRIBUTED DENIAL OF SERVICE ATTACK, COMPUTER VIRUSES, OTHER TECHNOLOGICALLY HARMFUL MATERIAL AND/OR UNAUTHORIZED ACCESS.
- OMRON SHALL HAVE NO LIABILITY FOR SOFTWARE DEVELOPED BY THE USER OR ANY THIRD PARTY BASED ON THE SOFTWARE OR ANY CONSEQUENCE THEREOF.

## APPLICABLE CONDITIONS

---

USER SHALL NOT USE THE SOFTWARE FOR THE PURPOSE THAT IS NOT PROVIDED IN THE ATTACHED USER MANUAL.

## CHANGE IN SPECIFICATION

---

The software specifications and accessories may be changed at any time based on improvements and other reasons.

## ERRORS AND OMISSIONS

---

The information in this manual has been carefully 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 Sysmac Studio and an NJ-series Robot Integrated CPU unit.

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.



Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.



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.

## WARNING

Check operations of the created user programs, data, and setting values carefully before proceeding to normal operation.



When building a robot system that includes this CPU Unit or an Omron robot, be sure to ensure compliance with the laws and regulations on the safety of industrial robots in the country or region where the robot is operating in design and operation of the system. Refer to Robot SAFETY GUIDE (I590) for details.



Ensure the enough safety before making any changes that may affect the operation of the robot.



Make sure that there are no hazards caused by robot's movements before operating the robot using the V+ Jog Control function.



Take a particular attention to the robot speed setting when you operate the robot using the V+ Jog Control function. Get ready to bring the robot to an emergency stop at an emergency. Make sure that there are no hazards caused by robot's movements before operating the robot.



Confirm that you are operating the right robot before conducting a jog operation using V+ Jog Control function.



If an execution program and task number are specified and the automatic execution of V+ program is enabled, it may possibly happen that the robot operates after the CPU Unit and robot are turned on. Make sure that the movement of the robot does not cause a danger.



## CAUTION

The simulator, which uses the 3D Visualizer, simulates the operations of a PLC and a robot. There are differences in movement and timing between actual PLC and robot. In addition to debugging the program in the simulator, be sure to check the operation on the physical machine before operating it.



Unexpected operation of the equipment may occur an accident.

# Regulations and Standards

## Software Licenses and Copyrights

This product incorporates certain third party software. The license and copyright information associated with this software is available at [http://www.fa.omron.co.jp/nj\\_info\\_e/](http://www.fa.omron.co.jp/nj_info_e/).

# Versions

---

Hardware revisions and unit versions are used to manage the hardware and software in NJ/NX-series Units, NY-series Industrial PCs, and EtherCAT slaves.

Refer to *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details on versions.



# Related Manuals

The followings are the manuals related to this manual. Use these manuals for reference.

Manual name	Cat. No.	Model numbers	Application	Description
Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual	W595	SYSMAC-SE2□□□ SYSMAC-SE200D-64	Learning about the operating procedures and functions of the Sysmac Studio to configure Robot Integrated System using Robot Integrated CPU Unit.	Describes the operating procedures of the Sysmac Studio for Robot Integrated CPU Unit.
Sysmac Studio Robot Integrated System Building Function with IPC Application Controller Operation Manual	W621	SYSMAC-SE2□□□ SYSMAC-SE200D-64	Learning about the operating procedures and functions of the Sysmac Studio to configure Robot Integrated System using IPC Application Controller.	Describes the operating procedures of the Sysmac Studio for IPC Application Controller.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC-SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.
Sysmac Studio 3D Simulation Function Operation Manual	W618	SYSMAC-SE2□□□ SYSMAC-SA4□□□-64	Learning about an outline of the 3D simulation function of the Sysmac Studio and how to use the function.	Describes an outline, execution procedures, and operating procedures for the 3D simulation function of the Sysmac Studio.
Sysmac Studio Project Version Control Function Operation Manual	W589	SYSMAC-SE2□□□ SYSMAC-TA4□□□	Learning the overview of the Sysmac Studio project version control function and how to use it.	Outlines the Sysmac Studio project version control function, and describes how to install, basic operation, and how to operate its major functions.
NJ-series Robot Integrated CPU Unit User's Manual	O037	NJ501-R□□□	Using the NJ-series Robot Integrated CPU Unit.	Describes the settings and operation of the CPU Unit and programming concepts for OMRON robot control.
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ/NX-series CPU Unit. CPU Unit operation CPU Unit features Initial settings Programming based on IEC 61131-3 language specifications
NJ/NX-series Troubleshooting Manual	W503	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	Learning about the errors that may be detected in an NJ/NX-series Controller.	Concepts on managing errors that may be detected in an NJ/NX-series Controller and information on individual errors are described.

Manual name	Cat. No.	Model numbers	Application	Description
eV+3 User's Manual	I651	NJ501-R□□□	Operating the OM- RON robot with the V + program.	Describes the V+ language to control the OMRON robots.
eV+3 Keyword Reference Manual	I652	NJ501-R□□□	Operating the OM- RON robot with the V + program.	Describes V+ keywords that are used in the V+ language.
Robot Safety Guide	I590	RL4-□□□□□□□ RL6-□□□□□□□	Learning how to use the OMRON robot safely.	Describes how to use the OMRON robot safely.
Teaching Pendant T20 User's Manual	I601	10046-010	Operating the OM- RON robot with a teaching pendant.	Describes the setup, operation, and user maintenance for the Teaching Pendant T20.

# Revision History

---

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

**Cat. No. W621-E1-01**

↑  
Revision code

Revision code	Date	Revised content
01	August 2020	Original production





# 1

# 1 Features and Specifications

---

This section provides an overview and lists the specifications and features of the Sysmac Studio Robot Integrated System Building Function.

# 1-1 Introduction

---

The Sysmac Studio Robot Integrated System Building Function is a software function that helps you build an automated system with robots.

It includes the following functions and supports the building of a system with robots in a broad range of phases, from planning to design, maintenance, and product type change.

- Wizards that easily help you build automated systems with robots
- Automated system simulations that include robots, conveyors, and peripheral devices such as cameras.
- Various programming
- Recipes for product type changes

The Sysmac Studio Robot Integrated System Building Function supports systems based on the NJ-series Robot Integrated CPU Units, which consist of OMRON robots, conveyors, the IPC Application Controller with the Application Manager, Feeders, OMRON cameras, sensors, and so on.

# 1-2 Features

---

The Sysmac Studio Robot Integrated System Building Function provides the following features.

## Easy System Building with Wizard

---

The Sysmac Studio Robot Integrated System Building Function provides the application sample wizard that helps you build systems with robots easily.

You can just select and set up components such as robots, Feeders, cameras, and conveyors according to the wizard instructions to easily build a pick-and-place system or the like.

This enables smooth system building because you can visually determine the positions of robots and the image capture positions of cameras.

## Offline Check of the Entire System with Simulation

---

The Sysmac Studio Robot Integrated System Building Function has the entire system simulation functions. And you can use the Pack Manager Application sample wizard on the computer to check the operation of the automated pick-and-place system that you have built. You can also check the operations of systems that consist of robots, End-Effectors, cameras, Feeders, and conveyors in the 3D Visualizer, and estimate the takt time by checking the interference between the robots and peripheral devices and simulating the cooperative operation with parts.

These allow you to check the feasibility of the entire system requirements without having a complete set of physical components.

In addition, you can simulate the system operation for product type changes of parts. This enables early decision on investment in introducing robots into a multi-product, variable-volume production system that is difficult to automate.

## Setup and Programming of System Configuration Devices with Single Support Software

---

The Sysmac Studio can be used for setting up and programming not only the controller to control peripheral devices, but also peripheral devices as conveyors, robots, Feeders, sensors, and HMIs. You can use only the Sysmac Studio to set up system configuration devices, instead of using separate Support Software for each device.

## Easy Change of the Product Type

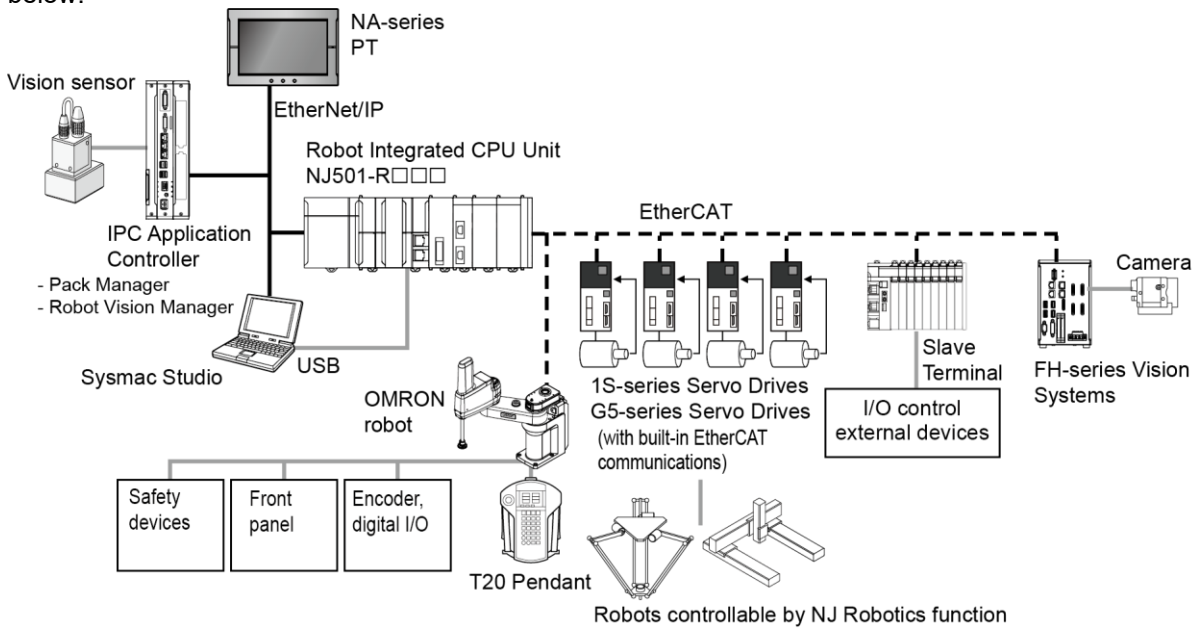
---

When you modify a system to change the product type, you can use the wizard to add or change the peripheral devices easily and then perform simulation to check the operation of the system.

In addition, you can manage the setup data for changing the product type as recipe data, which facilitates changing the product type in the physical system.

# 1-3 Robot Integrated System

The Sysmac Studio Robot Integrated System Building Function covers robot control systems as shown below.



Component		Description
Robot Integrated CPU Unit		The Robot Integrated CPU Unit provides the functionality of previous OMRON PLCs and also the functionality that is required for robot control. Control of I/O devices on high-speed EtherCAT can be applied to robots, safety devices, vision systems, motion equipment, discrete I/O, and more.
Sysmac Studio		The Sysmac Studio provides a wizard for building the basic configuration of a robot integrated system and generating process data and recipe data that Pack Manager uses to control the system. The generated process data and recipe data are then transferred to the IPC Application Controller for execution by Pack Manager. It is also used for setting up cameras and Vision Systems. The setup data is then transferred to the IPC Application Controller for execution by Robot Vision Manager.
IPC Application Controller	Pack Manager	The Pack Manager application controls automated pick-and-place systems with cameras, conveyors, and robots based on the data that you set up in the Sysmac Studio. It is executed on the IPC Application Controller to control the process data and recipe data.
	Robot Vision Manager	The Robot Vision Manager application processes images captured by cameras based on the data that you set up in the Sysmac Studio. It is executed on the IPC Application Controller.
OMRON robot		Consists of the robot amplifier and the robot arm connected to the robot amplifier. It connects with a Robot Integrated CPU Unit through EtherCAT communications. It has digital I/O ports to enable control for the external devices.
Vision sensor		Cameras that capture workpiece images. You can use cameras made by OMRON SENTECH or Basler. The captured images are processed by the Robot Vision Manager installed on the IPC Application Controller based on the settings in the System Studio. You can also use FH-series Vision Sensors.
NA-series PT		Displays various information and performs operation as required. It is used when you instruct a recipe change to the Robot Integrated CPU Unit.
T20 Pendant		The pendant is used to directly operate robot arms. It is also used to teach robots.



1S-series Servo Drives G5-series Servo Drives	Servo Drives with built-in EtherCAT communications.
Robots controllable by NJ Robotics function	Robots that can be controlled from the Robot Integrated CPU Unit that controls Servomotors/Servo Drives with built-in EtherCAT communications.
Slave Terminal	Consists of the NX-ECC20□ Communications Coupler Unit and NX Units that are connected to EtherCAT communications. It exchanges I/O data with a Robot Integrated CPU Unit through EtherCAT communications. Various Units such as digital I/O, analog I/O are covered, therefore, you can use the NX Units depending on the system demand.
FH-series Vision Systems	Vision systems connected to the EtherCAT communications.
Front panel	Changes the operating mode of OMRON robot and executes an emergency stop.

# 1-4 Specifications

## 1-4-1 Product Model Numbers

The Sysmac Studio Robot Integrated System Building Function supports Sysmac Studio (64 bit) version 1.42 or higher.

To use the Sysmac Studio Robot Integrated System Building Function, the following Sysmac Studio licenses are needed. In addition to the above, to execute the 3D simulation of the Mechanical Component, the following Sysmac Studio option licenses are needed.

To install the Sysmac Studio (64 bit), the following DVD is needed.



### Additional Information

Even when you have not registered the Sysmac Studio 3D Simulation Option license number, you can use the Robot Integrated System Building Function.

### ● Sysmac Studio License

Product name	Number of licenses	Model number
Sysmac Studio Standard Edition Ver.1. □□	1 license	SYSMAC-SE201L
	3 licenses	SYSMAC-SE203L
	10 licenses	SYSMAC-SE210L
	30 licenses	SYSMAC-SE230L
	50 licenses	SYSMAC-SE250L

### ● Sysmac Studio Options

Product name	Number of licenses	Model number
Sysmac Studio 3D Simulation Option	1 license	SYSMAC-SA401L-64
	3 licenses	SYSMAC-SA403L-64
	10 licenses	SYSMAC-SA410L-64
	30 licenses	SYSMAC-SA430L-64
	50 licenses	SYSMAC-SA450L-64

### ● DVD

Product name	Media	Model number
Sysmac Studio Standard Edition Ver.1. □□	64-bit edition DVD	SYSMAC-SE200D-64

## 1-4-2 Supported Languages

The supported languages confirm to the specifications of the Sysmac Studio. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for details.

## 1-4-3 Applicable Models

The Sysmac Studio Robot Integrated System Building Function can be used with the following models.

- Robot Integrated CPU Unit NJ501-R□□□

Refer to the *NJ-series Robot Integrated CPU Unit User's Manual (Cat. No. O037)* for devices controlled by the Robot Integrated CPU Unit.

## 1-4-4 Applicable Computers

The computer on which the Sysmac Studio (64 bit) can be installed. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for details.

Refer to the *Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618)* for recommended system requirements to use the 3D Simulation Option functions.



## 2 Software Setup

This section describes the procedures for setting up the software to use the Sysmac Studio Robot Integrated System Building Function.

# 2-1 Installing the Sysmac Studio

---

Install the Sysmac Studio from the DVD. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for details of the installation procedure.

# 3

## 3 Building the Robot Integrated System

This section describes the Robot Integrated System targeted by the Sysmac Studio Robot Integrated System Building Function and the basic design flow of the Robot Integrated System.

# 3-1 Types of Robot Integrated Systems

This manual describes how to operate the Sysmac Studio for the following Robot Integrated Systems.

Robot Integrated System example	Description	Center of control	Reference
Dynamic Pick and Place system	A camera or sensor detects the part on a moving conveyor, and the Robot picks it up and places it on another conveyor. Use the Pack Manager application sample wizard to create a programless system.	IPC Application Controller	Refer to 3-2 and later for details.
Static Pick and Place system	A Robot picks a stationary part and places it on another conveyor.	Robot Integrated CPU Unit	Refer to the <i>Sysmac Studio Robot Integrated System Configuration Function Robot Integrated CPU Unit Operation Manual</i> (Cat. No. W595).

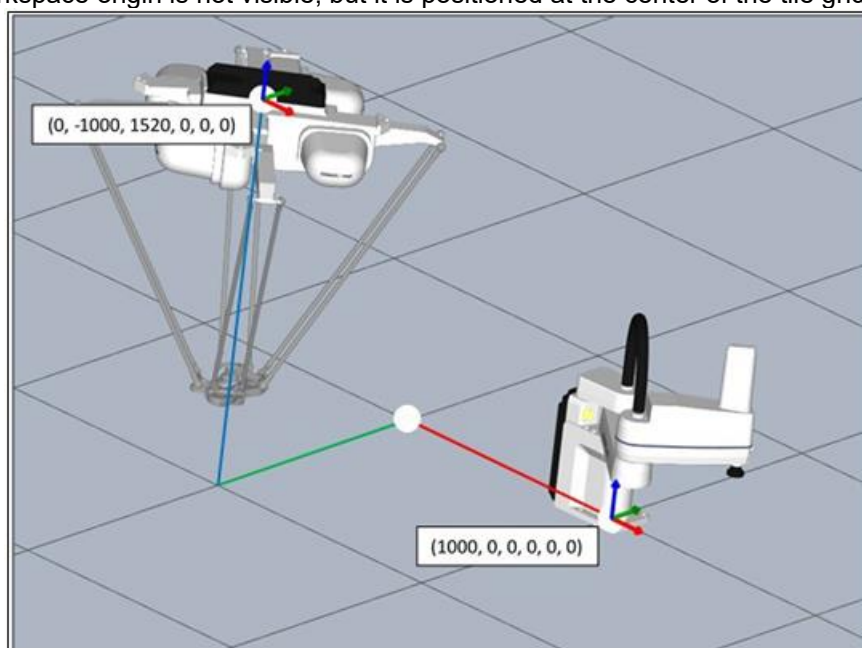
## 3-2 Coordinate Systems of Robot Integrated System

In Sysmac Studio, define the positions and directions using multiple coordinate systems for each motion element of the robot integrated system. The coordinate systems used in Sysmac Studio are as follows.

Coordinate System	Description	Reference
Workspace Coordinate System	This is the global coordinate system of the 3D Visualizer in Sysmac Studio. Workspace Coordinates are used to define the position of objects.	<i>3-2-1 Workspace Coordinate System</i>
Robot - World Coordinate System	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.	<i>3-2-2 Robot – World Coordinate System</i>
Robot - Joint Coordinate System	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.	<i>3-2-3 Robot - Joint Coordinate System</i>
Robot - Tool Coordinate System	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.	<i>3-2-4 Robot - Tool Coordinate System</i>
Belt Coordinate System	This is the coordinate system describing the direction and orientation of a conveyer belt. Each robot may have a different reference to belt coordinates for the same physical conveyor belt.	<i>3-2-5 Belt Coordinate System</i>
Camera Coordinate System	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.	<i>3-2-6 Camera Coordinate System</i>

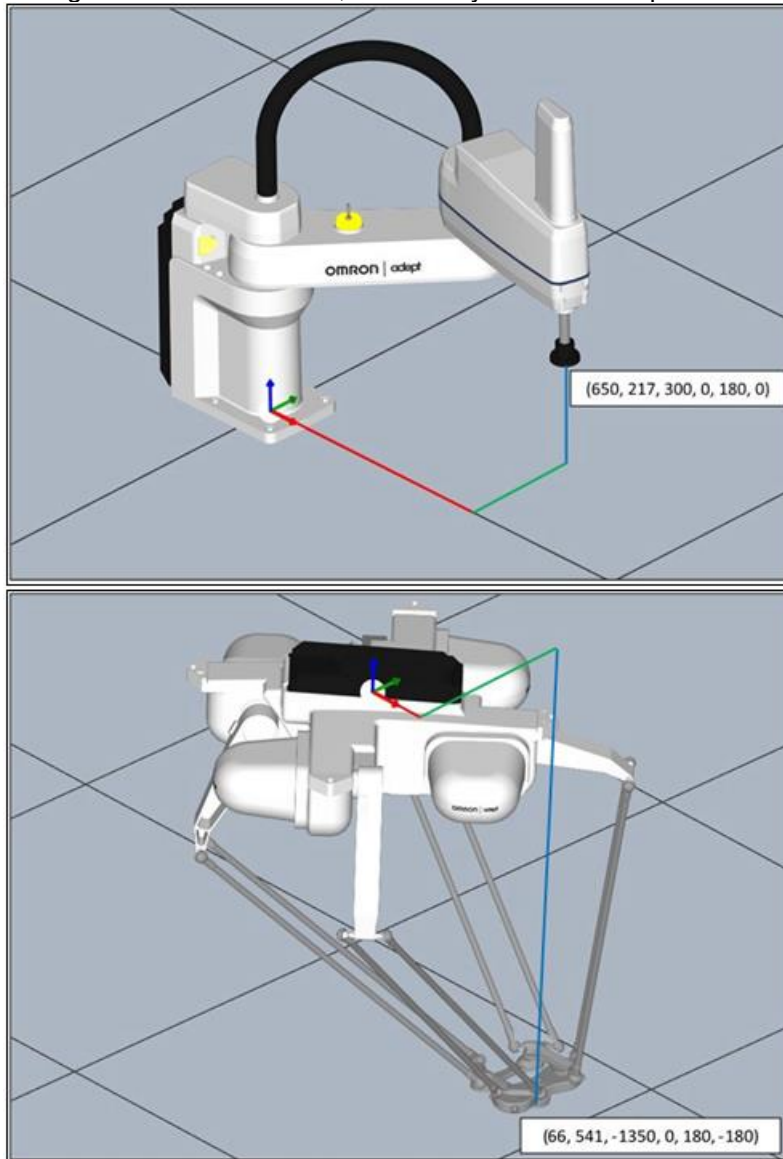
### 3-2-1 Workspace Coordinate System

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 below. 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 below.



### 3-2-2 Robot – World Coordinate System

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 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.



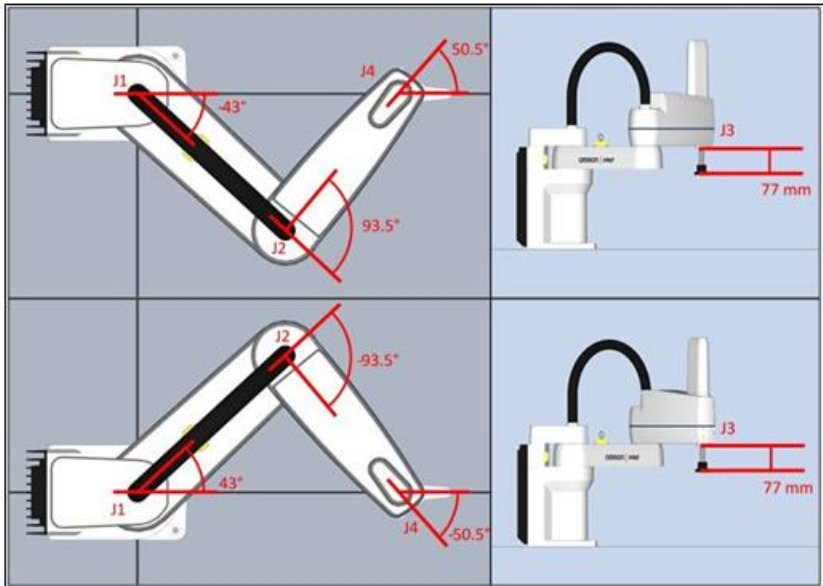
### 3-2-3 Robot - Joint Coordinate System

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.

Joint coordinates become useful when defining a point that can be reached in multiple orientations. For example, the two configurations shown 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  $(43, -93.5, 77, -50.5)$ .





## Precision Point

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 robots, since locations can be reached from multiple positions.

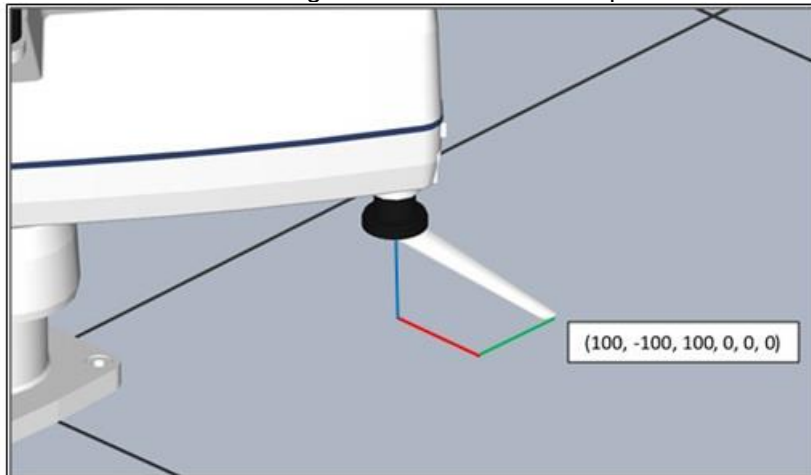


### Additional Information

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.

## 3-2-4 Robot - Tool Coordinate System

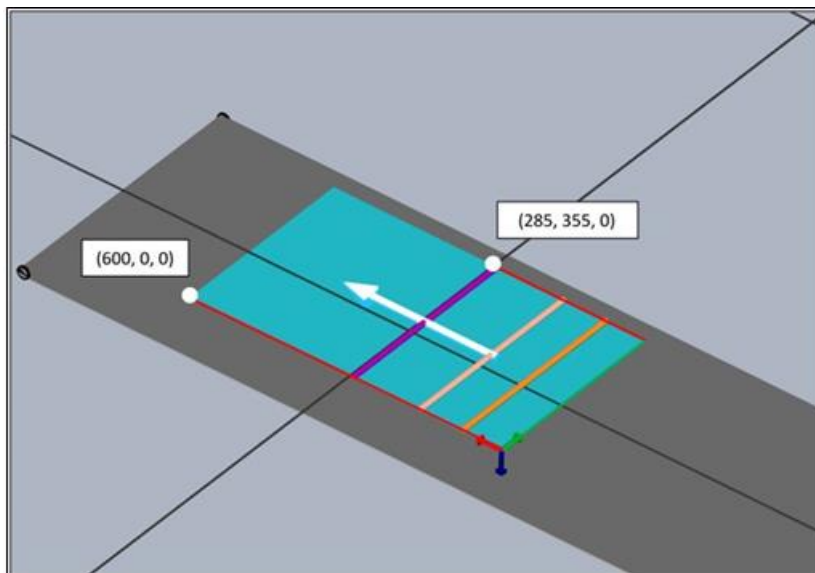
The tool coordinate system is used to define the position of tool tips. Its frame of reference is positioned at the tool flange itself. The tool Z-axis points opposite the other frames. This is because the main purpose of this system is to define the offset of tool tips. For example, a tool tip with 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.



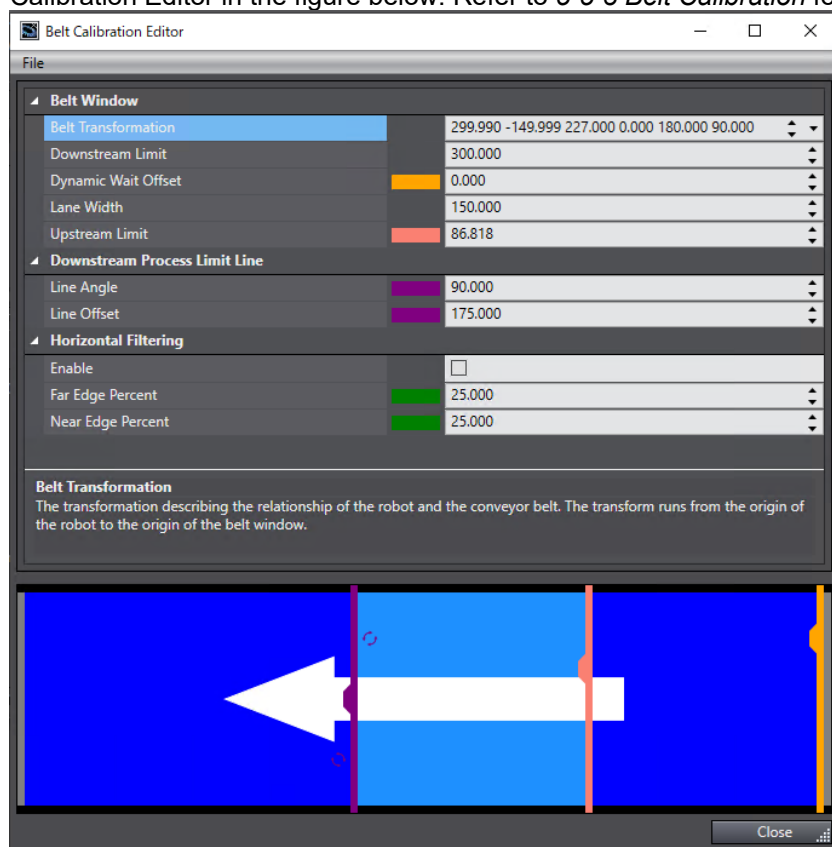
## 3-2-5 Belt Coordinate System

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 robot to belt 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 can be seen in the figure above based on the numbers shown in the Belt Calibration Editor in the figure below. Refer to *5-5-3 Belt Calibration* for more information.



### Additional Information

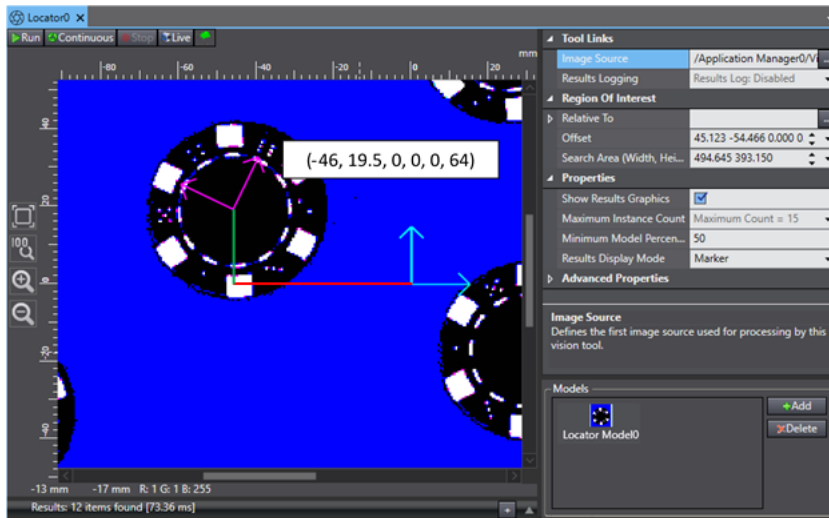
Belt coordinates do not apply to a Belt object created in the Process area of the Multiview Explorer (that is separate from a belt window). Belt objects are used to record information about the belt itself, such as encoders and signals, and provides a representation of a belt in the 3D Visualizer. Their 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.

## 3-2-6 Camera Coordinate System

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 returned as 6-element transformations and used, 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.



# 3-3 Calibration of Robot Integrated System

Robot Integrated System defines the positional relationship between equipment such as robots and belts. The calibration method may differ depending on whether the application uses ACE Sight or a Process Manager, but the function of calibration is the same. In applications using the 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.

## 3-3-1 Type of Calibration

There are the following types of calibration.

Type	Description	Reference
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.	3-3-2 <i>Robot-to-Belt Calibration</i>
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.	3-3-3 <i>Robot-to-Camera Calibration</i>
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.	3-3-4 <i>Robot-to-Latch Calibration</i>

When using Robot-to-Belt Calibration in combination with other calibrations in a robot system that uses belts, Robot-to-Belt Calibration needs to be performed first.

## 3-3-2 Robot-to-Belt Calibration

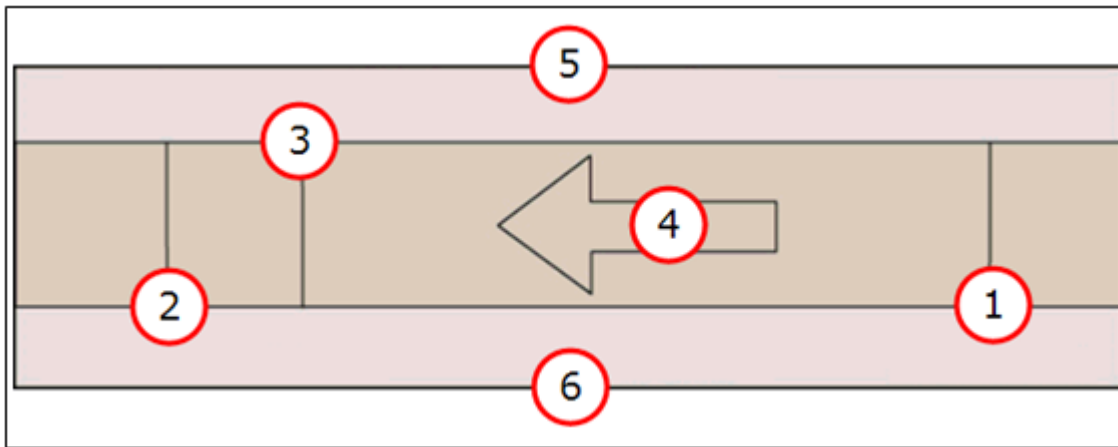
Robot-to-belt calibration will require three points to be defined on the surface of the belt, shown in order in the figure below. Use the following procedure to execute a robot-to-belt calibration.

- 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 factor 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.



### Precautions for Correct Use

The three points in this calibration also define other values, such as the upstream and downstream allocation limits, also shown in the figure below. 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 5-5-3 *Belt Calibration* for more information.



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

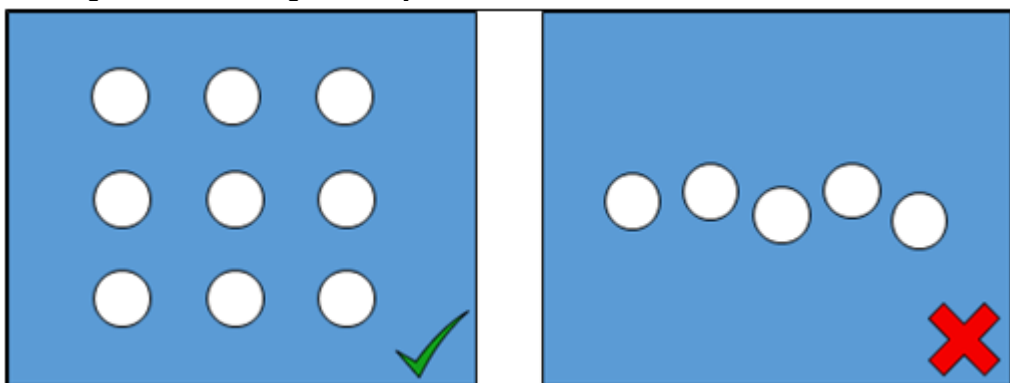
### 3-3-3 Robot-to-Camera Calibration

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 Grid Calibration for more information.

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

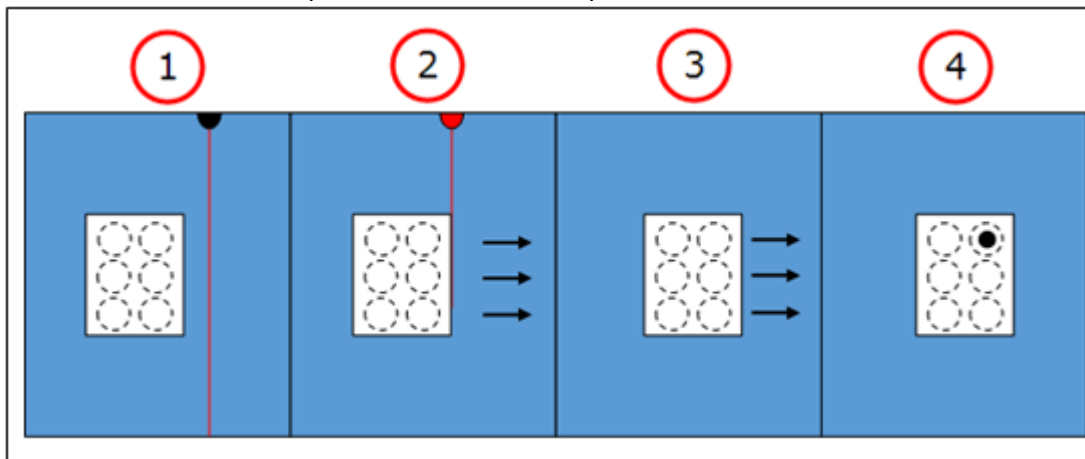
since the robot cannot reach the camera field of view. In the first phase, the targets are placed on the belt underneath the camera and their positions are recorded with respect to the camera. Then, the belt is advanced to where the robot can touch the targets and record their position in the robot world 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.

### 3-3-4 Robot-to-Latch Calibration

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 as follows.

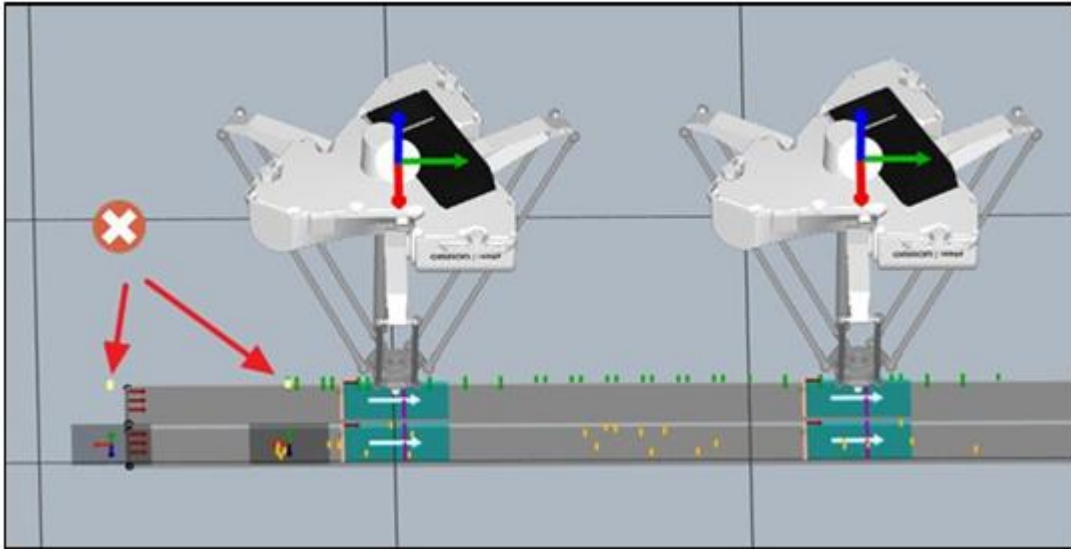


Item	Description
1	The pallet is positioned upstream of the latch sensor and the belt is not in motion.
2	The belt is advanced and the pallet is detected by the latch sensor. The belt encoder position at this position is recorded.
3	The belt is advanced to a position where the pallet is within the robot 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.

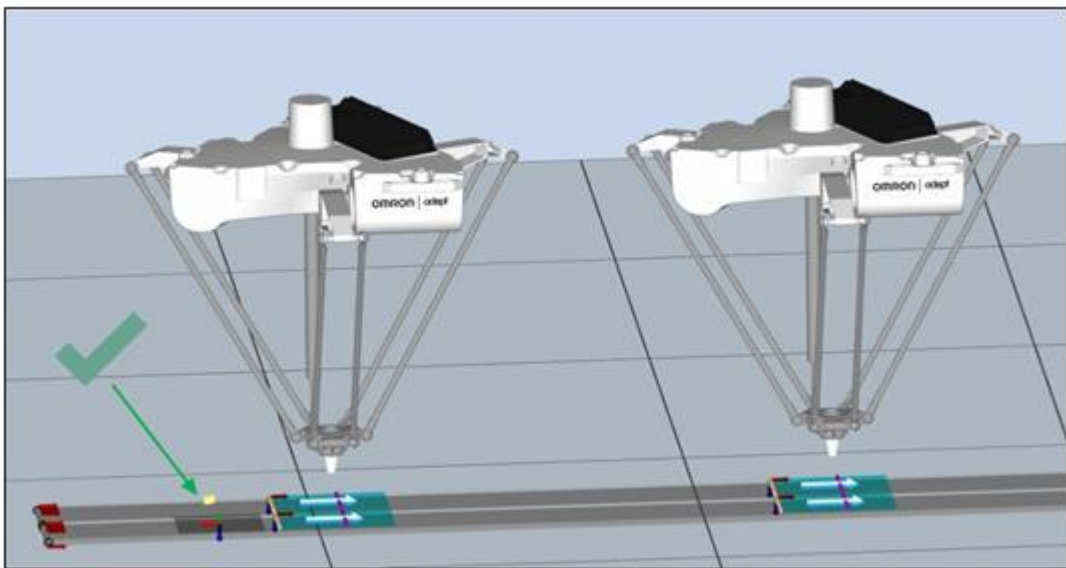


#### Precautions for Correct Use

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 figure below. 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.

## 3-4 Basic Robot Motion

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

### Speed, Acceleration, and Deceleration

Robot speed is usually specified as a percentage of normal speed, not as an absolute velocity. The speed for a single robot motion is set in the Speed parameter of the Pick Motion Parameters or Place Motion Parameters dialogs for each Part or Part Target location. The result obtained by the speed value depends on the operating mode of the robot (joint-interpolated versus straight-line). Refer to *Joint-Interpolated Motion vs. Straight-Line Motion* 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 joints 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.

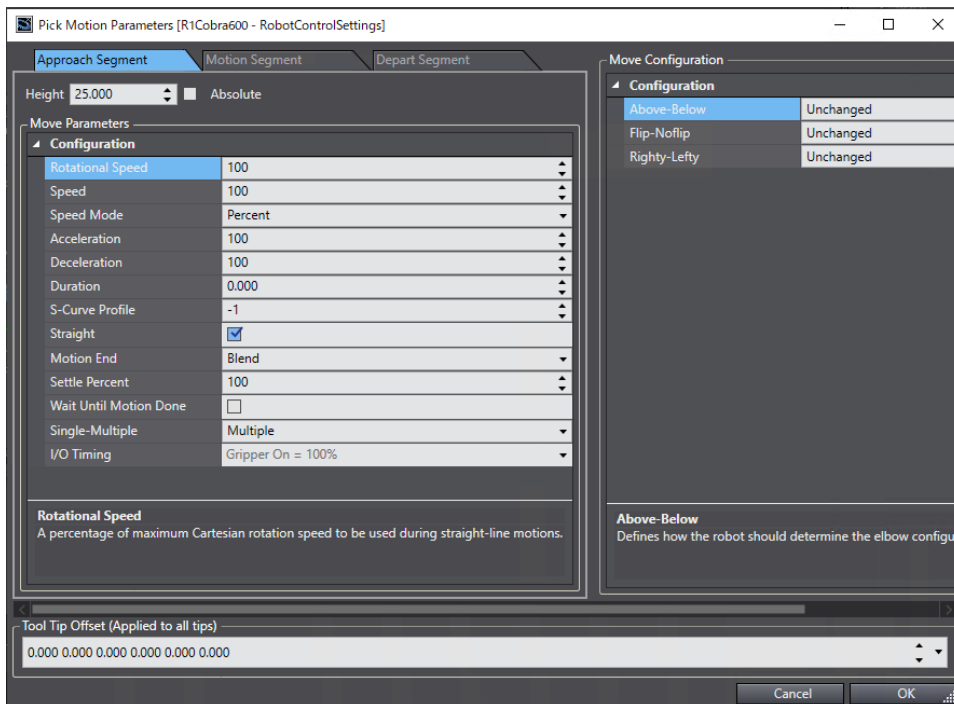
Higher acceleration / deceleration rates and higher speeds do not always result in faster cycle times due to positioning overshoot that may occur.

### 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.

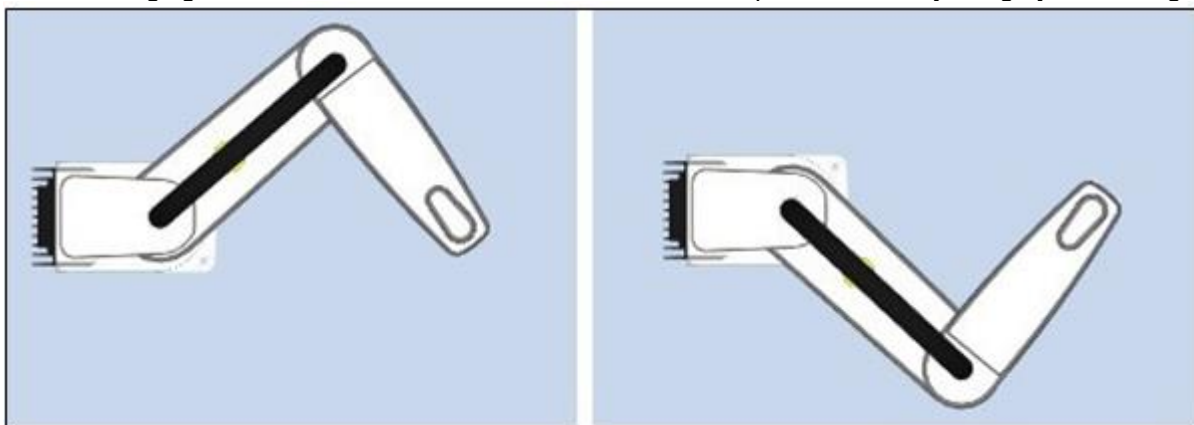
## Arm Configuration

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* or the eV+ Language Reference Guide for more information.

The following figure illustrates how a SCARA robot can reach a point with a lefty or righty arm configuration.



## Location Precision

When a robot moves to a location, it actually makes several moves, each of which is a closer approximation of the exact location. You can control the precision with which the robot moves to a location using the Motion End parameter (Settle Fine / Settle Coarse). If the 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.

## Continuous Path Motion

Making smooth transitions between motion segments without stopping the robot motion is called continuous path operation. When a single motion instruction is processed, the robot begins moving toward the location by accelerating smoothly to the commanded speed. Sometime later, when the robot is close to the destination location, the robot decelerates smoothly to a stop at the location. This motion is referred to as a single motion segment, because it is produced by a single motion instruction.

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

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.



### Precautions for Correct Use

Continuous-path transitions can occur between any combination of straight-line and joint-interpolated motions. Refer to *Joint-Interpolated Motion vs. Straight-Line Motion* for more information.

## 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 *Speed, Acceleration, and Deceleration* 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.

## 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 guide.

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

### ● Cell Layout Considerations

Regarding cell layout and jointed arms, the same point-to-point distance can result in different cycle times. Moving multiple joints combines the joint speeds for faster motion.

If the same distance is traversed using motion of a single joint, the motion of that joint will be longer, and therefore will take more time.

- **Part Handling Considerations**

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

- **Programming Considerations**

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

## 3-5 Belts (Conveyors)

---

This section describes basic belt (conveyor) concepts.

Conveyors are referred to as belts in the Sysmac Studio. Refer to *5-5-3 Belt Calibration* for more information.

### 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 “non-continuous 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**

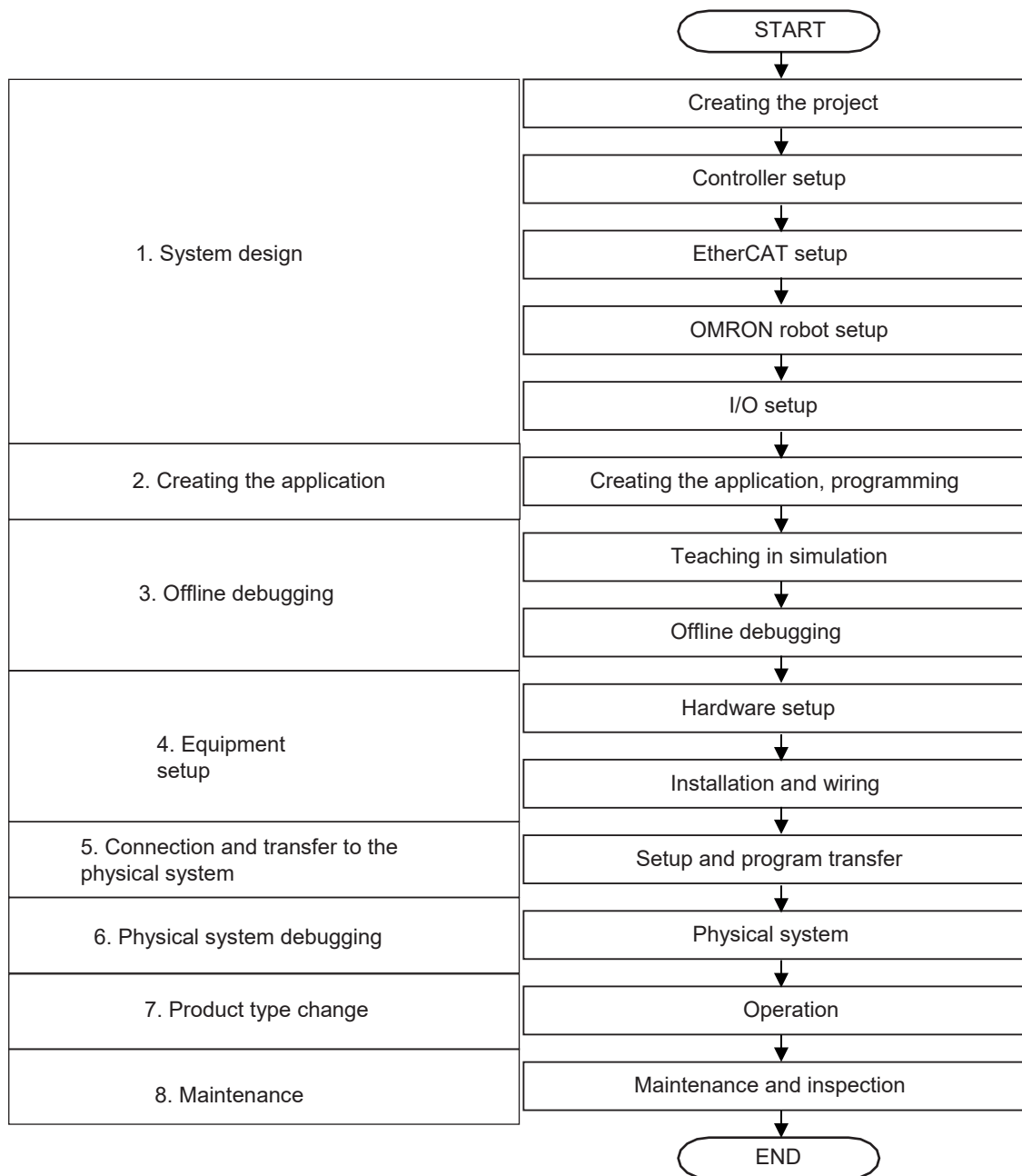
In a tracking conveyor system, the belt moves continuously and the robot tracks parts until the speed and location of the robot gripper match those of a part on the belt. The robot then accesses the part. Tracking conveyors must be equipped with an encoder that reports the movement of the belt and distance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

# 3-6 Flow of System Basic Building Controlled with IPC Application Controller

The building procedures for Robot Integrated System in the Sysmac Studio are as follows. Hereafter, operation procedures and functions are described according to this flow.



Procedure	Item		Description	Reference
1	System Design	New project creation	Creates a project for the Robot Integrated CPU Unit and adds a robot to the EtherCAT network.	<i>4-2-1 Creating a New Project</i>
		The Robot Integrated CPU Unit setup	Sets up the Robot Integrated CPU Unit for the robot control	<i>Sysmac Studio Robot Integrated System</i>

Procedure	Item	Description	Reference	
2	Creating the application		<i>Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)</i>	
		EtherCAT setup	Adds the EtherCAT slaves including the OMRON robots on the EtherCAT, and makes the settings.	<i>Sysmac Studio Version 1 Operation Manual (Cat. No. W504)</i>
		OMRON robot setup	Makes the robot control parameter settings.	
		I/O setup	Makes the setting to control I/O from a user program.	<i>Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)</i>
		Creating the application	Creates the Robot Integrated System application as the dynamic pick-and-place system. Sets up conveyors, cameras, process manager and its components that are the components of the Application Manager. And sets and various parameters and the robot relations of the Robot Integrated CPU Unit. You can also create application models with the Application sample wizard.	<i>5 IPC Application Controller Setup 8 Application Sample Wizard</i>
		Programming	Creates an application program by using C# language as required.	<i>5-11 C# Program</i>
3	Offline Debugging	Simulation of the Robot Integrated System	Simulates operations of the Robot Integrated System that includes robots and belt conveyors on the computer, and perform the debugging. Check takt time and interference between a robot and peripheral equipment as required.	<i>7 Debugging of the Robot Integrated System</i>

Procedure	Item		Description	Reference
4	Equipment setup	Hardware setting, installation and wiring	Make the settings for the hardware switches on the devices, installation and wiring for I/O and the network.	Refer to the <i>NJ-series Robot Integrated CPU Unit User's Manual</i> (Cat. No. O037), <i>IPC Application Controller User's Manual</i> (Cat. No. I632) , And the manuals of used OMRON robots and EtherCAT slaves.
5	Connection and transfer to the physical system	Connection and transfer to the Robot Integrated CPU Unit	Connect to the Robot Integrated CPU Unit online, and transfer the Controller settings and programs.	6 <i>Online connection with the IPC Application Controller</i>
		Connection and transfer to the IPC Application Controller	Connect to the IPC Application Controller online, and transfer the Application Manager setting.	
6	Physical system debugging	Operation check on the physical system	Perform operations such as camera calibration, belt calibration with the robot, and robot teaching on the physical system to check the system operation.	7 <i>Debugging of the Robot Integrated System</i>
7	Product type change	Setup for product type change	Sets up the IPC Application Controller and the Controller according to the product type to change to, create programs, and simulate the operation of the system.	5-7-6 <i>Recipe Manager</i>
8	Maintenance	Troubleshooting	If any device error occurs, check the error information on the IPC Application Controller and take action according to the error.	9 <i>Maintenance</i>







# 4

## 4 System Design

---

This section describes the robot system building procedures using the Robot Integrated CPU Unit and the IPC Application Controller in the Sysmac Studio.

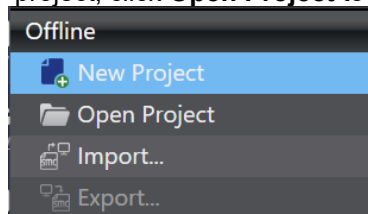
# 4-1 Starting and Exiting the Sysmac Studio

This section describes the starting procedure to use the Robot Integrated CPU Unit in the Sysmac Studio. For the basic operating procedure and precautions to start and exit the Sysmac Studio, refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*.

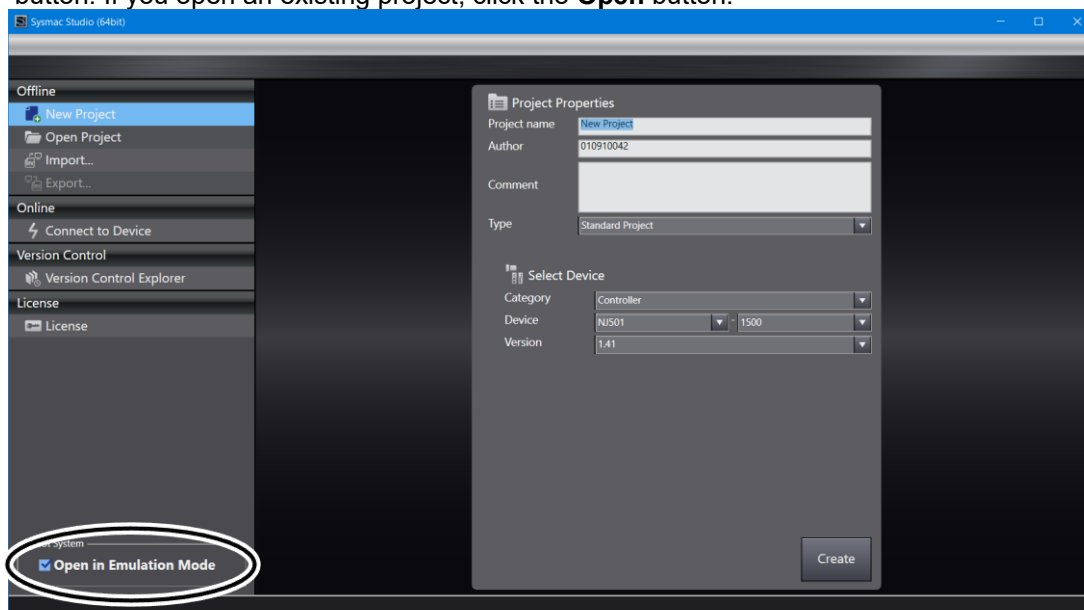
## 4-1-1 Starting the Sysmac Studio

You can perform a robot simulation for the Robot Integrated CPU Unit in the Sysmac Studio. Open a project in *Emulation Mode*.

- 1 Start the Sysmac Studio and click **New Project** on the start page. If you open an existing project, click **Open Project** to select the target project.

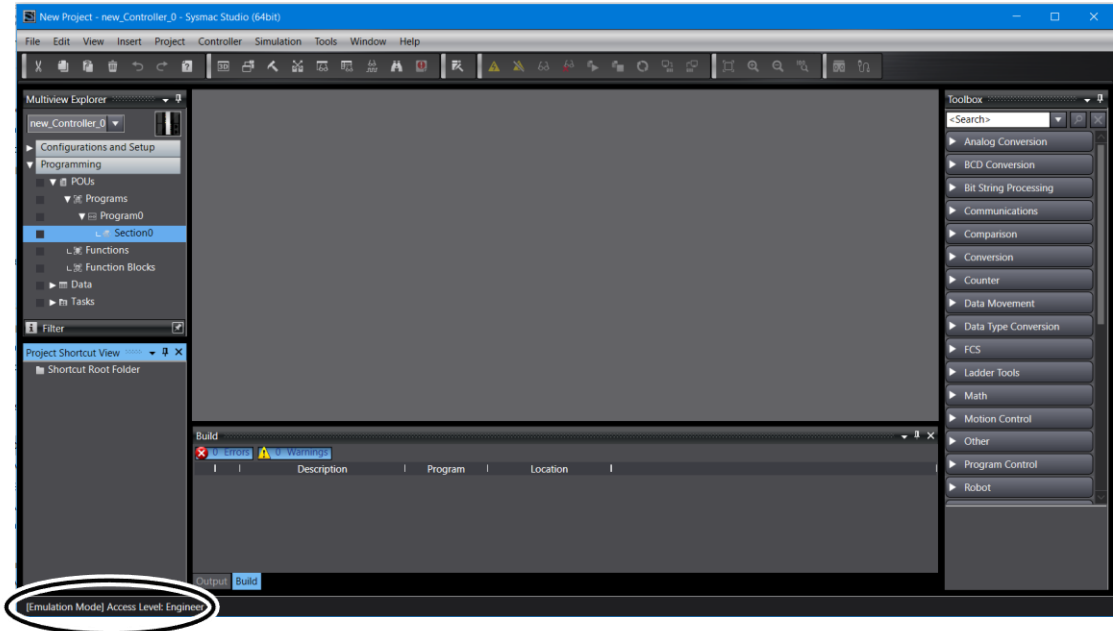


- 2 Select the **Open in Emulation Mode** check box under **Robot System** and click the **Create** button. If you open an existing project, click the **Open** button.



The project is opened in emulation mode.

At this time, the **Emulation Mode** is displayed in the status bar of the main Sysmac Studio window.



### Additional Information

- This option is valid only for the Robot Integrated CPU Unit.
- A project in the Robot Integrated System needs to run an emulator for the Robot Control Function Module, if you set up the Robot Control Function Module, edit the V+ program and perform offline debugging. Select the **Open in emulation mode** option to open a project.

Application	Open in emulation mode option
Setting up the Robot Control Function Module, editing V+ program and performing offline debugging	On
Physical component debugging, maintenance	Off

## 4-1-2 Exiting the Sysmac Studio

Use one of the following methods to exit the Sysmac Studio.

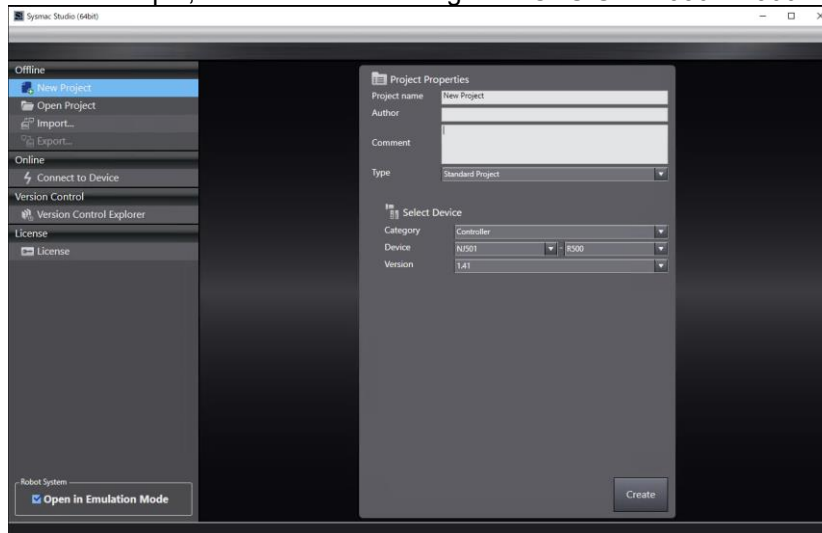
- Click the **Close** button on the right end of the title bar.
- Select **File - Close** from the main menu.

## 4-2 Creating a Project

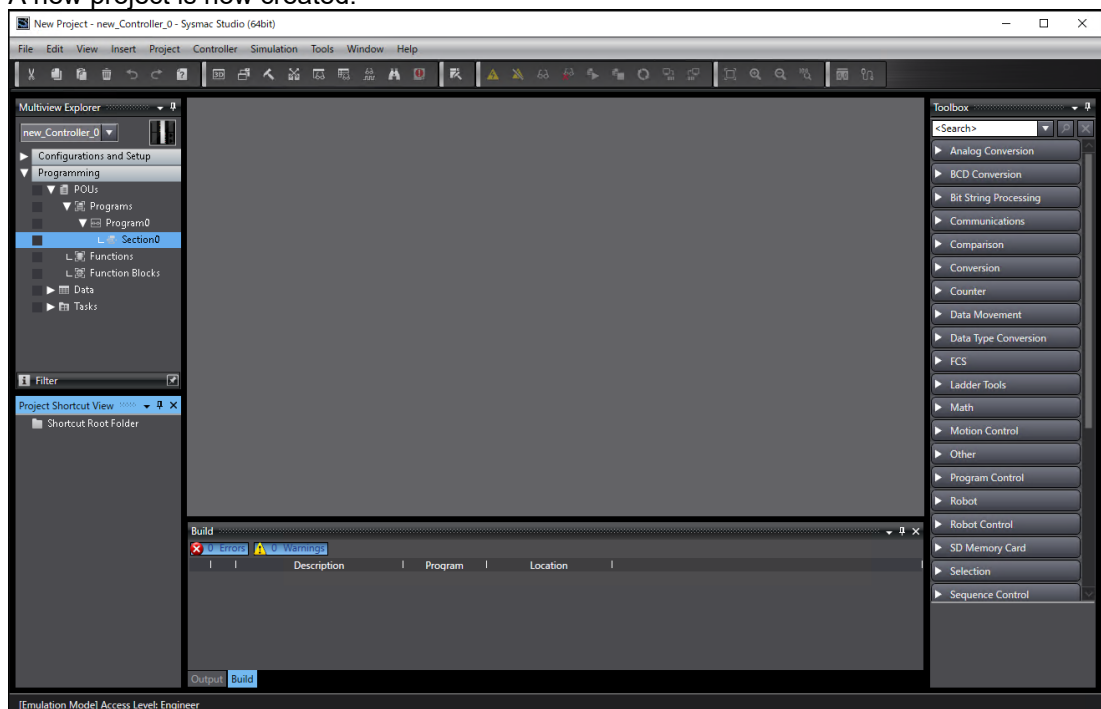
This section describes the procedure for creating a new project for a system that controls a robot with the IPC Application Controller. Create a new project for the Robot Integrated CPU Unit and register a robot to use.

### 4-2-1 Creating a New Project

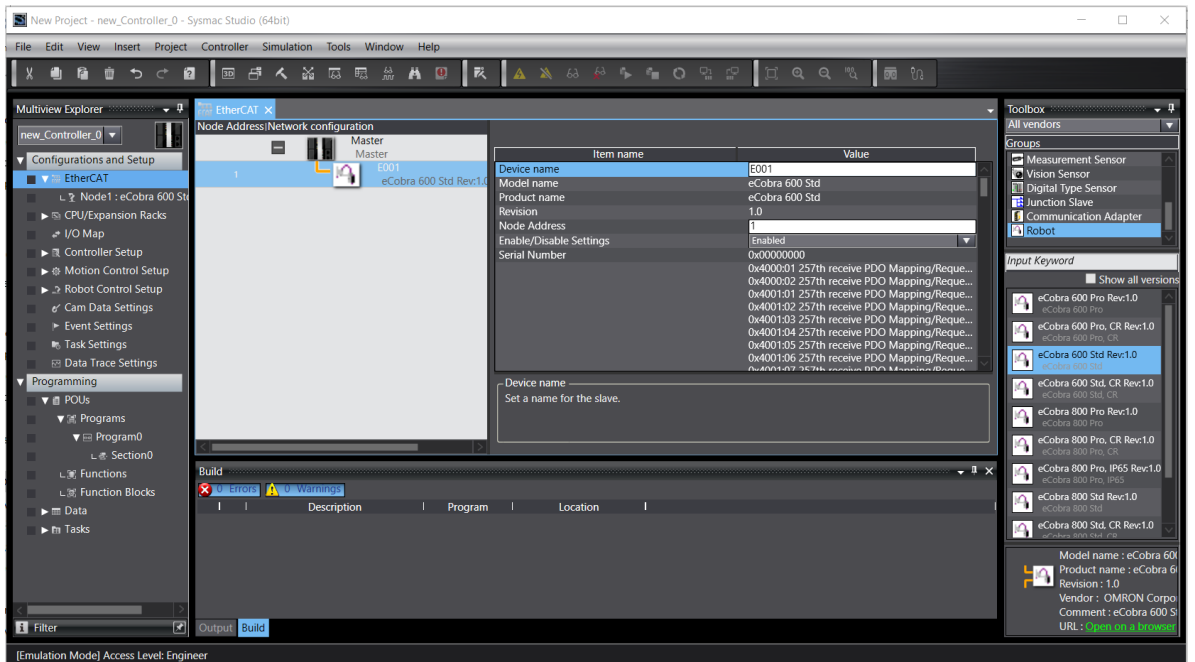
- 1 Select **New Project** on the Start menu of the Sysmac Studio. Then, select the Robot Integrated CPU Unit in **Select Device**.  
In this example, select the Robot Integrated CPU Unit NJ501-R500.



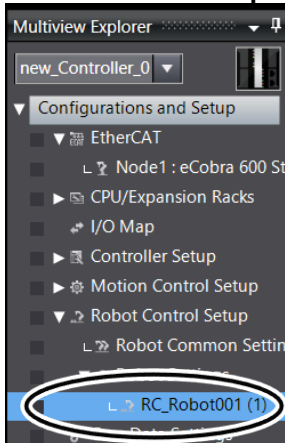
- 2 After entering data in the **Project Properties** dialog box, select **Open in Emulation Mode** check box and click the **Create** button.  
A new project is now created.



- 3 Display the EtherCAT Tab Page from **Configurations and Setup** in the Multiview Explorer, and register the robot to use.  
In this example, select the OMRON SCARA robot *Cobra 600*. Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details on the registration procedure of a device in the EtherCAT Tab Page.

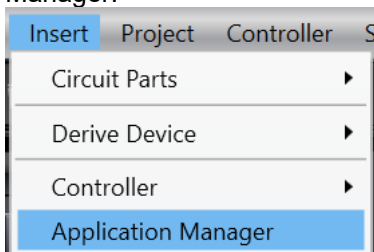


The robot is registered. The registered robot is displayed under **Configurations and Setup - Robot Control Setup - Robot Settings**.

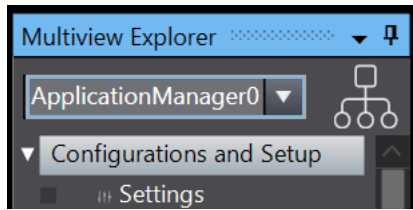


It is necessary to set the cycle of the primary periodic task to 2 ms or more in the task setting when you use a robot. Refer to the *Sysmac Studio Version 1 Operation Manual* (SBCA-470 No. W504) for how to operate the task settings.

- 4 In the EtherCAT Tab Page, register the peripheral devices to connect to the EtherCAT network as required.  
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for the registration procedure.
- 5 Register the Application Manager operated with the IPC Application Controller.  
There are the following registration methods: using the application sample wizard and registering an empty Application Manager.  
Refer to *8-1 Overview of Application Sample Wizards* for using the application sample wizard. Select **Application Manager** from the **Insert** menu in order to register an empty Application Manager.



The Application Manager is added.



## 4-2-2 Application Manager Device Configuration

This section describes setting items of the Application Manager.  
The items of the *Application Manager* device are as follows.

### ● Configurations and Setup

Item	Description	Reference
Setup	Sets the IP address and port number of the IPC Application Controller to be connected.	5-3 <i>IPC Application Controller Connection Settings</i>
3D Visualization	Adds 3D Shape Data in the 3D Visualizer of the robot system.	5-4 <i>3D Visualization</i>
Robot Vision Manager	Adds the settings for using Robot Vision Manager.	5-5 <i>Robot Vision Manager</i>
Camera	Adds cameras used in Robot Systems. Virtual cameras are added and used instead of actual cameras in offline simulation.	5-6 <i>Camera</i>
Configuration	Customizes the application operations by settings of the Application Manager starting, Data Mapper, and recipe.	5-7 <i>Configuration</i>
Feeder	Adds feeders supplying parts, to the robot system.	5-8 <i>Feeders</i>
Process	Makes various settings, related to the process, to pick a part and place it at the target position based on the specified execution order.	5-9 <i>Process</i>
Vision Tool	Adds tools for image processing use in the Robot Vision Manager.	5-10 <i>Vision Tools</i>

### ● Programming

Item	Description	Reference
Program	Creates an application program by using C# language.	5-11 <i>C# Program</i>
Variable	Is defined when various objects of Application Manager and variables shared by multiple C# programs, are required.	5-11-2 <i>C# Variable Objects</i>

## 4-2-3 Robot Integrated CPU Unit Configuration

This section describes the setting items of the Robot Integrated CPU Unit.  
They include the setting items of the controller function and the subdevice *RobotControlSettings* which controls the robot control function.

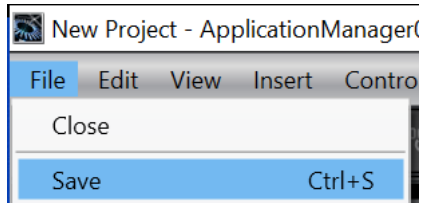
Setting Item	Setting outline	Reference
Controller setting	Performs the Controller setting (operation settings and built-in EtherNet/IP settings) of the Robot Integrated CPU Unit.	<i>Sysmac Studio Version 1 Operation Manual (Cat. No. W504)</i>
EtherCAT setting	Adds the EtherCAT slaves including the OMRON robots on the EtherCAT, and makes the settings.	<i>Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)</i>
The Robot Integrated CPU Unit setting	Sets robot numbers of OMRON robots, robot names specified by user program, and the I/O control by the V+ program.	
Robot subdevice setting	Sets the Robot Control Function Module, the Controller Setup, and the Save Configuration.	
Robot setting	Sets OMRON robots. In addition, sets the option settings/parameters and obstacle settings for each axis that operates the robots.	<i>4-4-1 Setting up IO EndEffectors Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)</i>

Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* and *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for other setting details.

## 4-3 Saving the Project

This section describes saving procedures of projects including Application Manager.

- 1 Select **File - Save** from the main menu. Or select **Save As**.



The current project is saved.

## 4-4 Robot Settings

Among the Robot Settings, this section describes the *IO EndEffector settings*, which are required in Robot Integrated Systems with an IPC Application Controller.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for other Robot Settings.

### 4-4-1 Setting up IO EndEffectors

Set up IO EndEffectors. End-Effectors, also known as tools and grippers, are used for pick, place, coating, and other common functions in Robotic applications.

End-Effectors perform actions such as grabbing, extension, retraction, and coating.

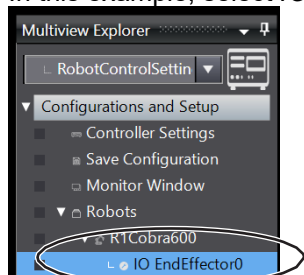
When you add a Robot, a single End-Effector object (IO EndEffector0) is automatically added. This is an I/O control gripper with one or more end-effector tips. To control each tool tip, digital input and output signals are used.

It is also possible to set up multiple End-Effectors so that the Robot can pick and place multiple objects at the same time. Add additional IO EndEffector objects to your application as needed. This includes calibration pointers that you use to calibrate systems with multi-tip grippers.

### Displaying the End-Effector Tab Page

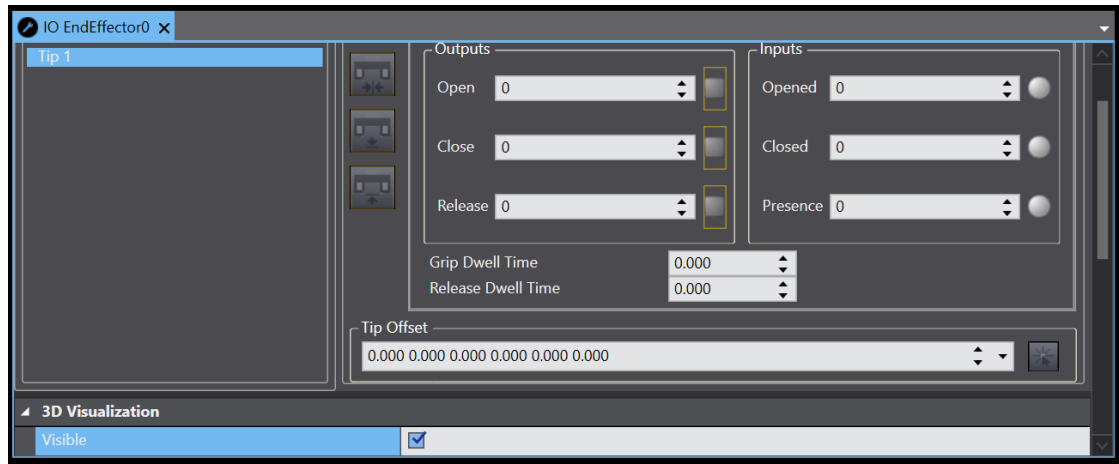
- 1 In the Multiview Explorer, select **RobotControlSettings** from the device list.
- 2 Double-click an End-Effector under **Configurations and Setup - Robot** in the Multiview Explorer. Or, right-click it and select **Edit** from the menu.

In this example, select *IO EndEffector0*.

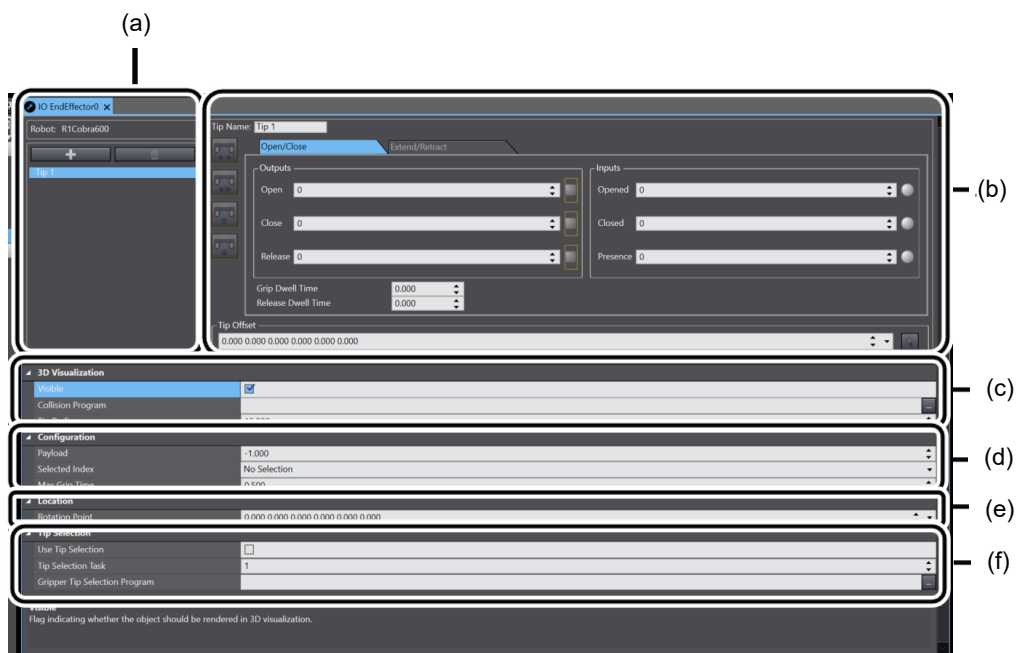


The setup tab page for *IO EndEffector0* is displayed.





## Setup Tab Page for IO EndEffectors



Item	Name	Description
(a)	<b>Add/Delete</b> buttons	Adds or deletes gripper tips.
	Tool tip list	Displays a 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) is used to pick up and release the part during the calibration procedure. The center line of the gripper is used as the reference tool in the calibration procedure.  NOTE: If multiple tips are defined, it can be useful to create a second IO EndEffector object that stores a single tool tip offset for use during calibrations. This is because the calibration process allows you to choose which IO EndEffector object to use, regardless of whether the physical End-Effector needs to be changed. Refer to <i>5-9-5Process Manager</i> for more information.
(b)	Outputs/Inputs	Defines the <b>Open/Close</b> or <b>Extend/Retract</b> activation signals and <b>Opened/Closed</b> or <b>Extended/Retracted</b> status signals for the selected tool tip. You can 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 this is specified, the status of the presence sensor is checked when a part is picked or placed. Before the Robot places a part in the target position, 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 are monitored.
	Grip Dwell Time/Release Dwell Time	The time (in seconds) that the Robot must wait before it continues with the grip or release operation. This is the actuation time for the gripper.
	<b>Open/Close</b> tab	Click this tab to access the Open/Close signal settings for the selected tool tip.
	<b>Extend/Retract</b> tab	Click this tab to access the Extend/Retract signal settings for the selected tool tip.
	<b>Open Tip/Close Tip</b> buttons	Use the <b>Open Tip</b> and <b>Close Tip</b> buttons to send an open tip ( ) or close tip ( ) signal to the selected tool tip.
	<b>Extend Tip/Retract Tip</b> buttons	Use the <b>Extend Tip</b> and <b>Retract Tip</b> buttons to send an extend tip ( ) or retract trip ( ) signal to the selected tool tip.
	Extend-Retract Dwell Time	The time (in seconds) that the Robot must wait before it continues with the extend or retract operation. The value represents the minimum dwell time. After the specified dwell time, the input signals are checked.
	Tip Offset	Shows the current offset for the selected tool tip in the 3D Visualizer. To change the offset, click the <b>Teach Tool Tip Wizard</b> button, which starts the <b>Tip Offset Wizard</b> .
(c)	Visible	Enables the display of the IO EndEffector tool tip positions in the 3D Visualizer. Refer to <i>5-4 3D Visualization</i> for more information.
	Collision Program	Select a program that is invoked when the 3D Visualizer detects a collision. Refer to <i>7-2-4 Collision Detection Settings</i> for more information.
	Tip Radius	The radius of the tool tip when the IO EndEffector is drawn in the 3D Visualizer. Refer to <i>5-4 3D Visualization</i> for more information.
(d)	Payload	The weight of the gripper plus the weight of the heaviest part the robot can carry (in kg). Setting a correct payload reduces tracking errors and settling time, which makes 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 tool tips are present.
	Max Grip Time	The maximum time (in seconds) allowed for grip verification and part presence check.
(e)	Rotation Point	The offset of the axis that becomes the center of the rotation operation.
(f)	Use Tip Selection	When this is enabled, the tip selection program is called when a tip is selected.
	Tip Selection Task	The task used when the tip selection program is executed.
	Gripper Tip Selection Program	The V+ program called when a tip is selected. This is used in the case where the gripper has some additional operation that is required to switch the tool tips. For example, you may need to set some additional I/O (or to move a motor) to physically move the tool tips.

## 4-5 Parts of the Window

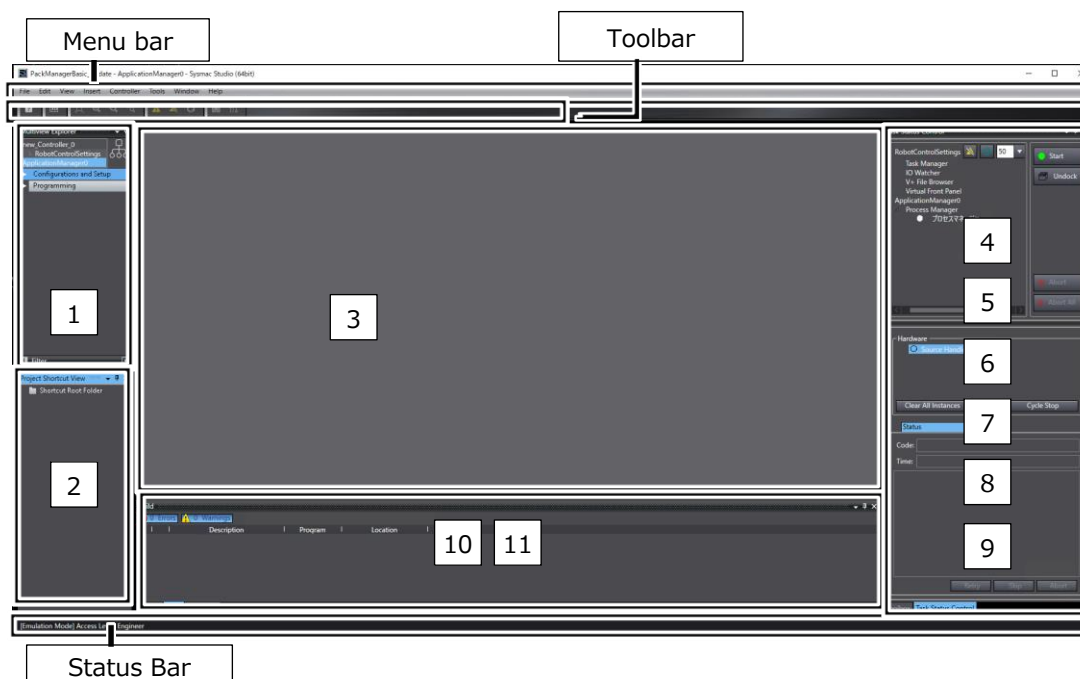
This section gives the names of the parts of the Sysmac Studio Window.

This section provides an overview of the windows and screens when you select *Application Manager* in the device list.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for basic display explanations.

### 4-5-1 Application Window

This is the main Sysmac Studio window.



Number	Name
1	Multiview Explorer
2	Project Shortcut View
3	Edit Pane
4	Toolbox
5	Search and Replace Pane
6	Task Status Control Pane
7	3D Visualizer
8	V+ Jog Control Pane
9	Vision Window
10	V+ Watch Tab Page
11	System Monitor Pane

### Menu Bar, Toolbar, and Status Bar

Hiding/Showing the Menu Bar, Toolbar, and Status Bar are common in the Sysmac Studio. Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details.

### Status Bar Display Information

When using the Application Manager, the following information is displayed in the status bar.

[Emulation Mode] Access Level: Engineer

When opening the project, the status of the emulation mode selected in **Robot System** is displayed. It also displays the access level set by the robot system operation authority verification function and authenticated by sign-in. Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual* (Cat. No. W595) for the robot system operation authority verification function and sign-in.

## 4-5-2 Project Shortcut View

You can create shortcuts for items, that are displayed under Configurations and Setup and Programming in the Multiview Explorer, to make them easier to access.

## 4-5-3 Edit Pane

The Edit Pane displays detailed data of Configurations and Setup items and the V+ Program editor window.

## 4-5-4 Toolbox

The Toolbox shows the objects that you can use to edit the data that is displayed in the Edit Pane.

## 4-5-5 Search and Replace Pane

In the Search and Replace Pane, you can search for and replace any string of the C# programs under Programming in the Multiview Explorer.

## 4-5-6 Task Status Control Pane

The Task Status Control Pane is the control screen for robot control. Refer to *7-5 Task Status Control* for details.

## 4-5-7 3D Visualizer

The 3D Visualizer shows robots, conveyors, and other equipment to check operations in the 3D simulation. Refer to *7-2 3D Visualizer* for details.

## 4-5-8 V+ Jog Control Pane

The V+ Jog Control Pane is for robot jog operations. Refer to *7-4 V+ Jog Control* for details.

## 4-5-9 Vision Window

Images of the vision tool and a camera set in the project can be displayed. Refer to *7-6 Vision Window* for details.

## 4-5-10 V+ Watch Tab Page

Displays a current value of a C# Variable object. Refer to *7-7-1 Adding C# Variable Objects to the V+ Watch Window* for details.

## 4-5-11 System Monitor Pane

Displays parameters of robots and the Application Manager graphically in real time, and outputs them to a log file. Refer to *10-2 System Monitor* for details.

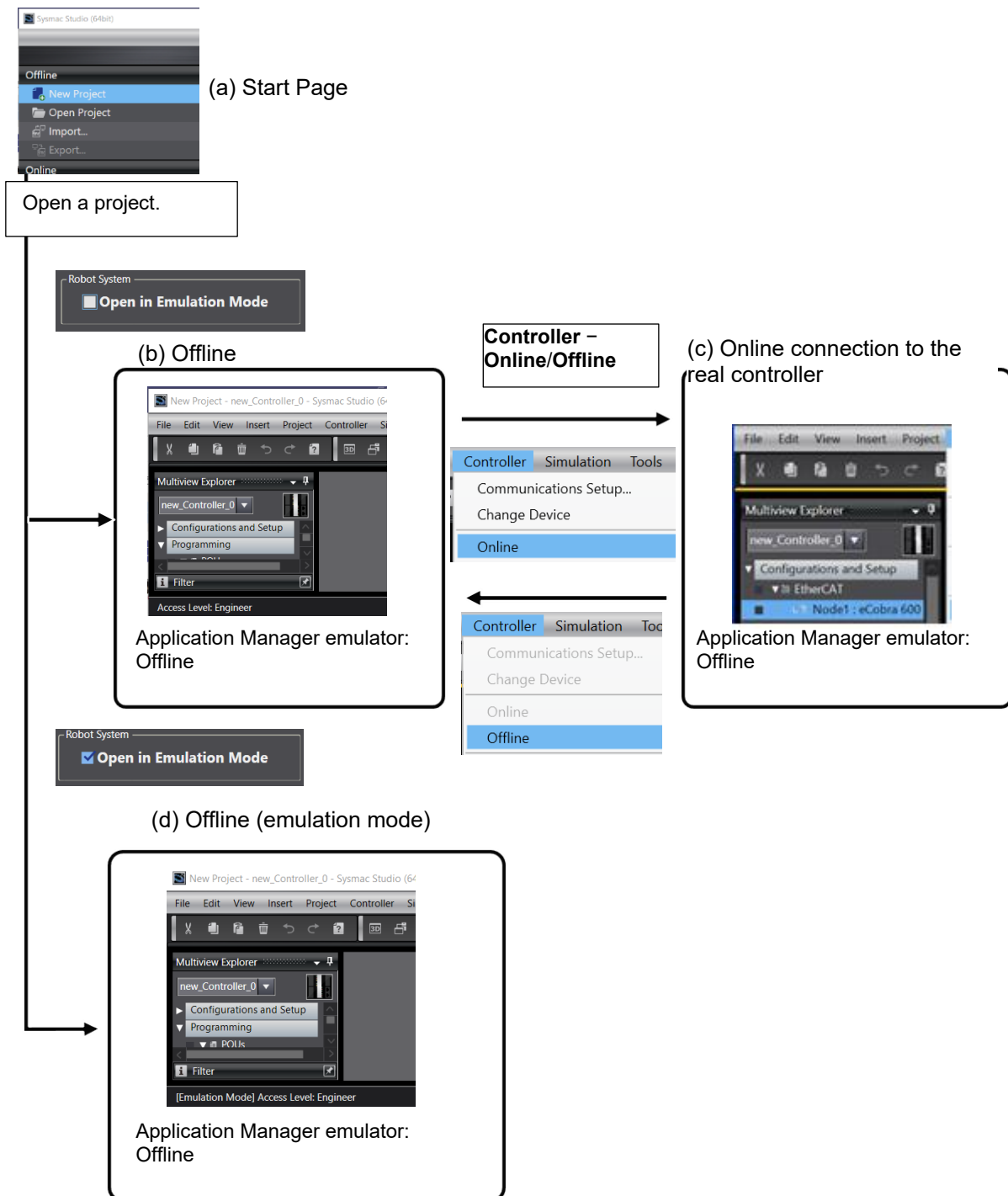
# 4-6 Offline and Online Modes of the Sysmac Studio and How to Use them

A project of the Robot Integrated System which uses the IPC Application Controller and the Robot Integrated CPU Unit has *emulation mode*, which executes the emulator of the application manager and the Robot Control Function. In *Emulation Mode*, you can set the Robot Control Function Module and edit V+ program and perform offline debugging. And you can also set the Application Manager and perform offline debugging.

The editable data and usable functions of the robot control function module and the Application Manager differ depending on the combination, of the connection status such as *simulator connection* and *online connection to physical system* and this *Mode*.

Regarding the mode and the connection status, the state transition diagram of the Application Manager starting from the start page is shown below.

Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual* (Cat. No. W595) for the state transition diagram of the Robot Control Function Module.



The operations and applications that can be used in each state are shown.

	Status	Application	Editable data and available operation	Non-editable data	State transition to	State transition operation
(a)	Start Page (Before opening a project)	The project selection	The project selection and whether to open the project in emulation mode	Project data	Offline	Opening the project by unchecking <b>Open in Emulation Mode</b>
					Offline (emulation mode)	Opening the project by checking <b>Open in Emulation Mode</b>
(b)	Offline	Configurations and Setup of the Application Manager and editing of programming (C# program, variable)	Configurations and Setup of the Application Manager and programming (C# program, variable)	None	Online connection to the physical system	Select <b>Online</b> from the <b>Controller</b> Menu. Or press the online button.
(c)	Online connection to the physical system	Application transfer/execution/monitor Configurations and Setup of the Application Manager and editing of programming (C# program, variable)	Application transfer/execution/monitor Configurations and Setup of the Application Manager and programming (C# program, variable)	None	Offline	Select <b>Offline</b> from the <b>Controller</b> Menu. Or press the online button.
(d)	Offline (emulation mode)	Configurations and Setup of the Application Manager and editing of programming (C# program, variable) Offline Debugging (*1)	Configurations and Setup of the Application Manager and editing of programming (C# program, variable) Application execution/monitor	None	-	-

\*1 The Simulator connection must already be performed with Robot Integrated CPU Unit to perform offline debugging of the Application Manager.



#### Additional Information

- Offline debugging can be performed only in emulation mode of a Robot Integrated System project.
- You can select **Enable emulation mode/ Disable emulation mode** from the **Controller** menu to enable the transferring if the status is between (b) *offline* and (d) *offline (emulation mode)*. At this time, the project is closed and reopened.

# 5

## 5 IPC Application Controller Setup

This section describes how to use the Sysmac Studio in a Robot system that uses the IPC Application Controller. It covers how to operate the functions of the Application Manager that is included in the IPC Application Controller, and how to operate the functions of the Pack Manager that controls the Robot system and the Robot Vision Manager that processes images captured by cameras.

# 5-1 Application Manager Overview

The Application Manager is software that runs on the IPC Application Controller.

It consists of the *Pack Manager* software that controls the pick and place system and the *Robot Vision Manager* software that processes the images captured by the camera connected to the IPC Application Controller. The operation settings of the Application Manager are configured in the Sysmac Studio.

This section describes the overview of the Pack Manager and Robot Vision Manager and how to work with the settings that make up the Application Manager in the Sysmac Studio.

## 5-1-1 Pack Manager

### Introduction

Pack Manager is a mechanism that can build a wide range applications from basic pick-and-place cells to complex pick -and place cells with multiple cameras, conveyors, and robots. You can build applications by using the application sample wizard of the Pack Manager without writing programming codes. For applications that require more complex control, you can achieve them by customizing the default V+ program generated by the application sample wizard as required.

### Characteristics

The robot, which is connected to the Robot Integrated CPU Unit, is controlled by the cooperation of the Robot Integrated CPU Unit and the Pack Manager in IPC Application Controller.

You do not have to write a V+ program for robot control.

The IPC Application Controller performs cell control. And the Robot Integrated CPU Unit controls the equipment that performs the sequence control.

The Pack Manager can perform process management, such as load adjustments, of multiple Robot Integrated CPU Units and OMRON robots that are connected to them.

## 5-1-2 Robot Vision Manager

### Introduction

Robot Vision Manager is a mechanism as follows. It calculates the taking image data of a vision camera, connected to the IPC Application Controller. Then, it processes to generate the output as the vision. In addition, it sets and saves calibration information for the belt (conveyor) and the vision camera for pick-and-place applications, and automatically generates a V+ program that is executed by the Robot Integrated CPU unit.

By using the Robot Vision Manager's application sample wizard, the Robot Vision Manager settings and V+ programs to realize the application are automatically generated. Thus, the load of robot program development can be reduced.

### Characteristics

The Robot Integrated CPU Unit performs the robot control that is connected to the Robot Integrated CPU Unit.

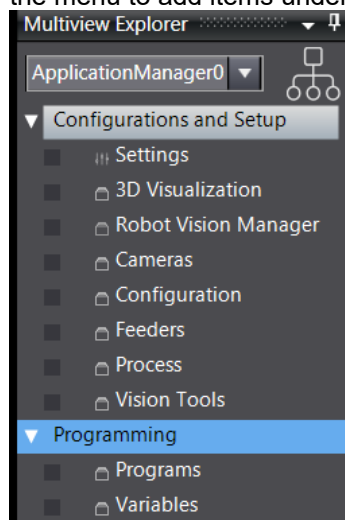
You must write V+ programs for the cell and robot control.

The Robot Integrated CPU Unit controls the cell, the robot and the equipment that performs the sequence control.



## 5-2 Application Manager Settings

The Application Manager has the following functions and settings. You can double-click an item in the Multiview Explorer to edit it, or right-click an item and select **Add** from the menu to add items under it.



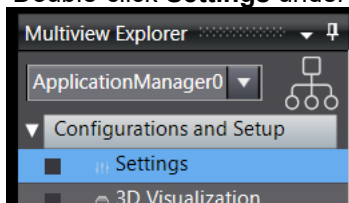
Item	Description	Reference
Settings	Set the IP address and port number of the IPC Application Controller to connect to.	<i>5-3 IPC Application Controller Connection Settings</i>
3D Visualization	Add 3D shape data to the 3D Visualizer for the Robot system.	<i>5-4 3D Visualization</i>
Robot Vision Manager	Add the settings required to use the Robot Vision Manager.	<i>5-5 Robot Vision Manager</i>
Cameras	Add camera that are used in the Robot system. In offline simulations, add virtual cameras in place of the real cameras.	<i>5-6 Camera</i>
Configuration	Customize the behavior of the application with the startup settings of the Application Manager, data mapper and recipe settings, etc.	<i>5-7 Configuration</i>
Feeders	Add feeders to the Robot system that feeds the part.	<i>5-8 Feeders</i>
Process	Configure various process-related settings for picking and placing parts at a target location based on a specified order of execution.	<i>5-9 Process</i>
Vision Tools	Add the tools used for image processing with the Robot Vision Manager.	<i>5-10 Vision Tools</i>

# 5-3 IPC Application Controller Connection Settings

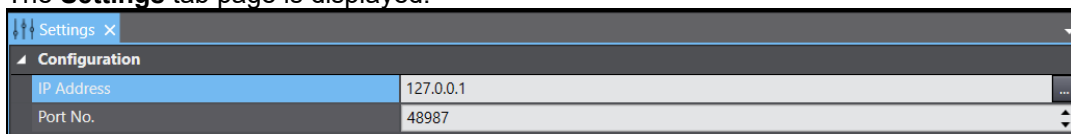
Configure the settings required to go online with the IPC Application Controller.  
Set the IP address and port number of the IPC Application Controller to connect to.

## Displaying the Settings Tab Page

- 1 Double-click **Settings** under **Configurations and Setup** in the **Multiview Explorer**.



The **Settings** tab page is displayed.



## Item

The following items are displayed in the **Settings** tab page.

Item	Description
IP address	Set the IP address of the IPC Application Controller to connect to.
Port No.	Sets the port number for the connection.

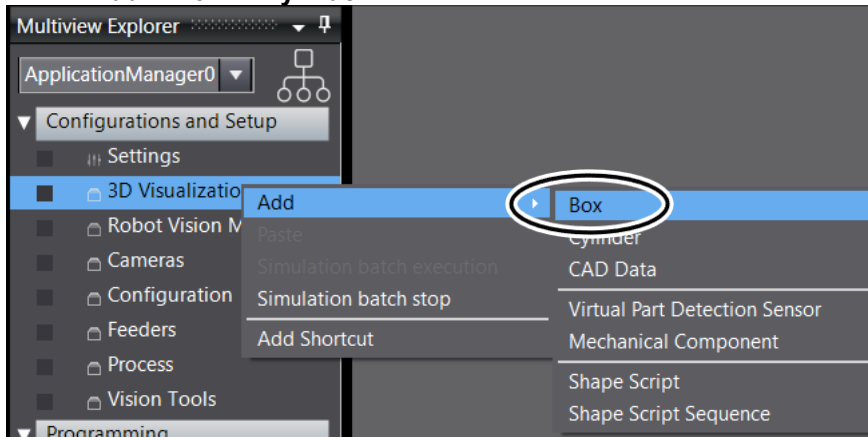
# 5-4 3D Visualization

Add the 3D shape data of parts and peripherals that make up the Robot system to 3D Visualization for the Application Manager.  
Refer to the *Sysmac Studio 3D Simulation Function Operation Manual* (Cat. No. W618) for details on the 3D Visualizer and 3D shape data.

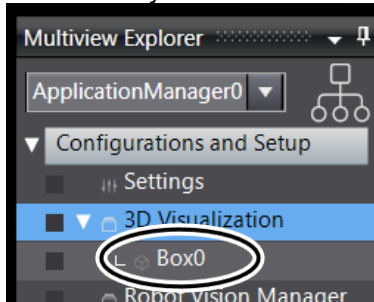
## 5-4-1 Boxes and Cylinders

Create boxes and cylinders that are used as 3D shape data for the part.

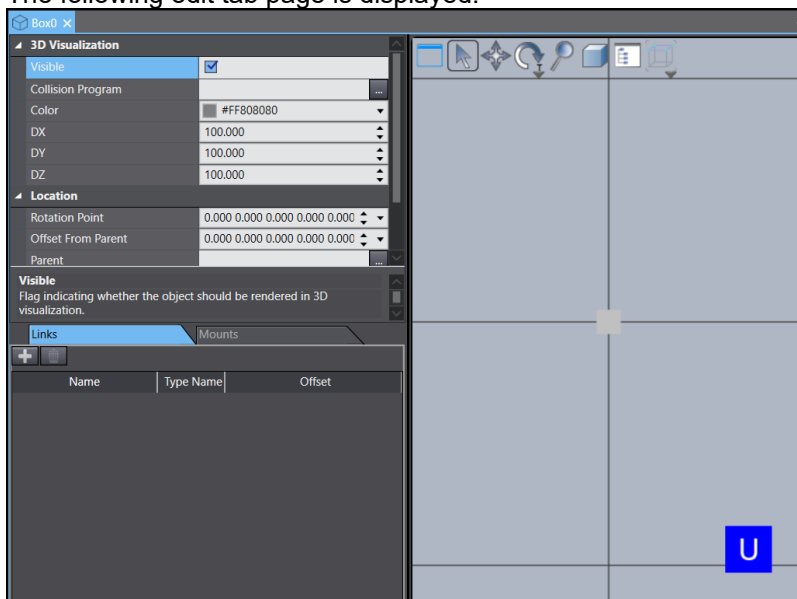
- 1 In the **Multiview Explorer**, right-click **3D Visualization** under **Configurations and Setup** and select **Add – Box** or **Cylinder** from the menu.



A box or a cylinder is added and displayed under **3D Visualization** in the Multiview Explorer.



- 2 Right-click the added box or cylinder and select **Edit** from the menu. The following edit tab page is displayed.

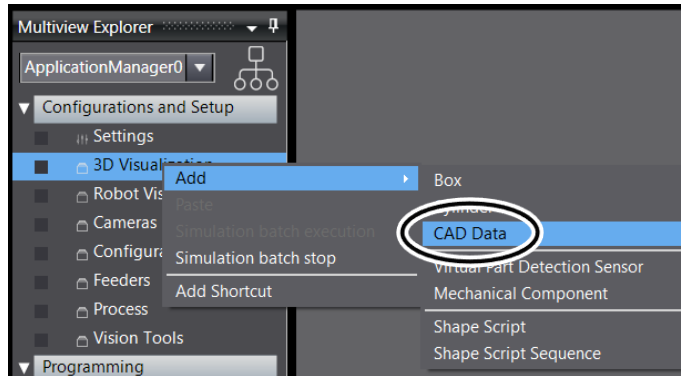


- 3 Configure the settings for the box or cylinder.

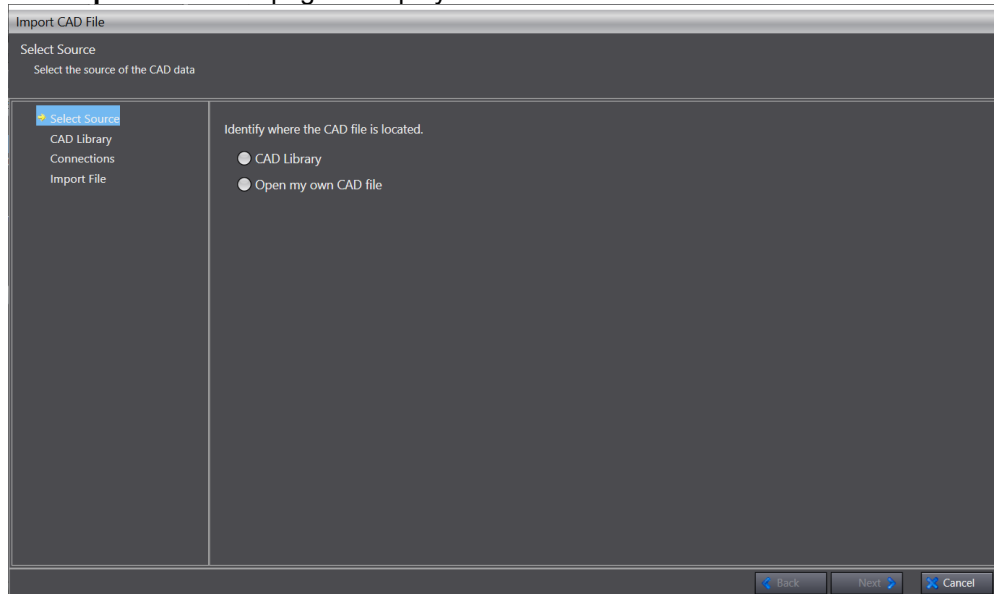
## 5-4-2 CAD Data

Load the CAD data used as 3D shape data such as grippers, tables, frames, parts, and peripherals from 3D CAD data files.

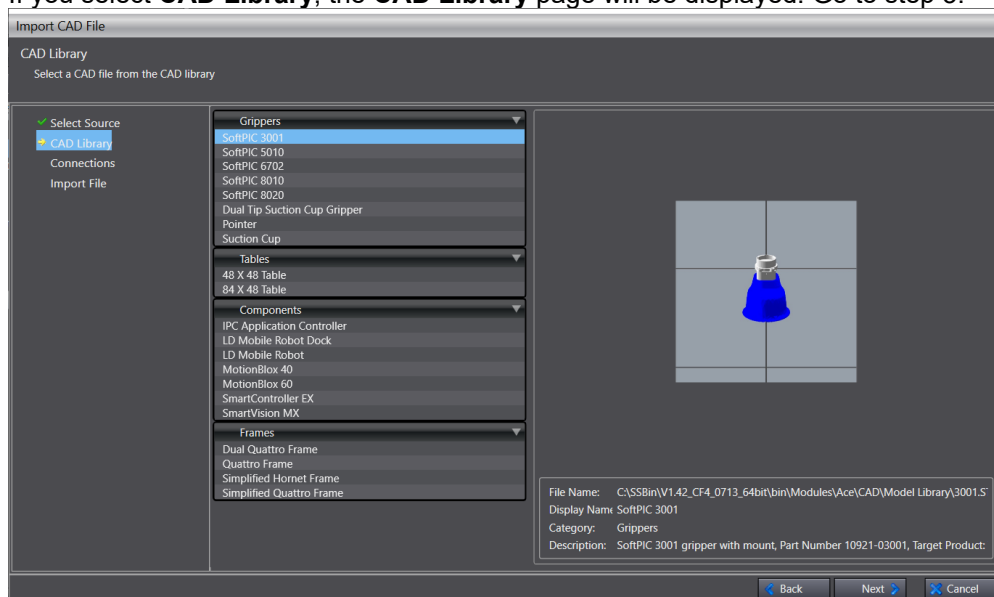
- 1 In the **Multiview Explorer**, right-click **3D Visualization** under **Configurations and Setup** and select **Add – CAD Data** from the menu.



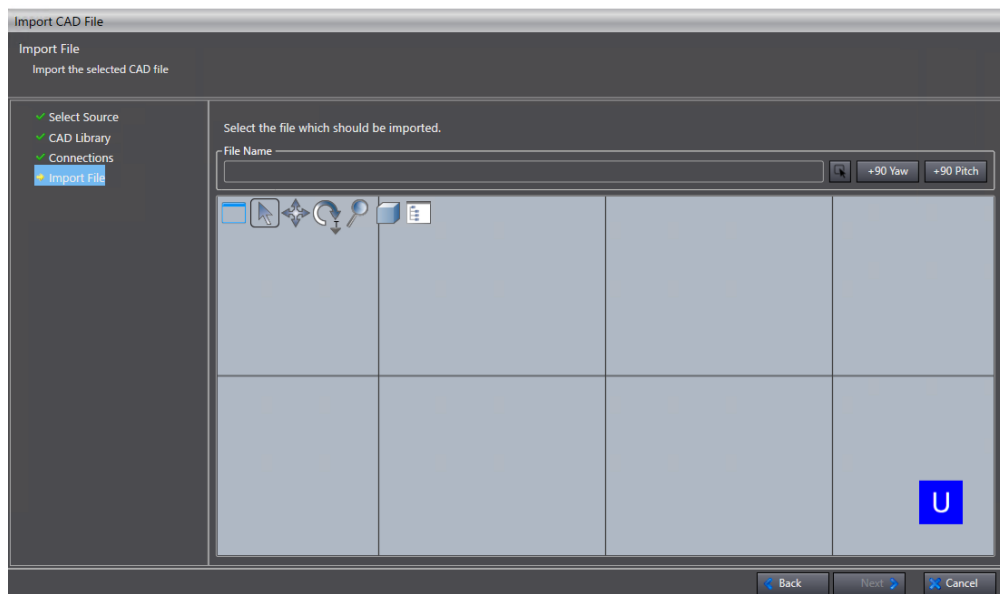
The **Import CAD File** page is displayed.



- 2 Select the CAD file import option, and then click the **Next** button. If you select **CAD Library**, the **CAD Library** page will be displayed. Go to step 3.



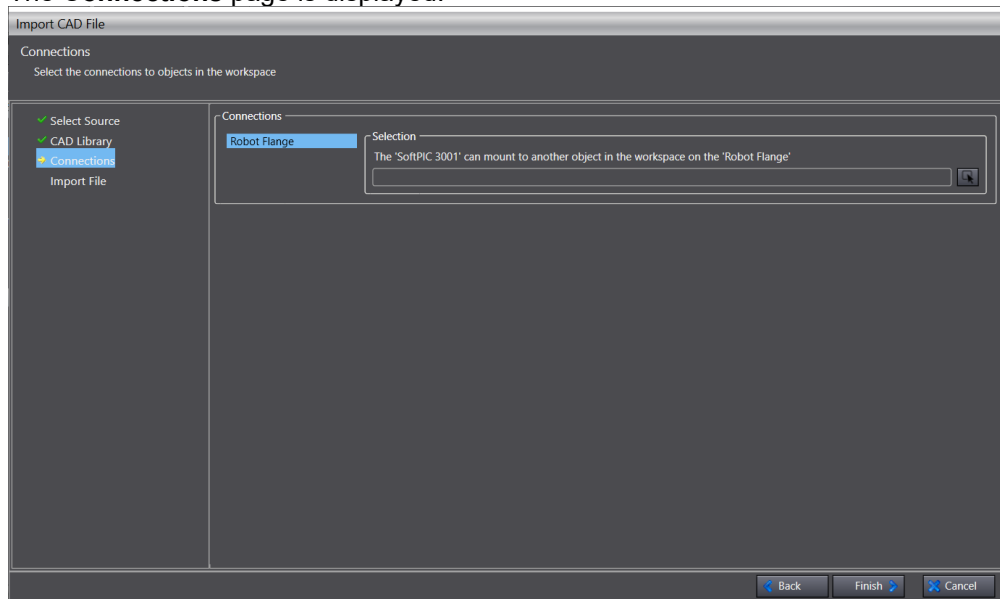
If you select **Open my own CAD file**, the **Import File** page will be displayed.



Select the CAD file to import, and then click the **Next** button. The specified CAD file is imported. The import of the CAD file is now complete.

Refer to the *Sysmac Studio 3D Simulation Function Operation Manual* (Cat. No. W618) for details on the operating procedure when you selected **Open my own CAD file**.

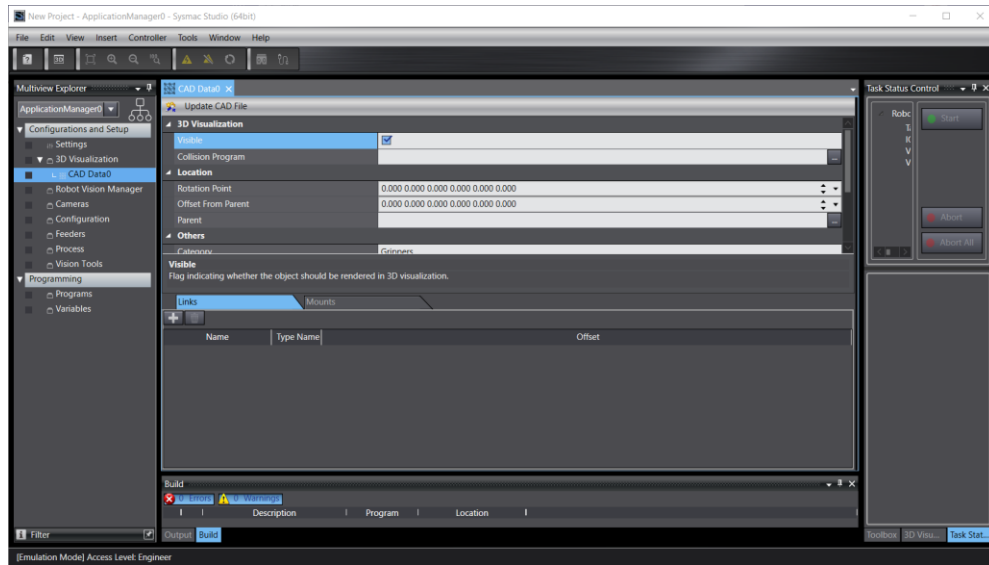
- 3** Select the CAD file to import from the library, and then click the **Next** button. The **Connections** page is displayed.



- 4** To attach 3D shape data from a CAD file to existing 3D shape data, specify the target 3D shape data and click the Finish button.

**CAD Data** is added and displayed under **3D Visualization**.

Various set values are registered under **CAD Data**. You can edit these values. Right-click the **CAD Data** to edit and select **Edit**. The following edit tab page is displayed.



Refer to the *Sysmac Studio 3D Simulation Function Operation Manual* (Cat. No. W618) for details on CAD data items.

# 5-5 Robot Vision Manager

Add the settings required to use the Robot Vision Manager.

## 5-5-1 Robot Vision Manager Settings

The Robot Vision Manager has the following settings.

Item	Description	Reference
Belt Latch Calibration	Set the Robot's behavior with respect to the belt conveyor latch. A sensor detects that a part has reached a specific position on the conveyor belt and generates a latch signal. This latch signal is used to synchronize the positions of the Robot and the part.	5-5-2 <i>Belt Latch Calibration</i>
Belt Calibration	Set the position of the Robot relative to the conveyor belt. Perform a calibration to establish a relationship among the conveyor belt, Encoder, and Robot. This calibration is necessary when the Robot handles parts that move on the conveyor belt.	5-5-3 <i>Belt Calibration</i>
Camera Calibration	Set the position of the Robot with respect to the camera. This calibration is necessary if a vision system is used with the Robot.	5-5-4 <i>Camera Calibration</i>
Gripper Offset Table	Define where on the part the Robot can pick, and define the relationship among the pick spot, the part model, and the Robot's flange center.	5-5-5 <i>Gripper Offset Table</i>
Vision Sequence	Display the execution order and dependencies of the Vision Sensor tools to execute. A means of retrieving results from the Vision Sensor tools is provided to the V+ program. The Vision Sequence displays a list of the Vision Sensor tools that are executed as part of the sequence, the order in which they are executed, and an index that shows the order in which each tool is executed. By executing the Overlap Tool and Communication Tool in the sequence, you can control when the Vision Sensor tools send their processing results to the Controller.	5-5-6 <i>Vision Sequence</i>
Overlap Tool	When the Vision Sensor tool detects a part on the conveyor, you can set it to recognize the moving part only once to avoid picking or processing the same part multiple times per image.	5-5-7 <i>Overlap Tool</i>
Communication Tool	The Communication Tool is called up and used in the Vision Sequence to send the results of the Vision Sensors associated with the conveyor to the Robot Integrated CPU Unit.	5-5-8 <i>Communication Tool</i>

## 5-5-2 Belt Latch Calibration

Set the Robot's behavior with respect to the belt conveyor latch.

Set the Belt Calibration to use for this setting beforehand. Refer to 5-5-3 *Belt Calibration* for details on how to set Belt Calibration.

The setting is made in the following order.

- Adding the calibration settings
- Performing a calibration
- Testing the calibration



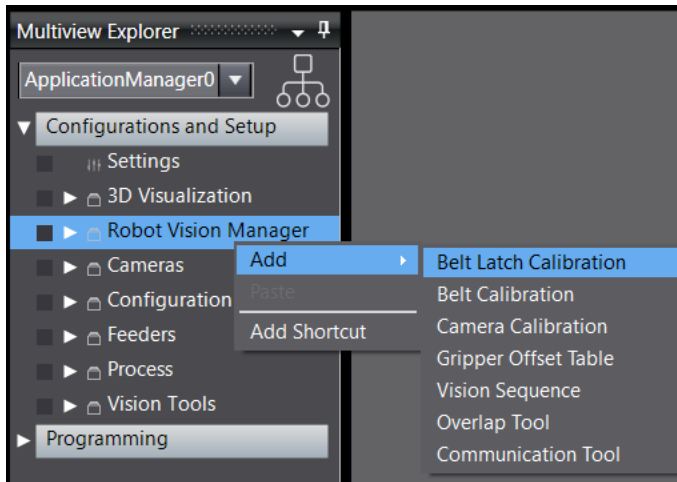
### Precautions for Correct Use

Belt Latch Calibration is only available in Emulation Mode.

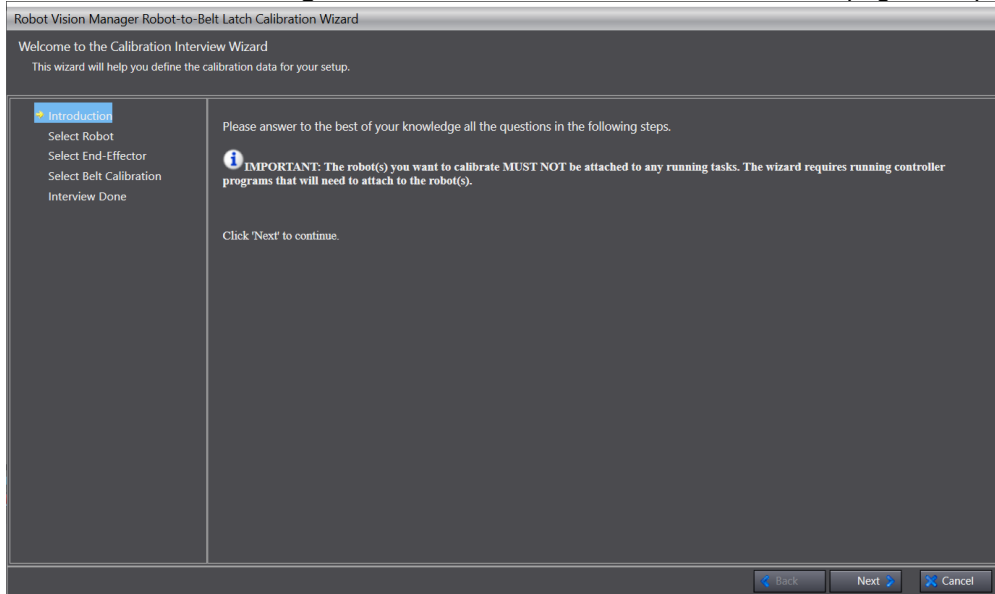
## Adding Belt Latch Calibration Settings

Set the data required to perform a calibration.

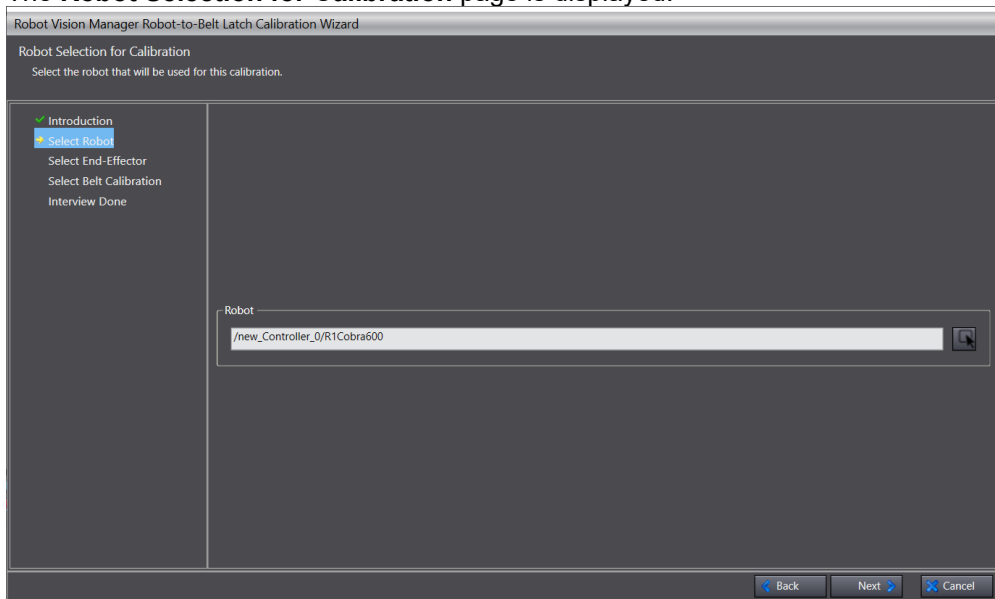
- 1 In the **Multiview Explorer**, right-click **Robot Vision Manager** under **Configurations and Setup** and select **Add – Belt Latch Calibration** from the menu.



The **Robot Vision Manager Robot-to-Belt Latch Calibration Wizard** page is displayed.

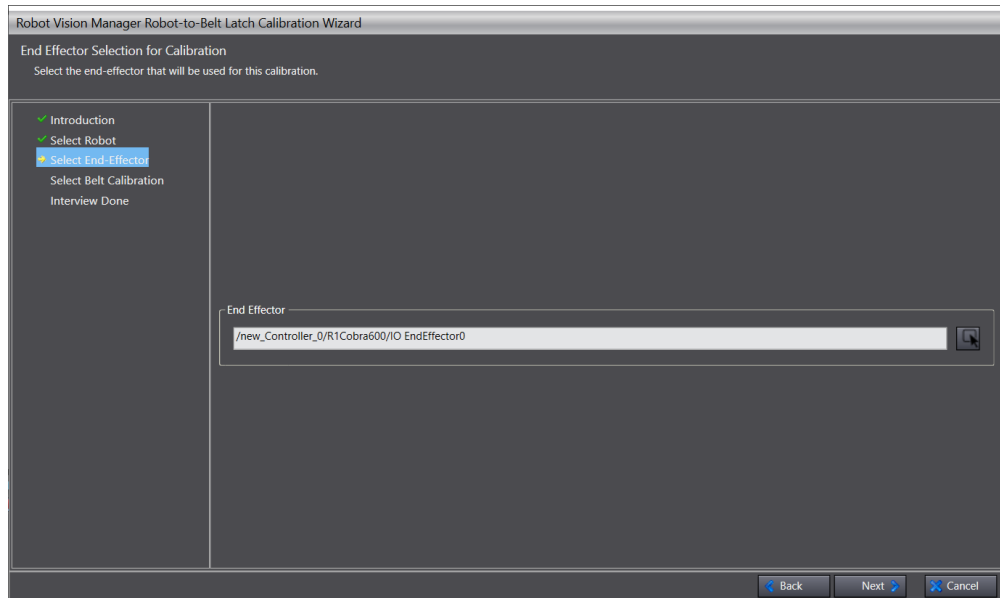


**2** Click the **Next** button.  
The **Robot Selection for Calibration** page is displayed.

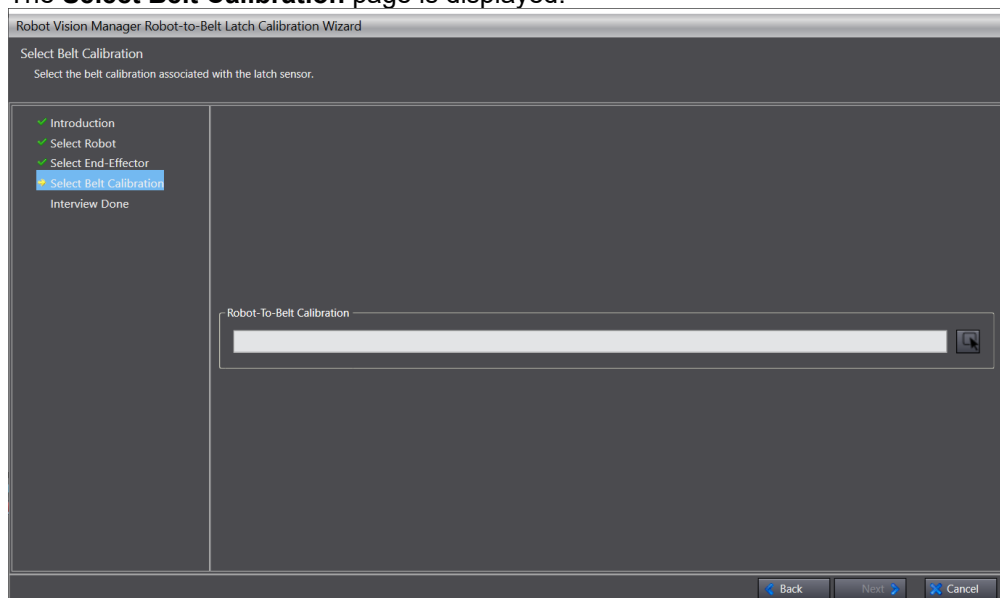


**3** Click the Robot selection button, specify the Robot to use for calibration, and click the **Next** button.  
The **End Effector Selection for Calibration** page is displayed.

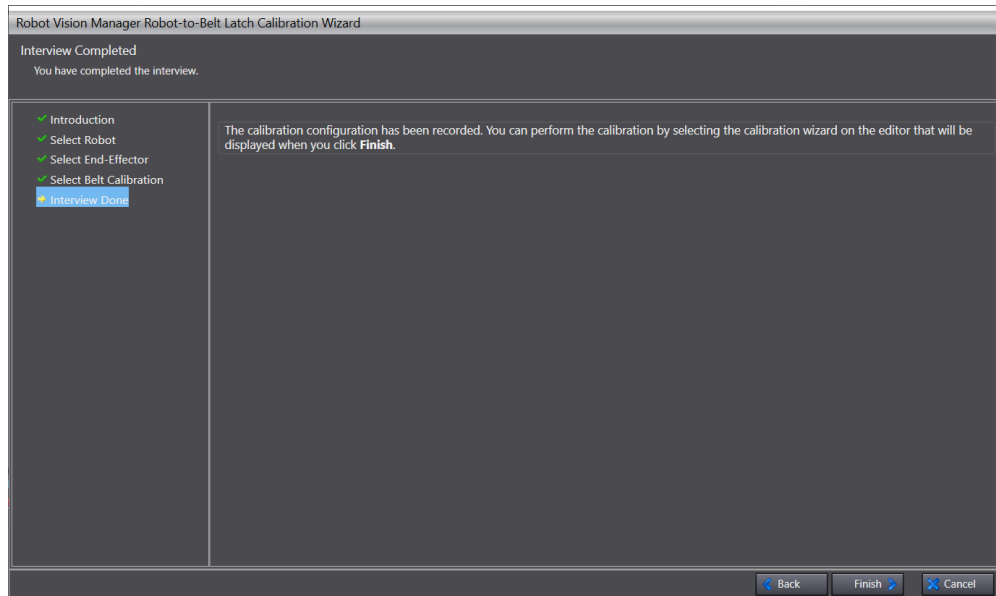




- 4** Click the End Effector selection button, specify the End-Effector to use for calibration, and click the **Next** button.  
The **Select Belt Calibration** page is displayed.



- 5** Click the Belt Calibration selection button, specify the Belt Calibration to associate with the latch sensor, and click the **Next** button.  
The **Interview Completed** page is displayed.

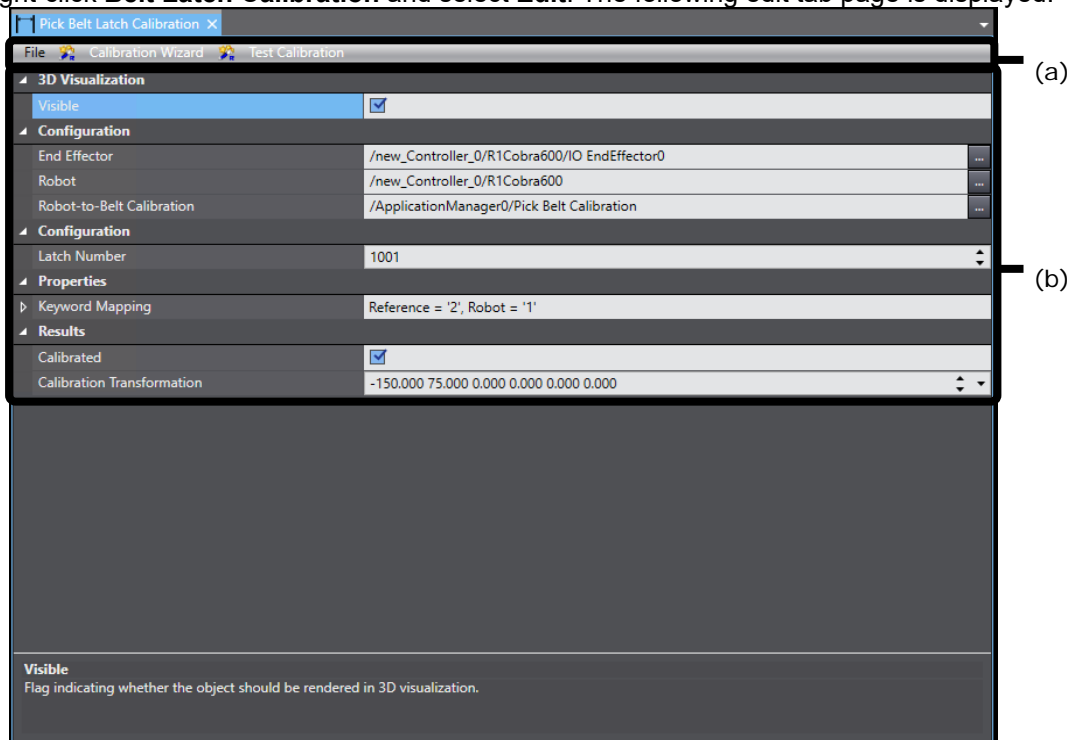


The calibration data setting is now complete.

Click the **Finish** button. **Belt Latch Calibration** is now added and displayed under **Robot Vision Manager**.

## Belt Latch Calibration Settings

Belt Latch Calibration allows you to edit various settings, perform a calibration, and test the calibration. Right-click **Belt Latch Calibration** and select **Edit**. The following edit tab page is displayed.



The items displayed in the edit tab page are as follows.

	Item	Description
(a)	File	After the calibration is complete, you can save the calibration data and load it for later use. <b>Save To:</b> Saves the data to a specified directory. <b>Load From:</b> Loads the saved data.
	Calibration Wizard	Use this button to perform a calibration according to the set conditions. Refer to <i>Performing a Calibration</i> for details on the Calibration Wizard.
	Test Calibration	Use this button to test the calibration according to the set conditions. Refer to <i>Testing the Calibration</i> for details on the calibration test..

	Item		Description
(b)	3D Visualization	Visible	Select this check box to display objects in the 3D Visualizer.
		Configuration	End-Effector
	Robot		Specify the Robot to use for calibration.
	Robot-to-Belt Calibration		Specify the robot-to-belt calibration data to use in the calibration process.
	Latch Number		Specify the latch number to use for calibration.
	Properties	Keyword Mapping	This is how to reference the calibration when the V+Robot Vision Manager keyword is used.
		Robot Index	Use the resultIndex parameter for the V+Robot Vision Manager keyword to pass the Robot index.
		Reference Index	Use the toolID parameter for the V+Robot Vision Manager keyword to pass this index.
	Results	Calibrated	This check box is selected if the robot-to-latch sensor calibration is successfully completed.
		Calibration Transformation	Associate the origin of the robot-to-belt calibration to the origin of the latch sensor.

## Performing a Calibration

You can perform a calibration. To do so, you need to connect to the simulator beforehand.

- 1** In the edit tab page, click the Calibration Wizard button. The **Robot-to-Belt Latch Calibration Sequence** page is displayed.
- 2** Specify the End-Effector to use for the application sample Robot and click the Next button. **Teach the Sensor Offset** page is displayed.
- 3** Teaches the sensor offset. The distance between the sensor and the origin of the belt is the sensor offset. In the 3D Visualizer, click and drag the edit handle to adjust the Belt Calibration. To rotate the calibration, click the edit buttons at the bottom of the 3D Visualizer to manually edit the transformation. When you finished teaching, click the **Finish** button. The calibration is now complete.

## Testing the Calibration

Test the calibration. To do so, you need to connect to the simulator beforehand.

- 1** In the edit tab page, click the Test Calibration button. The **Test Latch Calibration** tab page is displayed.
- 2** Specify the End-Effector tip to use during alignment and click the Next button. The **Position Part** page is displayed.
- 3** Confirm the contents and click the Next button. The **Advance Belt** page is displayed.
- 4** Move the belt so that the Robot reaches the target. Click the button to the right of On/Off to activate the belt.
- 5** When you finished checking, click the Next button. The **Position Robot** page is displayed.
- 6** Click the **Move** button. Confirm that the robot moves, then click the **Next** button. The **Continue testing the calibration** page is displayed.
- 7** Select whether to continue with the test and click the Next button. Select **Continue with the test** to return to the **Position Part** page. Select **Do not continue with the test** to close the page. The calibration is now complete.

### 5-5-3 Belt Calibration

Calibrate the belt that you connected to the Robot and determine the position of the belt with respect to the Sysmac Studio Robot Integrated System Building Function with IPC Application Controller Operation Manual (W621)

Robot.

This calibration is necessary when the Robot handles parts that move on the belt.  
Refer to *3-3-2 Robot-to-Belt Calibration* for technical information on the calibration.  
The setting is made in the following order.

- Adding the calibration settings
- Performing a calibration
- Testing the calibration



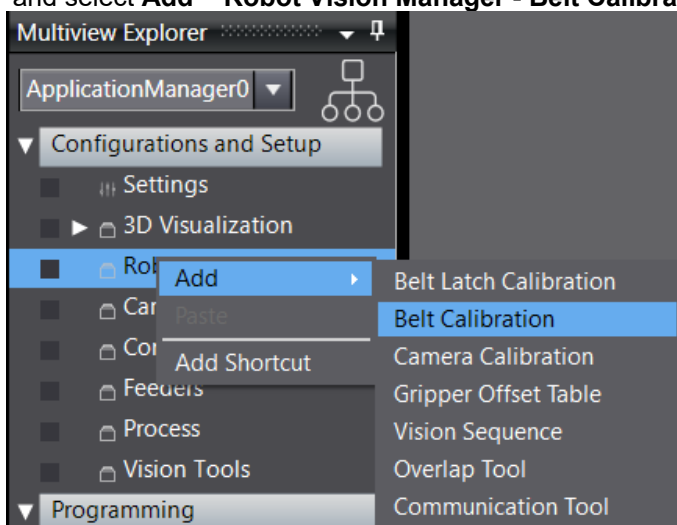
### Precautions for Correct Use

Belt Calibration is only available in Emulation Mode.

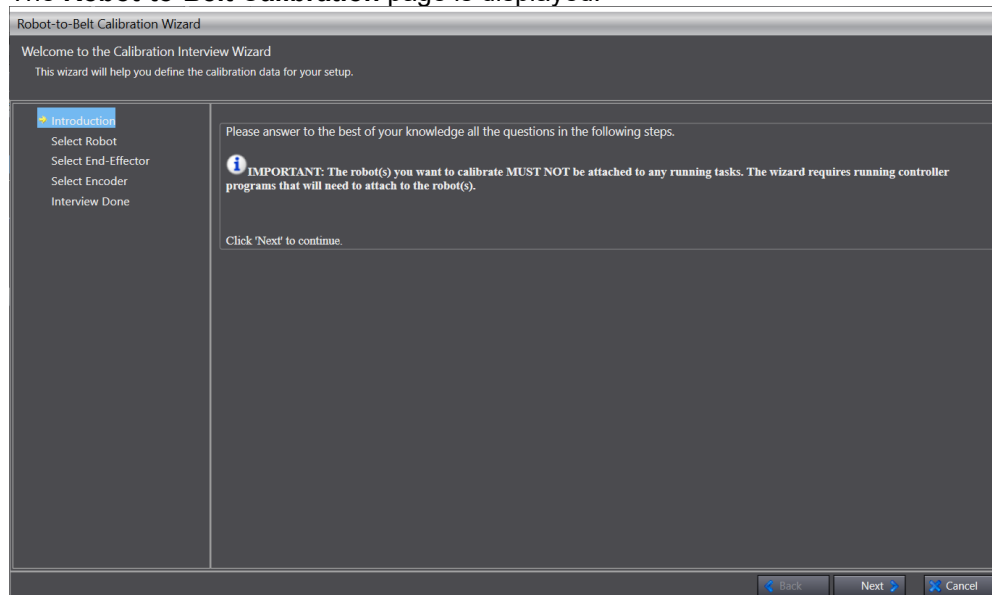
## Adding Belt Calibration Settings

Set the data required to perform a calibration.

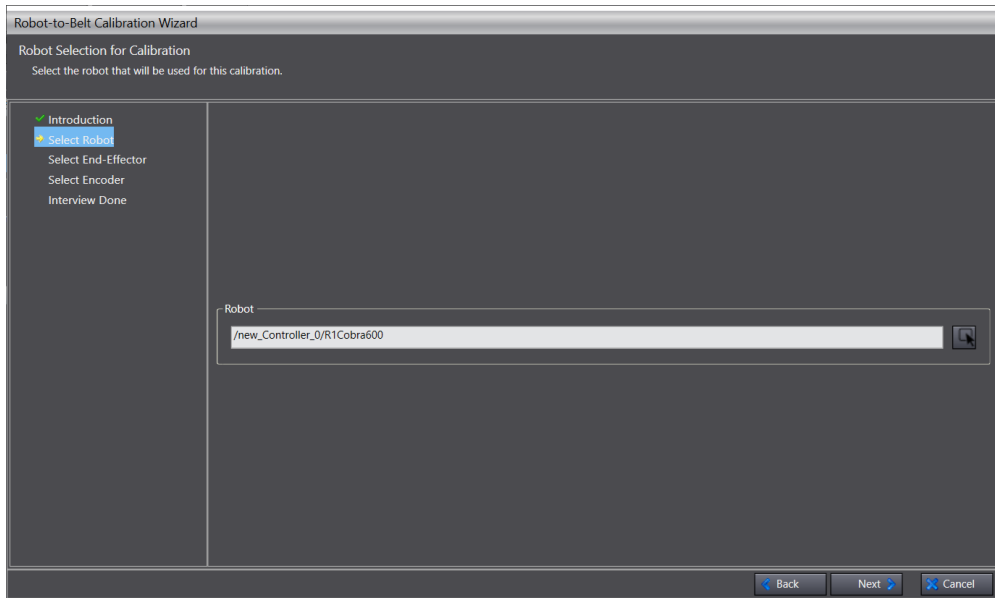
- 1 In the **Multiview Explorer**, right-click **Robot Vision Manager** under **Configurations and Setup** and select **Add – Robot Vision Manager - Belt Calibration** from the menu.



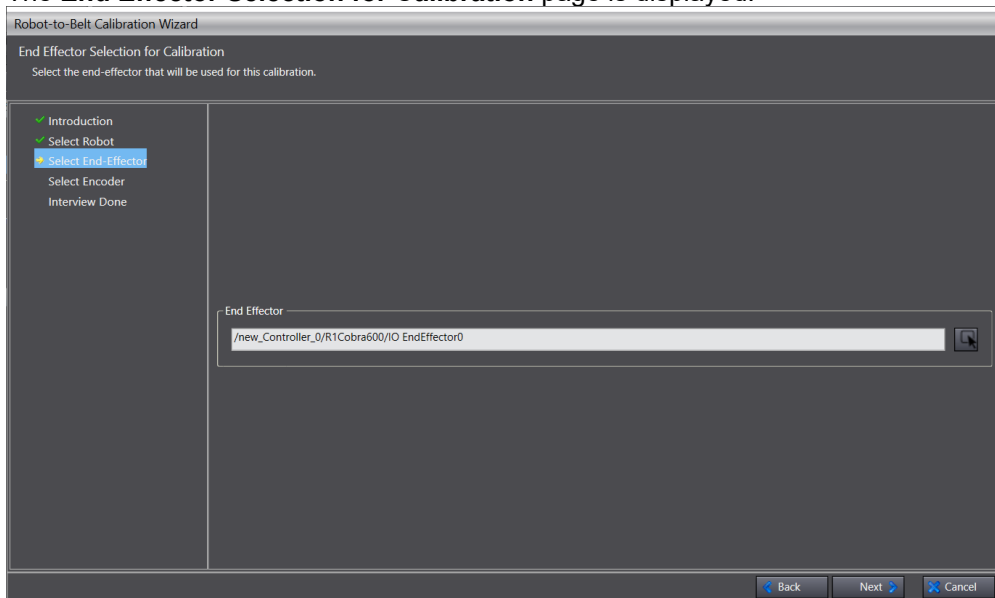
The **Robot-to-Belt Calibration** page is displayed.



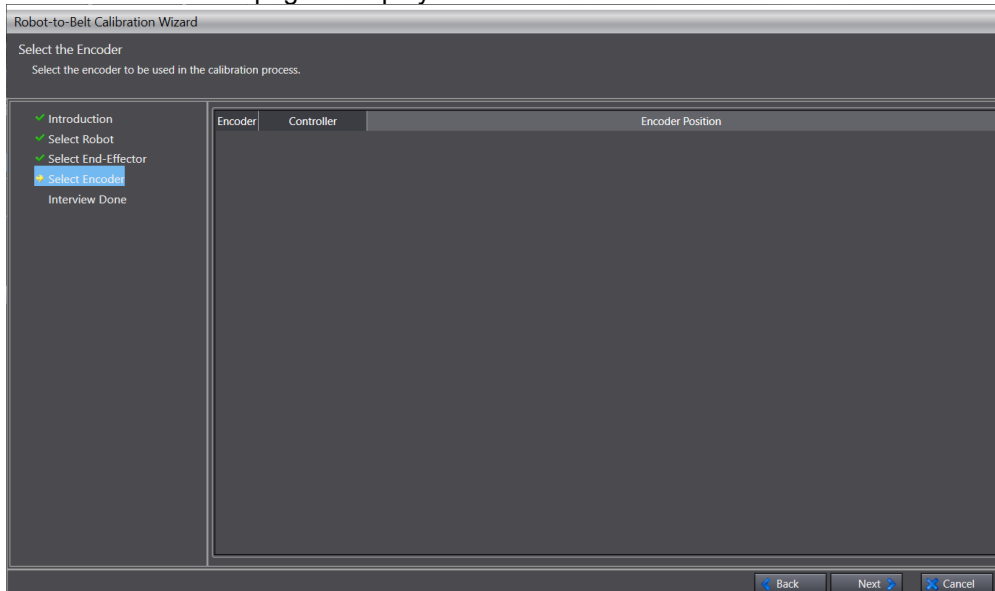
- 2 Click the **Next** button.  
The **Robot Selection for Calibration** page is displayed.



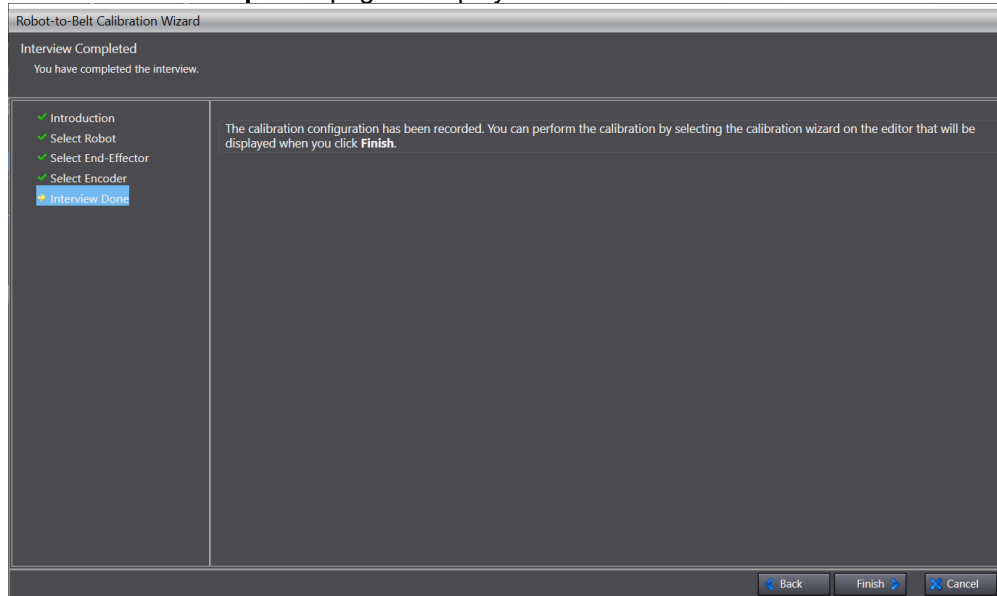
- 3** Click the Robot selection button, specify the Robot to use for calibration, and click the **Next** button.  
 The **End Effector Selection for Calibration** page is displayed.



- 4** Click the End Effector selection button, specify the End-Effector to use for calibration, and click the **Next** button.  
 The **Select the Encoder** page is displayed.



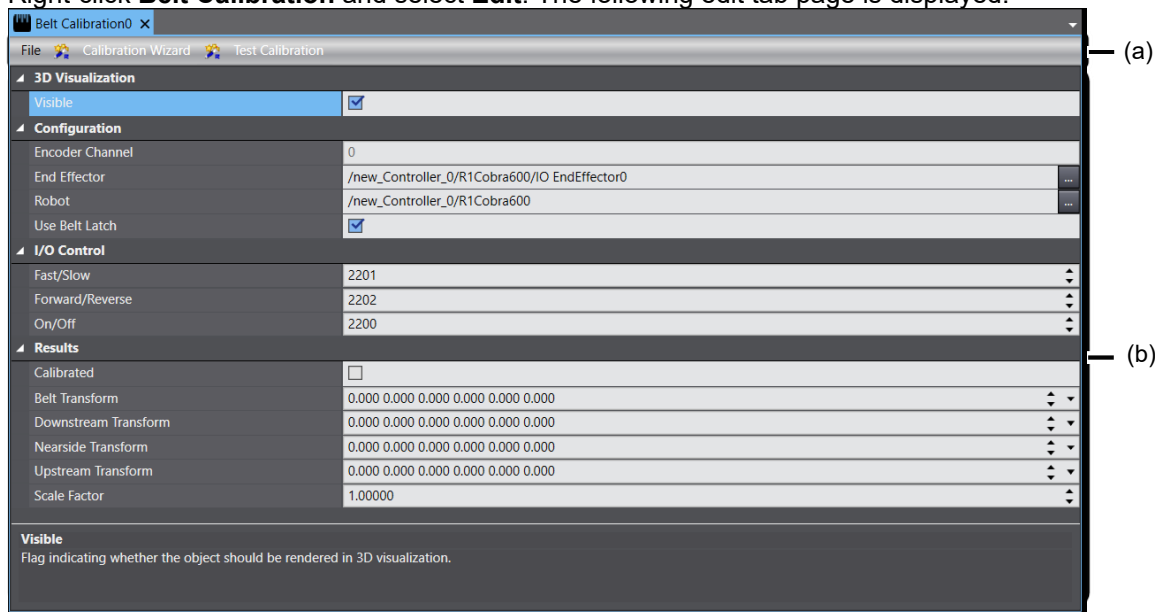
- 5 Specify the Encoder to use for the calibration process and click the Next button. The **Interview Completed** page is displayed.



The calibration data setting is now complete. Click the **Finish** button. **Belt Calibration** is now added and displayed under **Robot Vision Manager**.

## Belt Calibration Settings

**Belt Calibration** allows you to edit various settings, perform a calibration, and test the calibration. Right-click **Belt Calibration** and select **Edit**. The following edit tab page is displayed.



The items displayed in the edit tab page are as follows.

The items displayed vary depending on conditions, such as automatic or manual control, and the presence or absence of a belt. An example is given below.

	Item	Description
(a)	File	After the calibration is complete, you can save the calibration data and load it for later use. <b>Save To:</b> Saves the data to a specified directory. <b>Load From:</b> Loads the saved data.
	Calibration Wizard	Use this button to perform a calibration according to the set conditions. The items displayed in the wizard vary depending on the conditions, such as automatic or manual control and the presence or absence of a belt. Refer to <i>Performing a Calibration</i> for details on the Calibration Wizard.

	Test Calibration		Use this button to test the calibration according to the set conditions. The items displayed in the wizard vary depending on the conditions. Refer to <i>Testing the Calibration</i> for details on the calibration test.
(b)	3D Visualization	Visible	Select this check box to display objects in the 3D Visualizer.
		I/O Control	On/Off
		Fast Speed/Slow Speed	The fast-speed/slow-speed control signal. When the signal is asserted, the belt moves at high speed.
		Forward/Reverse	The forward/reverse control signal. When the signal is asserted, the direction of the belt is reversed.
	Configuration	Encoder Channel	The channel through which the belt encoder is wired to the Robot Controller.
		End-Effector	Specify the End-Effector to use for calibration.
		Use Belt Latch	Select this check box to read out the belt latch when the camera captures an image. Clear this check box if the current position of the belt is used instead of the latch position.
		Robot	Specify the Robot to use for calibration.
	Results	Belt Transform	A robot-to-belt calibration transformation.
		Downstream Transform	The position of the downstream Robot.
		Nearside Transform	The position of the Robot on the near side.
		Upstream Transform	The position of the upstream Robot.
		Scale Factor	The scale factor of the Encoder.
		Calibrated	This check box is selected if the robot-to-belt calibration is completed successfully, .

## Performing a Calibration

You can perform a calibration. To do so, you need to connect to the simulator beforehand.

- 1** In the edit tab page, click the **Calibration Wizard** button.  
The **Robot-to-Belt Calibration Sequence** page is displayed.
- 2** Specify the End-Effector to use for the application sample Robot and click the **Next** button.  
The **Test Encoder Operation** page is displayed.
- 3** Verify the operation of the Encoder by moving the belt.  
Click the button to the right of **On/Off** to activate the belt. When you finished the verification, click the **Next** button.  
The **Teach the belt window** page is displayed.
- 4** Teach the belt.  
In the 3D Visualizer, click and drag the edit handle to adjust the Belt Calibration. To rotate the Belt Calibration, click the edit buttons at the bottom of the 3D Visualizer to manually edit the transformation.  
When you finished teaching, click the **Next** button.  
The **Test Belt Calibration** page is displayed.
- 5** Place the Robot in the target position on the belt and click the **Start Tracking** button.  
The belt moves and the Robot tracks the part on the belt. When you finished testing, click the **Finish** button.  
The calibration is now complete.

## Testing the Calibration

Test the calibration. To do so, you need to connect to the simulator beforehand.

- 1** In the edit tab page, click the **Test Calibration** button.  
The **Test Belt Calibration** page is displayed.

- 2 Place the Robot in the target position on the belt and click the **Start Tracking** button. The belt moves and the Robot tracks the part on the belt. When you finished testing, click the **Finish** button.  
The calibration test is now complete.

## 5-5-4 Camera Calibration

Create a camera calibration object. Calibrate the camera that is connected to the Robot and determine the camera's field of view to the Robot.

The setting is made in the following order.

- Adding the calibration settings
- Performing a calibration
- Testing the calibration



### Precautions for Correct Use

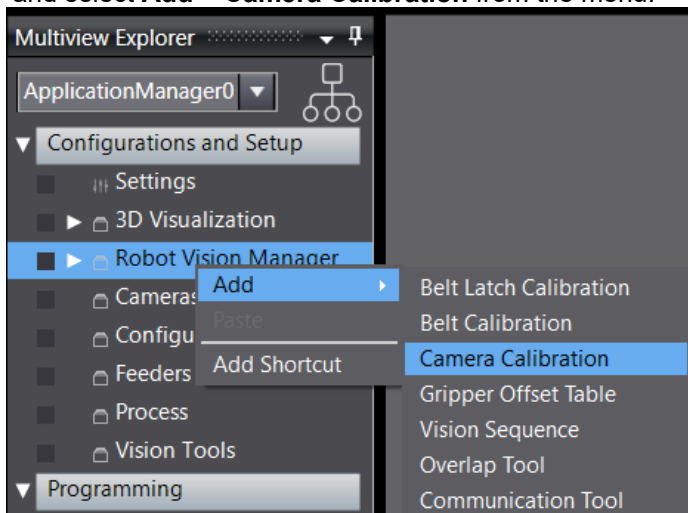
You need to calibrate the camera itself. Refer to 5-6-1 *Virtual Camera* for details.

To use a belt, the belt must be calibrated. Refer to 5-5-3 *Belt Calibration* for details.

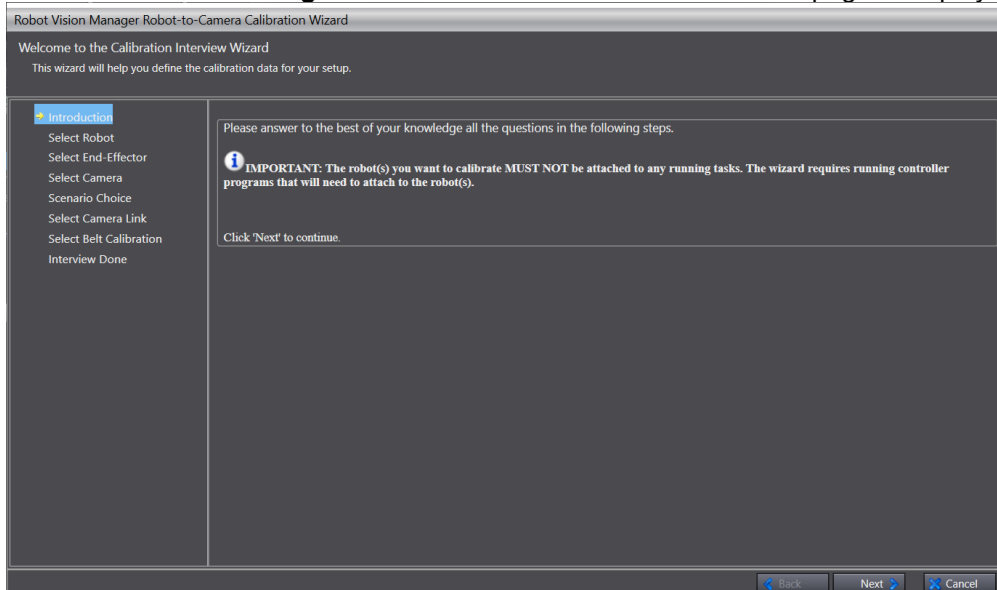
## Adding Camera Calibration Settings

Set the data required to perform a calibration.

- 1 In the **Multiview Explorer**, right-click **Robot Vision Manager** under **Configurations and Setup** and select **Add – Camera Calibration** from the menu.



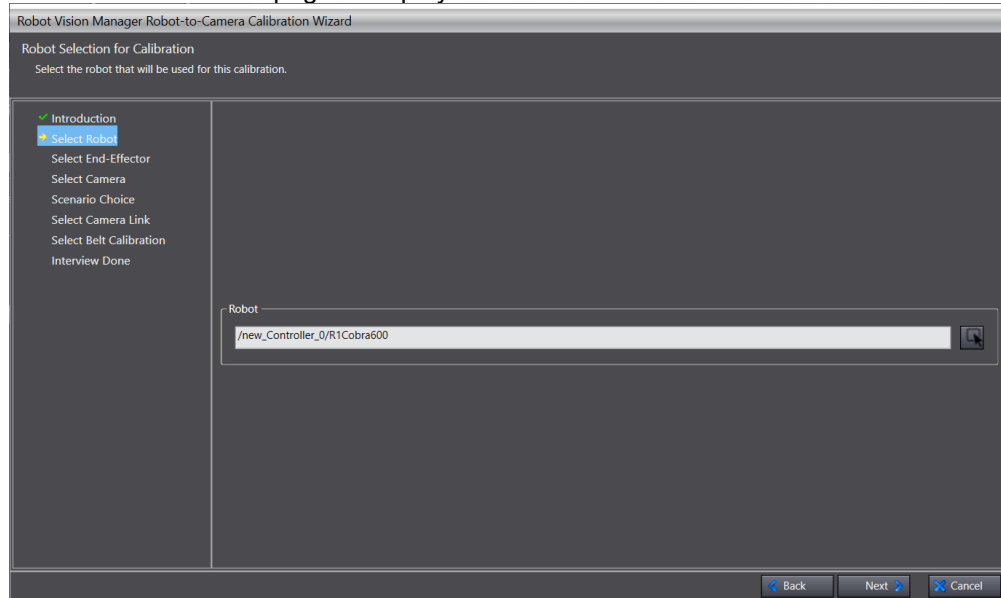
The **Robot Vision Manager Robot -> Camera Calibration Wizard** page is displayed.



- 2 Click the **Next** button.

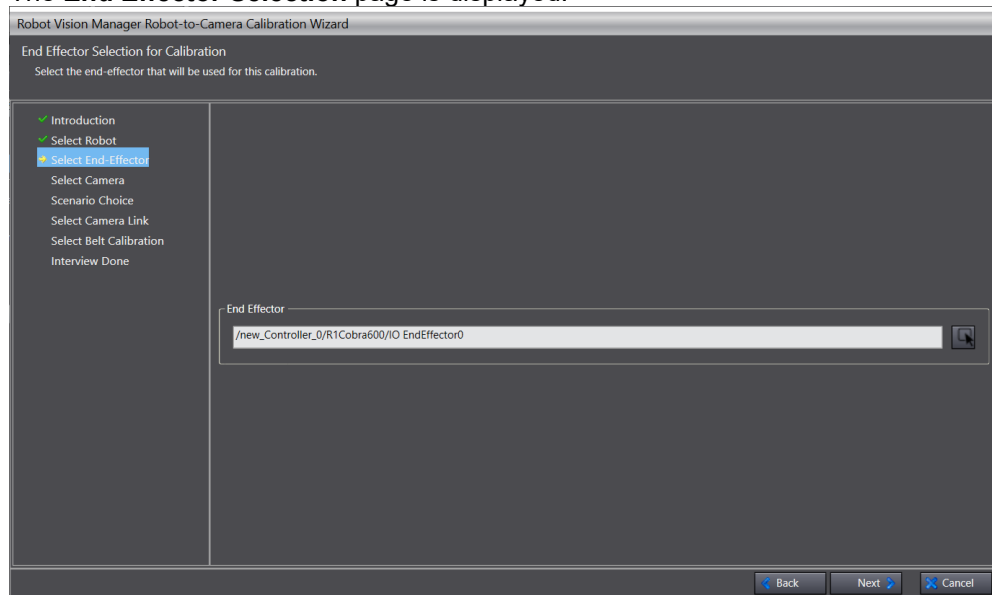


The **Robot Selection** page is displayed.



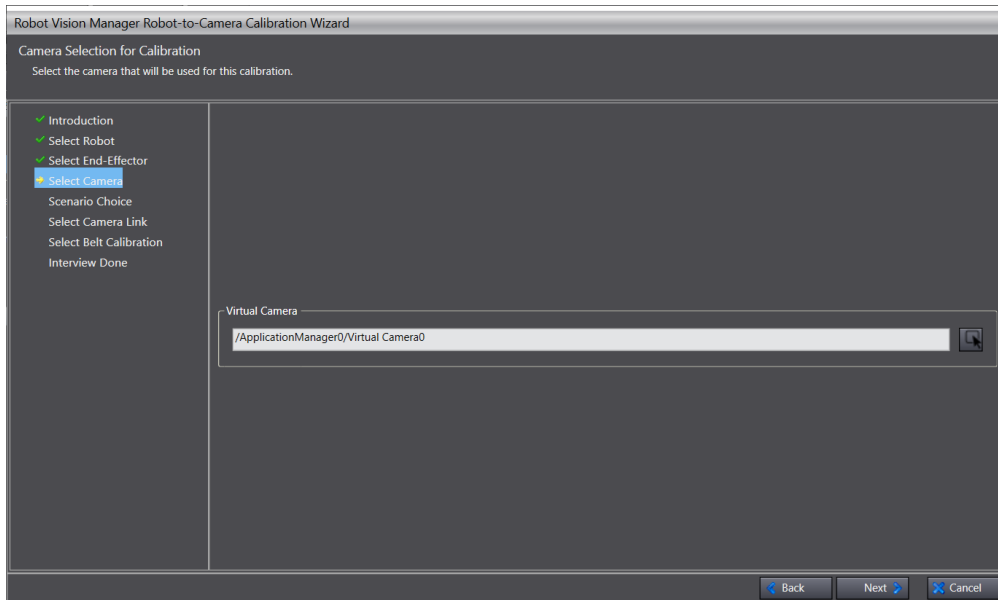
- 3 Click the Robot selection button, specify the Robot to use for calibration, and click the **Next** button.

The **End Effector Selection** page is displayed.

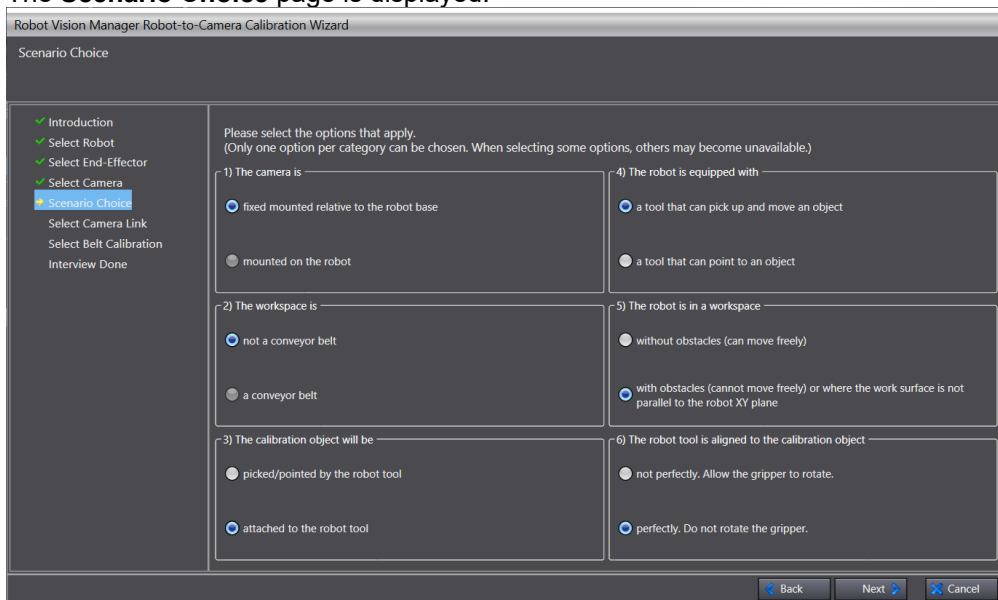


- 4 Click the End-Effector selection button, specify the End-Effector to use for calibration, and click the **Next** button.

The **Camera Selection** page is displayed.



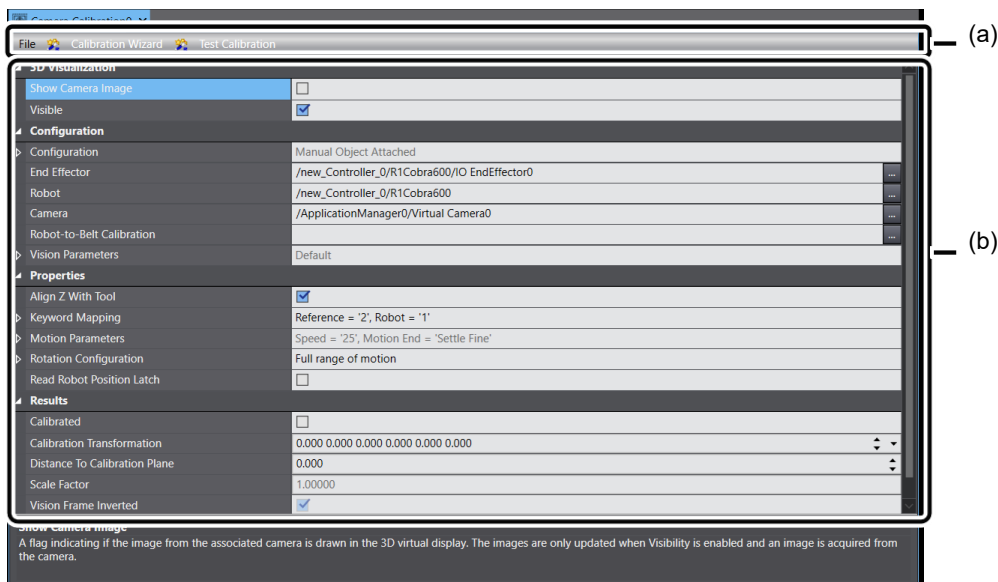
- 5 Click the Camera Selection button, specify the camera to use for calibration, and click the **Next** button.  
The **Scenario Choice** page is displayed.



- 6 Select the options to apply in order from 1 to 6 and click the **Next** button.  
The **Wizard Completed** page is displayed.  
Click the **Finish** button. **Camera Calibration** is now added and displayed under **Robot Vision Manager**.

## Camera Calibration Settings

Camera Calibration allows you to edit various settings, perform a calibration, and test the calibration. Right-click **Camera Calibration** and select **Edit**. The following edit tab page is displayed.



The items displayed in the edit tab page are as follows.

The items displayed vary depending on conditions, such as automatic or manual control, and the presence or absence of a belt. An example is given below.

	Item		Description	
(a)	File		After the calibration is complete, you can save the calibration data and load it for later use. <b>Save To:</b> Saves the data to a specified directory. <b>Load From:</b> Loads the saved data.	
	Calibration Wizard		Use this button to perform a calibration according to the set conditions. The items displayed in the wizard vary depending on the conditions, such as automatic or manual control and the presence or absence of a belt. Refer to <i>Performing a Calibration</i> for details on the Calibration Wizard.	
	Test Calibration		Use this button to test the calibration according to the set conditions. The items displayed in the wizard vary depending on the conditions. Refer to <i>Testing the Calibration</i> for details on the calibration test.	
(b)	3D Visualization	Show Camera Image	Select this check box to display images from the associated camera in the 3D Visualizer.	
		Visible	Select this check box to display objects in the 3D Visualizer.	
	Configuration	End-Effector	Specify the End-Effector to use for calibration.	
		Cameras	Specify the camera to use for calibration.	
		Configuration	Arm Mounted	Select this check box when an arm-mounted camera is connected. Clear this check box to connect a fixed mount camera.
			Attach Object	Select this check box to attach an object to the Robot's tool tip.
		Gripper	Select this check box if the gripper can be rotated during calibration.	
		Joint Number	Specify the joint number of the Robot that is equipped with the camera. You can specify this when an Arm Mounted camera is connected.	
		No Tool Offset	Select this check box if the tool is accurately centered. This is useful if the tool can pick and touch objects exactly at the model's origin with the tool radius regarded as zero.	
Belt	Select this check box to use the belt to perform a calibration.			
Pointer	Select this check box to use the pointer in the calibration			

			process.
		Move Automatically	Select this check box if the Robot can move around the workspace automatically and freely. Clear this check box if to move the Robot manually.
	Robot		Specify the Robot to use for calibration.
	Robot-to-Belt Calibration		Specify the robot-to-belt calibration data.
	Vision Parameters		Specify the Vision Sensor tool to use in the calibration process.
Properties	Align Z Axis to Tool		Select the check box to align the Z axis with the specified tool axis.
	Keyword Mapping		This is how to reference the calibration when the V+Robot Vision Manager keyword is used.
		Robot Index	Use the resultIndex parameter for the V+Robot Vision Manager keyword to pass the Robot index.
		Reference Index	Use the toolID parameter for the V+Robot Vision Manager keyword to pass this index.
	Motion Parameters		The parameters used for the calibration motion.
	Read Robot Position Latch		Select this check box to use the latched Robot position when the camera captures a image. Clear the check box to use the current Robot position. This parameter is only used if an arm-mounted or table-mounted camera is connected.
	Rotation Configuration		The movement range of the theta rotation used in the automatic Calibration Wizard. If the start angle is negative and the end angle is positive, the gripper rotates in a direction of -180 to +180 degrees. If the start angle is positive and the end angle is negative, the gripper rotates in a direction of 0 degrees.
Results	Calibrated		This check box is selected if the Robot-to-vision Sensor calibration is successfully completed.
	Calibration Transformation		Associate the origin of the Robot to the origin of the camera.
	Distance to Calibration Surface		The distance from the camera to the calibration target surface.
	Scale Factor		The scale factor from Vision Sensor coordinates to Robot coordinates.
	Flip Vision Sensor Frame		This check box is selected if the Z axis of the Vision Sensor frame is in the same direction as the Z axis of the Robot flange.

## Performing a Calibration

You can perform a calibration. To do so, you need to connect to the simulator beforehand.

- 1** In the edit tab page, click the **Calibration Wizard** button.  
The **Robot-to-Fixed/Arm Camera Manual Calibration Sequence** page is displayed.
- 2** Specify the End-Effector to use for the application sample Robot and click the **Next** button.  
The **Teach the Camera Location** page is displayed.
- 3** Teach the camera location.  
In the 3D Visualizer, click and drag the edit handle to adjust the Belt Calibration. To rotate the calibration, click the edit buttons at the bottom of the 3D Visualizer to manually edit the transformation.  
When you finished teaching, click the **Finish** button. The calibration is now complete.

## Testing the Calibration

Test the calibration. To do so, you need to connect to the simulator beforehand.

- 1** In the edit tab page, click the **Test Calibration** button.  
The **Calibration Test** page is displayed.

- 2 Specify the End-Effector tip to use during alignment and click the **Next** button.  
The **Move to Image Position** page is displayed.
- 3 Move the Robot to the image position and click the **Here** button.
- 4 When you finished moving the Robot, click the **Next** button.  
The **Teach the Model** page is displayed.
- 5 Detect the position of the model.
  - 1) Place the target part in the center of the registration area and click the **Execute** button.
  - 2) Move and adjust the green frame to specify the size and position of the area.
  - 3) Click the **Center** button to place the center point.
- 6 When you finished teaching, click the **Next** button.  
The **Detect Object Position** page is displayed.
- 7 Set the target area to an area that the Robot can access and click the **Next** button.  
The **Position Robot** page is displayed.
- 8 Align the Robot with the part and click the **Next** button.  
The **Continue with the test** page is displayed.  
Select **Continue with the test** to return to the **Move to Image Position** page. Select **Do not continue with the test** to close the page.

The calibration is now complete.

### 5-5-5 Gripper Offset Table

Define 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, and set them as the offset table.

A gripper can have two types of offsets:

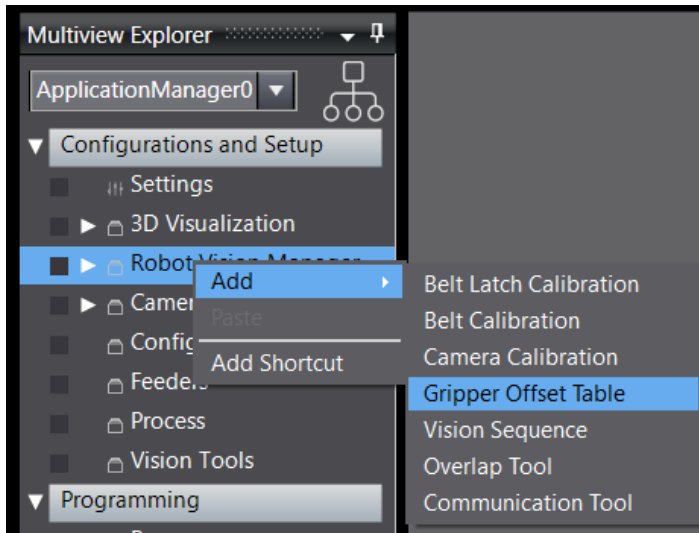
- A tip offset from the center of the tool flange to the gripper tip(s).  
This offset is established when the robot end effector is defined. This offset will be applied to that gripper whenever it is used, including calculating values for the Gripper Offset Table. Refer to *4-4-1 Setting up IO EndEffectors* for more information.
- The offset(s) from the actual pick point to the part origin  
It (They) 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.

Follow the procedure below to set.

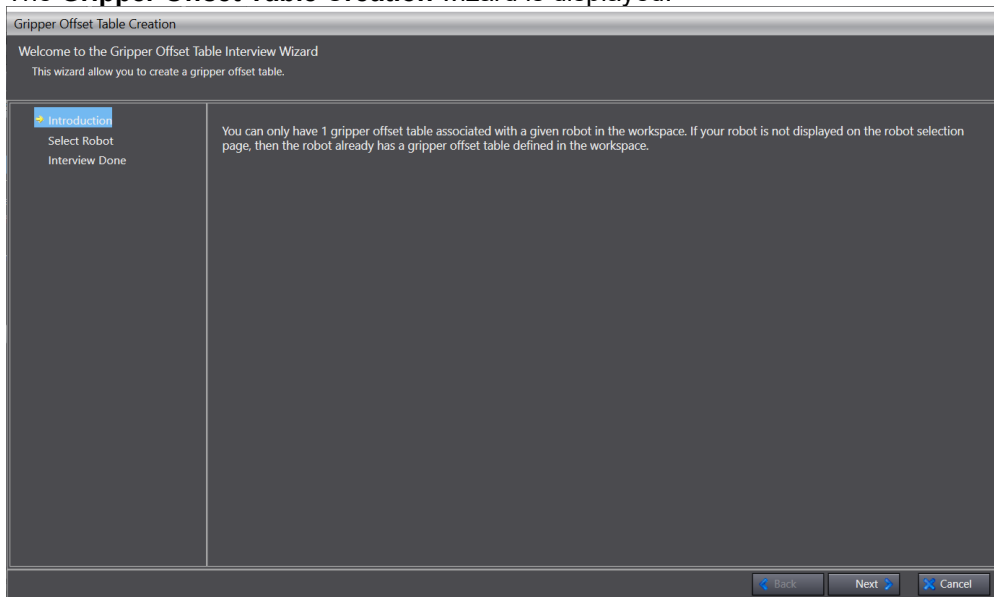
- Add a Gripper Offset Table
- Add a Gripper Offset Index
- Teach

## Adding a Gripper Offset Table

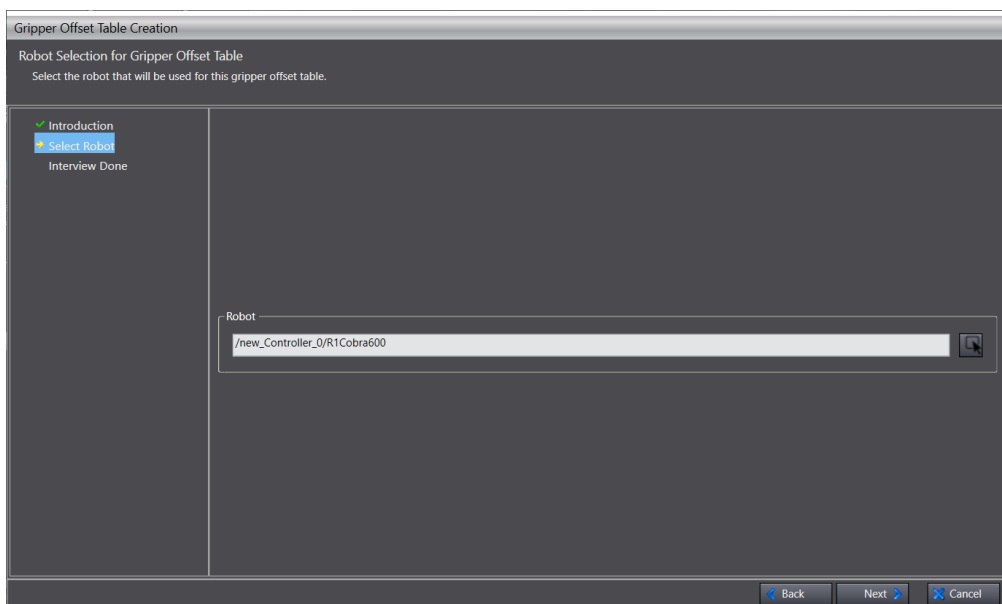
- 1 Right-click **Robot Vision Manager** under **Configurations and Setup**, and select **Add - Gripper Offset Table**.



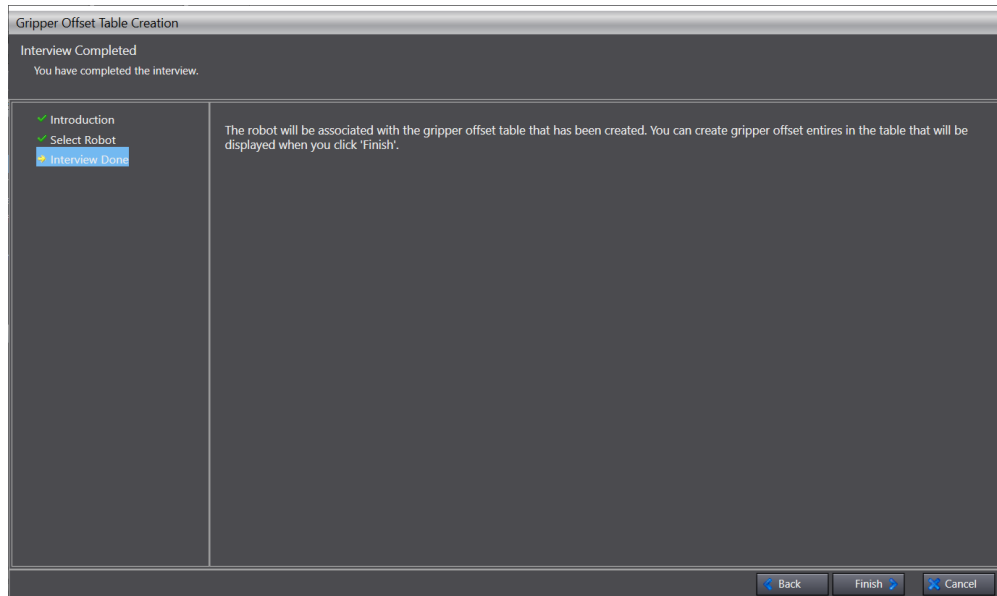
The **Gripper Offset Table Creation** wizard is displayed.



- 2 Click the **Next** button.  
The **Select Robots** page is displayed.



- 3 Specify the robot for the Gripper Offset Table, and then click the **Next** button.  
The **Interview Completed** page is displayed.



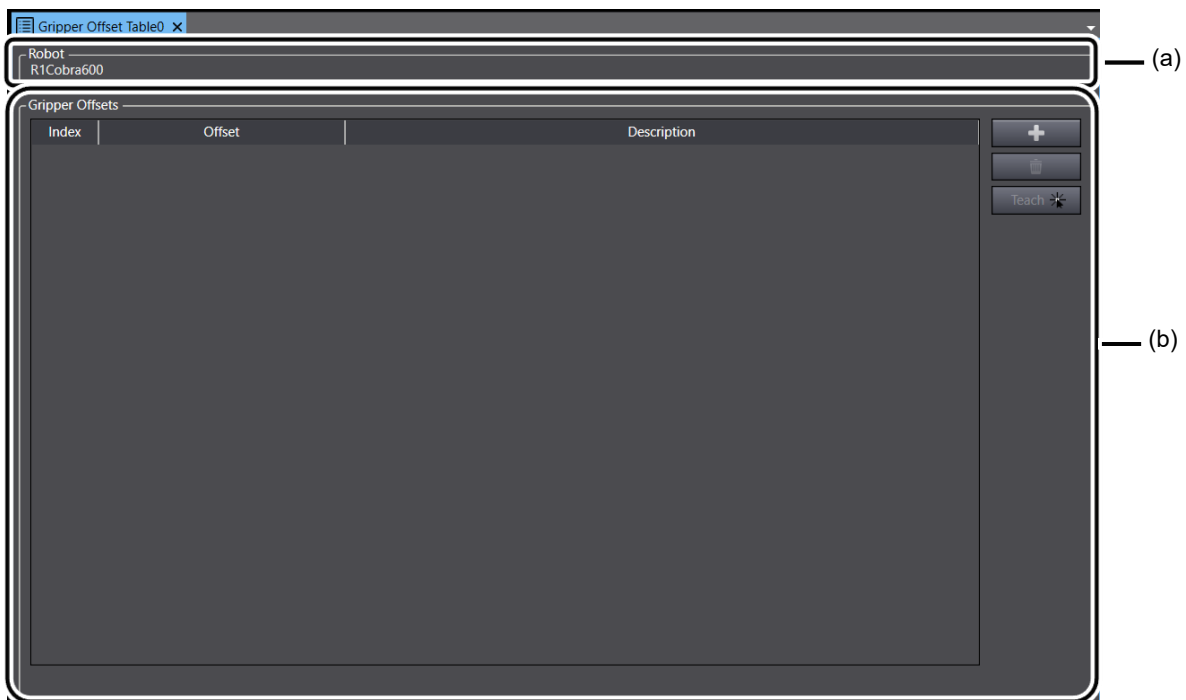
The robot is associated with the Gripper Offset Table created.

Click the Finish button, and then Gripper Offset Table is added under Robot Vision Manager.

## Setting Items of Gripper Offset Table

Various setting values are registered in **Gripper Offset Table**. You can edit the values.

Right click the target **Gripper Offset Table**, select **Edit**, and then the edit page is displayed.



The setting items on the edit pane are listed in the following table.

	Item	Description
(a)	Robot	Display the robot assigned to the Gripper Offset Table.
(b)	Gripper Offsets	Click the <b>Add</b> button to add a new offset. Click the <b>Delete</b> button to delete an existing offset. Change the value directly or click the <b>Teach</b> button to set the Gripper Offset. Refer to <i>Teach</i> for more information.
	Index	Enter a unique value to reference the Gripper Offset values.
	Offset	Enter gripper offset values expressed as (x, y, z, yaw, pitch, roll).
	Description	Enter a description for the Gripper Offset index entry.

## Teach

Set the Gripper Offset.

Click the **Teach** button, and then, the Gripper Offset Teach Wizard opens and guides you through several steps to teach an Offset value. Consider the following objects that may exist in the project.

- Specific end effectors
- Cameras and Camera Calibrations
- Vision Tools
- Belts and Belt Calibrations

In the Locate Object step of the Gripper Offset Teach Wizard, the **Next** button will not be available if the selected Vision Tool is not configured properly. For example, if a location is not obtained from the Vision Tool, you cannot proceed.

## 5-5-6 Vision Sequence

A Vision Sequence let you see 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 vision 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.



### Precautions for Correct Use

You need to register the tools to be used in advance.

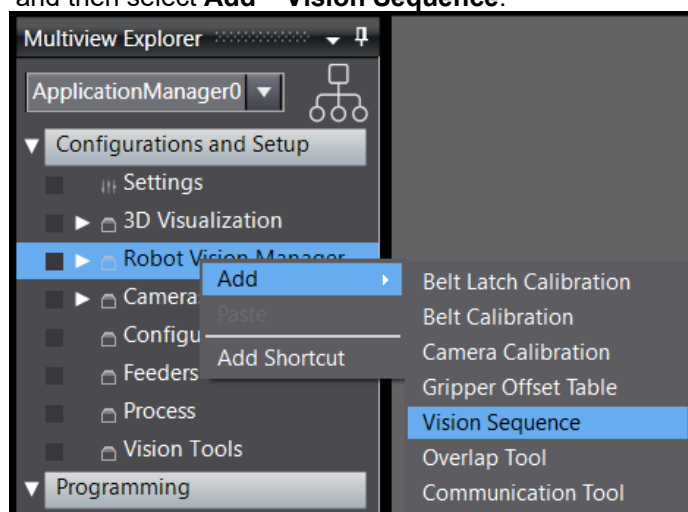


### Additional Information

You must set the **Sequence number** and **Index** when acquiring the results of the Images sensor with V+ programs.

## Adding Vision Sequence

- 1 Right-click **Robot Vision Manager** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Vision Sequence**.

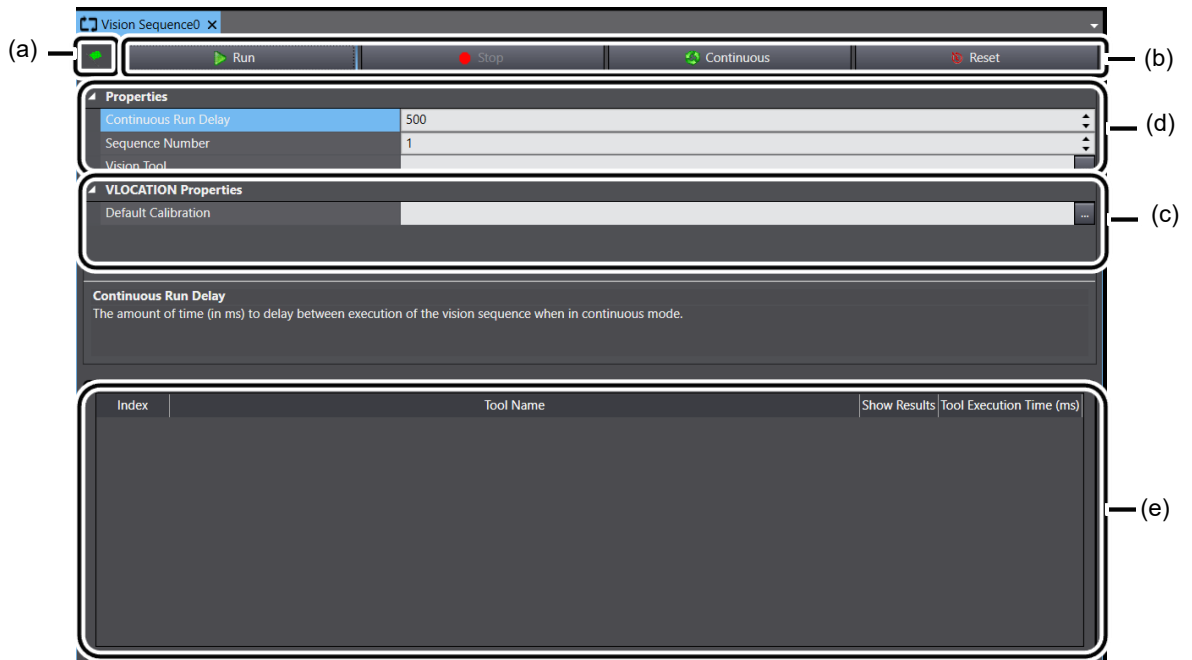


**Vision Sequence** is added under **Robot Vision Manager**.

- 2 Right-click **Vision Sequence**, and then select **Edit**.



The edit tab page is displayed.



The setting items on the edit pane are listed in the following table.

	Item	Description	
(a)	Configuration Status	A red flag indicates incomplete configuration. A green flag indicates the configuration is completed. The configuration status will update after the <b>Run</b> button is clicked. Hover over the red flag for information about the incomplete configuration.	
(b)	Sequence Control Buttons	Run	Run the sequence once.
		Stop	Stop the running sequence.
		Continuous	Run the sequence repeatedly.
		Reset	Reset the communication queue and the Overlap tool tracking of previously-located instances.
(c)	VLOCATION Properties	Default Calibration	Specify the default camera calibration used to apply to a VLOCATION transformation function from V+ programs.
		Vision Tool	Specify the top level vision tool the sequence references. The tool specified at the top is executed last.
		Sequence Number	The sequence number of the specified vision tool.
		Continuous Run Delay	Specify the time to delay the sequence execution interval in continuous mode. (Unit: ms)
(d)	Properties	Sequence Number	The number associated with the sequence. The sequencenumber is used in V+ programs to refer to this vision behavior.
		Vision Tool	The top level vision tool the sequence references. All the tools needed to operate the tool selected here are included in this sequence.
		Continuous Run Delay	The time to delay the sequence execution interval in continuous mode. (Unit: ms) This is the time interval to run the vision sequence for the V+ conveyor tracking program using Robot Vision Manager.
(e)	Sequence	Display the vision sequence for the specified Vision Tool.	

## Setting Vision Sequence

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 are shown on the edit pane. 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.

## 5-5-7 Overlap Tool

When taking images of the part moving on the conveyor by a camera to detect the part on a with the Vision Tool, the same part may exist in multiple Images. Set the Overlap Tool to recognize the moving part only once so that you do not pick or process the same part multiple times per images.

A part found by the Locator Tool (or other Vision Tool) 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.

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 5-5-6 *Vision Sequence* for more information.

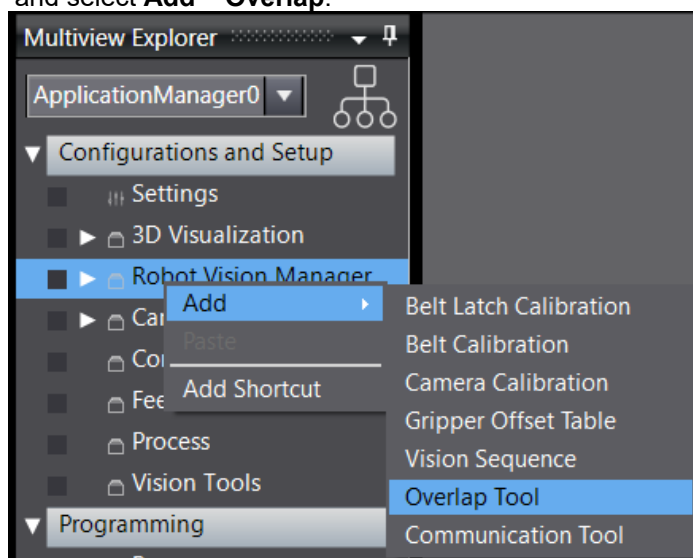


### Additional Information

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.

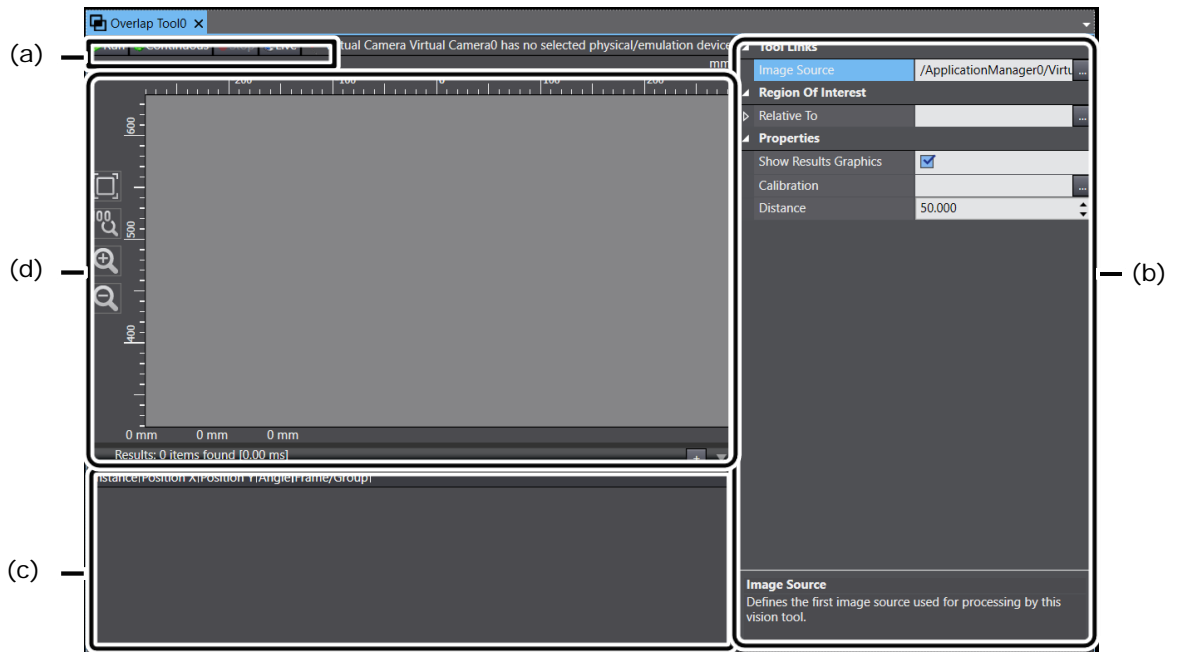
## Adding Overlap Tool

- 1 Right-click **Robot Vision Manager** under **Configuration and Setup** in the Multiview Explorer and select **Add – Overlap**.



**Overlap Tool** is added and displayed under **Robot Vision Manager**.

- 2 Right-click **Overlap Tool**, and then select **Edit** from the menu. The edit tab page is displayed.



The setting items on the edit pane are listed in the following table.

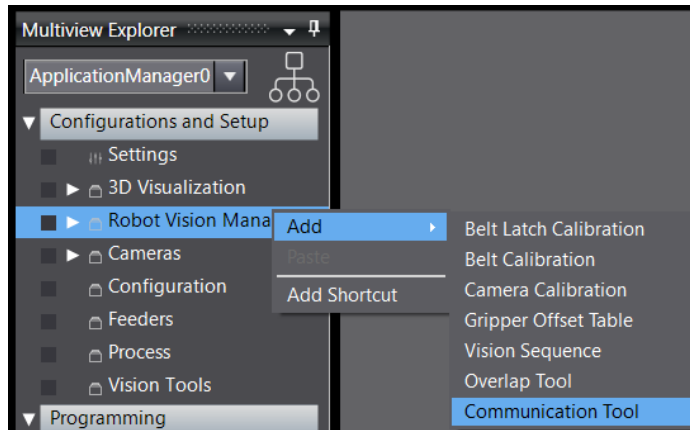
		Item	Description	
(a)		Run	Run the tool once.	
		Continuous	Run the tool continuously.	
		Stop	Stop the running tool or <b>Live</b> .	
		Live	Display input image without running the tool.	
(b)	Configu- ration	Tool Links	Image Source	Specify the Image Source for processing this tool.
			Properties	Calibration
		Distance		Specify the required minimum distance between parts in mm. 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.
		Show Results Graphics	Select this check box to display the image in the Vision Tool Window.	
	Region of Interest	Relative To	Specify the tool to input to this tool.	
		Link Name	Display the name of the linked property.	
(c)	Results	Display the results.		
(d)	Vision Window	Display the image specified in Image Source.		

- 3 Set each item.  
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.  
Refer to 5-5-6 *Vision Sequence* for more information on how to set Vision Sequence.

### 5-5-8 Communication Tool

Communication Tool is called and used in the vision sequence to send the results of the belt-related-vision to the Robot Integrated CPU Units. The It typically receives instances from an Overlap Tool, which prevents different images of the same instance from being interpreted as different instances.

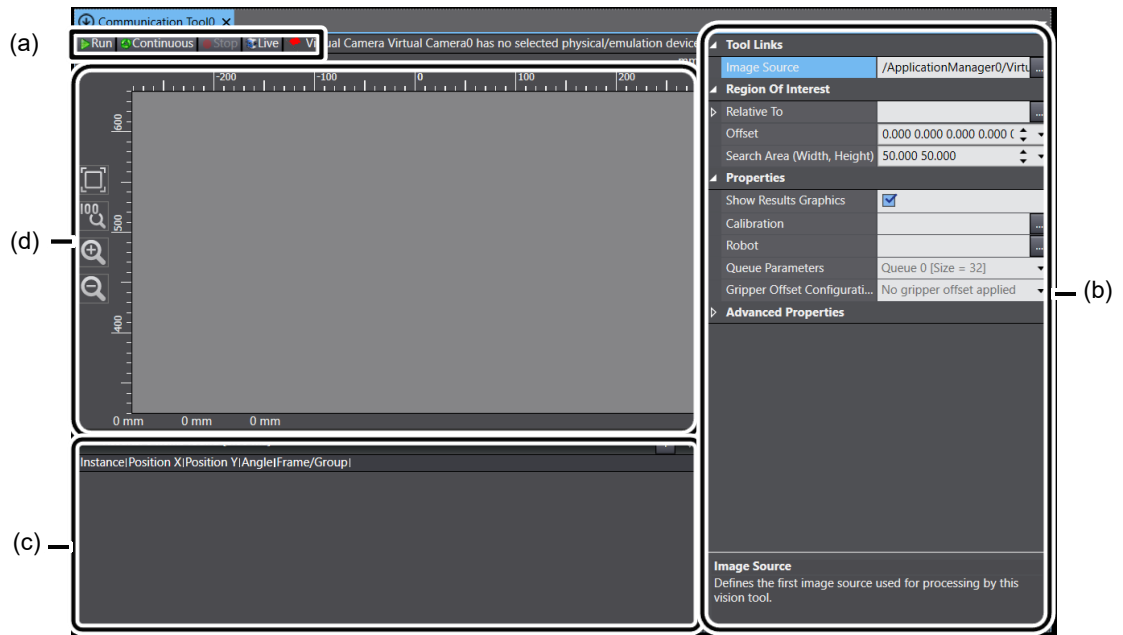
- 1 Right-click **Robot Vision Manager** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Communication Tool**.



A **Communication Tool** is added under **Robot Vision Manager**

- 2** Right-click **Communication Tool**, and then select **Edit**.  
The edit tab page is displayed.

The setting items on the edit pane are listed in the following table.



	Item		Description	
(a)		Run	Run the tool once.	
		Continuous	Run the tool continuously.	
		Stop	Stop the running tool or <b>Live</b> .	
		Live	Display input image without running the tool.	
(b)	Configu- ration	Tool Links	Image Source Specify the Image Source for processing this tool.	
		Properties	Calibration	Specify the Robot-to-Belt Camera Calibration used to calculate.
			Queue Parameters	Set the queue to which instances will be sent as follows. <b>Queue Index:</b> Identify the queue to which instances will be sent. <b>Queue Size:</b> Specify the number of instances that can be written to the queue in the range of 1-100. <b>Queue Update:</b> Select how often it will write instance data to the queue from <b>After Every Instance</b> or <b>After Last Instance</b> . The recommended setting is <b>After Every Instance</b> . <b>Use Soft Signal:</b> Select the check box to enable <b>Soft Signal</b> . <b>Soft Signal:</b> Set the value of the Soft Signal. V+ program uses the signal to synchronize the

		Item	Description
			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	Specify the offset in the Gripper Offset Table as the gripper offset. If not set, gripper offset will not be applied.  Set the following items. <b>Selection Mode:</b> Select the offset index of the gripper from <b>Disabled</b> , <b>Use Default</b> , or <b>Use Model Identifier</b> . <b>Default Offset Index:</b> If <b>Use Default</b> is selected as <b>Selection Mode</b> , the value in this item is used. <b>Transform:</b> Set the conversion value returned by the tool. <b>Model Offset Index:</b> The value of this item is used when <b>Use Model Identifier</b> is selected as <b>Selection Mode</b> .
		Robot	Specify the robot to send the instances to.
		Show Results Graphics	Select the check box to display the image in Vision Window.
	Advanced Properties	Tool Relative Coordinates	Selecting this option indicates that locations should be returned relative to the robot tip position when the picture was taken.
	Region of Interest	Offset	Specify the center and ration of the Region of Interest with X value, Y value, and angle value.
		Search Area (width, height)	Set the height and width of the Region of Interest.
		Relative To	Specify the tool to input to this tool.
		Link Name	Display the name of the property linked.
(c)	Instance		Display the instance.
(d)	Vision Window		Display the image specified in the <b>Image Source</b> .

- 3** Set each item.  
Place the communication tool that has been set after the overlap tool to receive the result of the overlap tool or after the inspection tool. 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.  
Refer to 5-5-6 *Vision Sequence* on how to set the vision sequence.

## Communication Tool Parameter Setting Example

This section describes effective setting examples of communication tool parameters.

### ● 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.

## ● 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 ACE Sight application sample for belt camera pick to 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, vision.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.

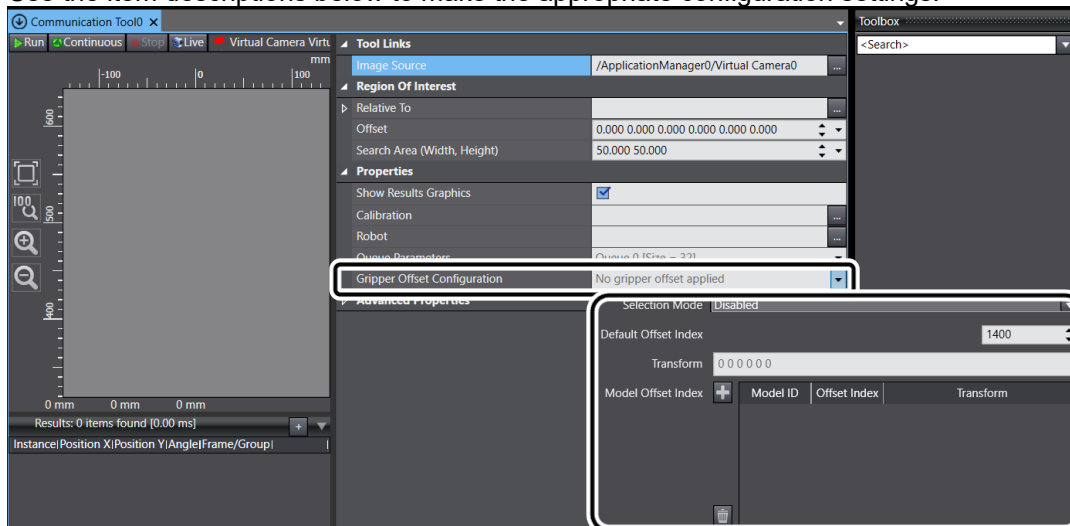
## ● Soft Signal

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 5-5-5 *Gripper Offset Table* 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).

# 5-6 Camera

Adds cameras used in Robot Systems. In the robot system, the image taken by the camera is processed by the vision tool to identify parts. An emulation camera is added and used instead of an actual camera in offline simulation. There are the following types of cameras.

Item	Description	Reference
Virtual Camera	This is a virtual camera that serves as an interface for accessing the camera image from the vision tool. The input images to the vision tool are generated by using part photographs of taken by the physical camera and photographs of the emulation camera as inputs.	5-6-1 <i>Virtual Camera</i>
Emulation Camera	This camera is a stored collection of images. By reading it from the virtual camera, you can process it like an image obtained from a physical camera.	5-6-2 <i>Emulation Camera</i>
Basler Camera	This is manufactured by Basler. Uses for taking an image. By assigning it to the virtual camera, you can use it actually taken by the camera.	5-6-3 <i>Basler Camera</i>
Sntech Camera	This camera is made by OMRON SENTECH. Uses for taking an image. By assigning it to the virtual camera, you can use it actually taken by the camera.	5-6-4 <i>Sentech Camera</i>
Custom Device	This device processes image data from an external camera or image system with a C# program. It is used by assigning it to the virtual camera.	5-6-5 <i>Custom Device</i>

## 5-6-1 Virtual Camera

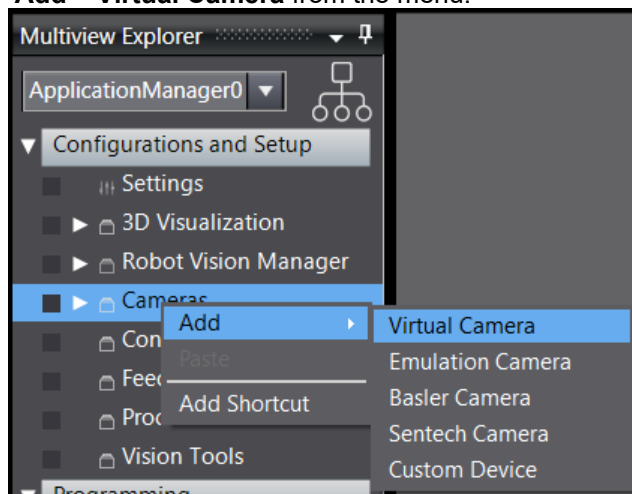
The Virtual Camera is a virtual camera that provides an interface for accessing camera Images from the Vision tool. Input photos of the part taken by the actual camera and photos of the emulation camera to generate the input Images to the Vision Tool.



### Precautions for Correct Use

In case you use a virtual camera, you need to add and associate the Virtual Camera when adding the camera or creating the emulation device. Refer to 5-6-3 *Basler Camera*, 5-6-4 *Sentech Camera*, 5-6-5 *Custom Device* for more information.

- 1 Right-click **Camera** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Virtual Camera** from the menu.



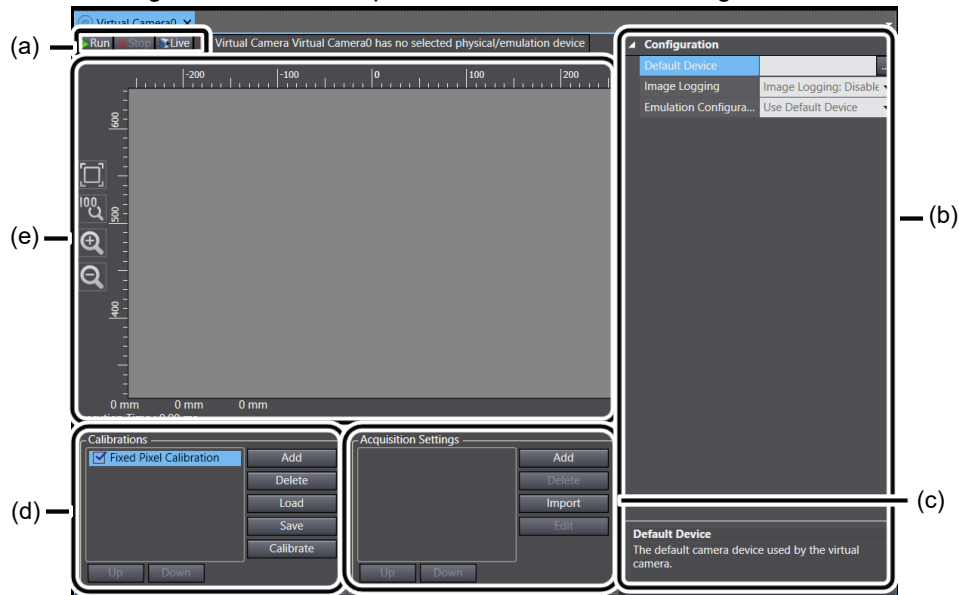
**Virtual Camera** is added under **Camera**.

- 2 Right-click **Virtual Camera** , and then select **Edit** from the menu.



The edit tab page is displayed.

The setting items on the edit pane are listed in the following table.



	Item		Description
(a)	Run		Acquire the image once from the camera specified.
	Stop		Stop continuous Live image acquisition.
	Live		Start continuous image acquisition from the camera device specified.
(b)	Configuration	Default Device	Specify the device used by the Virtual Camera.
		Image Logging	Set the following items to save the images used for offline setup and adjustment. <b>Enable:</b> Select the check box to enable Image Logging. <b>Directory Name:</b> Specify the directory to save. <b>Image Count:</b> Specify the number of images. Up to 1000 images can be saved.
		Emulation Configuration	Set the camera in emulation mode. Select <b>Emulation Configuration Behavior</b> from the following and configure each setting. <b>Use the default device:</b> Use the device specified in <b>Default Device</b> . <b>Random Instances:</b> Specify <b>Instance Count Minimum</b> and <b>Instance Count Maximum</b> . Acquire a random number of Images in a random orientation based on your specifications. <b>Use Alternate Device:</b> Specify another available <b>Camera Device</b> . <b>Images Replay:</b> Specify <b>Directory Name</b> to acquire images in hlg file format.
(c)	Acquisition Settings		The item is used to adjust image acquisition settings. Refer to <i>Acquisition Settings</i> for more information.
(d)	Calibrations		The item is used to calibrate the Virtual Camera. Refer to <i>Virtual Camera Calibration</i> for more information
(e)	Vision Window		Display the image specified in <b>Default Device</b> .

## Virtual Camera Calibration

When using vision tools, you need to calibrate the camera, such as the correction of perspective distortion and lens distortion, and definition of the relationship between the camera pixels and real-world dimensions. There are two methods available for virtual camera calibration as described below.

- Grid Calibration
- Fixed Pixel Calibration

## Grid Calibration

You can use a grid with known spacing to calibrate the camera. Sample dot target files are provided with the Sysmac Studio installation. Find these in the default installation directory with the following file names:

- DotPitchOthers\_CalibrationTarget.pdf
- DotPitch10\_CalibrationTarget.pdf

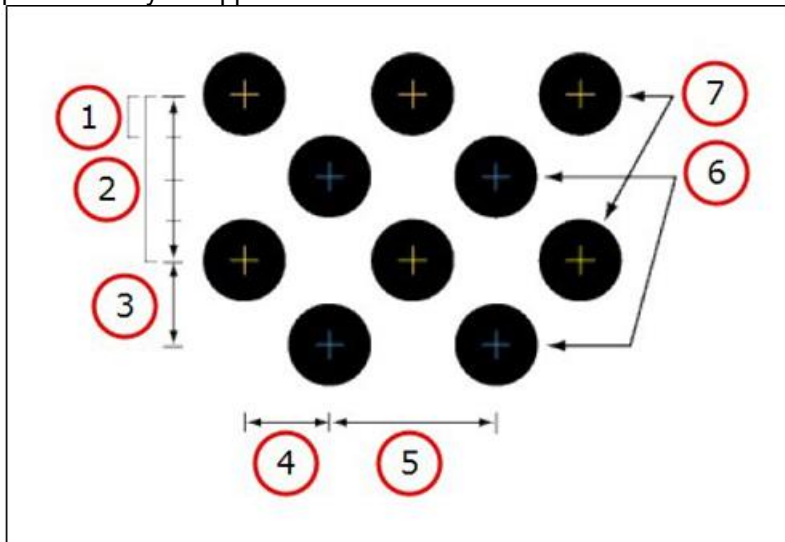


### Precautions for Correct Use

- 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

Dot targets are commercially available, but you can also create your own targets by following the guidelines provided below. The quality and precision of a grid of dots target have a direct impact on the overall precision of your 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)

## ● Dot Target Guidelines

- 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.



### Precautions for Correct Use

- You should measure the pitch of your dots after printing a grid to confirm that your printer did not change the scale of the grid. If the dots are not exactly the pitch you expect, 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.

## Fixed Pixel Calibration

Fixed-pixel calibration allows you to specify what physical distance is represented by each camera pixel. All camera pixels will be given the same dimension, which is not necessarily the case with a grid of dots. This method of camera calibration will not correct for lens distortion or perspective.

## 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, you can make several adjustments to the image such as shutter, gain, and exposure along with other camera related settings as described below.

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 lets you set the Pixel Format and Timeout value for the data being sent from the camera.

The available pixel formats will be displayed in the drop-down box when you click the down- arrow (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 you use a Timeout value. This is useful for time-critical applications in which fast operation is more important than the occasional occurrence of undetected object instances. This value is only approximate and the actual processing time may be slightly greater.

### ● Video Format

The **Video Format** tab lets you set Exposure, Gain, Black Level, and color balance.

Each line displays the minimum allowable value for that property, a bar indicating the current value, the maximum allowable value, and the numeric value of the current level.

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 you want to acquire images of still or moving objects. 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, you can 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 you enter an exposure time that your camera does not support, the time will be adjusted to be valid. If you haven't typed in an invalid exposure time, the left and right arrows will provide valid times.

## ● Gain 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. You can increase the contrast in the image by increasing the camera's gain setting.

Unless your application requires extreme contrast, make sure that detail remains visible in the brightest portions of the image when increasing gain. Noise is increased by increasing gain.

Increasing gain will increase the image brightness. 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 if you have a color camera connected.

On some Basler color cameras, such as the A601fc-2, the green balance is a fixed value that cannot be adjusted. In such cases, only the balance for blue and red will be enabled in this window (**Balance Green** will be grayed out).

It is recommended to consider white balance of the camera when using color camera models.

## ● Trigger

The **Trigger** tab lets you 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.

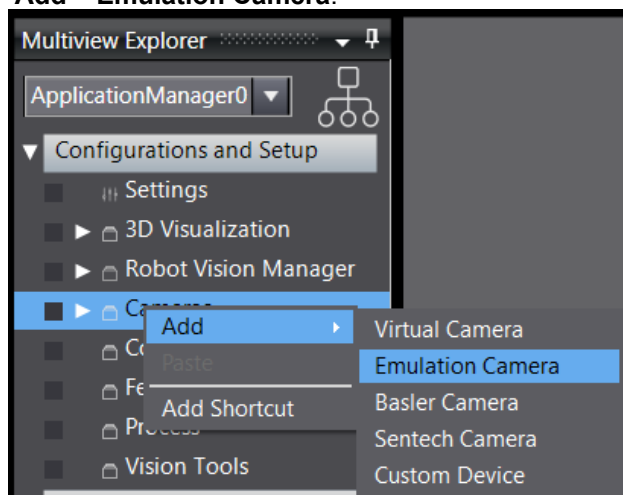
The **Edit** button opens a window for changing acquisition settings for the selected device. Use the Up and Down buttons to arrange multiple acquisition settings in a specific order.

## 5-6-2 Emulation Camera

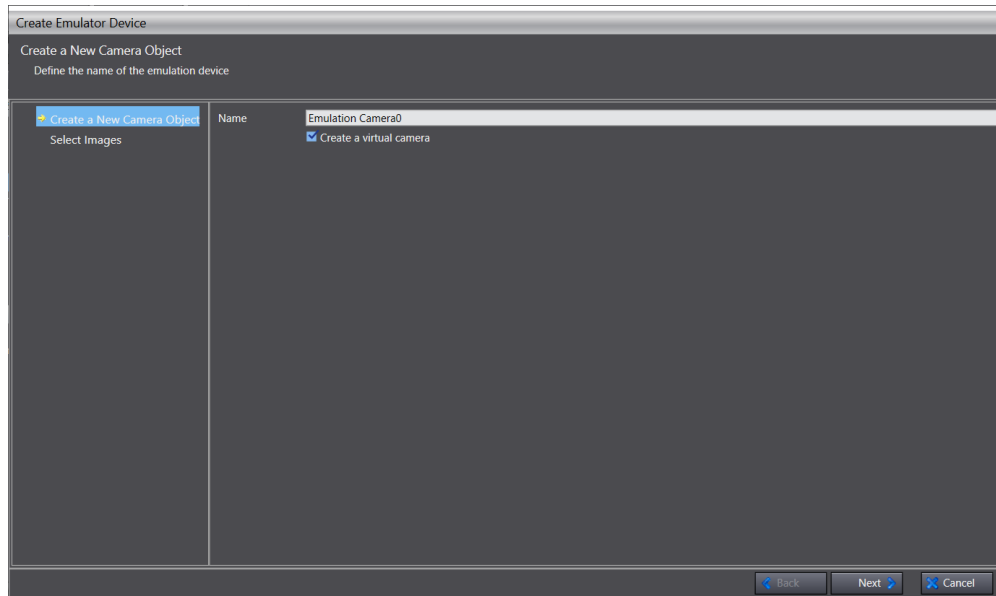
An emulation camera is Images data that has been captured and saved in advance. By assigning it to a virtual camera, it can be processed like Images acquired from an actual camera.

It is used when working offline, such as simulating a robot system before installing an actual equipment, or displaying Images from an application on a remote facility.

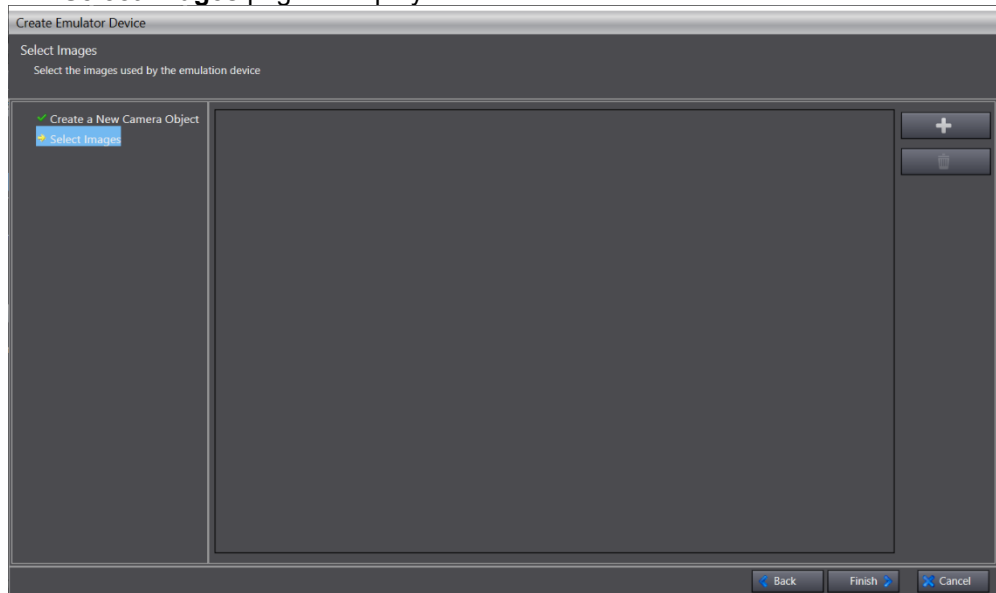
- 1 Right-click **Camera** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Emulation Camera**.



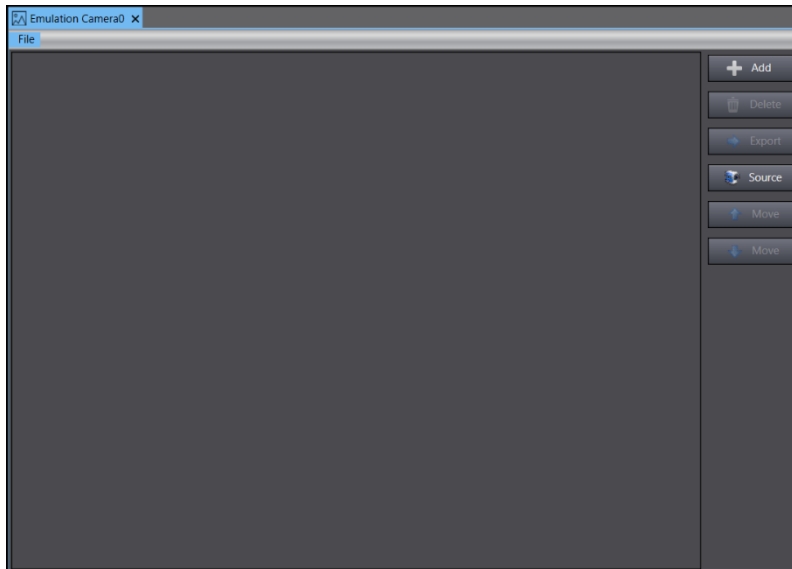
The **Create a New Camera Object** page is displayed.



- 2** Specify the name of **Emulation Camera**, and then click the **Next** button. By selecting the **Create a virtual camera** check box, you can use the virtual camera to exchange images with camera or emulation camera. Refer to *5-6-1 Virtual Camera* for more information. The **Select Images** page is displayed.



- 3** Specify the images used by **Emulation Camera**, and then click the **Finish** button. An **Emulation Camera** is added under **Camera**. The specified images are registered in **Emulation Camera**. You can add, export, and edit **Emulation Camera**. Right-click **Emulation Camera** to edit, select **Edit**, and then the edit tab page is displayed.



Item	Description
<b>Add/ Delete</b> button	Use these buttons to add or remove images.
<b>Export</b> button	Use this button to export images from the Emulation Camera to files on the PC.
<b>Source</b> button	Use this button to add images from another camera source within the Application Manager.
<b>Move</b> button	Use the Move button to arrange the images in a specific order.
<b>File Menu - Load From Database</b>	Load a collection of images from an emulation database file (.hdb).
<b>File Menu - Save All To Database</b>	Save a collection of images to an emulation database file (.hdb).
<b>File Menu - Sort Images</b>	Sort images in alphabetical order by file name.

### 5-6-3 Basler Camera

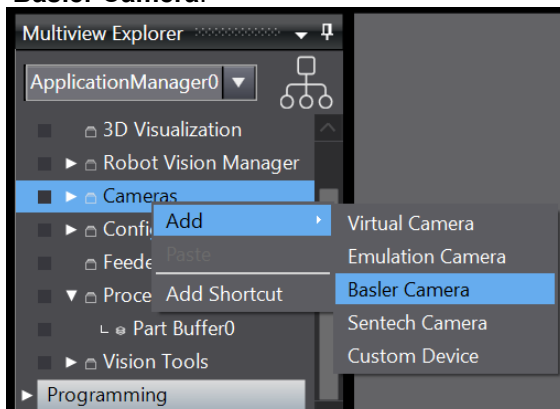
Basler Cameras are physical devices that are used for image acquisition. Allocate a Basler camera to a virtual camera to use images taken with a physical device.



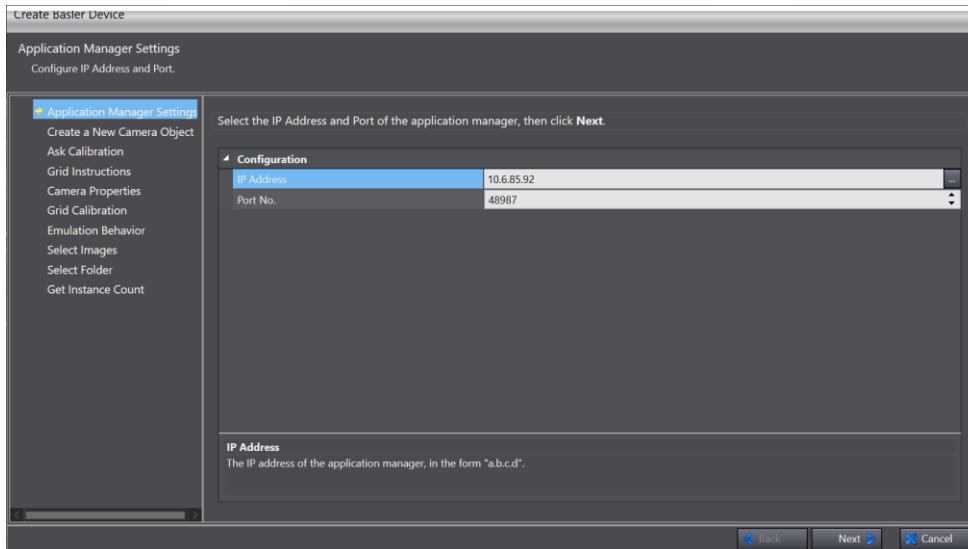
#### Additional Information

Robot Vision Manager supports a Basler Pylon Cameras.

- 1 Right-click **Multiview Explorer - Configurations and Setup - Cameras**, then select **Add - Basler Camera**.



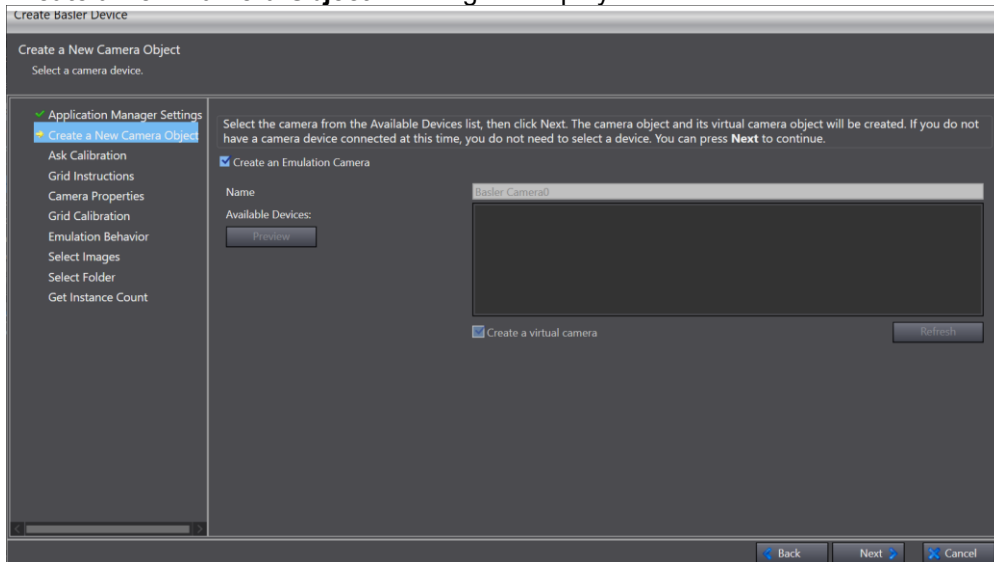
**Application Manager Settings** is displayed.



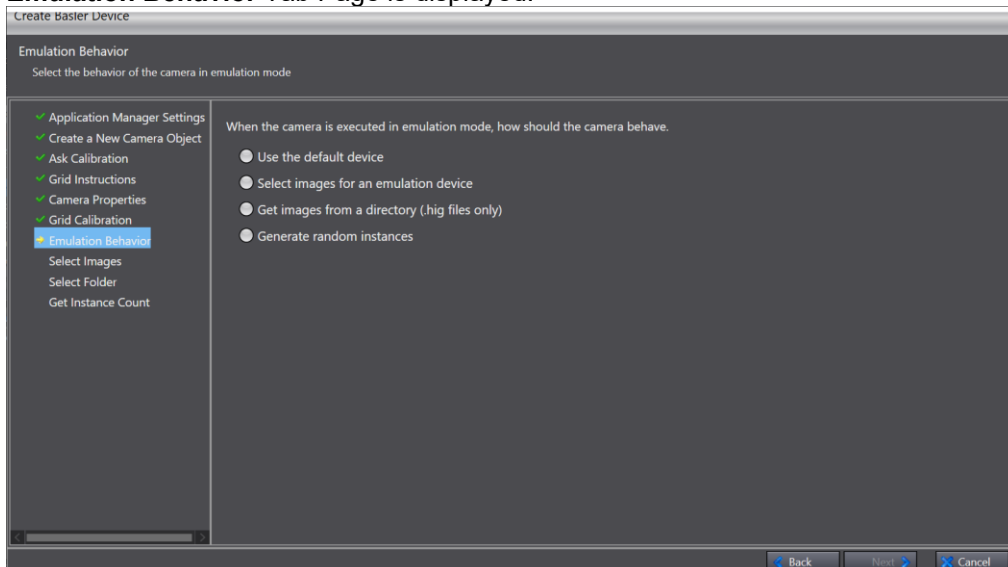
Settings tab pages differs from offline and online with an actual equipment. See *Offline* or *Online*, for each case.

- **Offline**

- 2 Click the **Next** button. **Create a New Camera Object** Tab Page is displayed.



- 3 Check the **Create an Emulation Camera** box. Then Click **Next**. **Emulation Behavior** Tab Page is displayed.



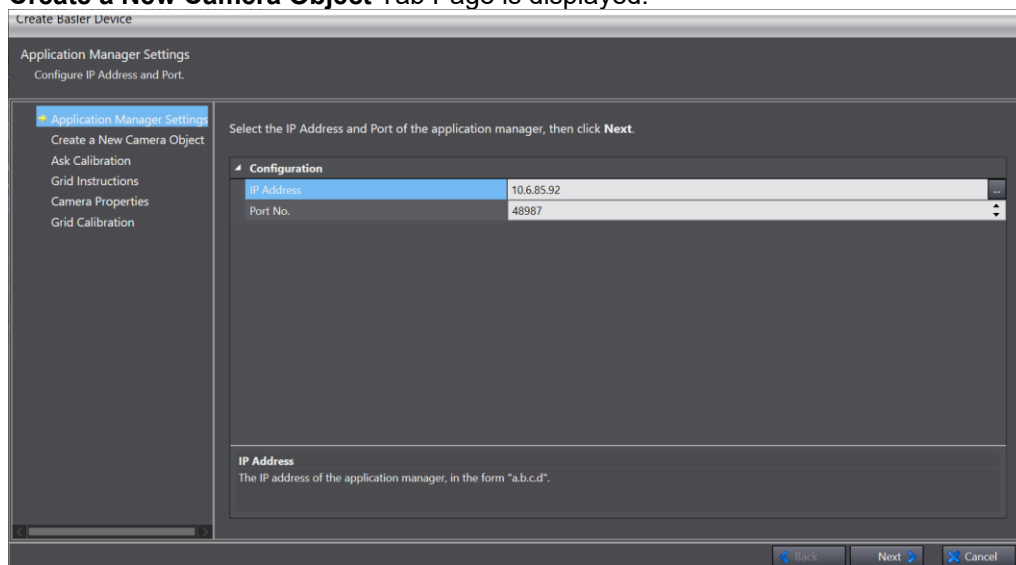
- 4 Select a camera behavior in the Emulation mode, then click the **Next** button.  
You can select the camera behavior in the Emulation mode from the table below.

Camera Behavior	Description
Use the default device	Not necessary to specify a camera. Images are processed in the Emulation mode.
Select images for an emulation device	<b>Select Images</b> Tab Page is displayed. You can select images used in the Emulation mode.
Get images from a directory (.hig files only)	<b>Select Folder</b> Tab Page is displayed. You can specify a folder where images used in the Emulation mode stored.
Generate random instances	<b>Get Instance Count</b> Tab Page is displayed. You can specify Instance Count Minimum and Instance Count Maximum. Random number of images are acquired in the range of the specified values.

- 5 Configure required settings following the guidance of the selected item. Then Click the **Finish** button.  
**Basler Camera, Emulation Camera, and Virtual Camera** are added under **Cameras**.

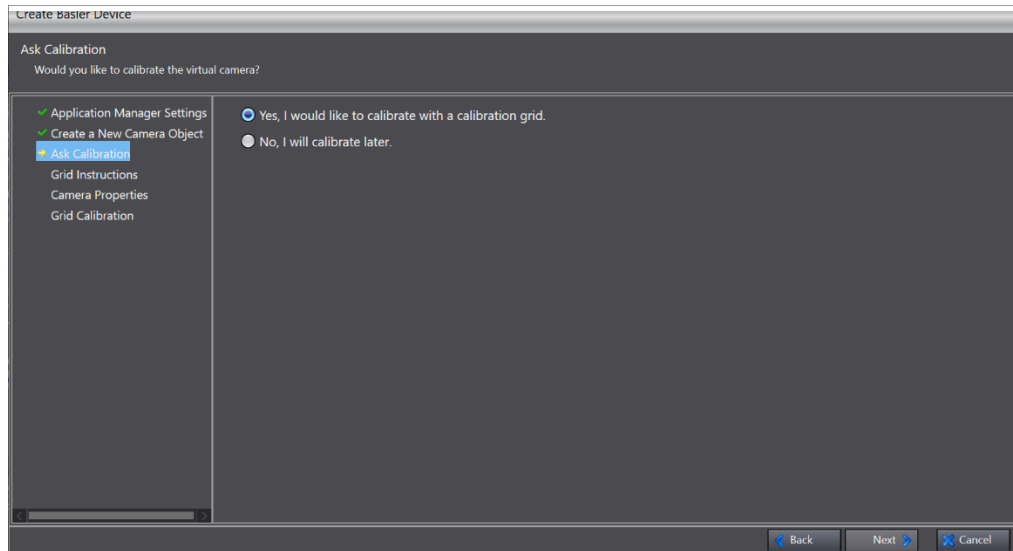
- **Online**

- 2 Click the button to select connection destination. The **Select a controller address** dialog box is displayed. Select an IPC Application Controller to which the target camera is connected, then click **Next**.  
**Create a New Camera Object** Tab Page is displayed.

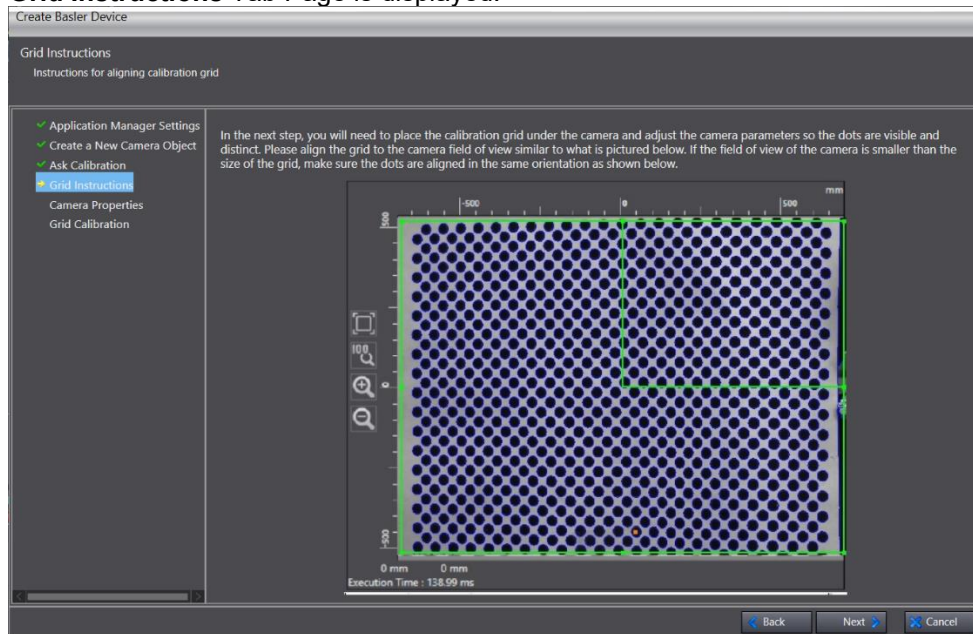


- 3 Select a camera you want to connect, then click the **Next** button.  
Check **Create a virtual camera** to use a virtual camera. Refer to 5-6-1 *Virtual Camera* for details.  
**Ask Calibration** Tab Page is displayed.

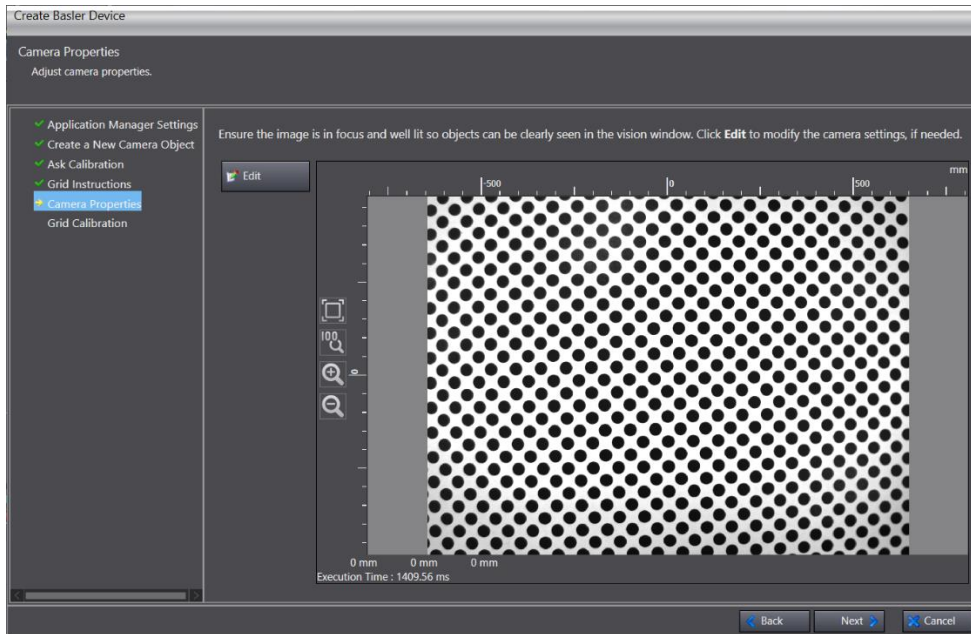




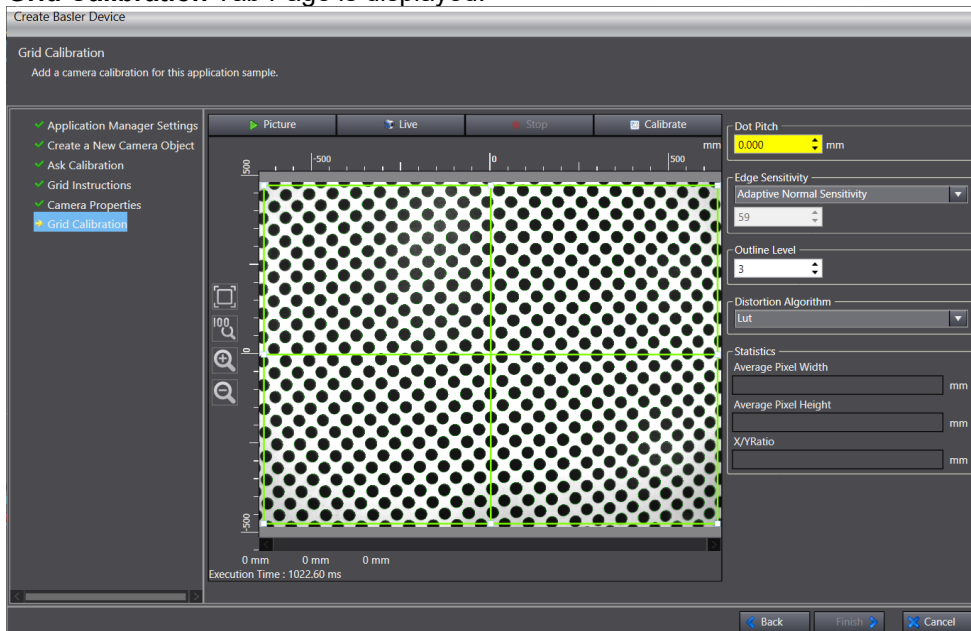
- 4** Check the radio button for **Yes, I would like to calibrate with a calibration grid**, then click the **Next** button.  
 If you check the button for **No, I will calibrate later**, go to the step 7.  
**Grid Instructions** Tab Page is displayed.



- 5** Confirm the description, then click the **Next** button.  
**Camera Properties** Tab Page is displayed.



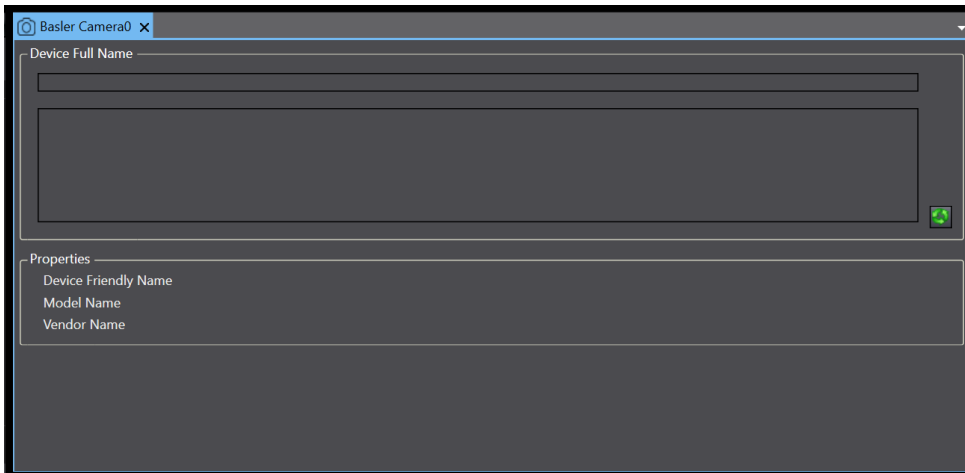
- 6 Click the **Edit** button to modify camera settings, if necessary. Then Click the **Next** button. **Grid Calibration** Tab Page is displayed.



- 7 Shoot a calibration sheet and execute a calibration. Then click the **Finish** button. **Basler Camera** is added under **Cameras**. If you have checked the **Create a virtual camera** box, **Virtual Camera** is added.

## Basler Camera Configuration

Properties of the specified camera are registered in the **Basler Camera** Tab Page. Right-click **Basler Camera** and select **Edit** to confirm properties of the connected camera.

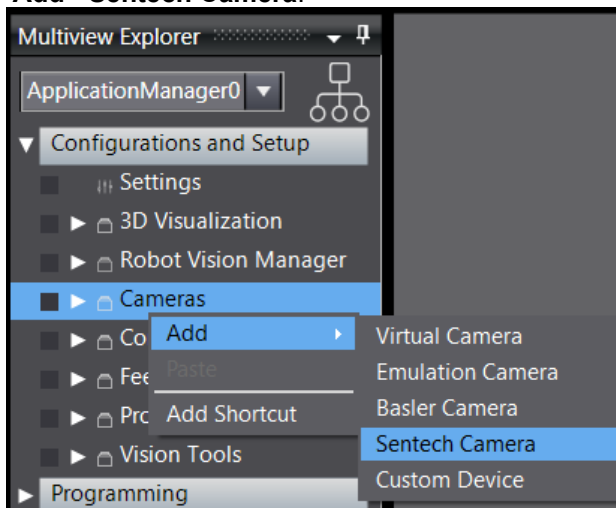


Displayed items differs depending on the selected Basler camera type. Refer to the manual of Basler camera for details.

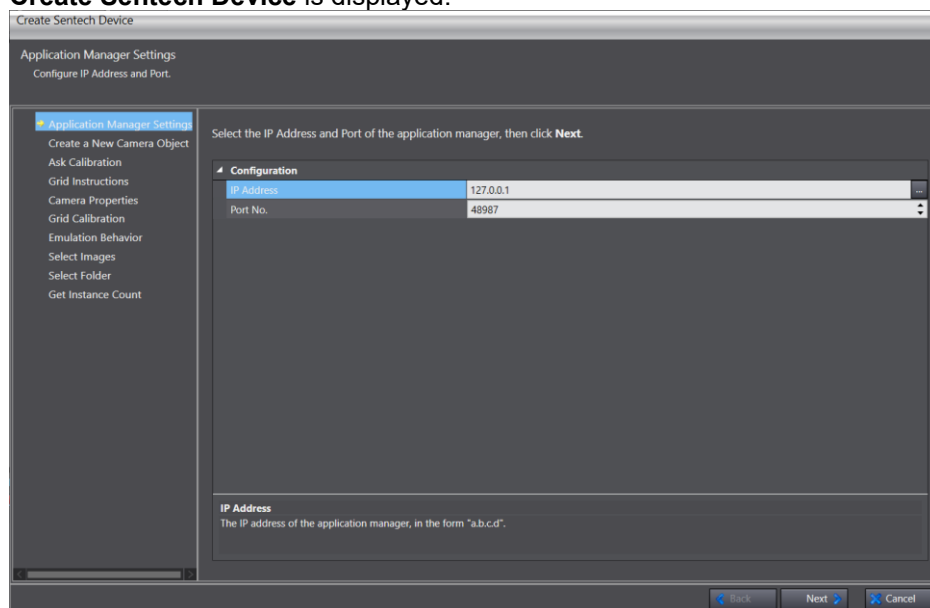
## 5-6-4 Sentech Camera

This camera is made by OMRON SENTECH. It is used for taking an image. By assigning it to a virtual camera, you can use the image actually taken by the camera.

- 1 Right-click **Cameras** under **Configurations and Setup** in the **Multiview Explorer**, and select **Add - Sentech Camera**.



**Create Sentech Device** is displayed.



- 2 Click the connection destination button to display the **Select a controller address** dialog box. And then select the IPC Application Controller, to which the target camera is connected, and click the **Next** button.

Subsequent screen operations are the same as adding the Basler Camera. Refer to *5-6-3 Basler Camera*.

Click the **Finish** button to add **Sentech Camera** under **Cameras**. **Emulation Camera** and **Virtual Camera** are added according to the settings on each screen.

## Sentech Camera Configuration

The properties of the specified camera are registered in **Sentech Camera**.

Right-click **Sentech Camera** and select **Edit** to check the properties of the connected camera.

The content of the displayed items depends on the type of Sentech camera selected. Refer to each Sentech camera manual for details.

### 5-6-5 Custom Device

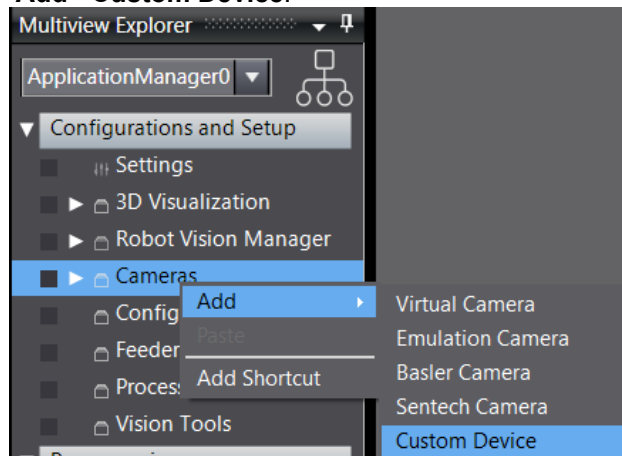
This device processes image data from an external camera or image system with a C# program. It is used by assigning it to a virtual camera.



#### Additional Information

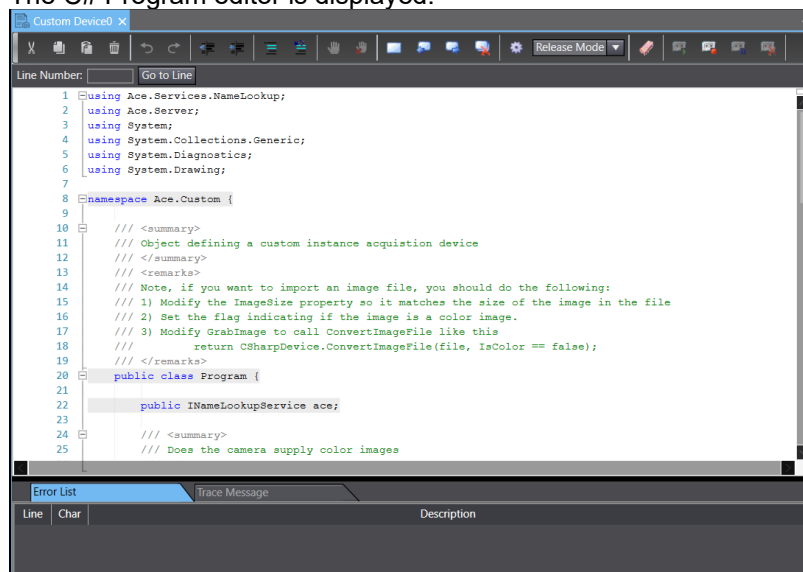
Refer to the instruction manual for an external camera or image system for details on how to detect images from the external camera or image system.

- 1 Right-click **Cameras** under **Configurations and Setup** in the **Multiview Explorer**, and select **Add - Custom Device**.



**Custom Device** is added under **Cameras**.

- 2 Right-click **Custom Device** and select **Edit**.  
The **C# Program** editor is displayed.



Refer to *5-11-1 C# Program Editor* for details on how to use the C# Program Editor.



# 5-7 Configuration

Makes various settings that the Application Manager has, such as the recipe manager function used to create and manage production recipes and the data mapping function that associates the inputs and outputs in the workspace.

The setting items are as follows.

Item	Description	Reference
Controller Connection Startup	Sets whether to automatically connect to the Robot Integrated CPU Unit module or to execute the program when the Application Manager on the IPC Application Controller starts.	5-7-1 <i>Controller Connection Startup</i>
Data Mapper	The Data Mapper provides a method to associate different data items within the Sysmac Studio project.	5-7-2 <i>Data Mapper</i>
Note	Creates a new Note object. It is used to enter notes about the application.	5-7-3 <i>Note</i>
OPC Container	Creates a new OPC Container object. 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.	5-7-4 <i>OPC Container</i>
Program System Startup	Makes settings for starting the Robot Integrated System. Specifies a C# program to run when the Robot Integrated System project opens.	5-7-5 <i>Program System Startup</i>
Recipe Manager	The Recipe Manager is used to define all the sources that will be used in creating recipes.	5-7-6 <i>Recipe Manager</i>
Recipe Manager Script	A Recipe Manager Script allows customization of C# programs invoked when certain events occur in the lifetime of a Recipe.	5-7-7 <i>Recipe Manager Script</i>

## 5-7-1 Controller Connection Startup

The Controller Connection Startup item is used to establish a connection, monitor, and maintain the connection to one or more controllers upon the startup of Application Manager which runs on IPC Application Controller. It can also be configured to start a V+ program on a specified task number upon startup.

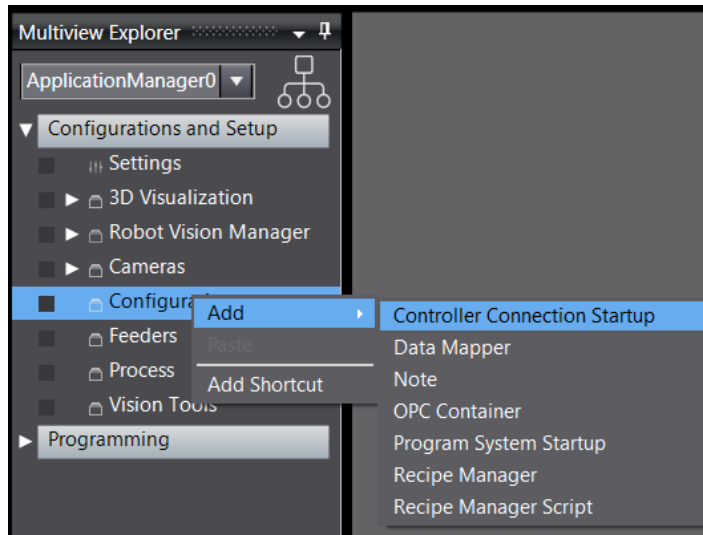
When the Controller Connection Startup object is configured, opening the ACE project will automatically initiate a connection to a controller and program execution, once the connection is fully established.

### WARNING

If an execution program and task number are specified and the automatic execution of V+ program is enabled, it may possibly happen that the robot operates after the CPU Unit and robot are turned on. Make sure that the movement of the robot does not cause a danger.

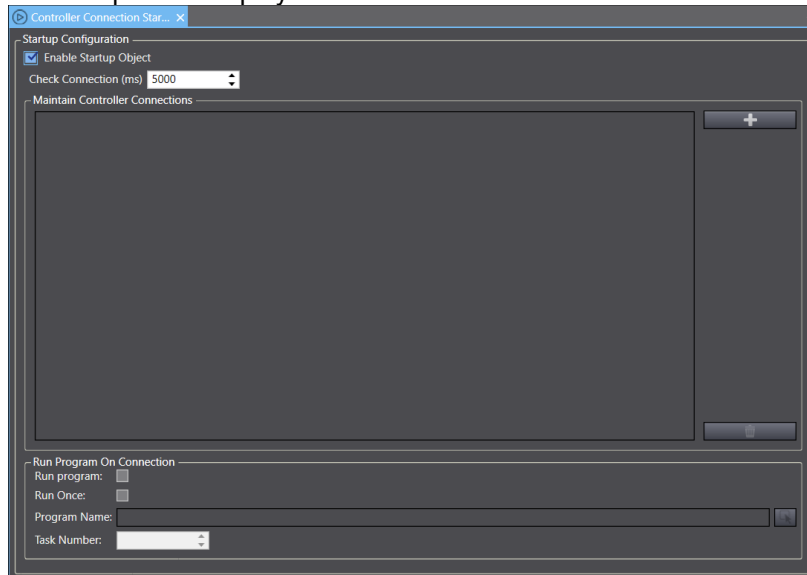


- 1 Right-click **Configuration** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Controller Connection Startup**.



A **Controller Connection Startup** is added under **Configuration**.

- 2** Right-click **Controller Connection Startup**, and then select **Edit**. The edit pane is displayed.



The setting items on the edit pane are listed in the following table.

Group	Item	Description
Startup Configuration	Enable Startup Object	Enable or disable the function of the Controller Connection Startup object when the Application Manager is opened on IPC Application Controller.
	Add / Delete buttons	Add or remove controllers to/from the <b>Maintain Controller Connections</b> list. You can only add a controller that already exists in the Robot Integrated CPU Units of the project.
	Check Connection (ms)	Specify the interval in milliseconds for checking the connections to the controllers specified on the <b>Maintain Controller Connections</b> list.
	<b>Maintain Controller Connections</b> List	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 the Application Manager of the IPC Application Controller is first connected to the controller. It will not rerun when disconnected and reconnected. If the program needs to run again, you will need to shut-down the Application Manager and restart it.
	Program Name	Specify the program to run when the selected controller is connected.
	Task Number	Specify the V+ task number for the V+ program. The specified V+ task number must be idle. Otherwise, the program will not run on connection.

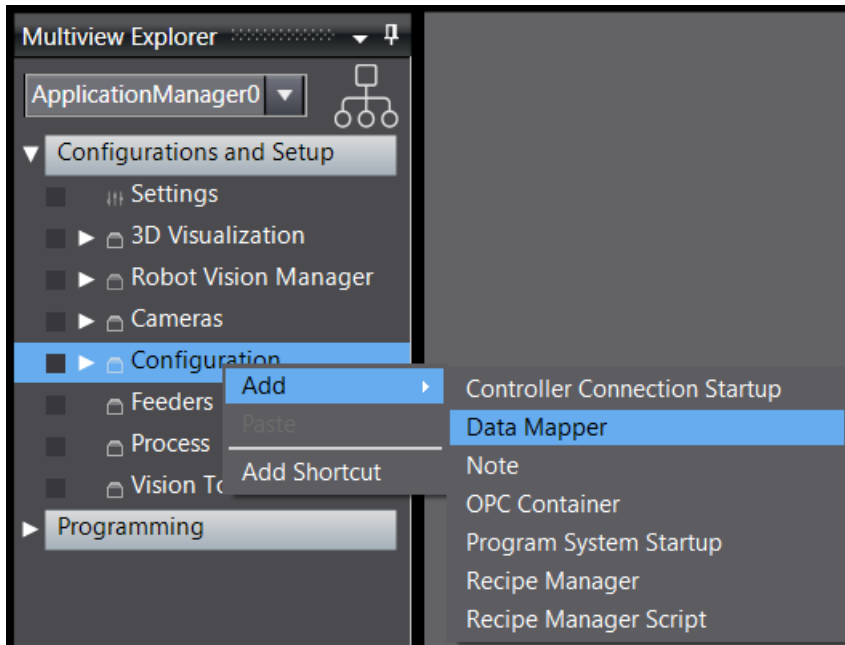
## 5-7-2 Data Mapper

The Data Mapper provides a method to associate different data items within the Sysmac Studio project. For example, you can trigger a Process Manager object 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 Sysmac Studio project is open.

A Data Mapper object is used to map information between objects within the workspace, and is created as shown below.

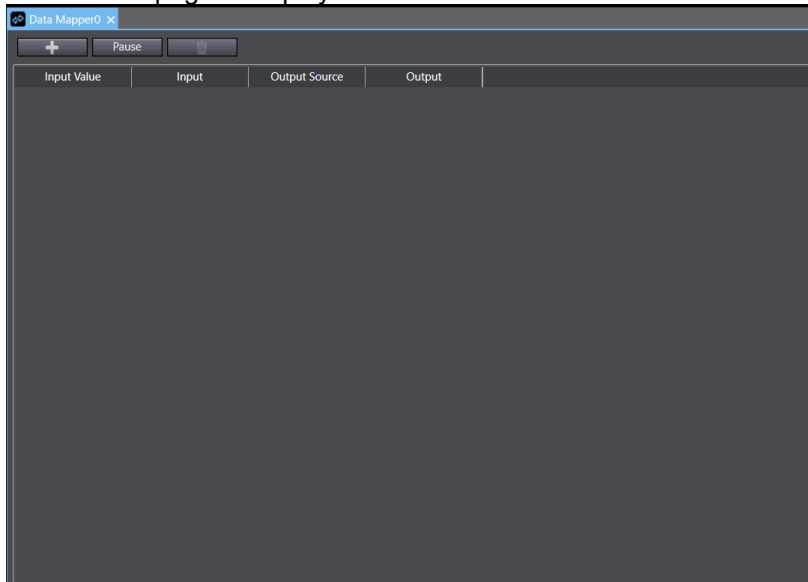
- 1 Right-click **Configuration** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Data Mapper**.





A **Data Mapper** is added under **Configuration**.

- 2 Right-click **Data Mapper**, and then select **Edit**.  
The edit tab page is displayed.



The setting items on the edit tab page are listed in the following table.

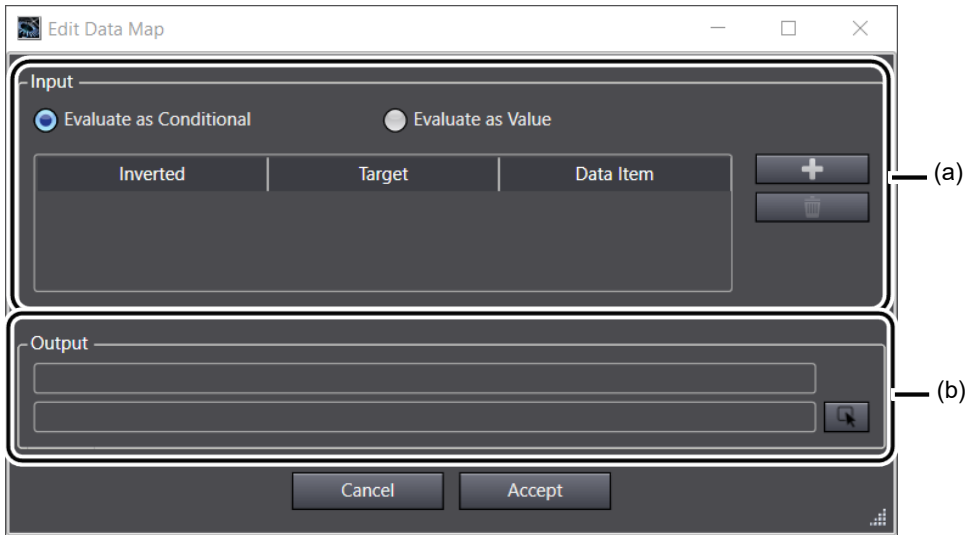
Item	Description
The <b>Add</b> button	Add a Data Mapper. Refer to <i>Data Mapping</i> for more information
The <b>Pause</b> button	Pause processing the data mapper.
The <b>Delete</b> button	Delete the selected data mapper from the list.
List	It is a list of created data mappers.

## 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.

The Data Mapping input and output items that are available depend on the objects present in the project.

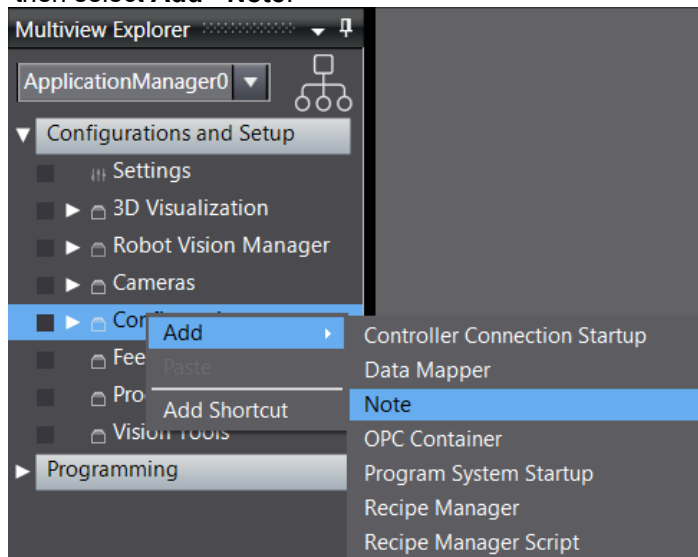


	Item	Description	
(a)	Input	Evaluate as Conditional	When <b>Evaluate as Conditional</b> is selected, the Data Mapper will interpret all input conditions as a boolean item. If the value of the input item is 0, the condition is considered OFF. If the value is non-zero, the condition is considered ON. If 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. Additionally, when <b>Evaluate as Conditional</b> is selected, you can invert the expected value of an input item. In that case, if the value is 0, the condition is considered to be ON.
		Evaluate as Value	When <b>Evaluate as Value</b> is selected, the value of all input conditions is added together and written to the output value.
		Add/ Delete button	Click the <b>Add</b> button, and then, the <b>Data Item Selection</b> dialog box is displayed. Select the input item from the <b>Robot Control Settings</b> of Robot Control Function Module of Robot Integrated CPU Units.
		Data Item List	It is a list of input items to be registered in the data map.
(b)	Output	Click the Output Item Selection button, and then the <b>Data Item Selection</b> dialog box is displayed. Select the output item from the <b>Robot Control Settings</b> of Robot Control Function Module of Robot Integrated CPU Units.	

### 5-7-3 Note

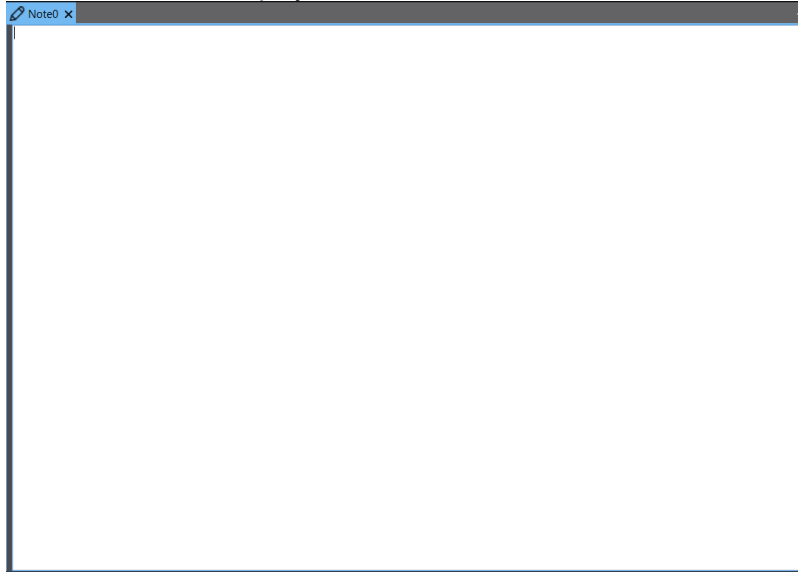
Creating a new Note object. It is used to enter notes about the application.

- 1 Right-click **Configuration** under **Configurations and Setup** in the **Multiview Explorer**. And then select **Add - Note**.



The **Note** object is newly added under **Configuration**.

- 2 Right-click **Note** and select **Edit**.  
The Note editor is displayed in the Edit Pane. You can enter notes.



## 5-7-4 OPC Container

Create OPC Container object. 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.

An OPC container can be configured for the following functions.

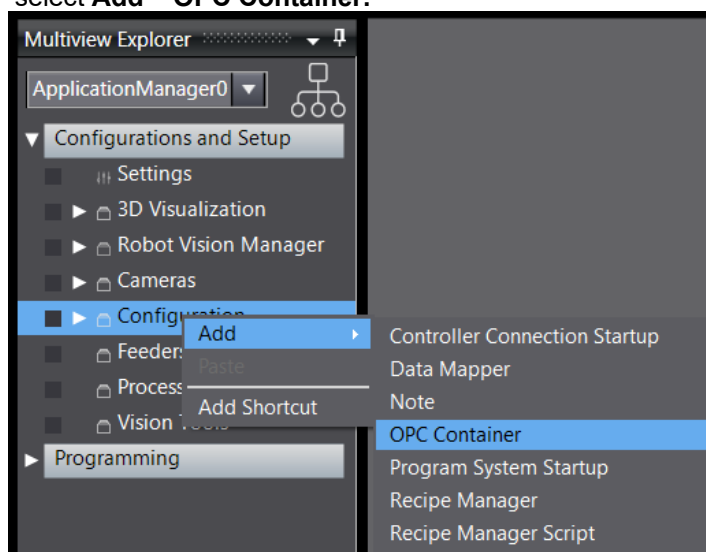
- Communicate values of V+ Global Variables and Process Manager statistics.
- Start or stop a process within an application of Robot Integrated System.

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.

For more information on OPC, please see the “OPC Foundation” website at the following URL:

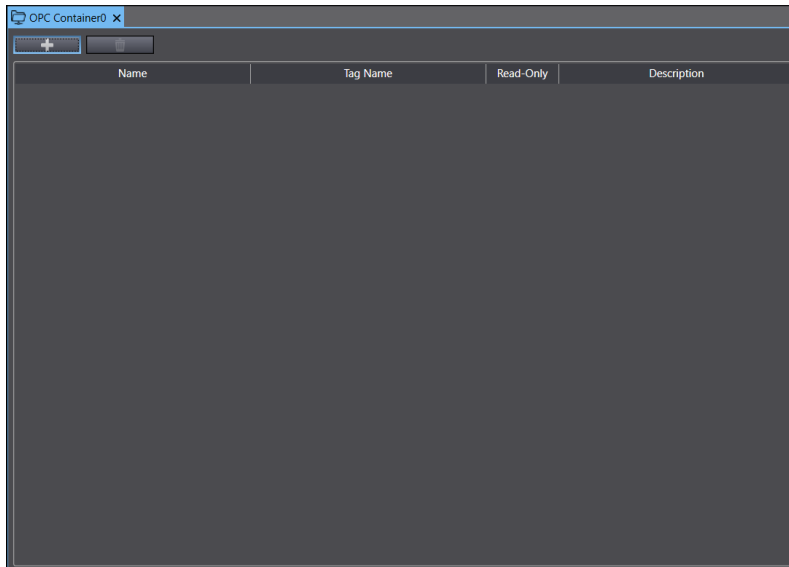
<https://www.opcfoundation.org>

- 1 Right-click **Configuration** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – OPC Container**.



An **OPC Container (number)** is added under **Configuration**.

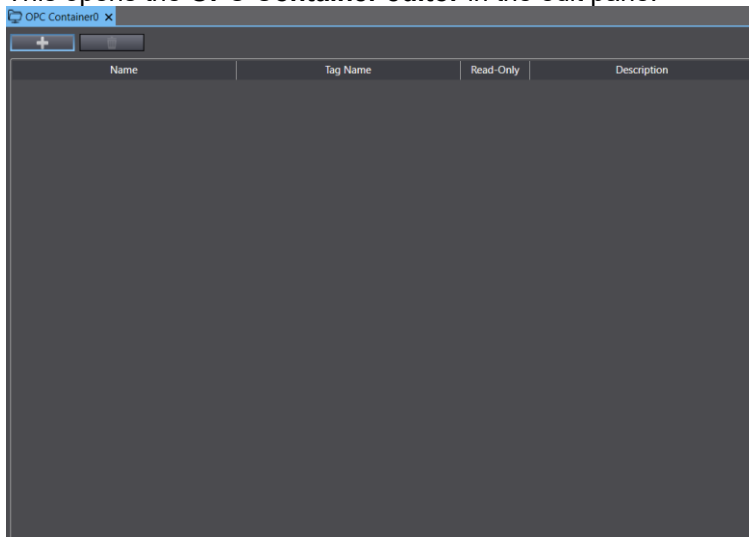
- 2 Right-click **OPC Container (number)**, and then select **Edit**.  
The edit pane is displayed.



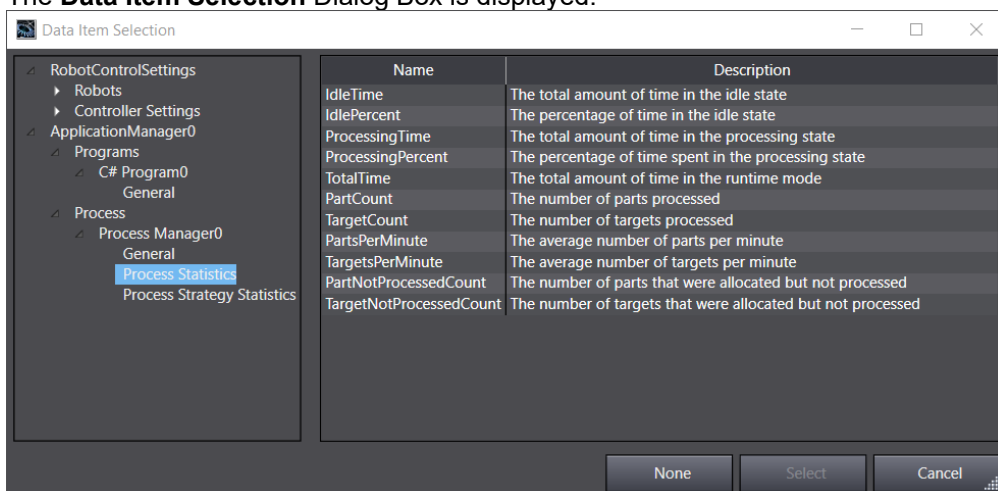
## OPC Container Configuration Procedure

Use the following procedure to configure an OPC Container object. Depending on the project configuration, the items available in the Data Item Selection list will vary. The procedure below sets the robot configuration that was generated using Robot Vision Manager Application Sample with a fixed pick and a fixed place arrangement.

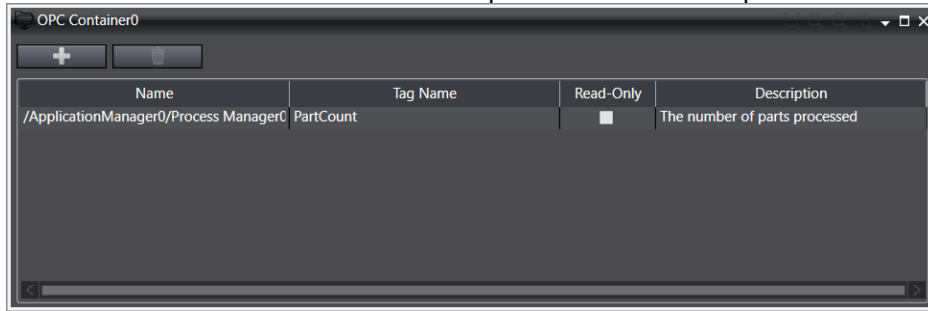
- 1 To access the OPC Container configuration, right-click the object in the Multiview Explorer and then select **Edit**, or double-click the object. This opens the **OPC Container editor** in the edit pane.



- 2 Click the **Add** button to select a Data Item. The **Data Item Selection Dialog Box** is displayed.



- 3 Select an item from the list and then click the **Select** button.  
The item is 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.  
Once all items have been added to the OPC Container publish list, the OPC client can be configured as shown in the next steps.  
Subsequent OPC client settings operate on the IPC Application Controller.  
Refer to the *Automation Control Environment (ACE) Version 4 User's Guide (I633)*.

## 5-7-5 Program System Startup

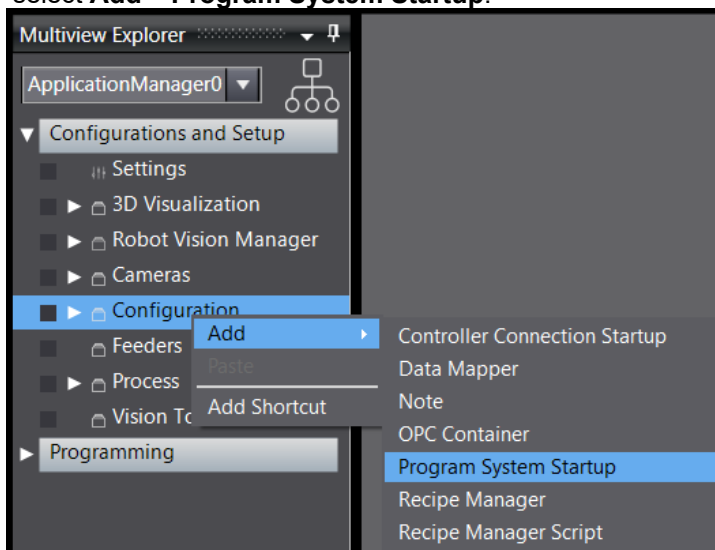
Use the following procedure to configure the Program System Startup of Robot Integrated System. Specify a C# program to run when the Robot Integrated System project opens.



### Additional Information

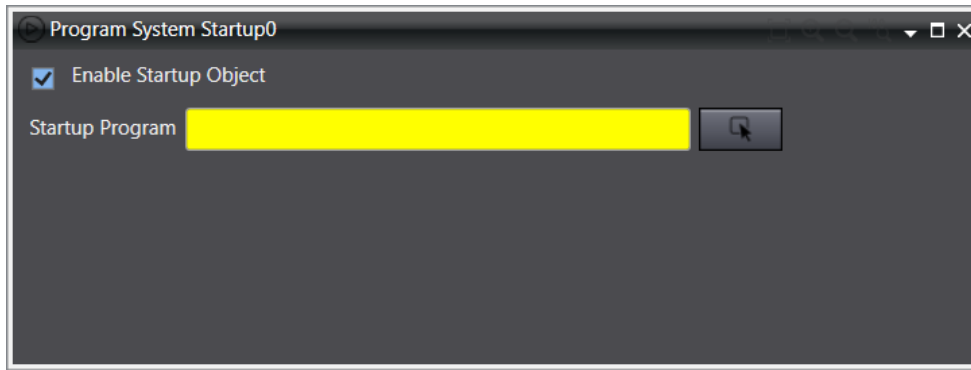
Only one system startup object can be created in the workspace.

- 1 Right-click **Configuration** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Program System Startup**.



A **Program System Configuration** is added under **Configuration**.

- 2 Right-click **Program System Startup**, and then select **Edit**.  
The edit tab page is displayed.



The setting items on the edit pane are listed in the following table.

Item	Description
Enable Startup Object	Select the check box to run the C# program specified by the <b>Selector</b> button at system startup.
<b>Selector</b> button	Specify the C# program to run at system startup. Only one object can be specified.

## 5-7-6 Recipe Manager

The Recipe Manager is used to define all sources that will be used when creating individual recipes. You must add sources to the Recipe Manager before creating a Recipe.

You can save information set in the controller or the Vision Tool in the object. The saved information is used to create and manage production recipes.

### 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 Application Manager 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 Application Manager, 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, you would 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. Alternatively, if two types of product must be recognized by the same Locator, you could have two Locator Model objects and include both as sources.

#### ● Recipe Source

The following objects in an Application Manager project can be used as sources for the Recipe Manager.

Item	Description
V+ Variables	Specify V+ Variables to be included in a recipe. You can identify how the variable is displayed to the user and what access level a user will have.
Vision Tools	All vision tools can be accessed with the Recipe Manager, except the following tool types. <ul style="list-style-type: none"> <li>· Calculation Tools</li> <li>· Image Process Tools</li> <li>· Custom Tools</li> </ul>
Virtual Camera	Specify Virtual Camera object data to be included in a recipe.
AnyFeeder	Specify AnyFeeder object data to be included in a recipe.

#### ● 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 Application Manager 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.



### Precautions for Correct Use

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 *Recipe Editor* for more information regarding creating, modifying, and deleting recipes.

## ● Recipe Selection

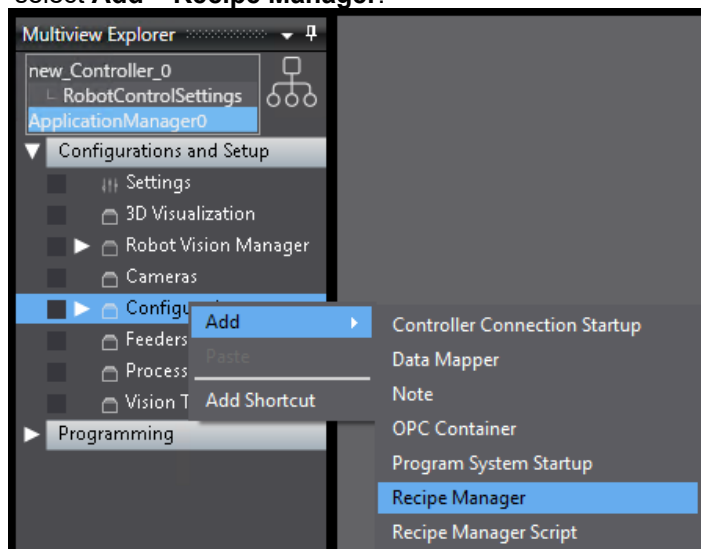
Recipe Selection is a single item selection process for applying the parameters stored in the recipe to the Source objects.

Once all recipes have been defined and optimized you may want to 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 C# program.

When a Recipe is selected, the parameters saved in the Recipe are applied to the Application Manager. 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 Sysmac Studio project.

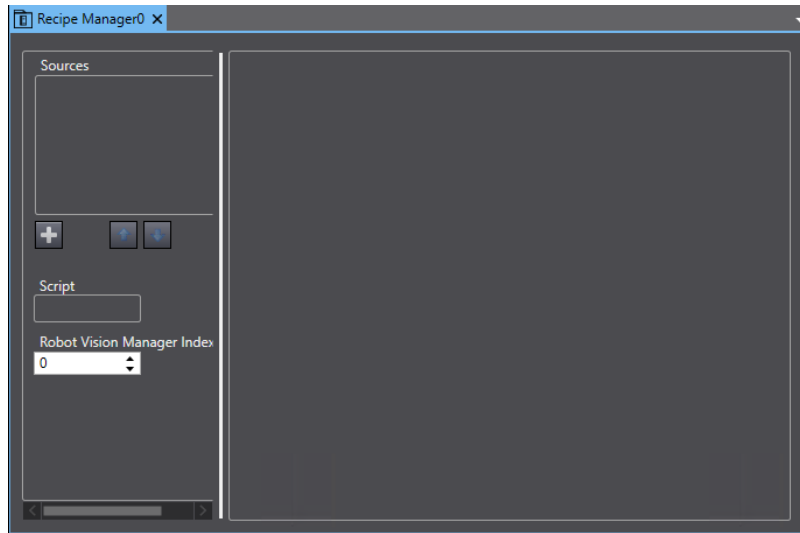
## Adding a Recipe Manager

- 1 Right-click **Configuration** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Recipe Manager**.



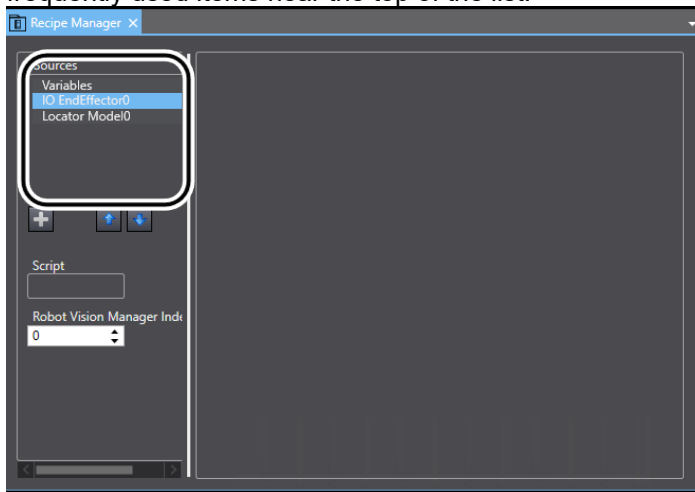
A **Recipe Manager** is added under **Configuration**.

- 2 Right-click **Recipe Manager**, and then select **Edit**.  
The edit pane is displayed.  
Click the **Configuration** tab, and then the following pane is displayed.



## 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.



When a data source is added to the Sources list, you can 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 - You can restrict write access to Recipe Manager sources by setting user access levels. Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for more information.

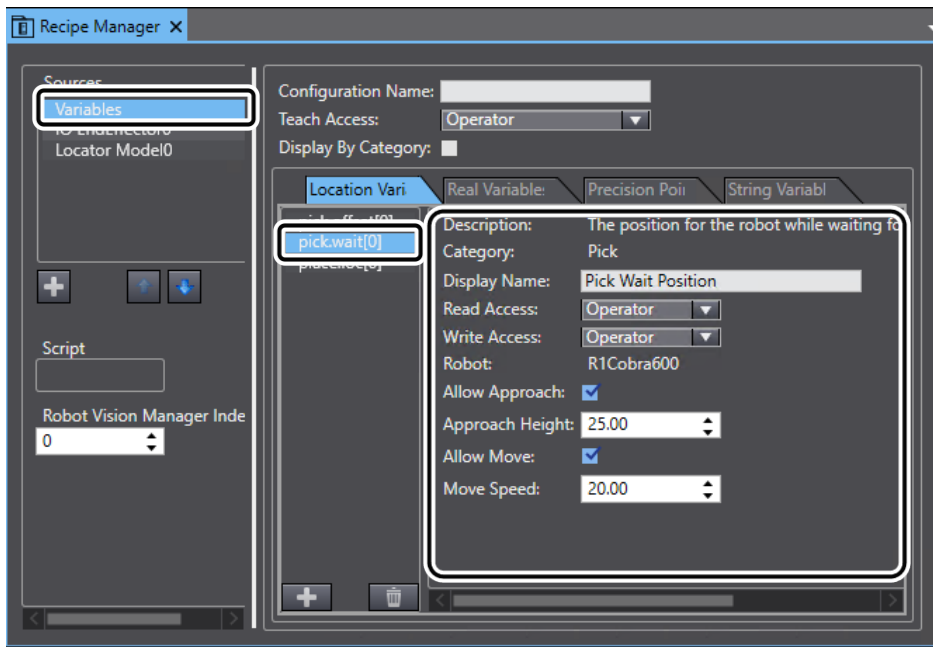
### ● V+ Variable Sources

When configuring a V+ Variable Source in the Recipe Manager, you must add individual variables 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 you in the Recipe Editor. For example, you can define 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. For each Recipe you create, 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 Sysmac Studio project that they correspond to. When a vision tool included in the Recipe configuration is modified in the Sysmac Studio 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 Sysmac Studio project vision tool object is automatically updated. Because of this linking between the Recipe and Sysmac Studio project object, you can configure a vision tool object and it 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 *Recipe Editor* for more information.

- **Recipe Script Selection**

Select a Recipe Script object created with the Recipe Manager Script editor. Refer to *5-7-7 Recipe Manager Script* for more information.

- **Robot Vision Manager Index Setting**

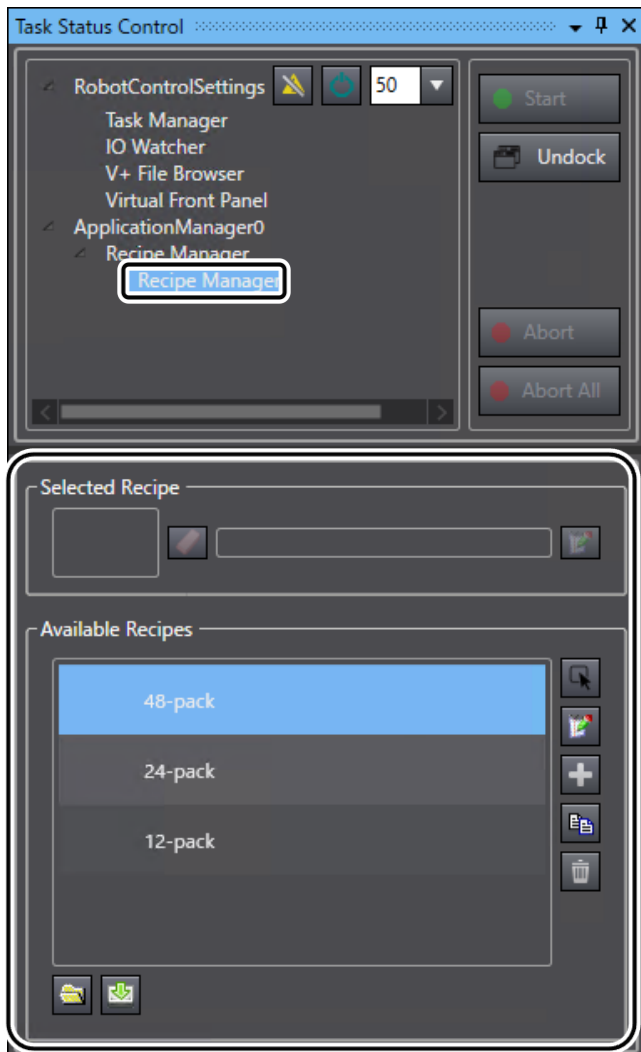
Robot Vision Manager index setting defines the index used as the “sequence\_id” when accessing the recipe manager object from a V+ program. Refer to *ACE Reference Guide* for more information.










## 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 7-5 for more information.

The Recipe Editor is described below.

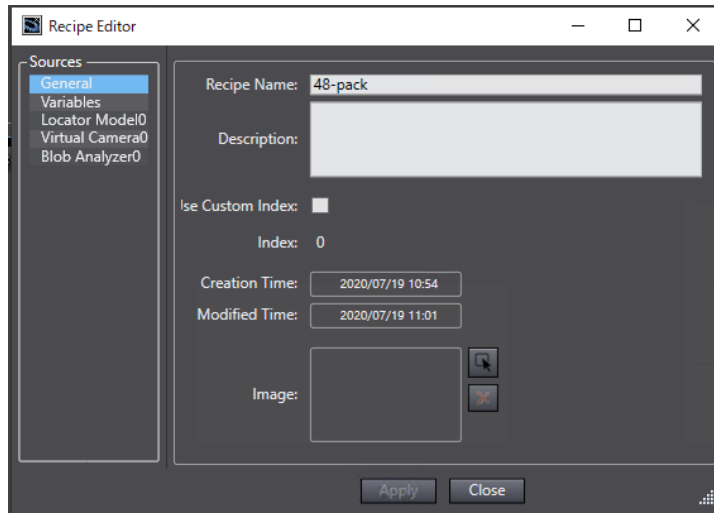


Item	Button	Description
Selected Recipe		Edit the selected Recipe.
		Clear the selected Recipe.
Available Recipes		Makes the highlighted Recipe in the Available Recipe list the active Recipe.
		Edit the selected Recipe.
		Add a new Recipe.
		Create a copy of the selected Recipe.
		Delete a selected Recipe.
		Open a Recipe from a saved file on the PC.
		Save a Recipe to a file on the PC.

## ● 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.



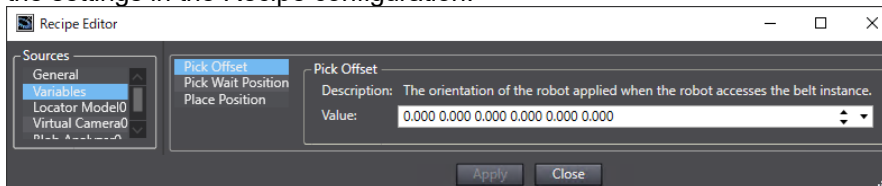
The setting items on the recipe editor are listed in the following table.

Item	Description
Recipe Name	Provide a unique name for the Recipe.
Description	Provide a description associated with the Recipe.
Index	If the Use Custom Index option is selected, you can 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 <i>ACE Reference Guide</i> 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.

- 3** Select **General** item in the Sources list and then input the general information and settings about 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.

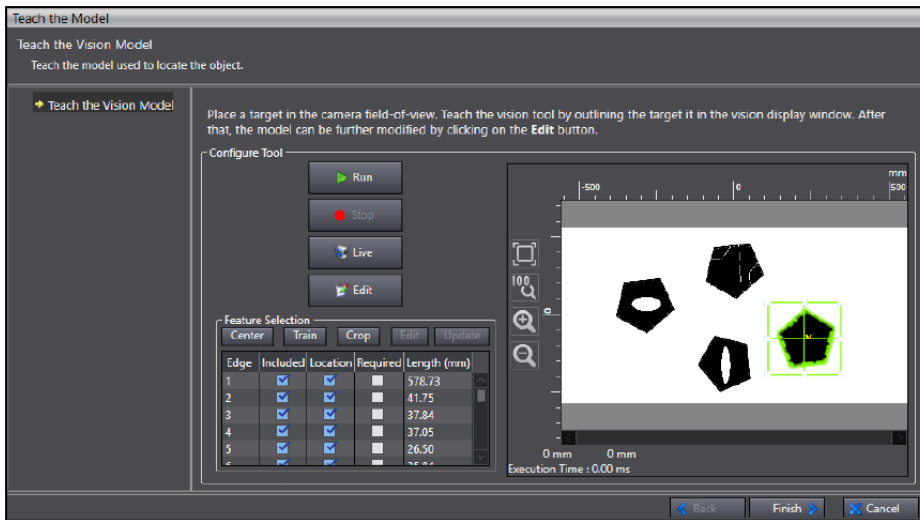
### ● V+ Variable Source

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.



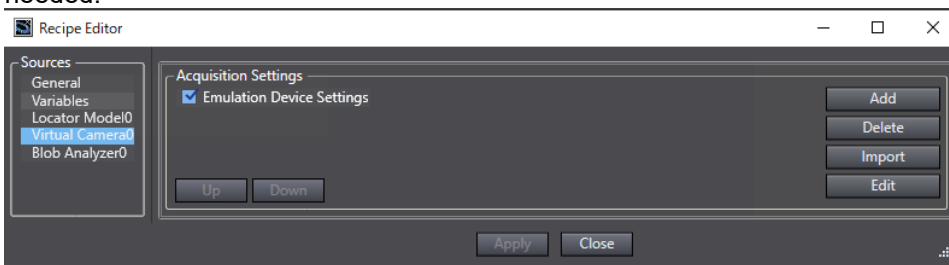
### ● Locator Model Source

You can see the currently trained locator model and can edit or retrain the locator model.



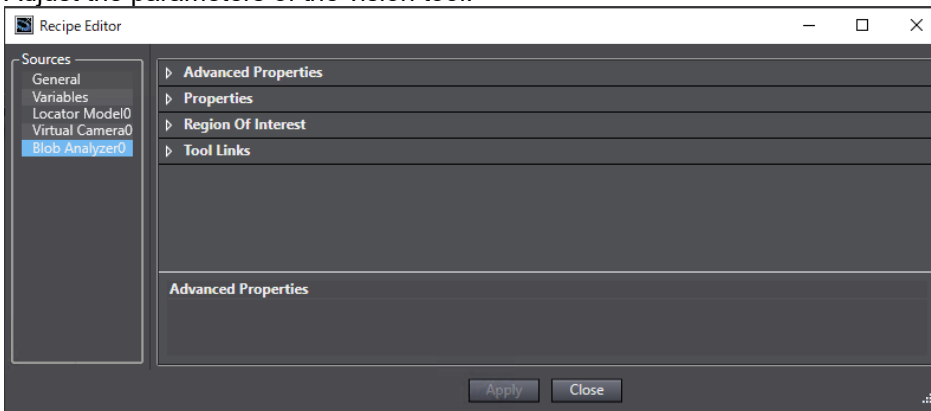
## ● Virtual Camera Source

The acquisition properties are displayed in a list. You can modify, add, or remove acquisition settings as needed.



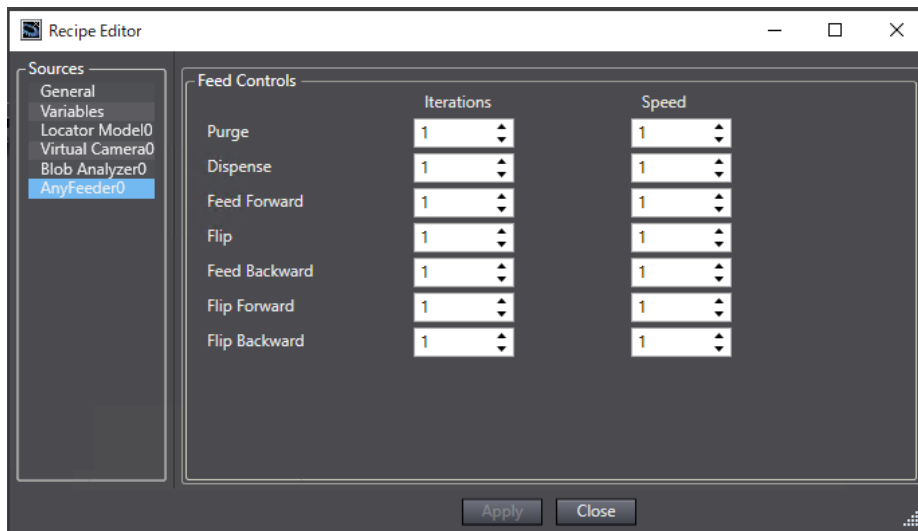
## ● Other Vision Tool Sources

Adjust the parameters of the vision tool.



## ● AnyFeeder Sources

Adjust the Feed Controls of the AnyFeeder.



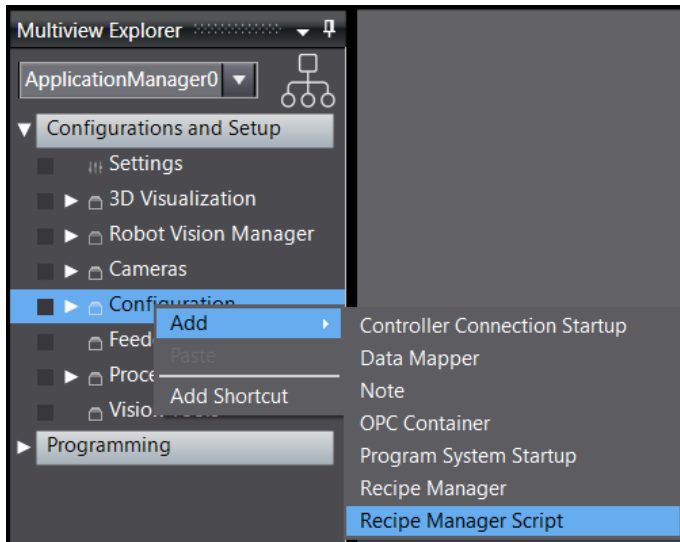
### 5-7-7 Recipe Manager Script

A Recipe Manager Script is C# program invoked when certain events occur in the lifetime of a Recipe. You can use it within Recipe Manager by selecting in **Script** of Recipe Manager. The following Script Methods are available in Recipe Manager.

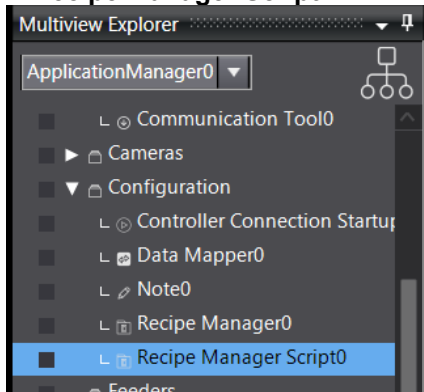
Script Method	Description
string CanEdit(Recipe recipe)	Called to check if a Recipe can be edited. If this method returns an empty string, the Recipe can be edited. If it returns a non-empty string, editing will be prevented and the string will be displayed.
void BeforeEdit(Recipe recipe)	If a Recipe 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

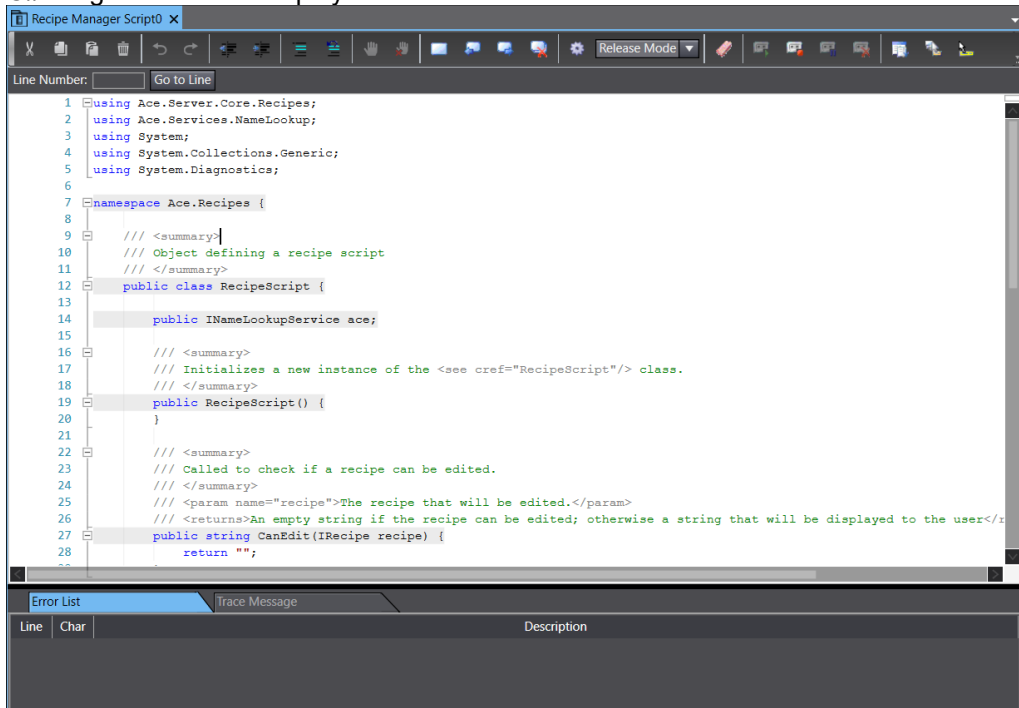
- 1 Right-click **Configuration** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Recipe Manager Script**.



A **Recipe Manager Script** is added under **Configuration**.



- 2 Right-click **Recipe Manager Script**, and then select **Edit**.  
C# Program Editor is displayed.



Refer to *5-11-1 C# Program Editor* on how to operate C# Program Editor.

# 5-8 Feeders

Feeders are objects that represent devices used to dispense and present parts to a robot. The usable Feeders are shown below.

Type	Description
AnyFeeder	AnyFeeder objects represent an integrated part feeding system optimized to work together with vision, motion, and robots.
IO Feeder	IO Feeder objects represent generic feeder devices that are controlled with input and output signals from the connected Robot Integrated CPU Unit.

## 5-8-1 AnyFeeder

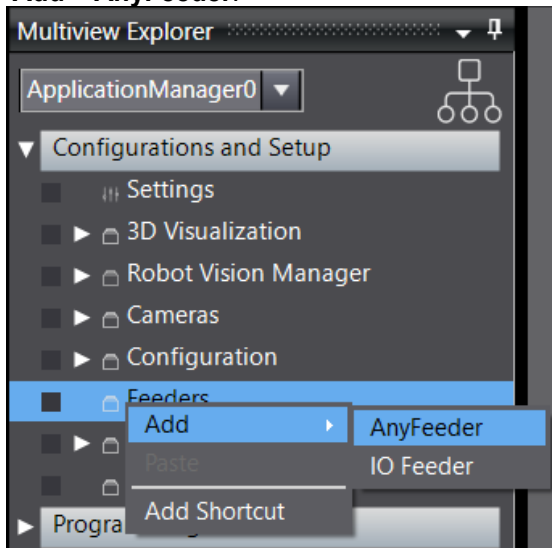
Represent an integrated parts feeding system optimized to work together with vision, motion, and robots. You can perform various settings and tests regarding AnyFeeder such as feed control, execution order, and etc.



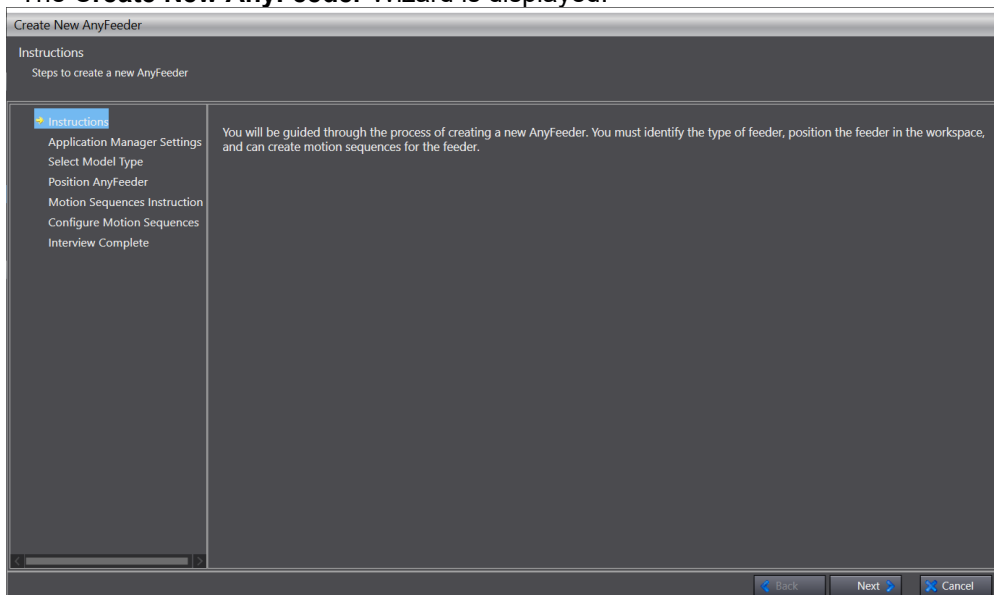
### Additional Information

Refer to the manual of AnyFeeder on its installation.

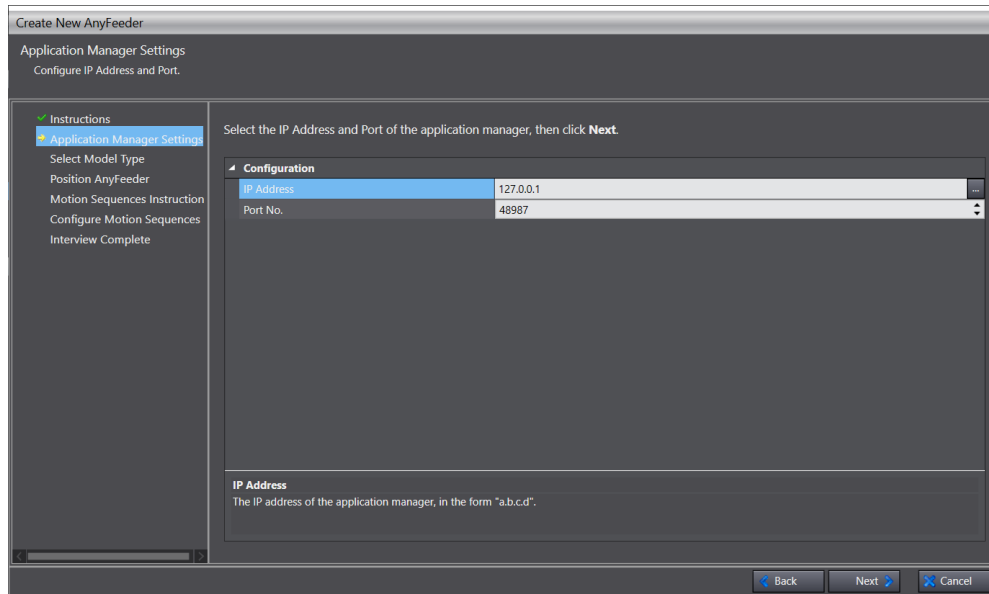
- 1 Right-click **Feeders** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – AnyFeeder**.



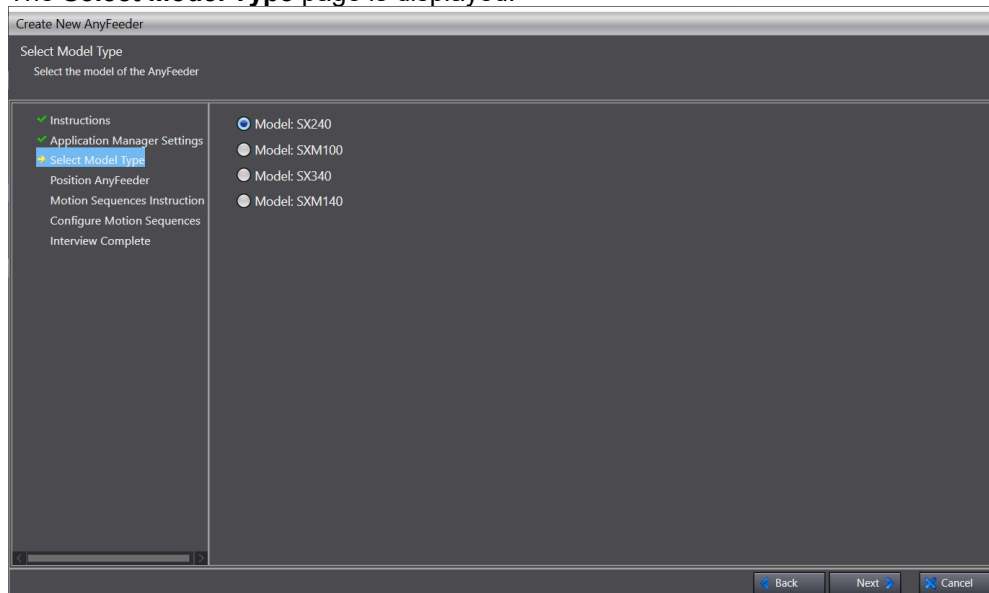
The **Create New AnyFeeder Wizard** is displayed.



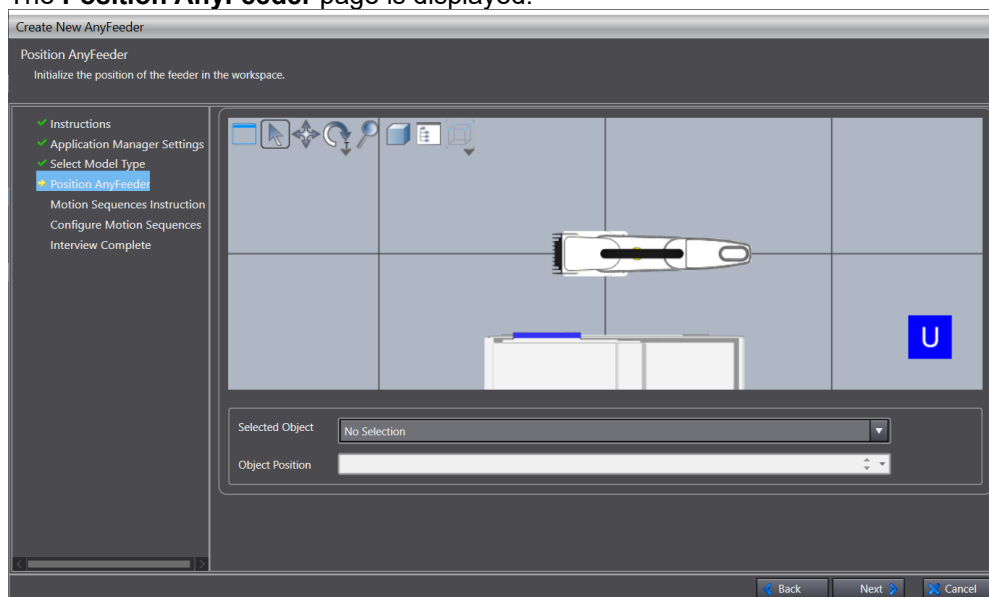
- 2 Click the **Next** button.  
The **Application Manager Settings** page is displayed.



- 3** Set the IP address and port of the Application Manager, and then click **Next** button. The **Select Model Type** page is displayed.

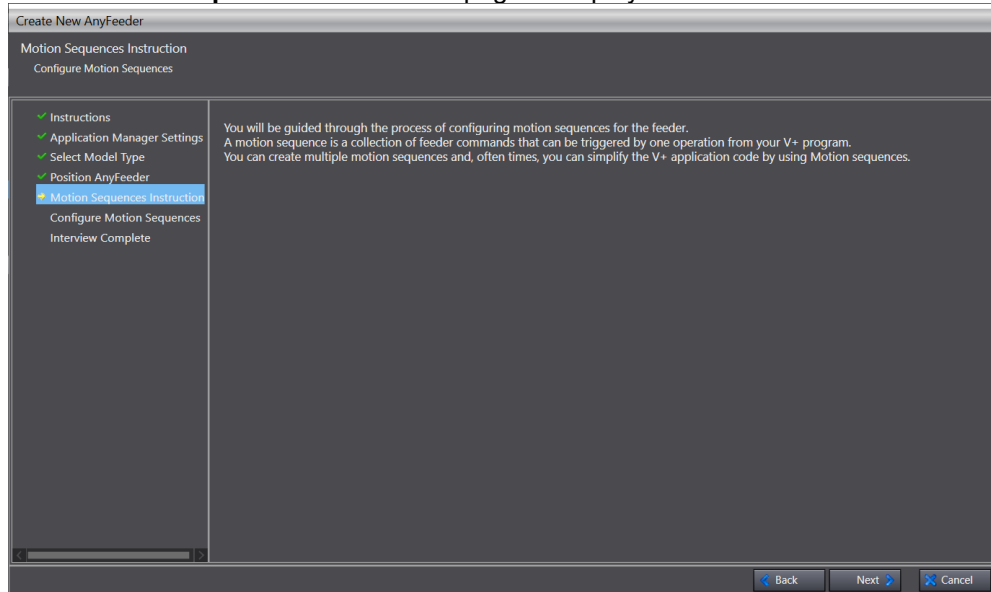


- 4** Select the check box of the type of AnyFeeder to use, and then click the **Next** button. The **Position AnyFeeder** page is displayed.

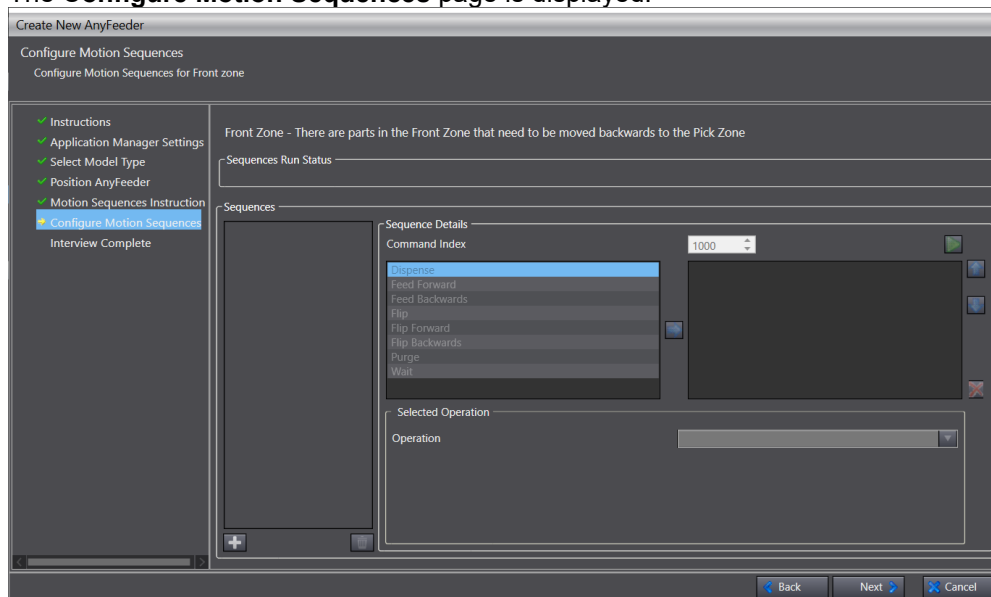




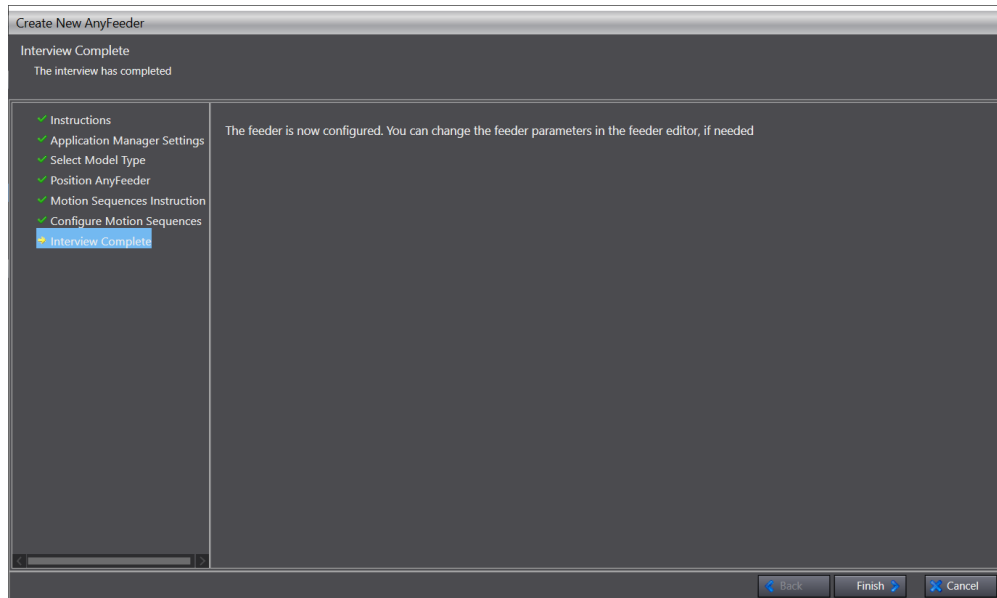
- 5 Specify the position of AnyFeeder in the workspace, and then click the **Next** button. The **Motion Sequences Instruction** page is displayed.



- 6 Click the **Next** button. The **Configure Motion Sequences** page is displayed.



- 7 Set the Motion Sequences of the front zones, and then click the **Next** button. Set the operations to be performed as a sequence. When the sequence is selected and 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# script, or with a V+ Robot Vision Manager protocol. The **Interview Complete** page is displayed.

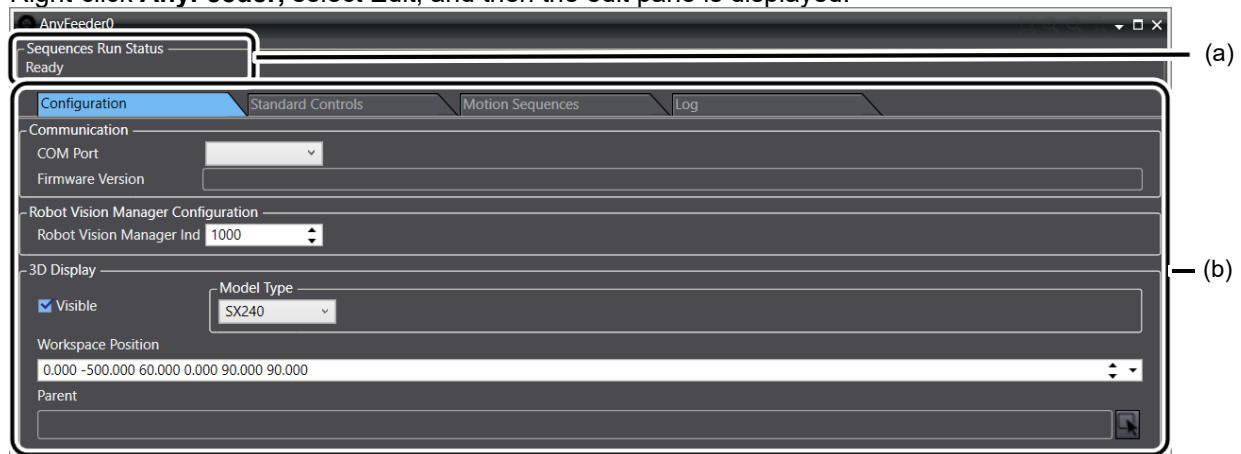


This completes the AnyFeeder settings.  
Click the **Finish** button, and then the **AnyFeeder** is added under **Feeder**.

## AnyFeeder Configuration

Various setting values are registered in **AnyFeeder** and can be edited.

Right-click **AnyFeeder**, select **Edit**, and then the edit pane is displayed.

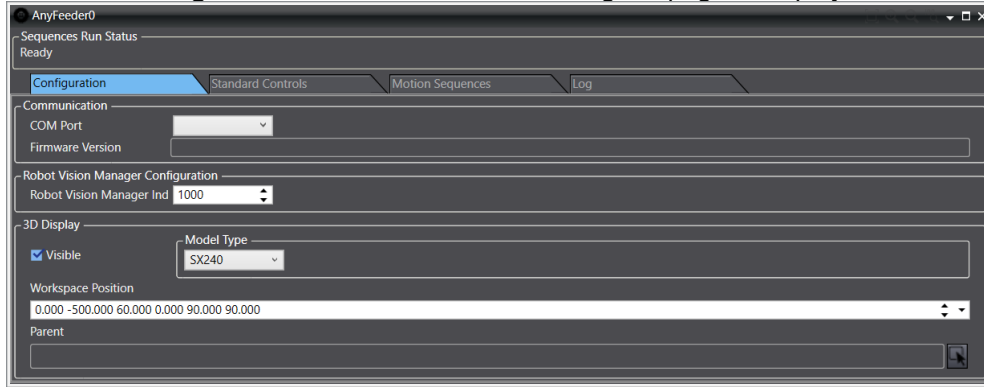


	Item	Description	Reference
(a)	Sequences Run Status	Information about operations and errors related to AnyFeeder object is displayed. The current status is shown in the top field. A description of the status is shown in the lower field.	-
(b)	Configuration	The general setting items of AnyFeeder are displayed.	<i>Configuration</i>
	Standard Controls	The setting items are used for manually controlling the AnyFeeder device.	<i>Standard Controls</i>
	Motion Sequences	The list of high level motion sequences associated with the AnyFeeder device is displayed. You can define a sequence as a collection of individual feeder functions. When a sequence is executed, all the operations are performed in the order defined in this area.	<i>Motion Sequences</i>
	Log	A history of the communications between the AnyFeeder device and the PC is displayed.	<i>Log</i>

### ● Configuration

There are general setting items of AnyFeeder

- 1 Click the **Configuration** tab, and then the following tab page is displayed.



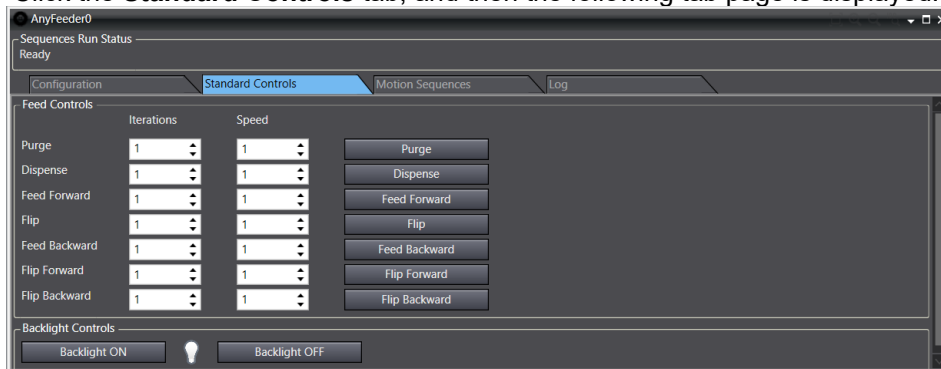
Item		Description
Communication	COM Port* <sup>1</sup>	Specify the serial communications port that is used to send commands to the AnyFeeder device.
	Firmware Version* <sup>1</sup>	Display the firmware version of the connected AnyFeeder device.
Robot Vision Manager Configuration	Robot Vision Manager Index	The index identifies Robot Vision Manager sequence and corresponds to the ID identifier of AnyFeeder. Specify a value for AnyFeeder that is different from the value used for the Robot Vision Manager sequence. Use a unique number from 0 to 9999.
3D Display	Visible	Select the check box to display AnyFeeder object in 3D Display.
	Model Type	Select the AnyFeeder Model type to display in 3D Display.
	Workspace Position	Enter the location of the AnyFeeder device in the workspace.
	Parent	Specify the 3D shape data that is the parent of AnyFeeder in the workspace.

<sup>1</sup>\* Not available in Emulation Mode.

## ● Standard Controls

There are setting items used for manually controlling AnyFeeder device.

- 1 Click the **Standard Controls** tab, and then the following tab page is displayed.



	Item	Description	
Feed Controls	Feeder Functions	Purge	Feed parts backward through purge gate.
		Dispense	Dispense parts from the part hopper to the conveyor belt
		Feed Forward	Feed parts forward along the conveyor belt.
		Flip	Flip parts on the belt without moving forwards or backwards.
		Feed	Feed parts backward along the conveyor belt.

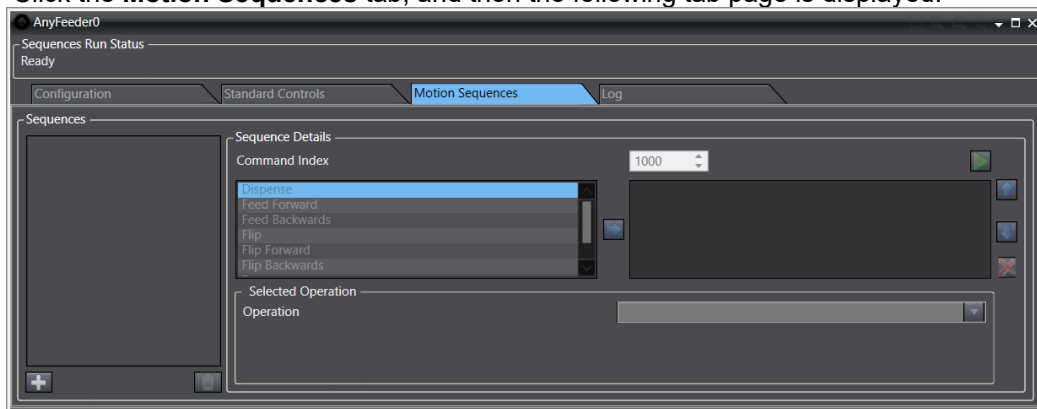
	Item	Description
	Backward	
	Flip Forward	Flip parts on the belt and move them forward.
	Flip Backward	Flip parts on the belt and move them backward.
	Iterations	Specify the number of times this action is performed. There are some actions that are designed to occur at the end of all motion and will only 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.
	Run button	Click the button to execute the selected function at the set speed and the set number of times.
Backlight Controls	Backlight ON/OFF	Turns the optional backlight on or off.

## ● Motion Sequences

The list of high level motion sequences associated with the AnyFeeder device is displayed. You can define a sequence as a collection of individual feeder functions. When a sequence is executed, all the operations are performed in the order defined in this area.

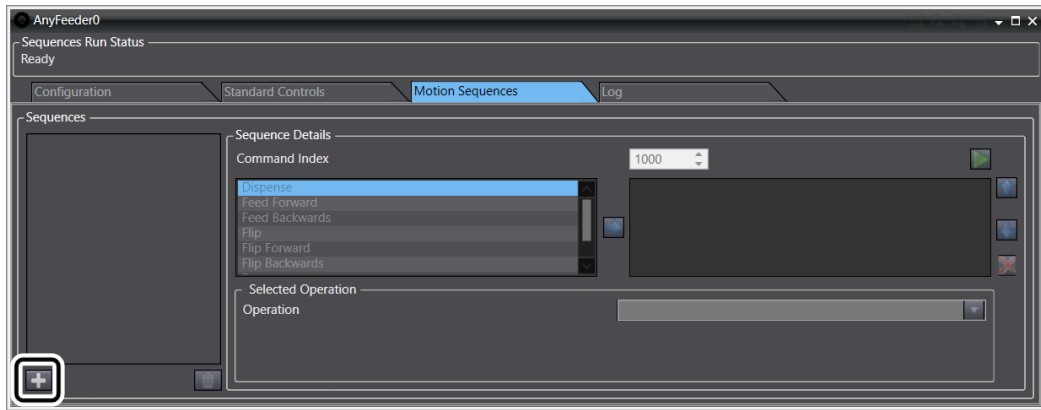
Motion sequences can be triggered through the AnyFeeder user interface, a C# program, or with a V+ program. Motion sequences are stored as Command Index numbers between 1000 to 10000. The motion sequence is referenced with this number in C# and V+ programs.

- 1 Click the **Motion Sequences** tab, and then the following tab page is displayed.



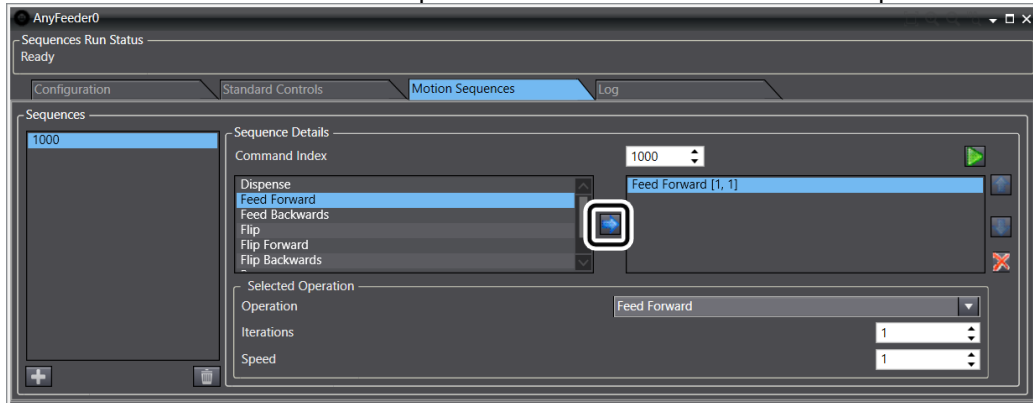
	Item	Description
	Sequences	It is the command index. Use Add button to add a new sequence.
	Sequence Details	Set sequence to each command. Select the feeder function to be executed first and click Select button for adding to the action list.
	Play button	Test the sequence.
	Up/ Down button	Rearrange actions of the sequence.
	Delete button	Delete the sequence and its steps.
	Select button	Add the selected command to the action list.

- 2 Click **Add** button in the Sequences to add a new sequence.



A new sequence is allocated in the sequence list.  
Adjust the command index if necessary.

- 3 Select the feeder function to be executed first, and then click **Select** button. It is added to the action list. The Flip function has been added in the example below.



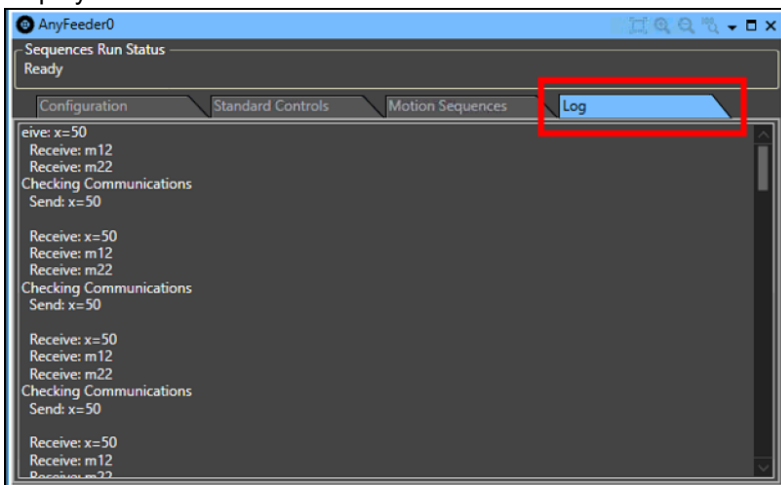
- 4 Select the action from the Selected Operation list and then make any necessary adjustments to Iterations and Speed.

- 5 Repeat steps 3 through 4 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

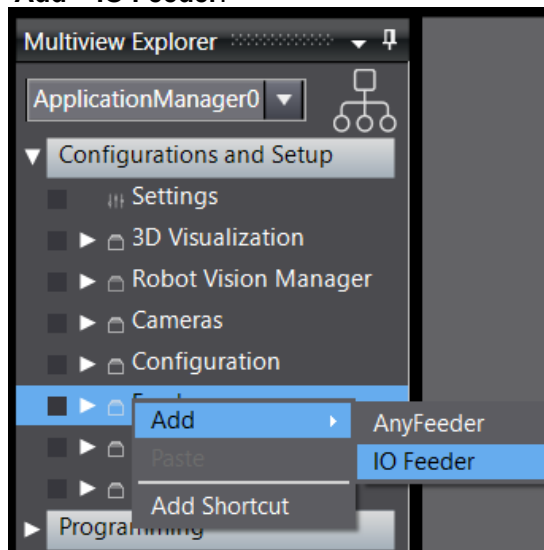
The log tab page shows A history of the communications between the AnyFeeder device and the PC is displayed.



## 5-8-2 IO Feeder

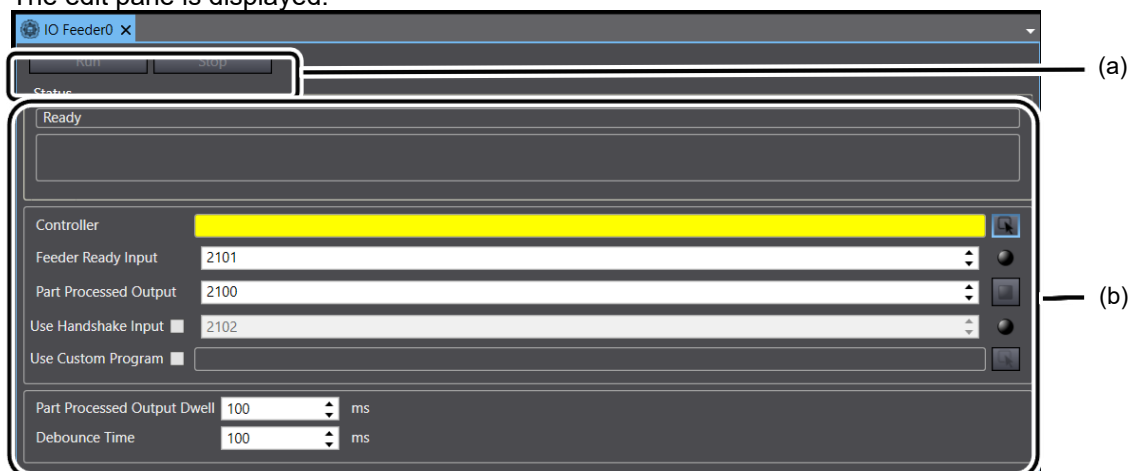
IO Feeder objects represent generic feeder devices that are controlled with input and output signals from a connected Robot Integrated CPU Units.

- 1 Right-click **Feeder** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – IO Feeder**.



An **IO Feeder** is added under **Feeder**.

- 2 Right-click **IO Feeder**, and then select **Edit**. The edit pane is displayed.



The setting items on the edit pane are listed in the following table.

	Item	Description
(a)	<b>Run</b> button	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.
	<b>Stop</b> button	Stop (interrupts) the test cycle.
(b)	Status	Provide operation and error information about the IO Feeder object. The current status is shown in the top field. A description of the status is shown in the lower field.
	Controller	Specify the controller that will process the feeder signals.
	Feeder Ready Input <sup>*1</sup>	Specify the input signal that indicates the feeder is ready and available to present a part instance.
	Part Processed Output <sup>*1</sup>	Specify the output signal that indicates the instance has been processed (acquired) by the robot.
	Use Handshake Input <sup>*1</sup>	If enabled, the feeder will assert a signal indicating it has acknowledged the part processed signal.

	Item	Description
	Use Custom Program *1	If you enable this option, the selected 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.
	Part Processed Output Dwell	Specify the dwell time (time to wait) in milliseconds after the Part Processed output signal is turned ON before turning it OFF.
	Debounce Time	Specify the amount of time (in milliseconds) that a signal must be detected in the ON state before it is considered logically ON.

\*1 Not available in Emulation Mode. You can run tests regardless of signal conditions and settings.

# 5-9 Process

Based on the specified execution order, configure various settings of Process to pick parts and place on the target position.

The setting items of Process are listed in the following table.

Item	Description	Reference
Part Buffer	Define an object that is an overflow area where parts can be temporarily stored when part targets are unavailable to accept more parts.	<i>5-9-1 Part Buffer</i>
Part Target	Define an object that is a possible destination for a part.	<i>5-9-2 Part Target</i>
Part	Define the shape of the part and the conditions for picking and placing the part with the robot.	<i>5-9-3 Part</i>
Belt	Define a physical conveyor belt used by the system. The Belt object maintains a list of encoders that are associated with the physical conveyor.	<i>5-9-4 Belt</i>
Process Manager	Process Managers are the central control point for developing packaging applications. A Process Manager allows you to create complex applications without having to write any programming code. It provides access to automatically generated V+ and C# programs that allow you to customize the default behavior to meet the requirements of your 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.	<i>5-9-5 Process Manager</i>
Allocation Script	It is used to create and edit custom part-allocation programs for use with the Process Manager.	<i>5-9-6 Allocation Script</i>
Pallet	Define the layout of a pallet, which can be used to pick parts from or place parts to.	<i>5-9-7 Pallet</i>
Vision Refinement Station	Define an object that is used to refine the part to gripper orientation for improved placement accuracy.	<i>5-9-8 Vision Refinement Station</i>

## 5-9-1 Part Buffer

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.

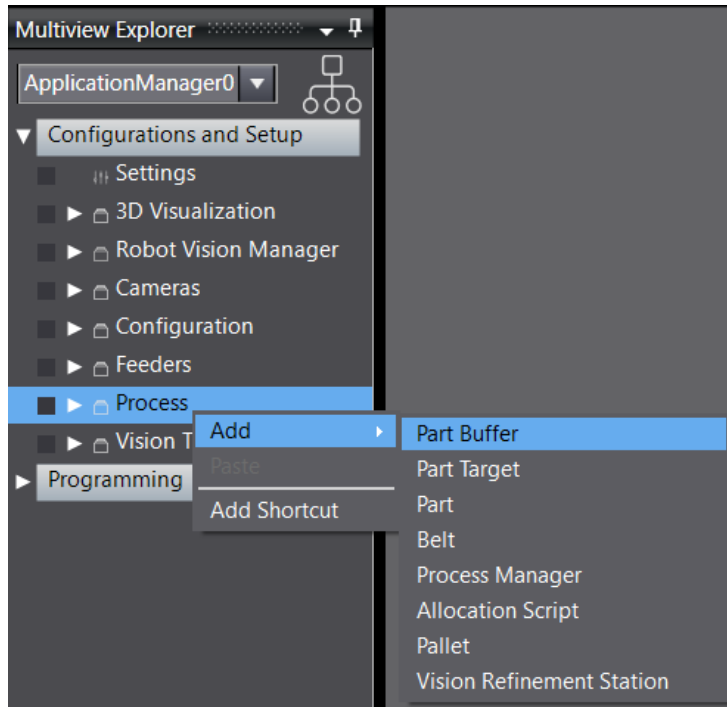


### Additional Information

Pallets can be used to carry parts. When using pallets, you can manage access methods such as LIFO (Last In, First Out) and FIFO (First In, First Out). If you want to use a pallet, register the pallet in advance. Refer to *5-9-7 Pallet* for more information.

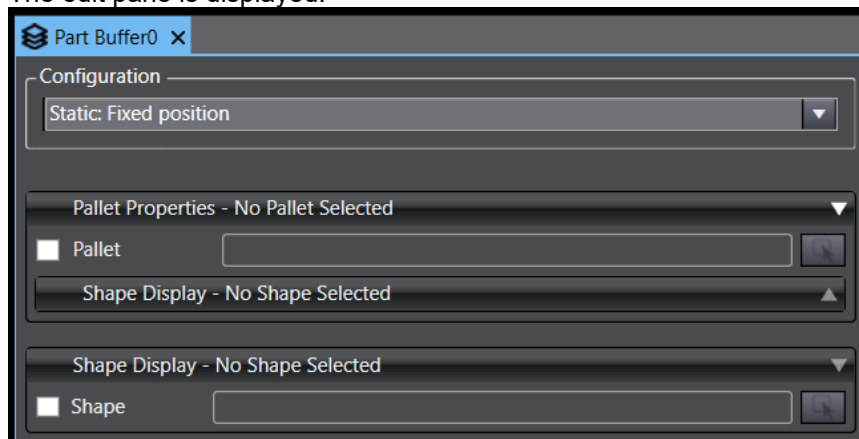
- 1 Right-click **Process** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Part Buffer**.





A **Part Buffer** is added under **Process**.

- 2 Right-click **Part Buffer**, and then select **Edit**.  
The edit pane is displayed.



The setting items on the edit pane are listed in the following table.

Item		Description
Configuration		Select Static: Fixed Position. Parts are placed at a static location.
Pallet Properties	Pallet	Select the check box and specify the pallet to use to move part.
	Shape Display	Select the check box to display the pallet in 3D Visualizer.
Shape Display	Shape	Select the check box and specify the shape of part to display in 3D Visualizer.

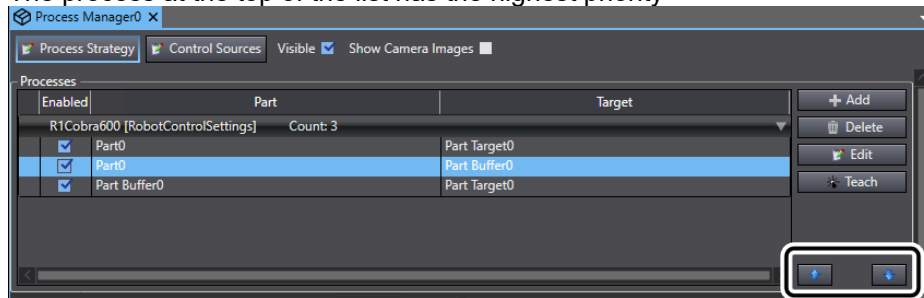
## Adding a Part Buffer to a Process

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

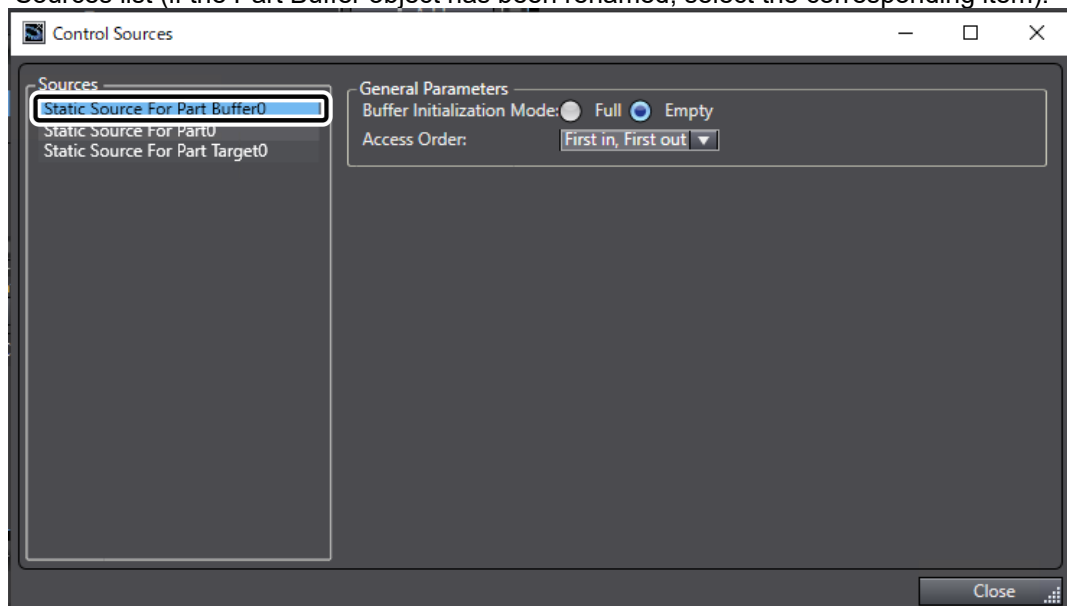
When a pallet is used for the part buffer, you need to specify how the parts will be accessed as the buffer is being emptied. You can choose between the following options.

- 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* 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).

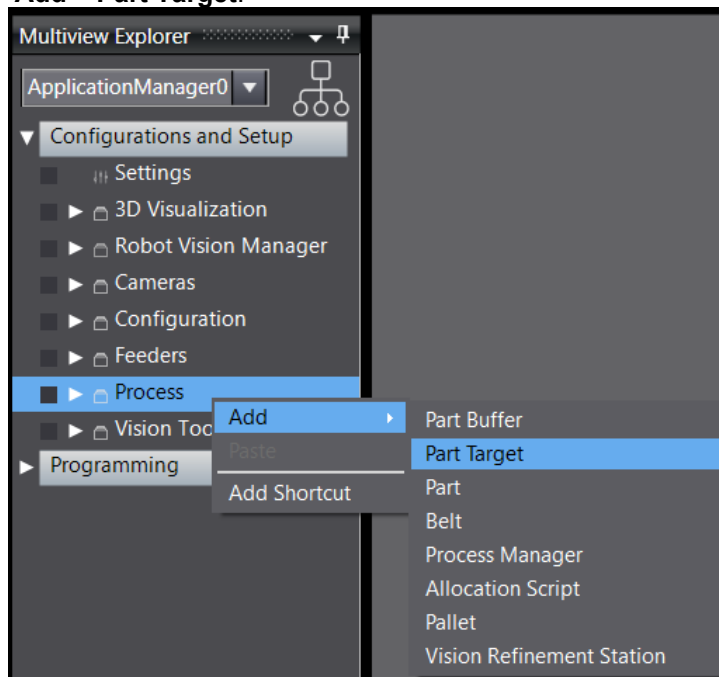


- 2 If necessary, select the desired Buffer Initialization Mode to indicate the state of the part buffer when it is initialized.  
The default state is Empty, which means the buffer is empty when initialized.
- 3 Select the required Access Order from the list and then the Part Buffer access order setting procedure is complete.

## 5-9-2 Part Target

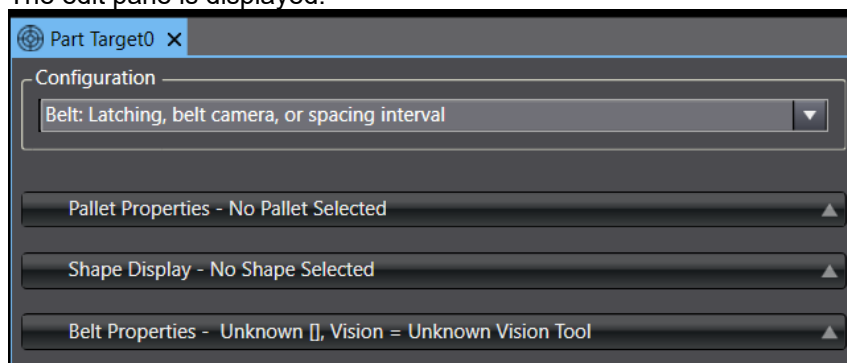
Defines a location that is a possible destination for a part.

- 1 Right-click **Process** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Part Target**.



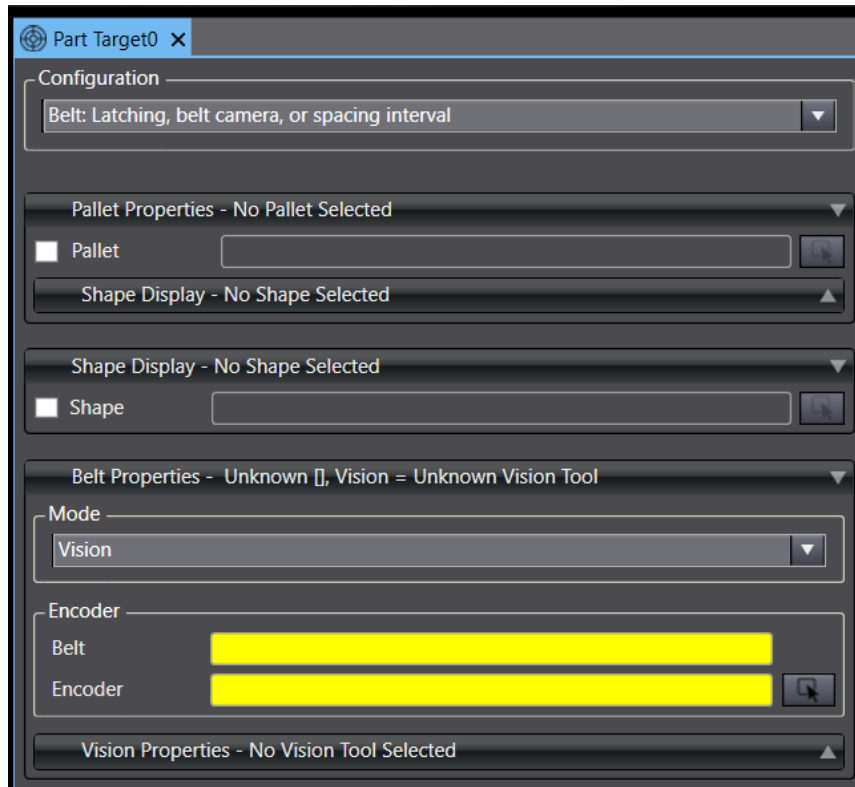
A **Part Target** is added under **Process**.

- 2 Right-click **Part Target**, and then select **Edit**.  
The edit pane is displayed.



### Part Target Configuration

The setting items on the edit pane are listed in the following table.



Item		Description	
Configuration		Select a method to pick and place the part.	
Pallet Properties (Option) <sup>*1*2*</sup> <sub>3</sub>	Pallet	Select the check box to use pallet to move part and specify the pallet.	
	Shape Display	Shape	Select the check box and specify the shape of pallet to display the pallet in 3D Visualizer.
Shape Display <sup>*1*2*3</sup>	Shape	Select the check box and specify the shape of part to display the part in 3D Visualizer.	
Belt Properties <sup>*1</sup>	Mode		Select the operation mode of belt.
	Encoder <sup>*4*5*6</sup>	Belt	Specify the belt to use for handling part.
		Encoder	Specify the belt encoder to use for controlling belt.
	Vision Properties <sup>*4</sup>	Vision Tool	Specify the Vision Tool to detect part on the belt.
		Use Named Instance (Option)	Select the check box to display the instance with name when detecting part. It has usually the same name as the model name.
	Latch Period in Emulation Mode <sup>*7</sup>		Specify the latch period.
Spacing <sup>*6</sup>		Specify the spacing on the belt between parts in mm.	
Vision Properties <sup>*3</sup>	Vision Tool	Specify the Vision Tool to detect part on the belt.	
	Use Named Instance	Select the check box to display the instance with name when detecting part. It has usually the same name as the model name.	

\*1. Displayed when **Belt** is selected in **Configuration**.

\*2. Displayed when **Static** is selected in **Configuration**.

\*3. Displayed when **Vision** is selected in **Configuration**.

\*4. Displayed when **Vision** is selected in **Belt Properties - Mode**.

\*5. Displayed when **Latch** is selected in **Belt Properties - Mode**.

\*6. Displayed when **Spacing** is selected in **Belt Properties - Mode**.

\*7. Displayed when **Latch** is selected in **Belt Properties - Mode** when running in Emulation Mode.

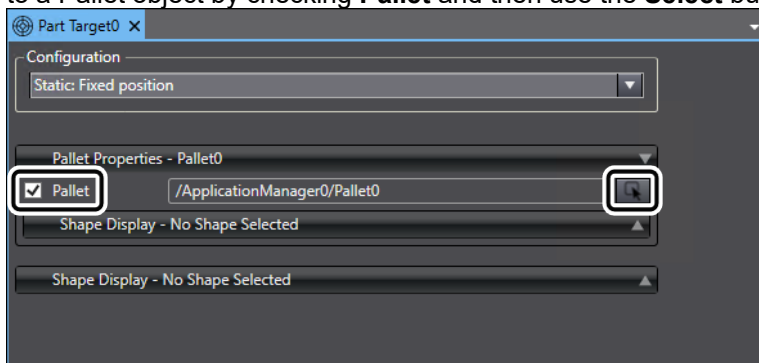
## Configuration

**Configuration** drop-down list box is used to specify how the target is input to the system. The options are described below. The options are described below.

Item	Description
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 <i>Belt: Latching, Belt Camera, or Spacing Interval Configuration</i> for more information.
Static: Fixed position	Targets / instances are acquired from a static location such as a part feeder or a pallet. Refer to <i>Static: Fixed Position Configuration</i> 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 <i>Vision: Fixed camera not relative to belt</i> for more information.

## Pallet Properties

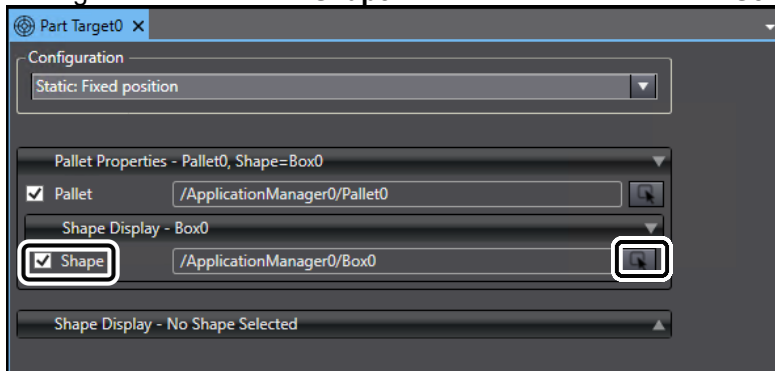
Use the Pallet Properties area if you need to specify a pallet that is used to hold the part(s). Make reference to a Pallet object by checking **Pallet** and then use the **Select** button to specify that object as shown below.



A Pallet object must already exist in the project. Refer to *5-9-7Pallet* for more information.

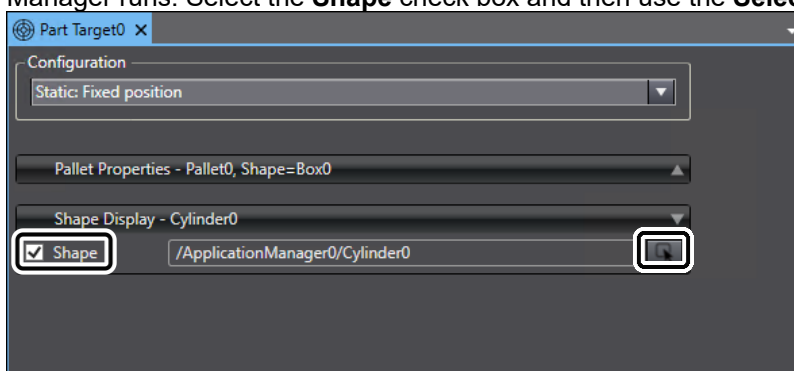
### ● Shape Display (Pallet Properties)

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.

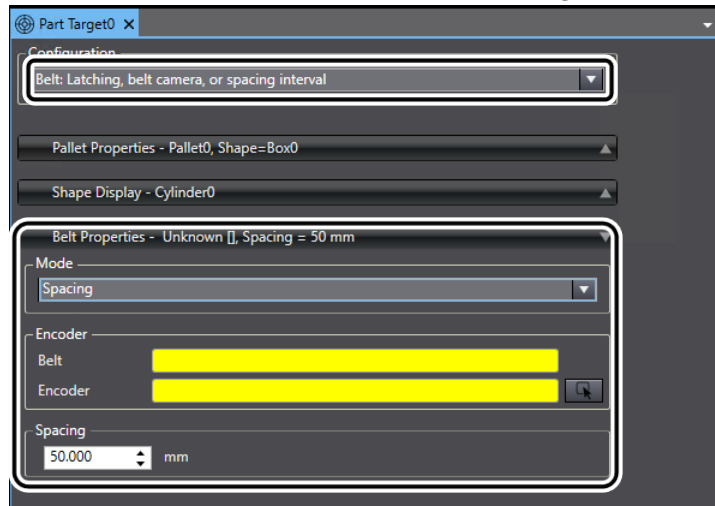
Vision: A vision tool is used to detect the part on the belt.

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. You also 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.

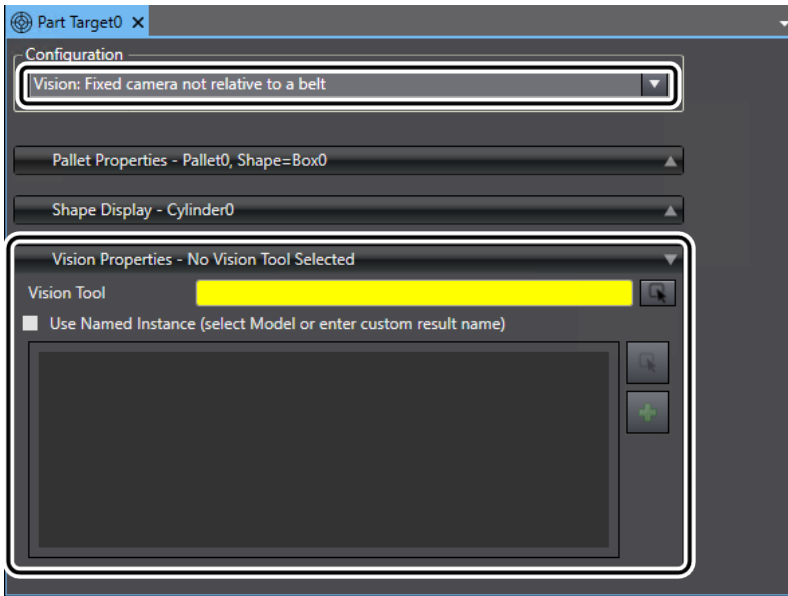
## Vision: Fixed camera not relative to belt

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.

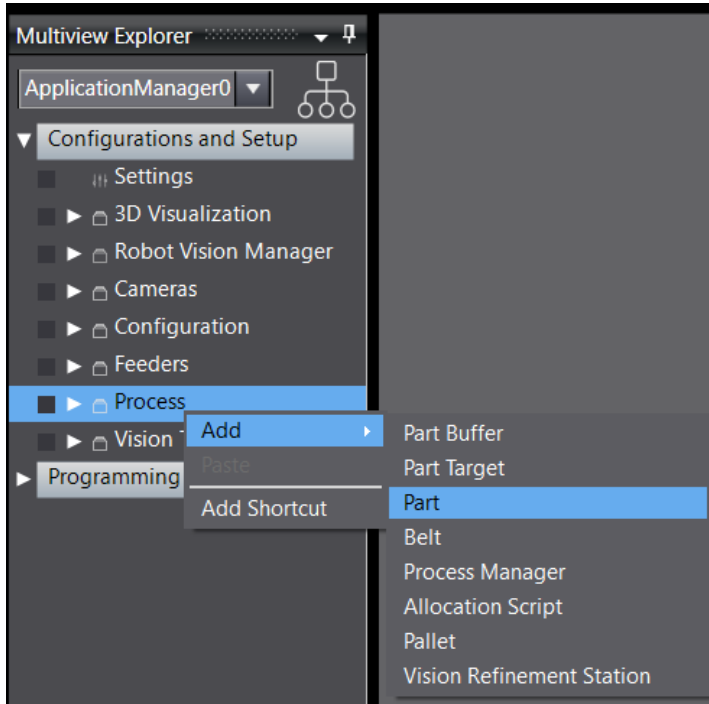
You can optionally specify a named instance and then select a Model or enter a custom result name.



### 5-9-3 Part

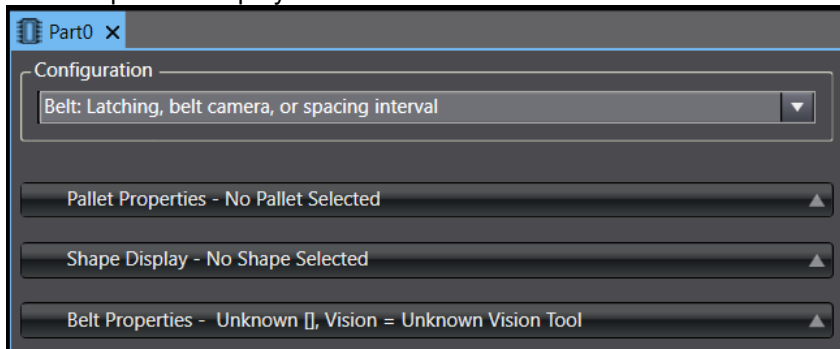
Define the shape of the part and the conditions for picking and placing the part with the robot.

- 1 Right-click **Process** under **Configuration and Setup** in the Multiview Explorer, and then select **Add - Part**.



A **Part** is added under **Process**.

- 2 Right-click **Part**, and then select **Edit**.  
The edit pane is displayed.



The setting items on the edit pane are listed in the following table.

Item		Description	
Configuration		Select a method to pick and place the part. The available methods are: Belt: Pick the part from the belt or place it on the belt. Use the latch, camera, or spacing to determine position. Static: Pick the part from the fixed position or place it on the fixed position. No camera or belt is used due to fixed position. Vision: Use a fixed position camera to determine position to pick or place the part. No belt is used. The properties that can be set vary depending on the selected mode.	
Pallet Properties (Option) <sup>*1*2*3</sup>	Pallet	Select the check box to use pallet to move part and specify the pallet.	
	Shape Display	Shape	Select the check box and specify the shape of pallet to display the pallet in 3D Visualizer.
Shape Display <sup>*1*2*3</sup>	Shape	Select the check box and specify the shape of part to display the part in 3D Visualizer.	
Belt Properties <sup>*1</sup>	Mode		Select the belt to use for handling part and enter description about its mode. The properties that can be set vary depending on the selected mode.
	Encoder <sup>*4*5*6</sup>	Belt	Specify the belt to use for handling part.
		Encoder	Specify the belt encoder to use for controlling belt.
	Vision Properties <sup>*4</sup>	Vision Tool	Specify the Vision Tool to detect part on the belt.
		Use Named Instance (Option)	Select the check box to display the instance with name when detecting part. It has usually the same name as the model name.
	Latch Period in Emulation Mode <sup>*7</sup>		Specify the latch period.
Spacing <sup>*6</sup>		Specify the spacing on the belt between parts in mm.	
Vision Properties <sup>*3</sup>	Vision Tool		Specify the Vision Tool to detect part on the belt.
	Use Named Instance		Select the check box to display the instance with name when detecting part. It has usually the same name as the model name.

\*1. Displayed when Belt is selected in Configuration.

\*2. Displayed when Static is selected in Configuration.

\*3. Displayed when Vision is selected in Configuration.

\*4. Displayed when Vision is selected in Belt Properties - Mode.

\*5. Displayed when Latch is selected in Belt Properties - Mode.

\*6. Displayed when Spacing is selected in Belt Properties - Mode.

\*7. Displayed when Latch is selected in Belt Properties – Mode when running in Emulation Mode.

## 5-9-4 Belt

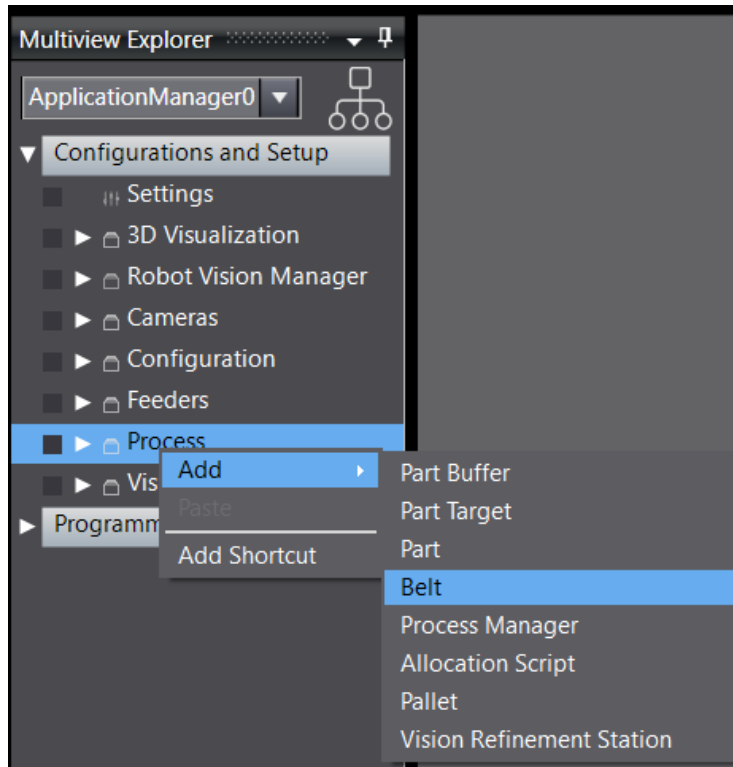
A Belt object represents a physical conveyor belt in the workcell or packaging line. A belt may be tracked by multiple robots that may be controlled by a single or multiple controllers. Belts may also be related to multiple part or part target objects. After setting the belt control, place it on the 3D Visualizer and associate the latch and controller with the belt.

### Adding a Belt

Use the following procedure to add a belt.

- 1 Right-click **Process** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Belt**.



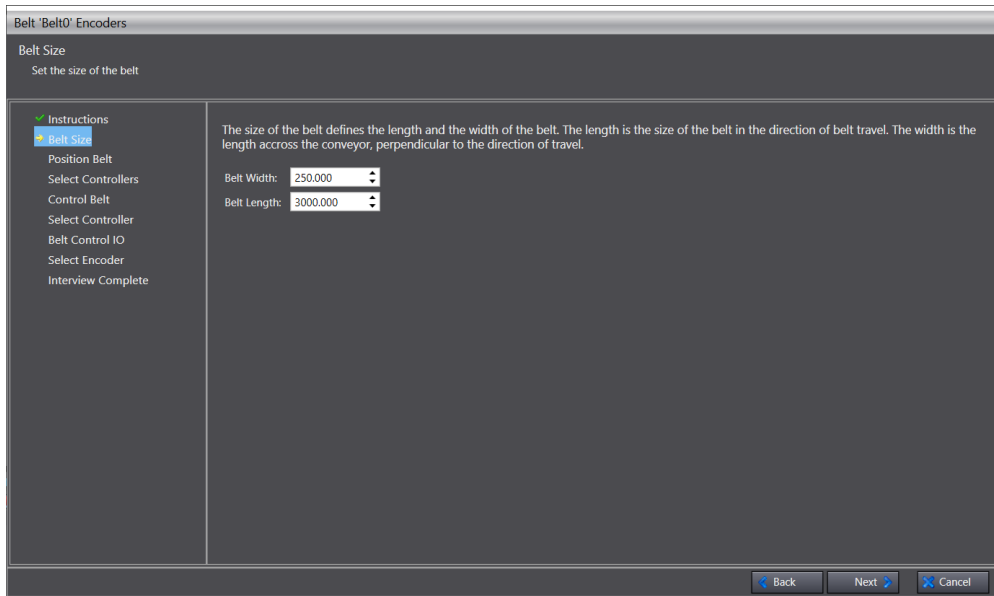


A confirmation message is displayed while offline. Follow the message and click the **OK** or **Cancel** button.

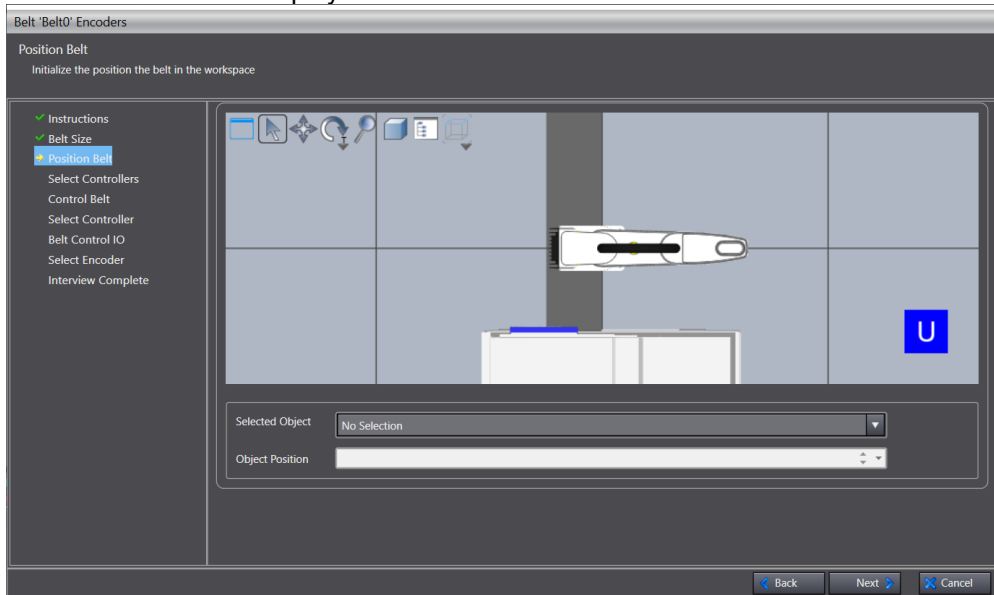
Click the **OK** button, then the Belt wizard is displayed.



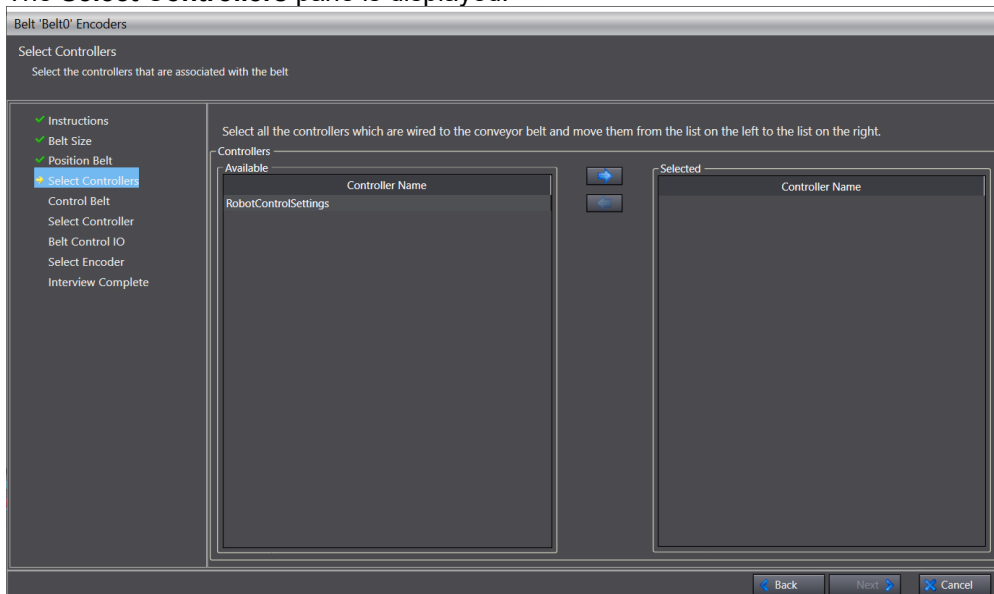
- 2** Click the **Next** button.  
The **Belt Size** pane is displayed.



- 3 Specify the belt width and the belt length, and then click the **Next** button. The **Position Belt** is displayed.



- 4 Specify the position of the belt in the workspace, and then click the **Next** button. The **Select Controllers** pane is displayed.



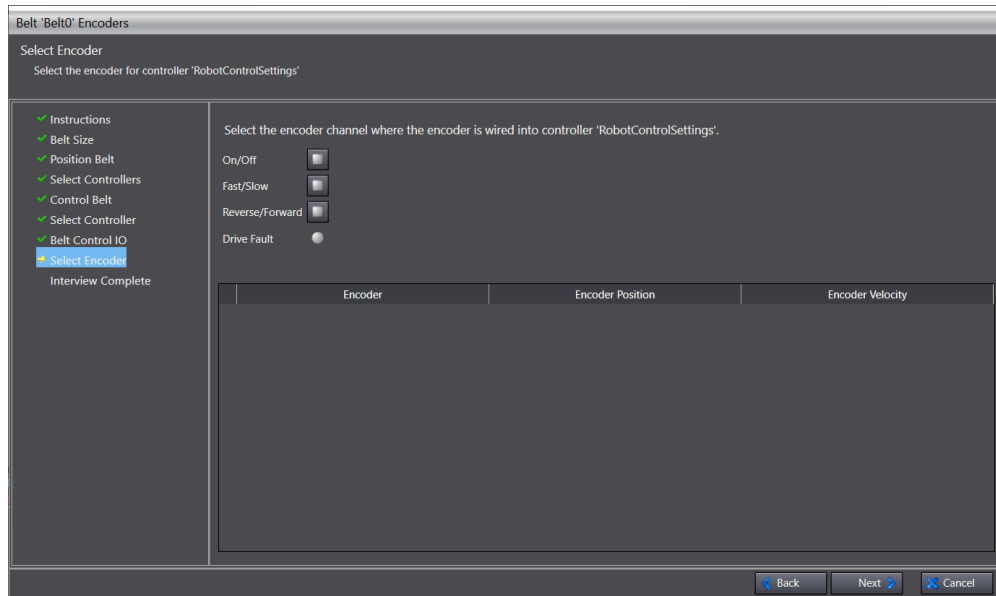
- 5 Specify the controller to be associated with the belt, and then click the **Next** button. The **Control Belt** pane is displayed.

The screenshot shows a window titled "Belt 'Belt0' Encoders" with a sub-header "Control Belt". Below the sub-header is the question: "Is the belt controlled by digital I/O wired into a controller?". On the left side, there is a vertical list of steps: "Instructions", "Belt Size", "Position Belt", "Select Controllers", "Control Belt" (highlighted in blue), "Select Controller", "Belt Control IO", "Select Encoder", and "Interview Complete". On the right side, there are two radio button options: "Yes, the belt is being controlled" (which is selected) and "No, the belt is not being controlled". At the bottom right, there are three buttons: "Back", "Next", and "Cancel".

- 6 Check if the belt is controlled by the digital I/O connected to the controller, select the corresponding check box, and then click the **Next** button. When the **Yes, the belt is being controlled** is checked, the **Select Controller** pane is displayed. Go to No.7.

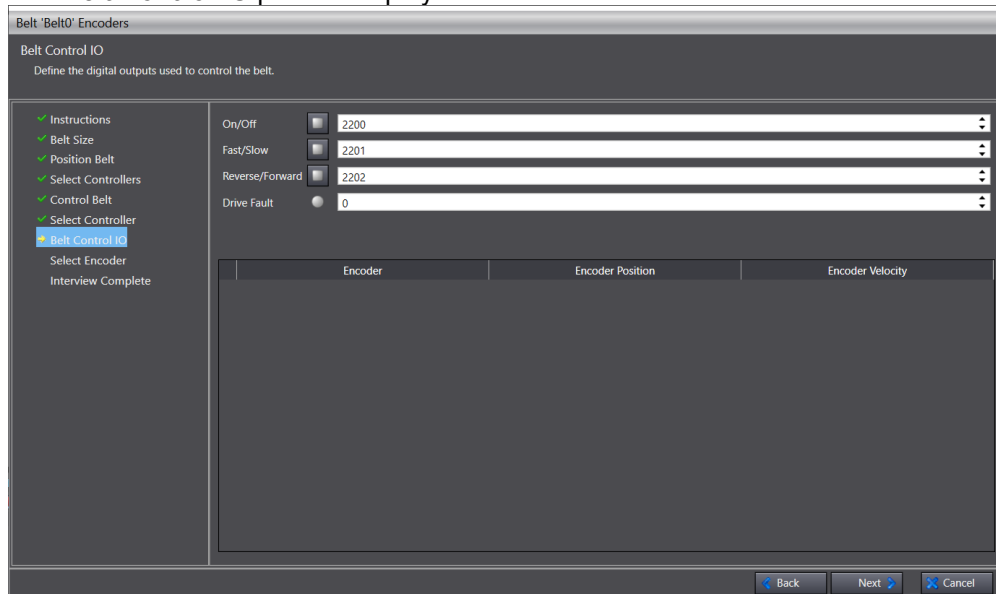
The screenshot shows a window titled "Belt 'Belt0' Encoders" with a sub-header "Select Controller". Below the sub-header is the instruction: "Select the controller that the belt I/O is wired into.". On the left side, there is a vertical list of steps: "Instructions", "Belt Size", "Position Belt", "Select Controllers", "Control Belt", "Select Controller" (highlighted in blue), "Belt Control IO", "Select Encoder", and "Interview Complete". On the right side, there is a text input field labeled "Controller" containing the text "/new\_Controller\_0/Controller Settings". At the bottom right, there are three buttons: "Back", "Next", and "Cancel".

When the **No, the belt is not being controlled** is checked, the **Select Encoder** pane is displayed.



Go to No.9.

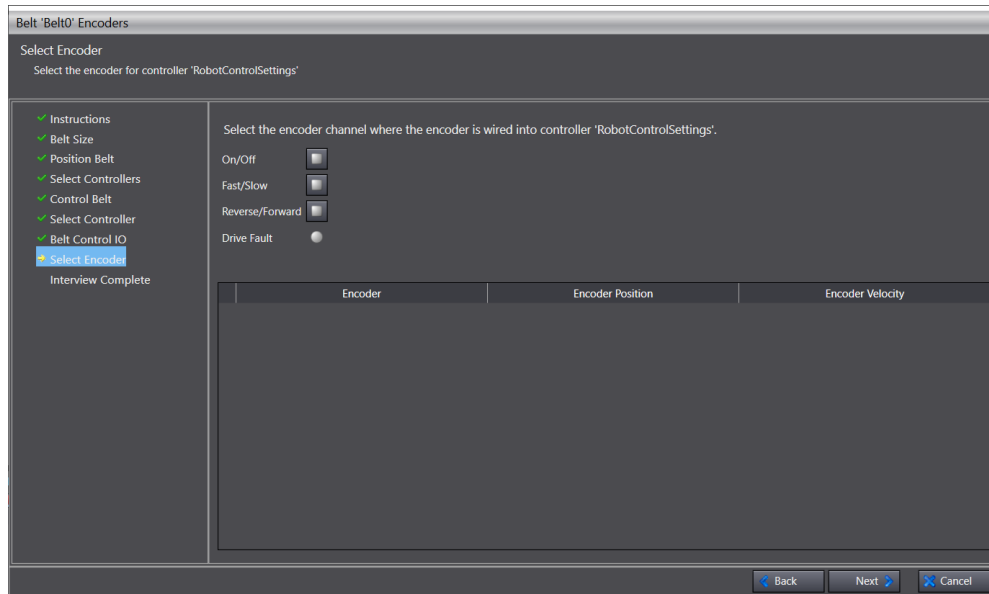
- 7** Specify the controller connected, and then click the **Next** button. The **Belt Control IO** pane is displayed.



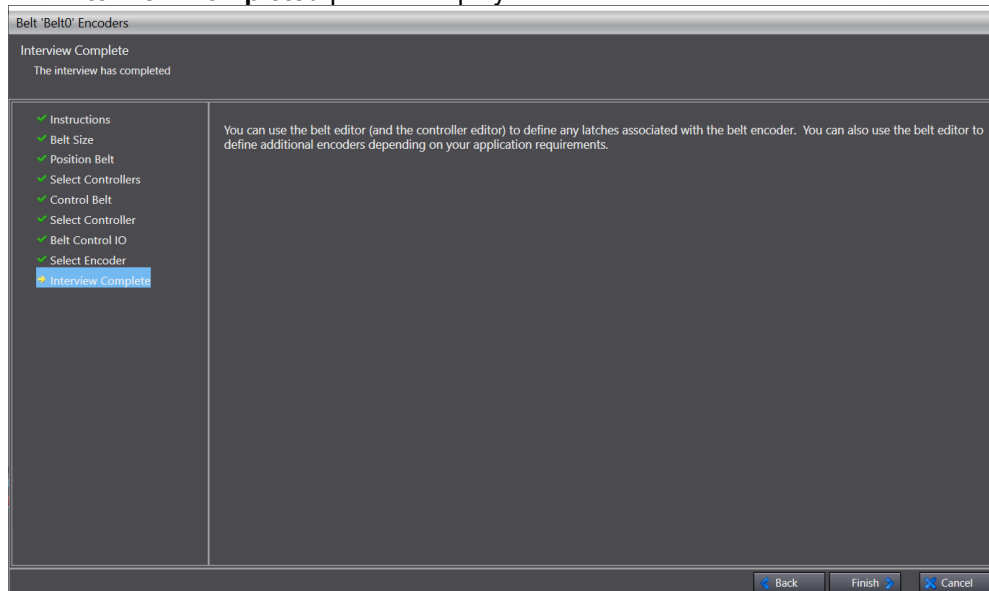
- 8** Set the digital output used to control the belt, and then click the **Next** button. The items and contents that can be set are as follows.

Item	Description
On/Off	Specify the digital I/O signal number used to start and stop the belt. When the signal is turned on, the belt moves.
Fast/Slow	If the belt supports high speed (Fast) and low speed (Slow), specify the digital I/O signal number used to switch between Fast and Slow of the belt. When the signal is turned on, the belt moves at high speed.
Reverse/Forward	If the belt supports forward and reverse, specify the digital I/O signal number used to switch between forward and reverse of the belt. When the signal is turned on, the belt moves in the opposite direction.
Drive Fault	If the belt control fails, this display turns on to indicate that an error has occurred.

The **Select the Encoder** pane is displayed.



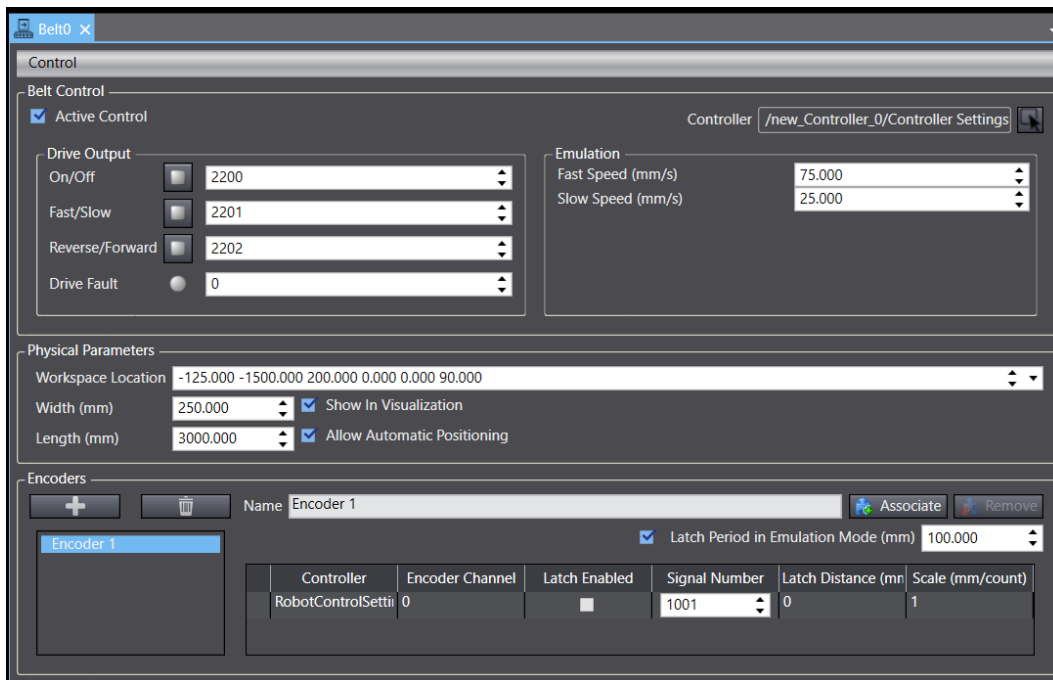
- 9 Select the encoder channel to connect to the controller, and click the **Next** button. The **Interview Completed** pane is displayed.



This completes the belt object creation. When you create a new belt object, a **Belt** is added under **Process**.

## Belt Configuration

Various setting values are registered in **Belt**. Right-click the **Belt** to edit, and select **Edit**, and then the edit pane is displayed.



The setting items on the edit pane are listed in the following table.

Item		Description	
Control	View Belt Encoders	Display the edit pane of the belt encoder and check the value when moving the belt.	
	Test Latch Signal	Display the wizard to test the encoder latch signal.	
Belt Control	Active Control	Select the check box to control the robot by the specified controller.	
	Controller	Specify the robot controller used to control the belt.	
	Drive Output	On/Off	Specify the digital I/O signal number used to start and stop the belt. When the signal is turned on, the belt moves.
		Fast/Slow	If the belt supports high speed (Fast) and low speed (Slow), specify the digital I/O signal number used to switch between Fast and Slow of the belt. When the signal is turned on, the belt moves at high speed.
		Reverse/Forward	If the belt supports forward and reverse, specify the digital I/O signal number used to switch between forward and reverse of the belt. When the signal is turned on, the belt moves in the opposite direction.
		Drive Fault	If the belt control fails, this display turns on to indicate that an error has occurred.
Emulation *1	Fast Speed (mm/s)	When the Fast Speed is enabled, specify the distance to mover per second.	
	Slow Speed (mm/s)	When the Slow Speed is enabled, specify the distance to move per second.	
Physical Parameters	Workspace Location	Enter the position of the belt in the workspace. The position is shown in the world coordinate system.	
	Width (mm)	Specify the width of the belt.	
	Length (mm)	Specify the length of the belt.	
	Show In Visualization	Select the check box to display the belt in 3D Visualizer.	
	Allow Automatic Positioning *1	The belt object in the workspace is automatically positioned.	
Encoders	<b>Add</b> button	Add a new encoder.	
	<b>Delete</b> button	Delete the encoder.	
	Name	Edit the name of the encoder.	

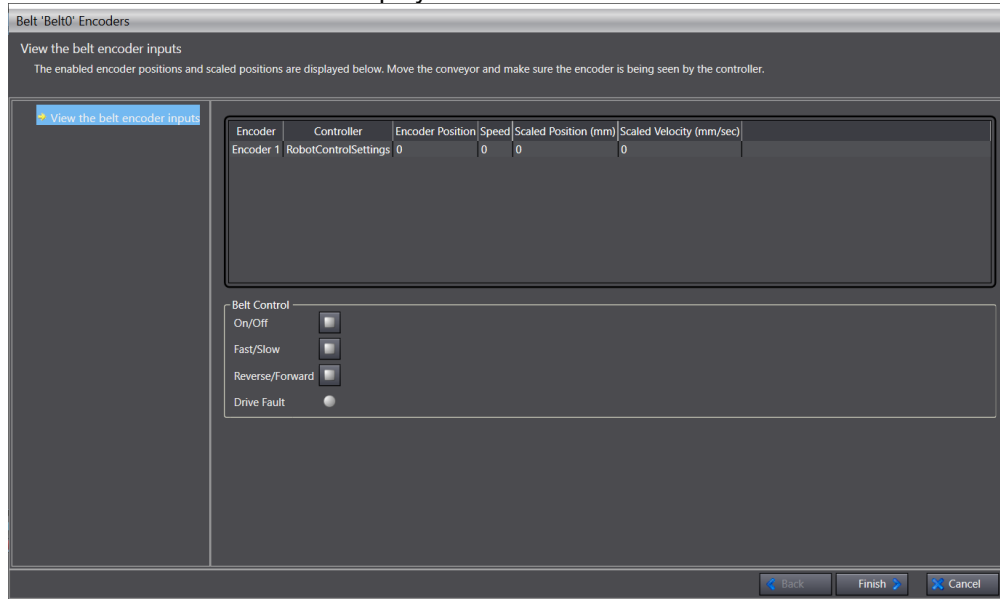
Item		Description
	Associate/Remove	Specify the robot controller to associate with the selected encoder. Select the encoder, click the <b>Associate</b> button, and then specify the robot controller to associate. Select the row of the controller, and then click the <b>Remove</b> button to delete.
	Latch Period in Emulation Mode (mm) <sup>*1</sup>	Specify the latch period.

\*1. Displayed when the emulation mode is enabled.

## View Belt Encoders

Display the edit pane of the belt encoder, and check the value when moving the belt.

- 1 Click the **Control** in the **Belt** pane, and then select **View Belt Encoders**. The **Belt Encoder** wizard is displayed.



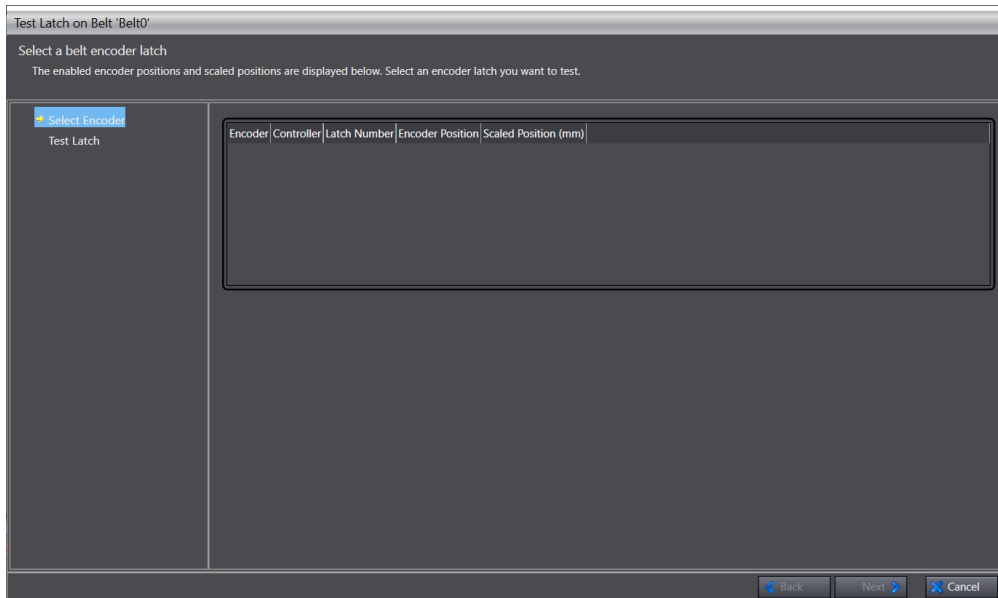
- 2 Select the belt encoder from the list, enable the belt, and then verify the value. These values change as the belt moves. The belt control can be used in the wizard when active control is enabled or in Emulation Mode. When you have finished verifying the values, click the **Finish** button to exit the wizard.

## Test Latch Signal

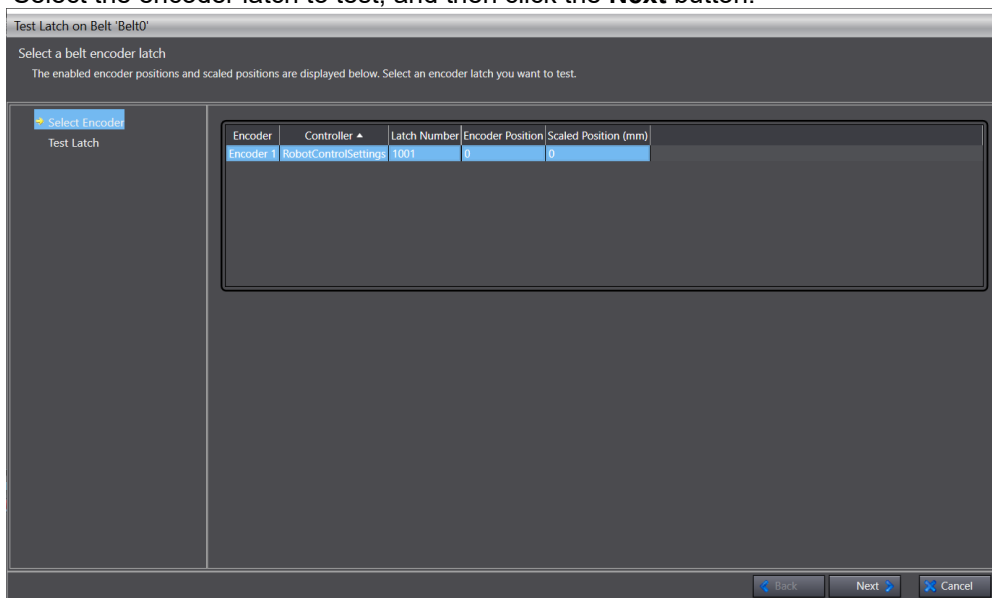
Test the encoder latch signal.

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 *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual* (Cat. No. W595) for more information.

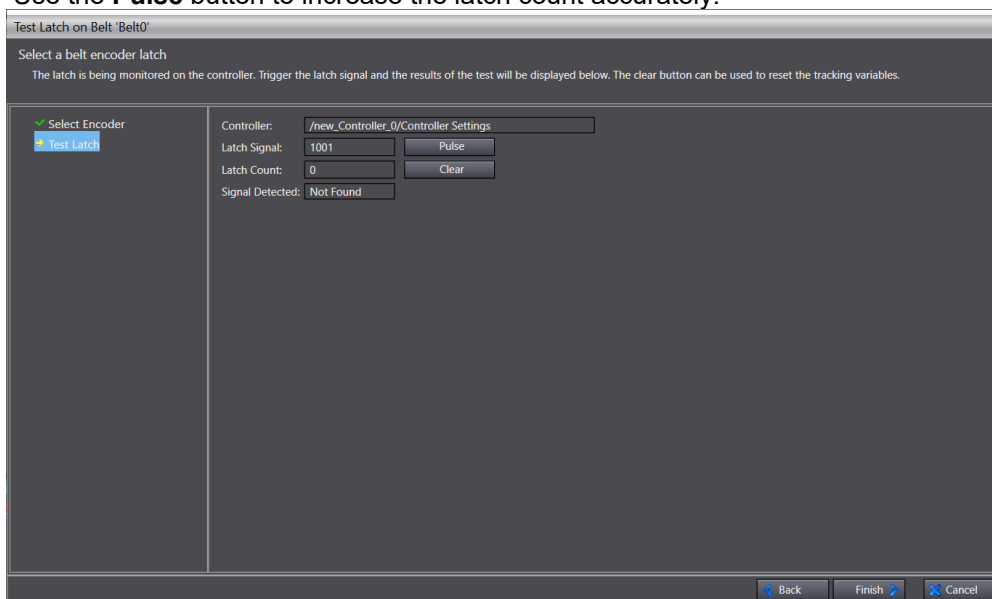
- 1 Click the **Control** in the **Belt** pane, and then select **Test Latch Signal**. The **Test Latch Signal** wizard is displayed.



**2** Select the encoder latch to test, and then click the **Next** button.



**3** Use the **Pulse** button to increase the latch count accurately.



**4** When the test is completed, click the **Finish** button to exit the wizard.



## 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 SmartController 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 367 for more information. Refer to *Process Strategy Belt Control Parameters* for more information.

## Physical Parameters

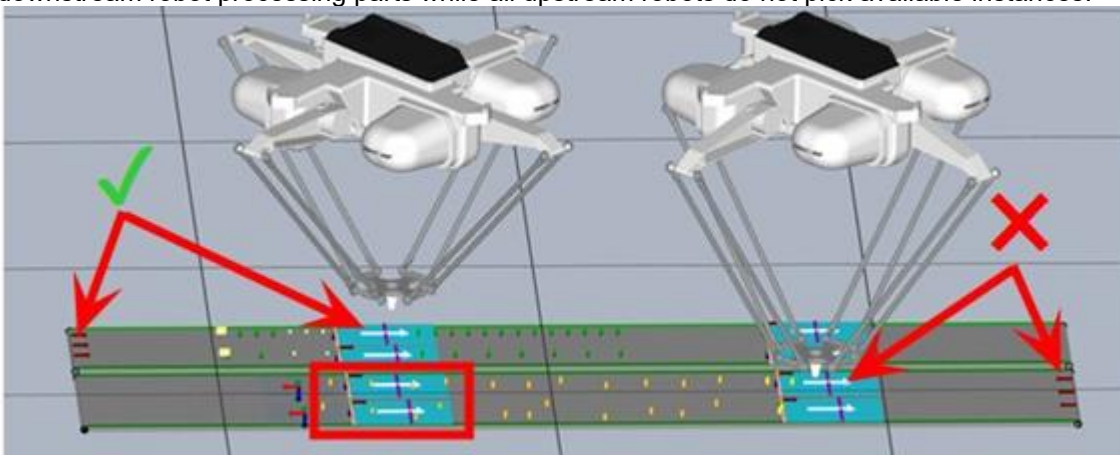
The Physical Parameters are generally set when creating the Belt object, but they can be modified as needed when the Process Manager is not executing. The Workspace Location, Width, and Length of the Belt object should be configured to closely approximate the position of the physical belt relative to 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.

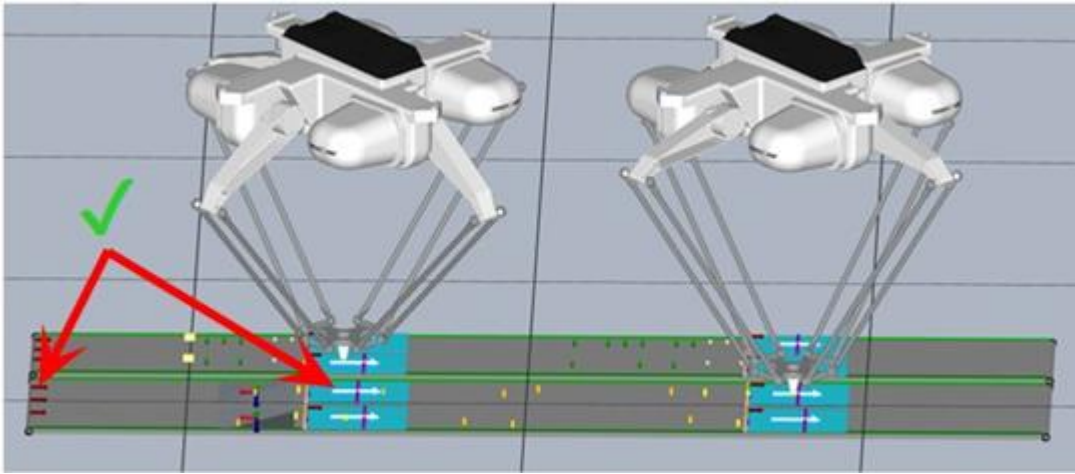
## Belt Direction of Travel

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 figure below.



## 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.

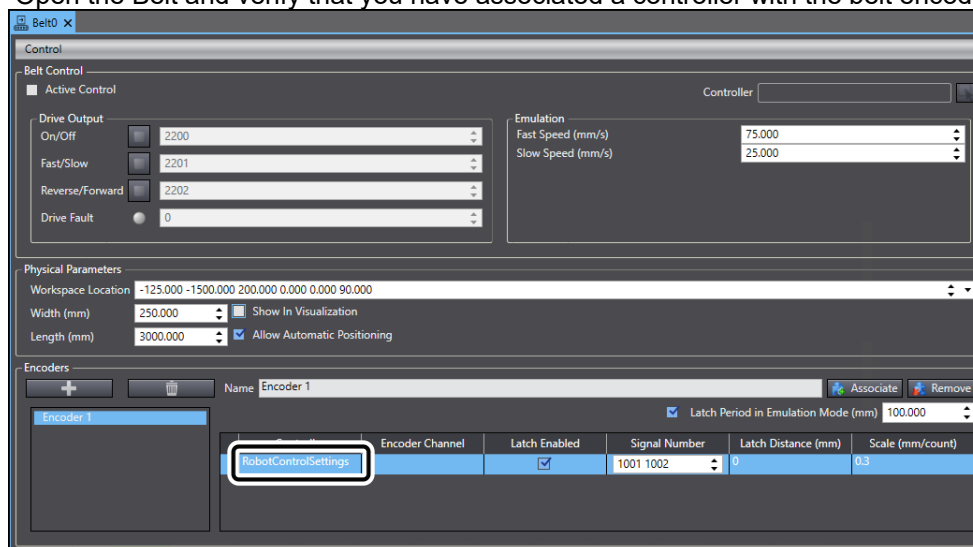
Belt object Encoders are effectively virtual encoders for the purpose of constructing tracking structures and allocation limits independently for each part or part target object type.

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, you may need to configure Encoders, Encoder Associations, or both. This section will describe the different situations that are supported.

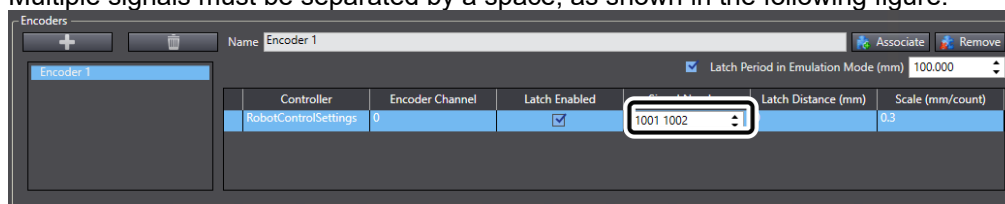
## Multiple Cameras Relative to a Single Robot-to-belt Calibration

Use the following procedure to configure multiple cameras relative to a single robot-to-belt calibration.

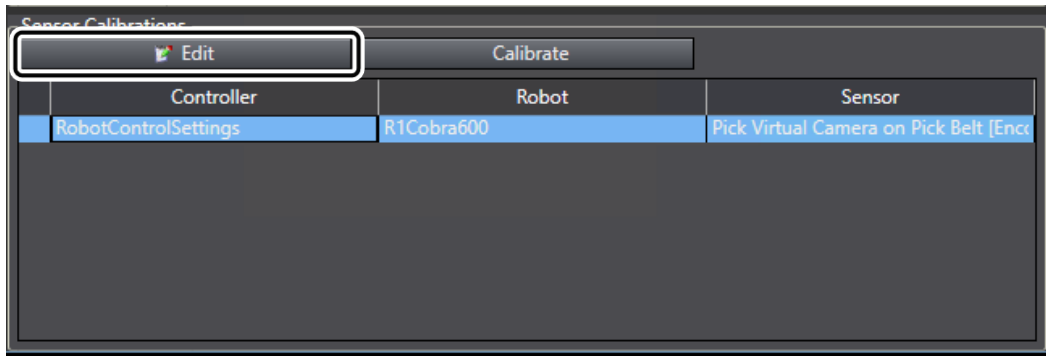
- 1 Open the Belt and verify that you have associated a controller with the belt encoder.



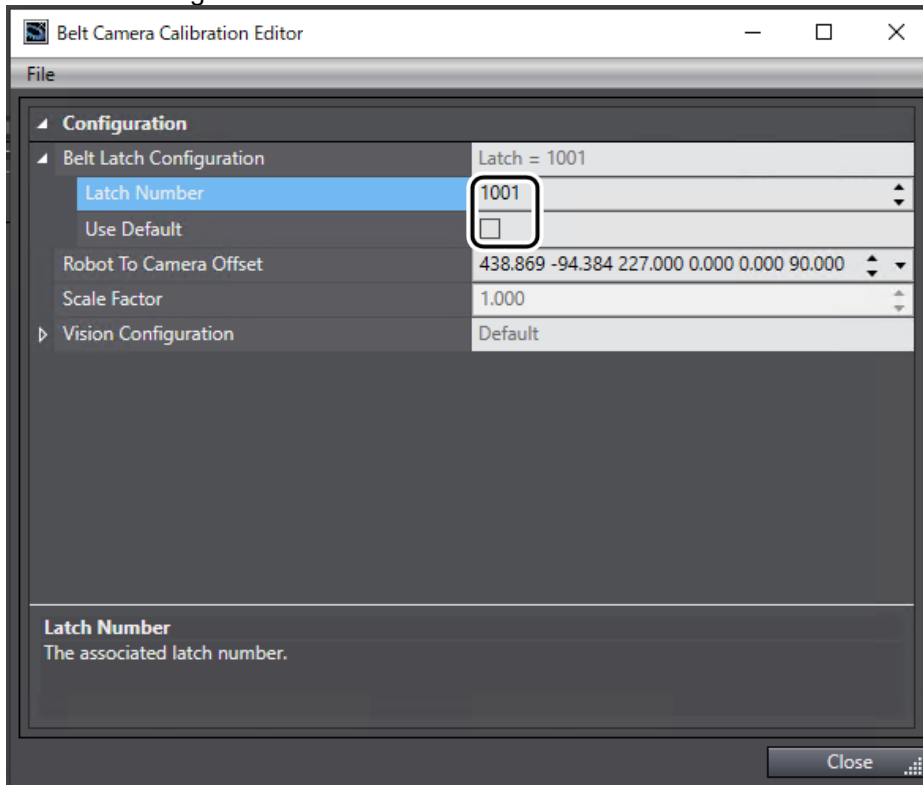
- 2 Specify the latch signals used by each camera in the Signal Number field. Multiple signals must be separated by a space, as shown in the following figure.



- 3 Locate the Robot-to-Sensor belt camera calibrations in the Process Manager. Select a Sensor Calibration and click the **Edit** button.



- Expand the Belt Latch Configuration parameter group, uncheck **Use Default**, and then enter the desired latch signal number associated with that camera.



- 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.

## 5-9-5 Process Manager

Process Manager is the feature to control the operation process of your robot integrated system. It manages processes for a part, for example, picking up a part and placing it on the target point, following the specified execution order.



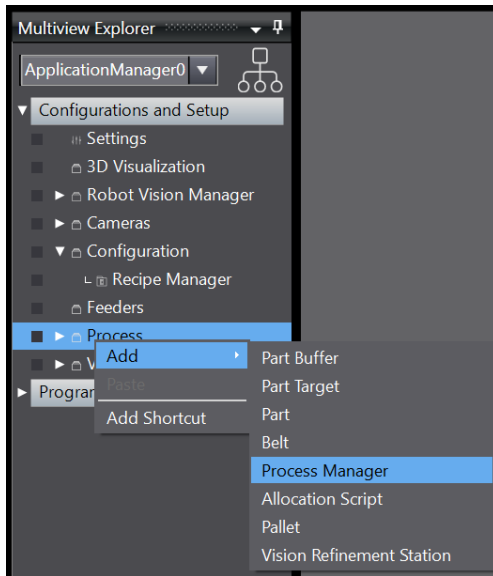
### Additional Information

Process Manager allows you to create and configure complex applications without writing any programming code.

If the application requires advanced modification, it provides an access to V+ programming controls that allow you to modify your application.

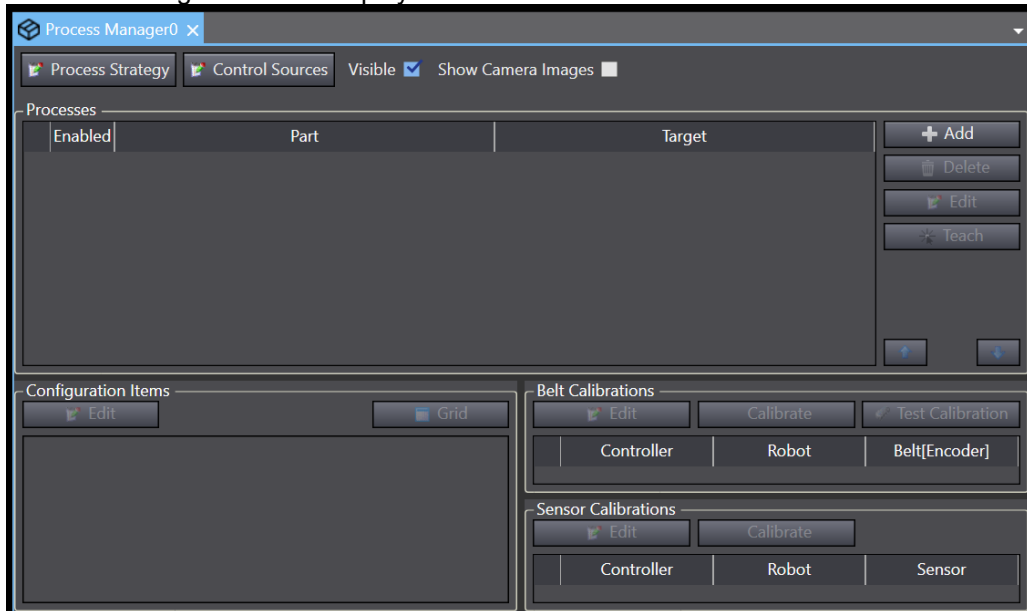
## Adding Process Manager

- Select **Configurations and Setup** from **Multiview Explorer**. Right-click **Process**. Select **Add**, and then, click **Process Manager**.



**Process Manager** is added under **Process**.

- 2 Right-click **Process Manager** to select **Edit**.  
Process Manager Editor is displayed.



## Process Manager Setting Items

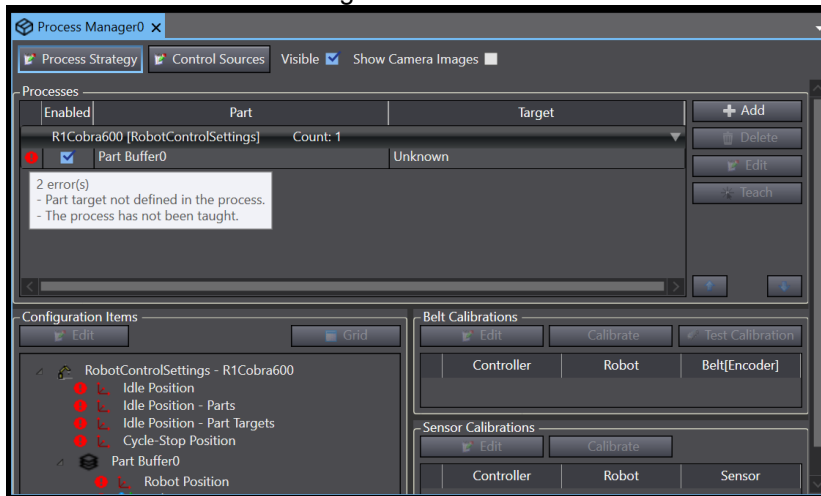
Items in the Process Manager Editor are shown in the table below.

Item	Description	Reference
Process Strategy	Edit Process Strategy in the editor. Process Manager allocates Parts and Part Targets to specific robots based on a set Process Strategy. An output process is passed to the controller queue, then runs according to the Process Strategy.	<i>Process Strategy</i>
Control Sources	Edit properties of selected item.	<i>Control Sources</i>
Visible	Display calibrations or Part/Part Target detection results on the 3D Visualizer. Check the box to show all calibrations.	---
Show Camera Images	Check the box to display camera images on the camera calibration or view field of the 3D Visualizer.	---
Processes	Configure robots, picking points, and placing points controlled by Process Manager.	<i>Process List</i>
Configuration Items	Shows relations among workcell items, in a tree structure.	<i>Configuration Items</i>
Belt Calibrations	Calibrate the robot respecting to the conveyor belt.	<i>Belt Calibrations</i>

Item	Description	Reference
Sensor Calibrations	Configure robot-to-sensor calibration for the selected workcell process.	<i>Sensor Calibrations</i>

## ● Process Manager Configuration Errors

If there is a configuration error, an alert icon displays in the corresponding item. Hover the mouse cursor over the icon to see a message that describes the error as shown below.



## Process Strategy

Process Manager invokes a Process Strategy to determine how to allocate Parts and Part Targets identified by the Process Manager. It allocates the Parts and Part Targets to specific robots using the list of Part Processes. An 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.

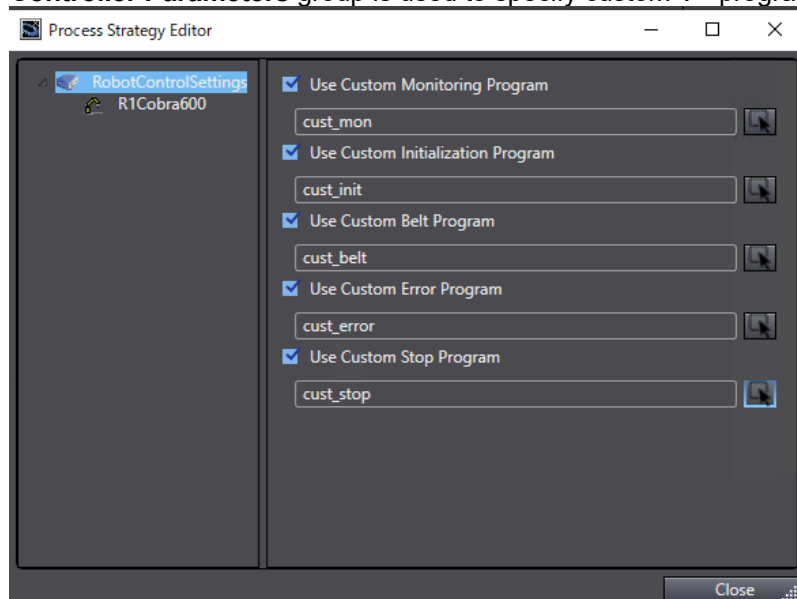
The Process Strategy Editor provides access to the following parameters editors.

- Controller Parameters: Refer to Process Strategy Controller Parameters for details.
- Robot Parameters: Refer to Process Strategy Robot Parameters for details.
- Belt Control Parameters: Refer to Process Strategy Belt Control Parameters for details.

To access the Process Strategy Editor, click the **Process Strategy** button. An appropriate editor is shown according to 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.

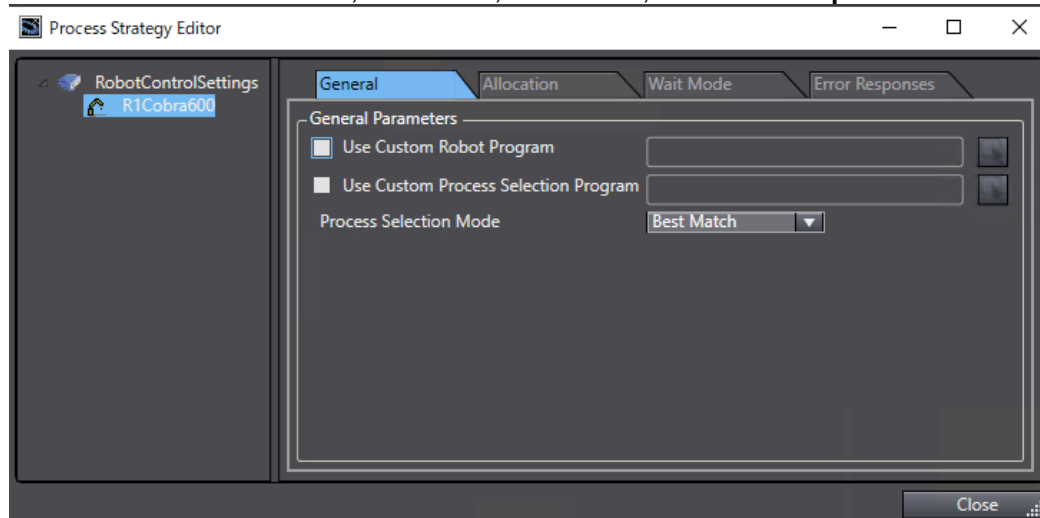


Item	Description
Use	The default process monitoring has the following functions:

Item	Description
Custom Monitoring Program	<ul style="list-style-type: none"> <li>• Checks for updates to process strategies</li> <li>• Handles belt monitoring</li> <li>• Monitors Parts and Part Targets</li> </ul> <p>You can copy the default V+ monitoring program for editing, or select an existing program.</p> <p>Most applications do not require the Custom Monitoring Program.</p>
Use Custom Initialization Program	<p>The default initialization program that executes before the control programs (robot, belt, process strategy, etc.) are started.</p> <p>This program 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. You can copy the default V+ initialization program for editing, or select an existing program.</p>
Use Custom Belt Program	<p>The default belt program monitors the speed and ON/OFF status of all belts. You can copy the default V+ belt program for editing, or select an existing program.</p> <p>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.</p>
Use Custom Error Program	<p>The default error program handles the processing and reporting of errors during the execution of a process. You can 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.</p> <p>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.</p>
Use Custom Stop Program	<p>The custom stop program can be used to perform certain operations after the application has stopped. You can copy the default V+ stop program for editing, or select an existing program.</p>

## ● Process Strategy Robot Parameters

**Robot Parameters** is displayed when a robot is selected in the Process Strategy Editor. There are four tabs: **General**, **Allocation**, **Wait Mode**, and **Error Response**.



Group	Item	Description
General	Use Custom Robot Program	Allows you to specify a custom V+ main robot-control program. For example, you can precisely customize the timing when a given robot executes each process. You can copy the default V+ program for editing, or select an existing program.
	Use Custom Process Selection Program	Allows you to specify a custom V+ process-selection program. For example, you can customize the selection and ordering of processes. You can 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 down- stream 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.
Allocation	Part/Target Filtering Mode	Specifies how instances are identified for processing by the robot. 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: Specifies relative distance between process instances separated by a certain distance. Pick/Skip Pallets: Process a certain number of pallets then skip a certain number of pallets. Pick/Skip Instances: Processes a certain number of instances then skips a certain number of instances.

	Queue Size	<p>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.</p> <p>Each Part and Target is associated with a different queue for a given robot. This parameter defines the size of each queue.</p> <p>The default value of this parameter is 10.</p> <p>To change the parameter value, enter a new value into the <b>Queue Size</b> field, or adjust the value pressing the up or down arrow.</p>
	Allocation Distance	<p>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.</p> <p>This distance can be considered the maximum distance upstream that an instance can be allocated to the robot.</p>
	Allocation Limit	<p>Specifies the distance upstream from the Process Limit Line that a Part instance must be to allocate that Part instance to the robot. This distance can be considered the minimum distance upstream that an instance must be for allocation.</p> <p>If set to zero, instances will be allocated if they are at or upstream of the process limit line.</p> <p>If greater than zero, it represents the upstream distance (in mm) from the process limit line an instance must be for it to be allocated.</p> <p>This parameter is useful in the case of fast-moving belts where the robot needs additional look ahead to process instances.</p> <p>For example, if a Part pallet has four instances and the robot queue size is two, when the robot picks instance No.1, now there is a space for another instance in the queue. If instance No. 3 is at the upstream of the allocation limit, it will be allocated. If it is at the downstream of the allocation limit (<math>&gt; 0</math>), it will not be allocated.</p> <p>It prevents instances at the downstream of the allocation area from being allocated.</p>
	Use Custom Allocation Script	<p>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.</p> <p>With this program, you can manipulate a list showing the system what should be allocated to a given robot. It is invoked by the Process Manager when it needs to allocate a Part.</p> <p>If this option is enabled, click the <b>Select</b> button to select a necessary Custom Allocation Script. Refer to <i>ACE Reference Guide: Custom Allocation Scripts</i> for details.</p>
Wait Mode	Stay at current Position	Makes the robot stay at the current position while waiting for a Part or Target to be processed.
	Move to idle Position	<p>Makes the robot move to the idle position after the specified <b>After waiting for</b> time (in milliseconds) while waiting for a Part or Target to be processed.</p> <p>For example, if the <b>After waiting for</b> 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.</p> <p>The <b>Idle Position Parts</b> and <b>Idle Position Part Targets</b> must be enabled. Refer to Idle Positions for details.</p>



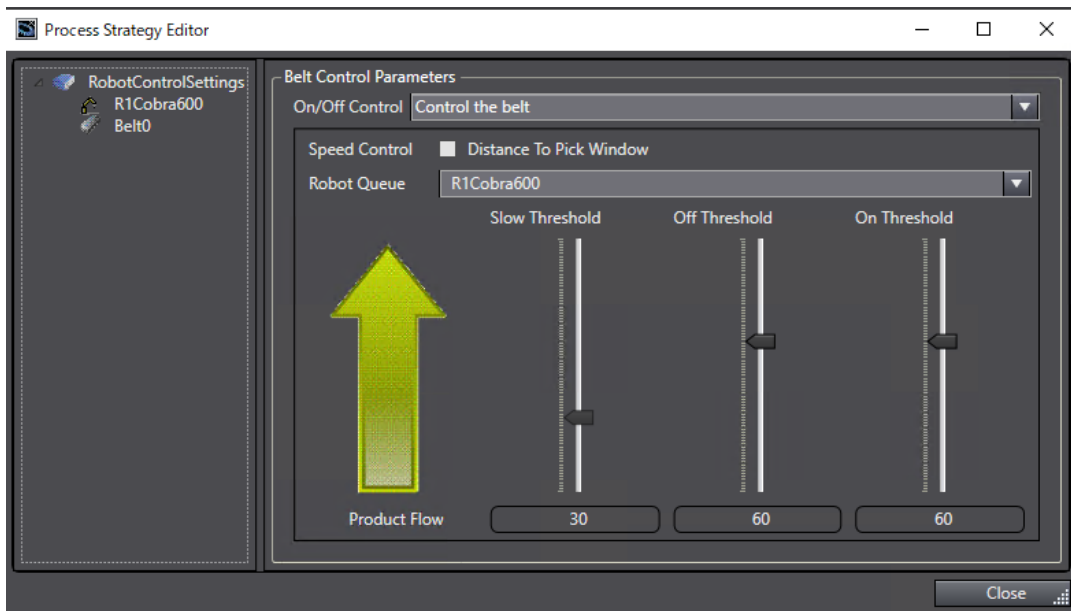
	Use Custom Wait Program	Allows you to specify a custom V+ wait program. The program would be called when the robot does not have a Part or Target available. With this program, you can check to see if the robot needs to be moved to the idle location or if it should stay at the current location. This program runs with the logic specified by one of the selections above. You can copy the default V+ program for editing, or select an existing program.
	Use Signal at Cycle Stop	If a cycle stop is issued and this option is enabled, the specified I/O signal will be turned ON when the robot has reached a cycle stop state. When the cycle is resumed (i.e. the cycle stop state is canceled), it will turn the specified signal OFF and will attempt to enable power, if high power has been disabled.
Error Responses	Output Signal— On Error	Defines a digital signal to assert when an error is detected for the selected robot.
	Error Listing (Using the <b>Add</b> , <b>Delete</b> , and <b>Up/Down</b> buttons, arranges a list)	<p>Specifies how specific error conditions are to be handled.</p> <p>By default, all errors will be reported to you and the software will wait until you respond 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.</p> <p>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 Error: Problems with power being enabled or enabling power  Gripper Error: All gripper actuations  All Errors: All other errors</p> <p>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. Refer to <i>Process Strategy Controller Parameters</i> for details.</p>

### ● Process Strategy Belt Control Parameters

The **Belt Control Parameters** is displayed when a belt is selected in the Process Strategy Editor. Belt Control Parameters are only available when the following items are configured:

- **Active Control** is enabled in the Belt configuration items
- **Controller** is selected in the Belt configuration items
- Defined process includes a Part or Part Target that references the Belt Object

The **Belt Control Parameters** group illustrated below is used to set the belt control parameters for the selected conveyor belt. You will set these parameters to determine when the conveyor belt is turned ON or OFF. An optional speed control parameter is also provided. The decision point for the belt I/O control depends 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	<p>Specifies an ON/OFF control for the belt. Following three options are available.</p> <p>Do not control the belt: Use this option if there are output signals that can control the belt, but you do not want the Process Manager to control the belt during run time. If the belt control is provided by the Robot Integrated CPU Unit and output signals are not able to control the belt, disable Active Control in the Belt Object.</p> <p>Leave the belt always ON: The belt is turned ON when the process starts and OFF when the Process Manager is stopped.</p> <p>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.</p>
Speed Control	<p>If this is selected, you can 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.</p>
Robot Queue	<p>Specifies a robot for queue monitoring (the queue size for the robot that set in the Robot Parameters group)</p> <p>The selected robot will typically be the most downstream robot if multiple robots work at the same belt. If Parts get to the last robot, it needs to slow down or stop the conveyor belt to ensure all Parts or Part Targets are processed.</p>
Slow, OFF, and ON Thresholds	<p>These thresholds are used to control the belt based on instance position. It 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.</p> <p>Slow Threshold: Specifies the point in the belt window for slowing the conveyor belt if Parts reach that point. For example, at 50%, the conveyor belt will be slowed when a Part reaches the mid- point of the belt window.</p> <p>OFF Threshold: If an instance reaches this threshold, the belt will be turned OFF. This threshold 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).</p> <p>ON Threshold: When a belt is turned OFF by the <b>OFF Threshold</b>, the belt will remain OFF until all instances are removed between the <b>OFF Threshold</b> point and the <b>ON Threshold</b> point. Setting this threshold can minimize the number of times the belt is started and stopped. The threshold 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).</p>
Product Flow	<p>Shows the product flow (belt window) direction of travel corresponding to the <b>Slow Threshold</b> and <b>OFF Threshold</b> slide controls. It is the 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.</p>

## Control Sources

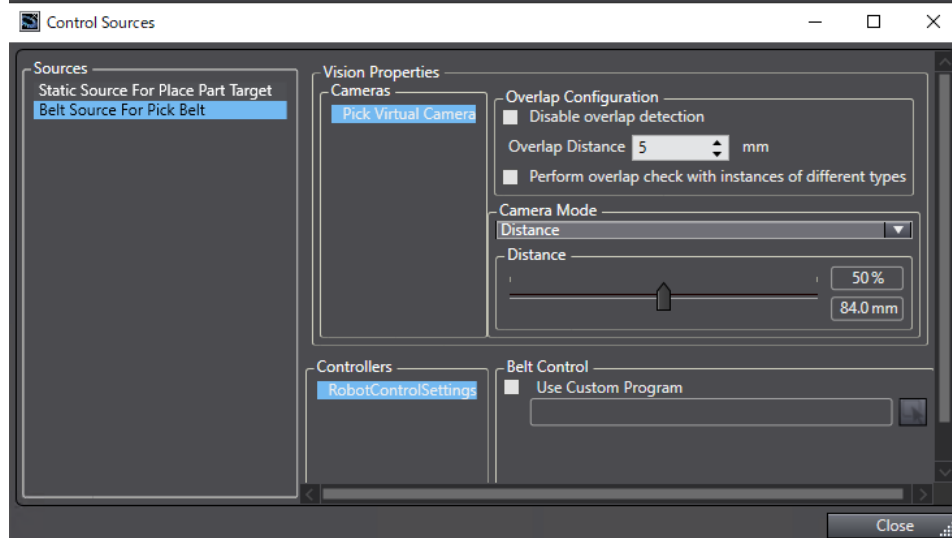
The Control Sources editor provides access to parameters that affect Part and Part Target sources for the defined processes. Sources are responsible for providing instances to a process. These are automatically created based on the Part and Part Target object configuration property. There are three types of Sources that are described in the following sections.

- 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.



#### Cameras

The Cameras list will display the Virtual Cameras that have been associated with this belt through the Part and Part Target object configurations. You can change the Vision Properties for the selected camera in this area.

#### 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 duplicated 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.

#### Camera Mode

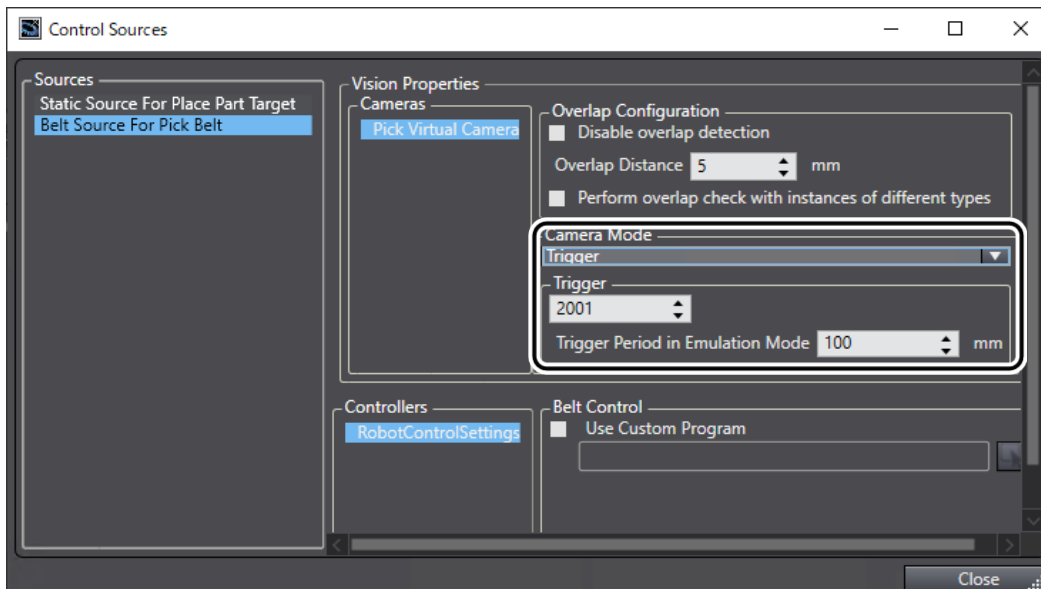
There are three camera modes available: Distance, Trigger, and Time that are described below.

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 the EMULATION mode, Trigger mode will use the **Trigger Period in Emulation Mode** distance value specified. It is used to simulate the trigger occurring based on the specified distance of belt travel.



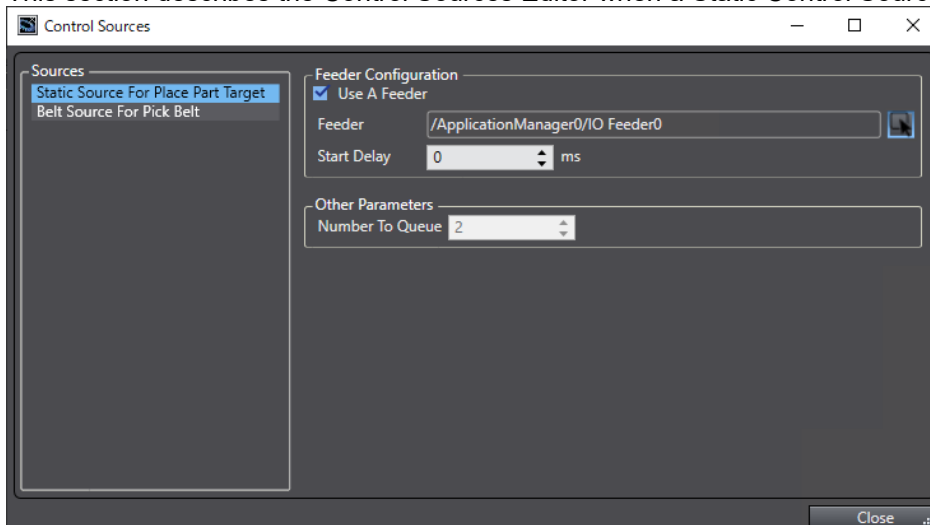
When **Time** is selected, an image will be requested on the specified time interval.

### Belt Control - 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, you may 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. You may need to make the same modifications 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 **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.

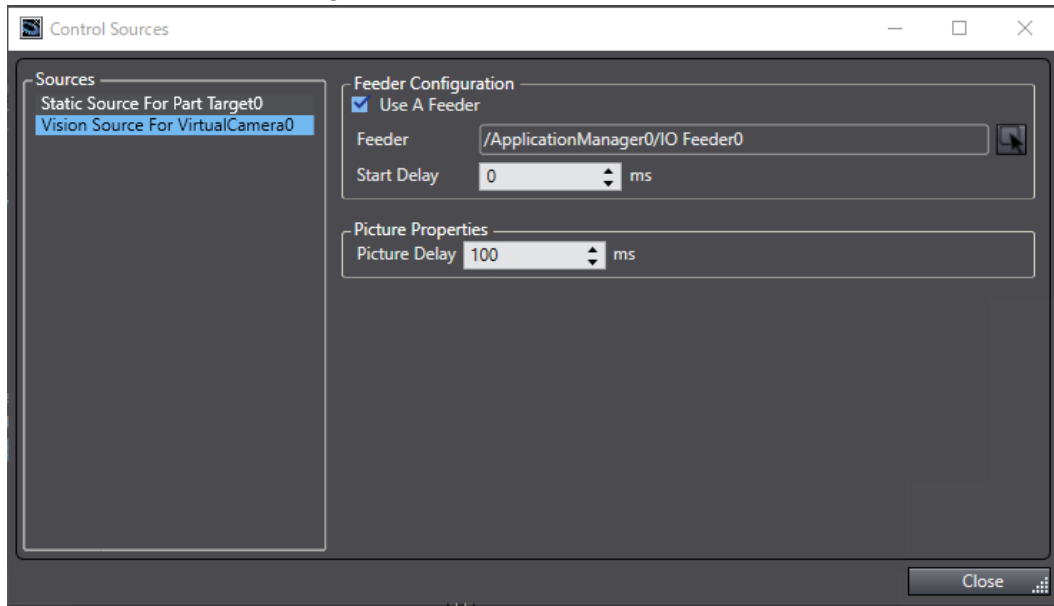
Alternatively, select **Use A Feeder** and choose an IOFeeder object that controls when parts are generated. When it is selected, another V+ program is executed for monitoring feeder activity. This can be used for individual parts or pallet configurations. For example, associate an IOFeeder with a target 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 to 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.

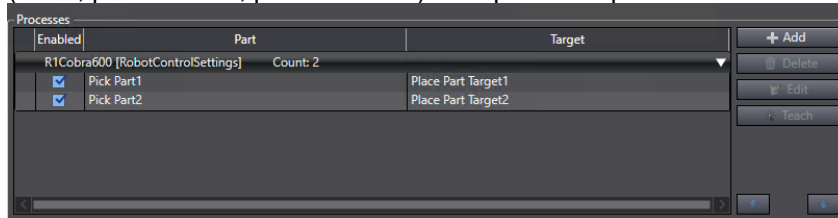


Vision Sources are used for fixed-mounted cameras. They can be associated with IOFeeder objects, similar to Static Sources.

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 List

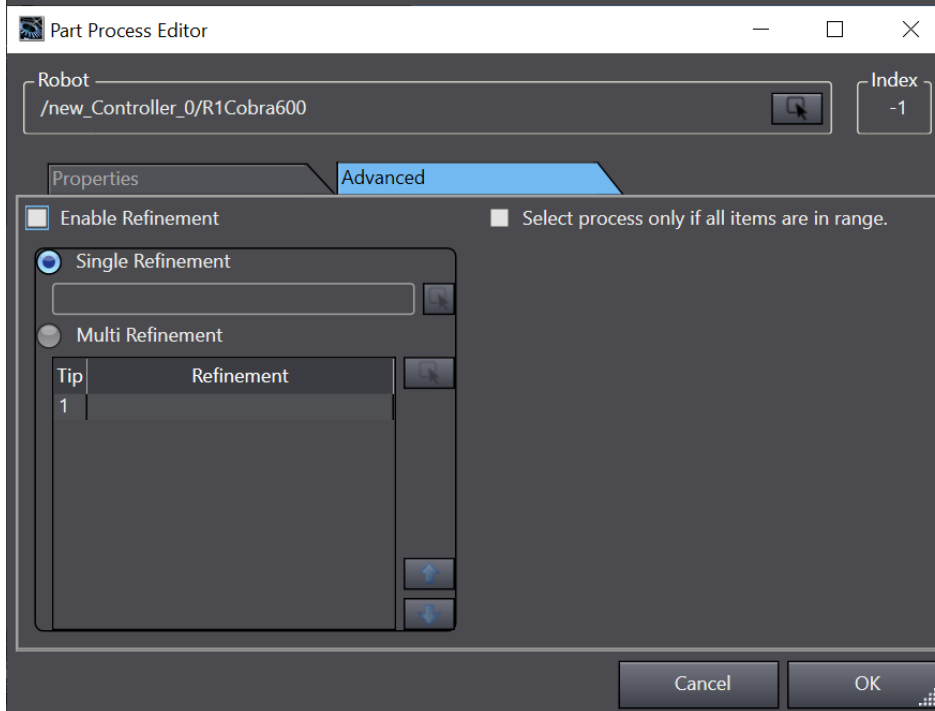
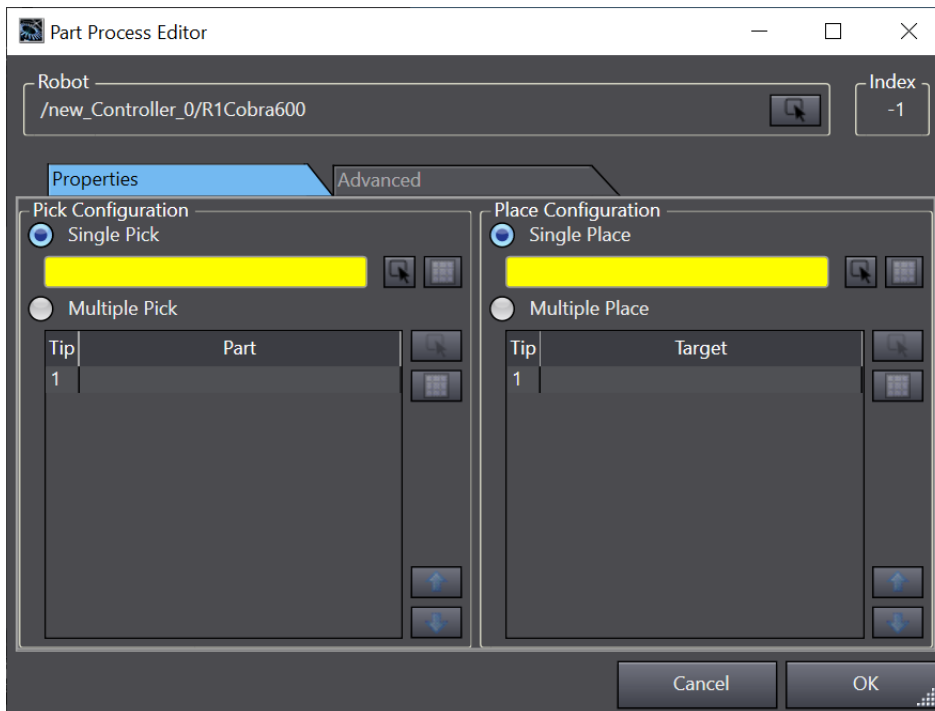
This section describes the Process List which the Process Manager provides. You can define the elements (robot, pick location, place location) for a particular process that will be controlled by the Process Manager.



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 for details.
Enabled check box	If checked, the process is enabled for use.
Part	The Part(s) specified for the process. You can double-click this item or press the <b>Edit</b> button change it.
Target	The Part Target(s) specified for the process. You can double-click this item or press the <b>Edit</b> button change it.
<b>Add</b> button	Add a new process to the Processes list.
<b>Delete</b> button	Remove the selected process from the Processes list.
<b>Edit</b> button	Edit the selected process. Refer to Part Process Editor for details.
<b>Teach</b> button	Teach the selected process. Refer to Teaching a Process for details.
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. Double-clicking an existing process, or clicking the **Add** button for a new process displays the **Part Process Editor** dialog box.



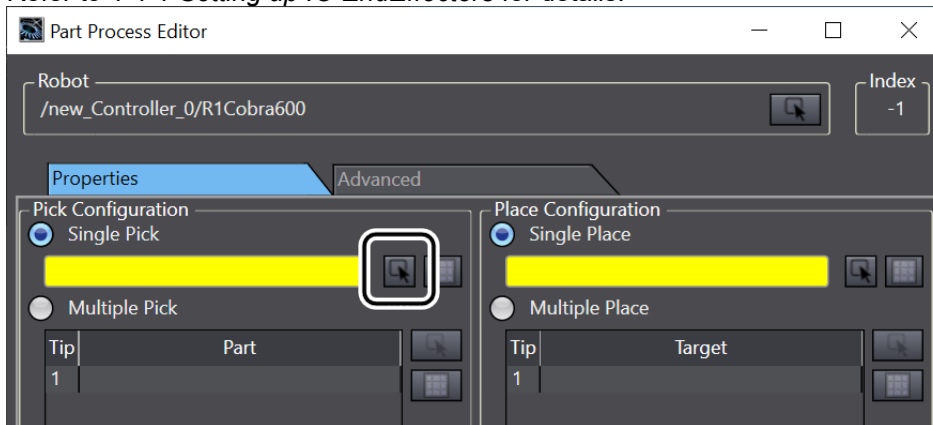
Item		Description
Robot		The Robot reference is used to specify the robot that will be used for the pick-and-place process. Use the <b>Select</b> button to open the <b>Select a Reference</b> dialog box, which allows you to create or specify the robot that will be used in the pick-and-place process.
Index		This item displays the index number of the process, which can be referenced in V+ programs and C# programs.
Properties Tab	Pick Configuration	The Pick Configuration group is used to specify the single or multi-pick Part items for the pick- and-place process. Single Pick/Place and Multiple Pick/Place can be used together. For example, if you want to pick multiple parts individually and then place them all at the same time, you would use Multiple Pick and Single Place. Refer to Single Pick and Multiple Pick for details.
	Place Configuration	The Place Configuration group is used to specify the single or multi-place Part Target items for the pick-and-place process. Refer to Single Place and Multiple Place for details.
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

Item	Description
	<p>accuracy. It is used when placement error needs to be smaller than error introduced during the pick operation.</p> <p>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 <i>5-9-8 Vision Refinement Station</i> for details.</p> <p>As for selection of an object to be refined, refer to <i>Single Refinement</i> and <i>Multi Refinement</i> for details.</p>
Select process only if all items are in range	<p>The <b>Select process only if items are in range</b> check box tells the system to only select this process if all parts and targets are in range of the robot. Refer to <i>Select Process Only if All Items are in Range</i> for details.</p>

## ● 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 IO EndEffector that has multiple tips defined, the center line Tip Offset is applied. This is the average of all Tip Offsets defined in the IO EndEffector object. Refer to *4-4-1 Setting up IO EndEffectors* for details.

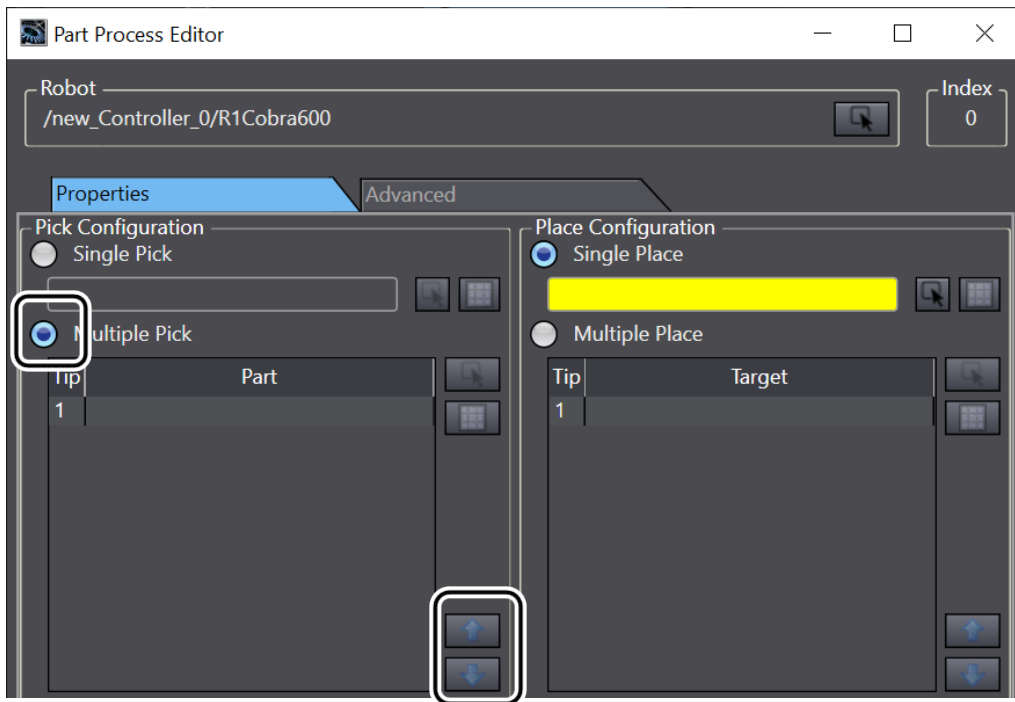


If you want to use a Pallet, use the **Pallet Slot Selection** button to select the pallet slots the robot is allowed to access the Pick/Place Configuration. This button is only available when a Pallet object is configured for the selected Part /Target.

## ● Multiple Pick

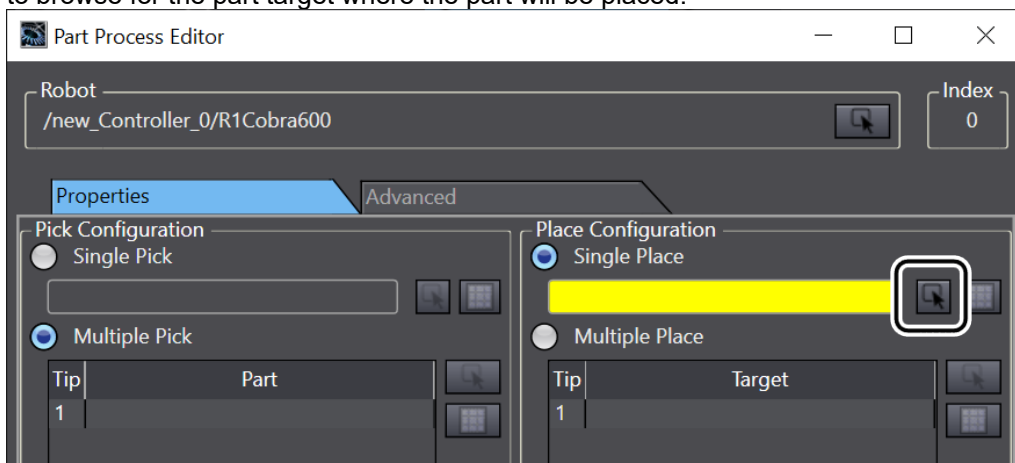
Select this item for a multiple-pick application where multiple parts will be picked before placement at a target. When multi-pick is enabled, the available tip indexes will be provided for the selected robot IO EndEffector. Use the **Select** icon to browse for the part that will be picked by each tip. The Up/Down buttons allow you to change the order of the tip processing.

Multiple Tips must be defined in the robot's IO EndEffector configuration in advance. The Process Manager will access the IO EndEffector object specified in the **Selected End-Effector** property of the robot object.



### ● 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.

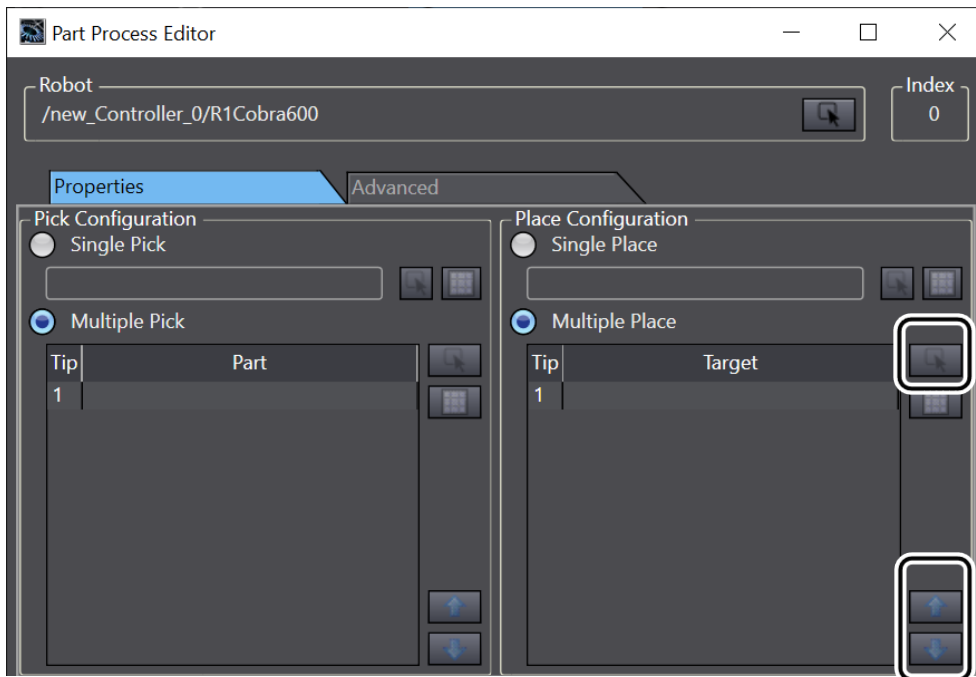


### ● Multiple Place

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 placed. The Up/Down buttons allow you to change the order of the tip processing.

Multiple Tips must be defined in the robot's IO EndEffector configuration in advance. The Process Manager will access the IO EndEffector object specified in the **Selected End-Effector** property of the robot object.





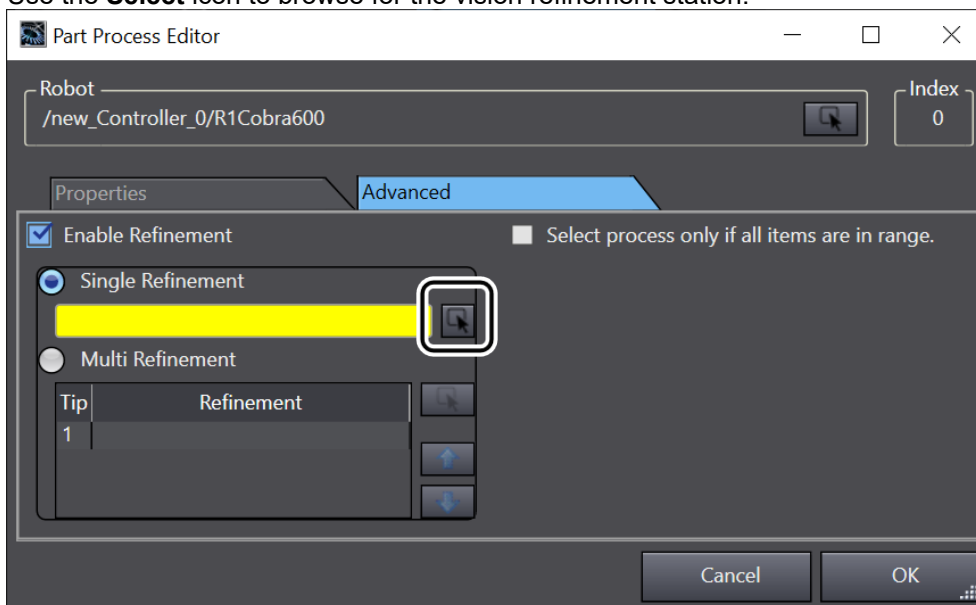
### ● Single Refinement

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. If each part needs a refinement, use the **Multi Refinement** option.

Use the **Select** icon to browse for the vision refinement station.

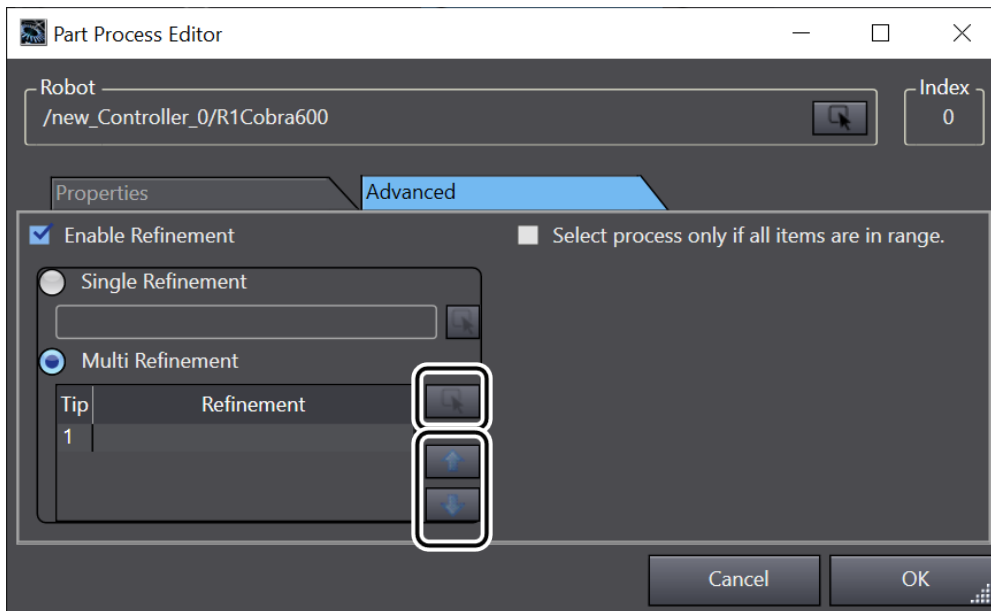


### ● Multi Refinement

Select this item for a multiple-tip-gripper application. 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.

The Up/Down buttons allow you to change the order of the tip processing.



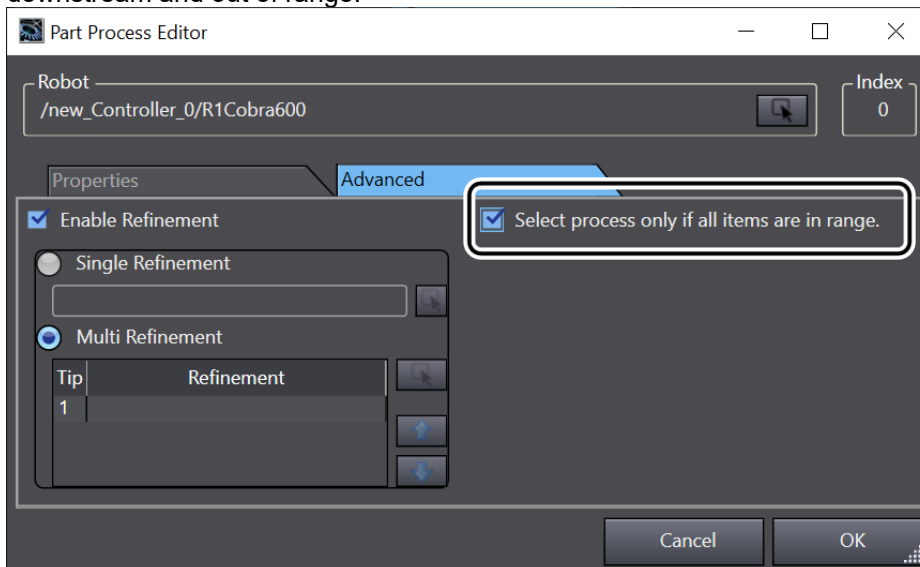
### ● Select Process Only if All Items are in Range

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 where you want to 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, you may need to consider part and target filtering to reduce the number of instances the robot has to process through when checking position of all instances.

It 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 upstream relative to Process B Part, but Process B Target is further downstream than Process A Target, you may want the system to select 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.

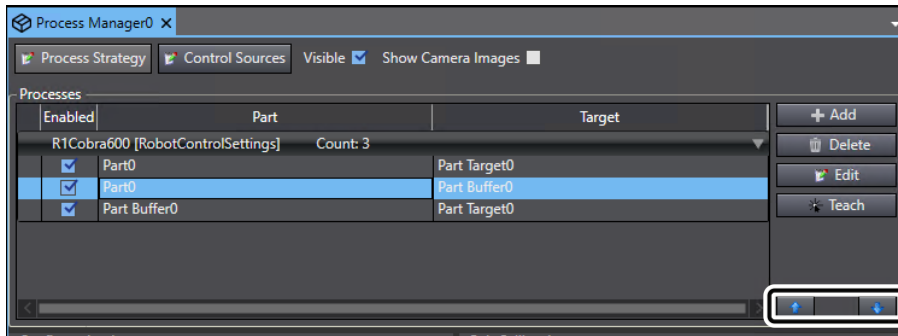


## Changing Process Priority

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.

You can change the priority for a process by using the **Up** and **Down** buttons on the Process list editor, shown in the following figure.

In addition to the arrows, you can also affect process priority through the **Process Selection Mode** setting of the **Robot** parameters in the Process Strategy Editor area. Refer to Process Strategy Robot Parameters for details.



For example, you have three processes defined in the Process Manager, as follows.

1. Pick from input Part, place to output Part Target
2. Pick from input Part, place to output Part Target
3. Pick from input Part, place to output Part Target

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, if you change the order (the priority), you will get a different behavior as follows.

1. Pick from Part Buffer, place to output (Part Target)
2. Pick from input (Part), place to output (Part Target)
3. Pick from input (Part), place to Part Buffer

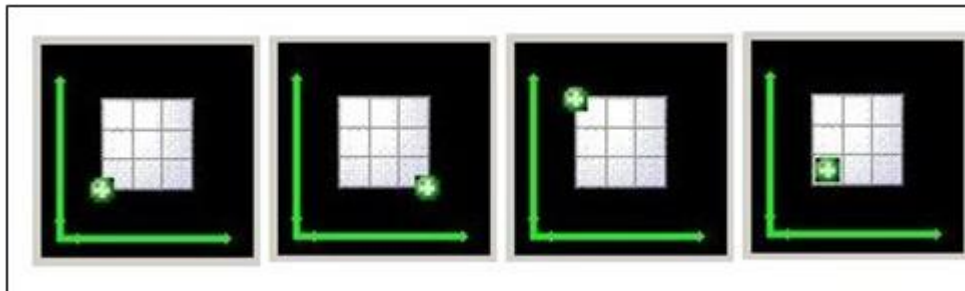
In this case, the Process Manager will always remove parts from the Part Buffer until no parts remain in the Part Buffer, before processing the input (Part).

## Teaching a Process

The last step in creating the Process is the Teach function. It 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 your wizard will vary depending on the application and types of Parts and Part Targets you have defined for your application.

If a Pallet object is used for a Part or Part Target, you will see additional screens in the wizard for teaching the pallet frame (the orientation of the pallet in the workspace) and the first position in the pallet. Each of these steps contain an image of the pallet item being taught, which provides a visual reference. Refer to 5-9-7Pallet for details.

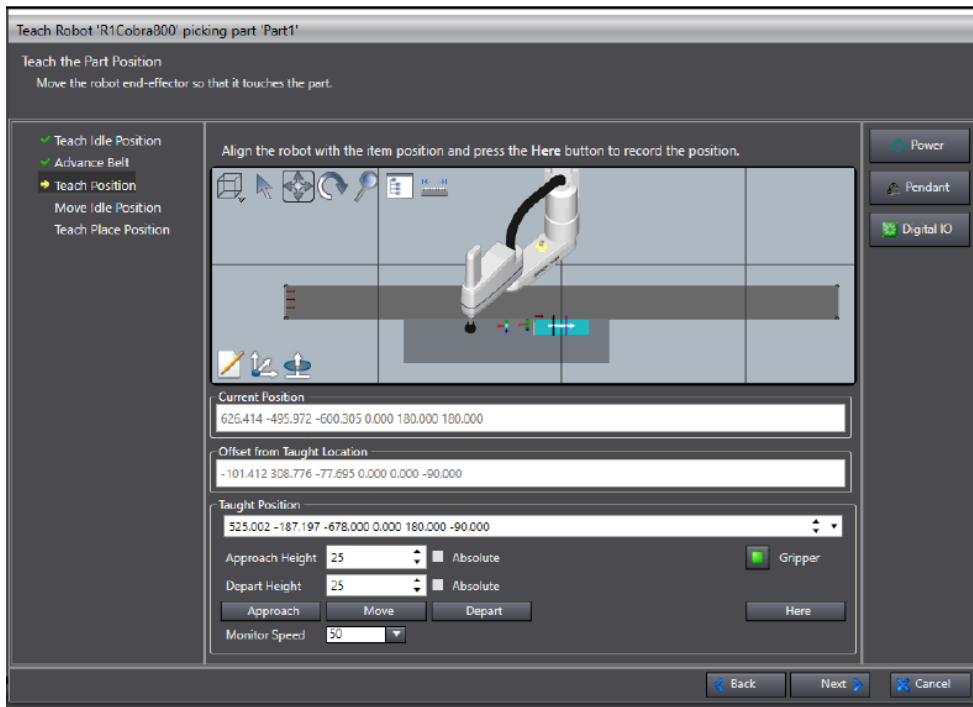


Use the Teaching a Process wizard to make selections for pick, place, and idle robot positions.



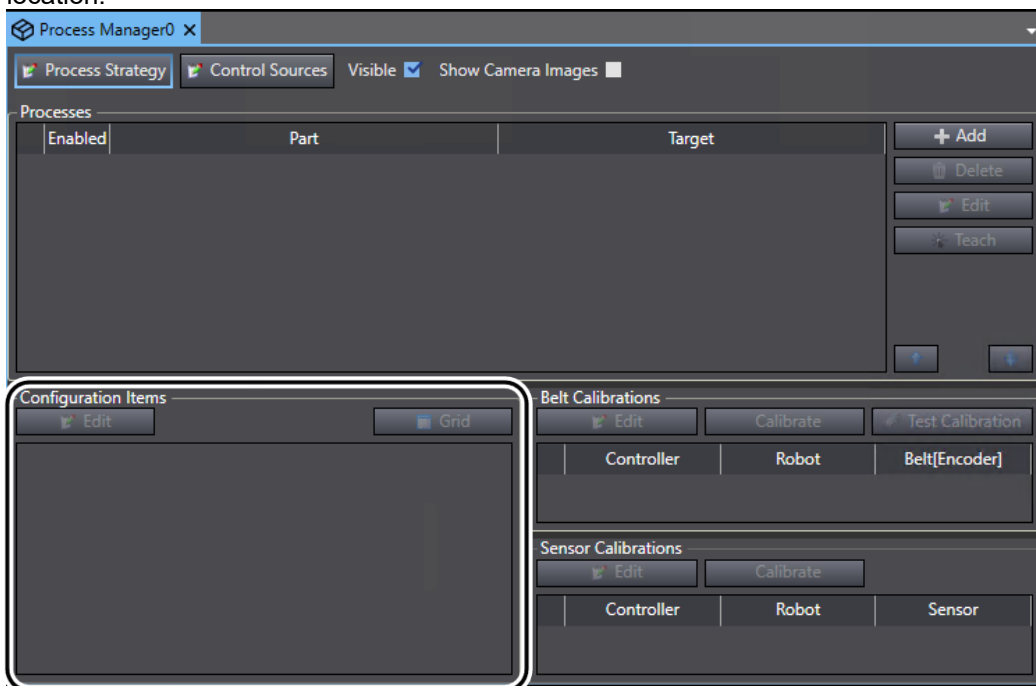
### Precautions for Correct Use

When the **Absolute** option is selected, the approach or depart heights will be an absolute Z coordinate in robot world coordinates. You must ensure that 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.



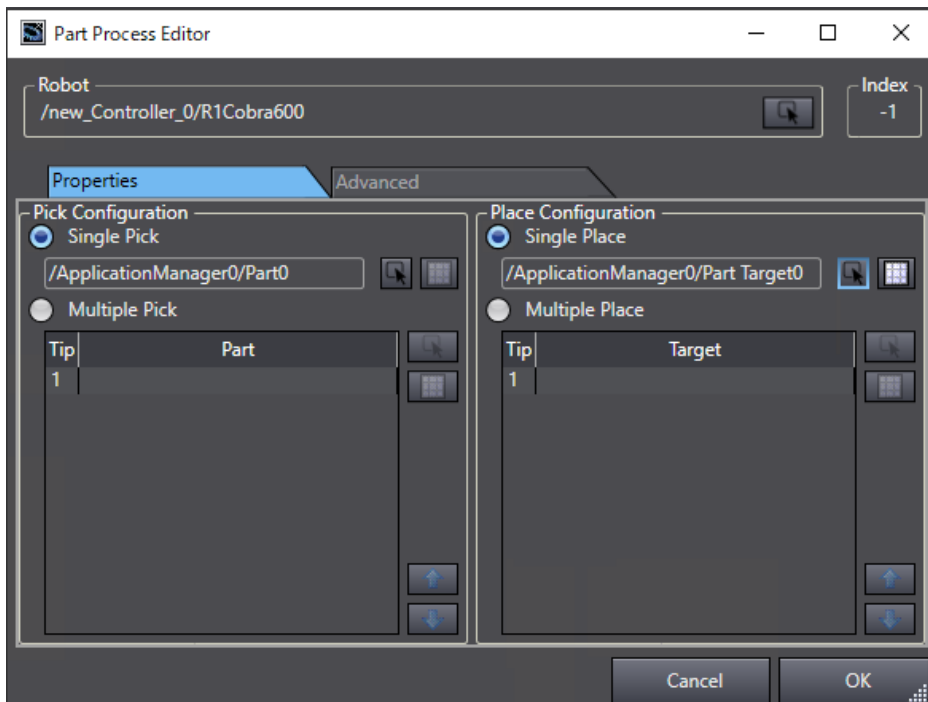
## Configuration Items

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.

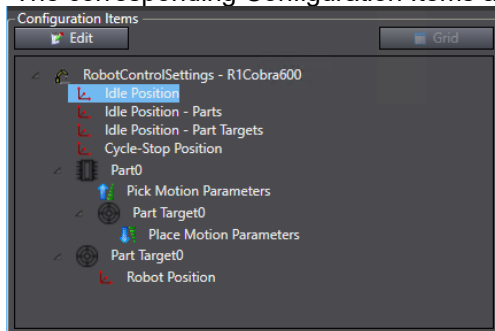


### ● Creating the Configuration Items

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 Items Area Structure and Features

The Configuration Items are arranged in a tree structure to show the relationships between the workcell items. The Configuration Items group contains the following features.

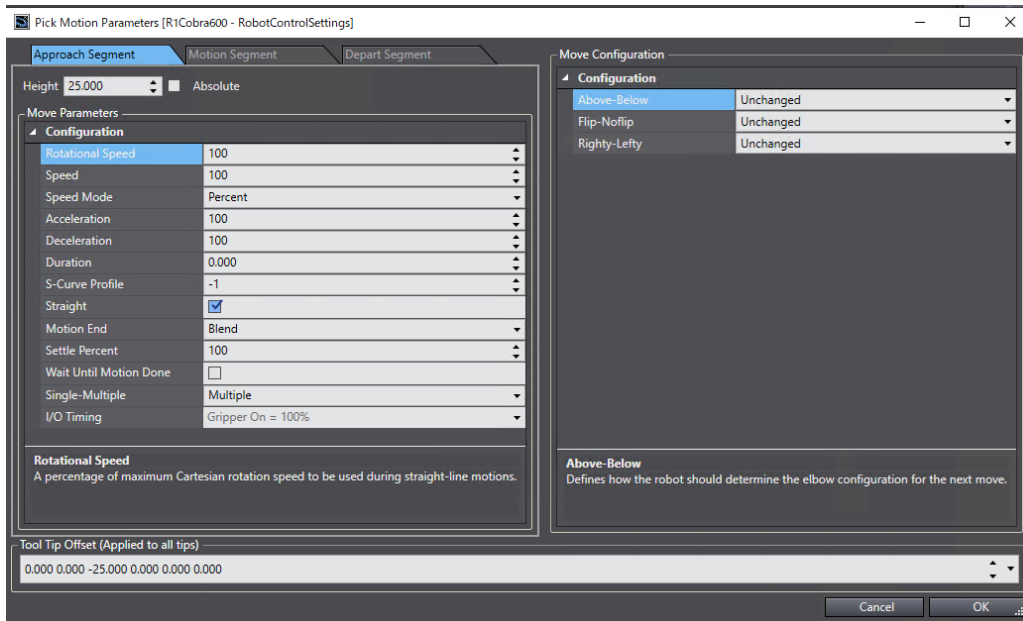
- Expand or collapse a tree branch by clicking the arrow icons next to an item name.
- Double-click any of the Position objects (idle or robot) or select the Position object and click the Edit button to open the location editor that can be used to manually enter the object location. Refer to the following section for more information.
- Click the **Grid** button 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 for more information.

## ● Location Editors

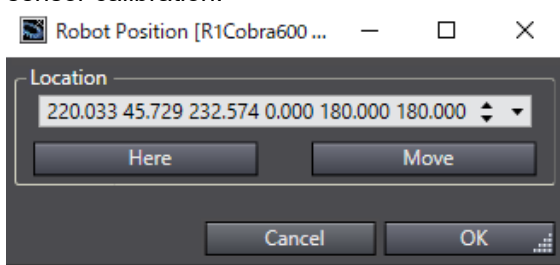
There are two types of location editors: a simple editor, which allows you to enter location information and an enhanced position editor, which contains additional sections such as **Move Parameters**, **Move Configuration**, **Approach/Depart Parameters**, etc.

For example, the Idle Position editor shown in the following figure is an enhanced position editor, which contains additional properties for **Move Parameters** and **Move Configuration**. Refer to Enhanced Location Editor Parameters for details.

The Location Editor title bar indicates the type of parameters being edited.



The Robot Position editor shown below is a simple position editor. The editor allows you to enter or teach the 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. The following section describes parameter editor functions of various enhanced location editors.

### Move Configuration Area

These parameters control the configuration of the robot at the selected location. For example, if your workcell contains a SCARA robot and you want to set it as **Lefty**, you would 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 allow you to fine-tune 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 allows you to enter a value for the **Approach** and **Depart** segments. The **Absolute** check box allows you to enter absolute height values for the **Approach** and/or **Depart** motion segments. You can enter positive or negative values, as needed.



### Precautions for Correct Use

When the **Absolute** option is selected, you must ensure the approach/depart heights are set correctly. Otherwise, the robot could move to an unintended location or out of range, which may damage the robot or other equipment in the workcell.

### 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, if you set the value to 25 mm, the gripper will activate at 25 mm from the pick position. If you set it at 25%, the gripper will activate at 25% of the distance from the approach start to the pick position. The time value allows you to set the gripper timing (in milliseconds).



### Additional Information

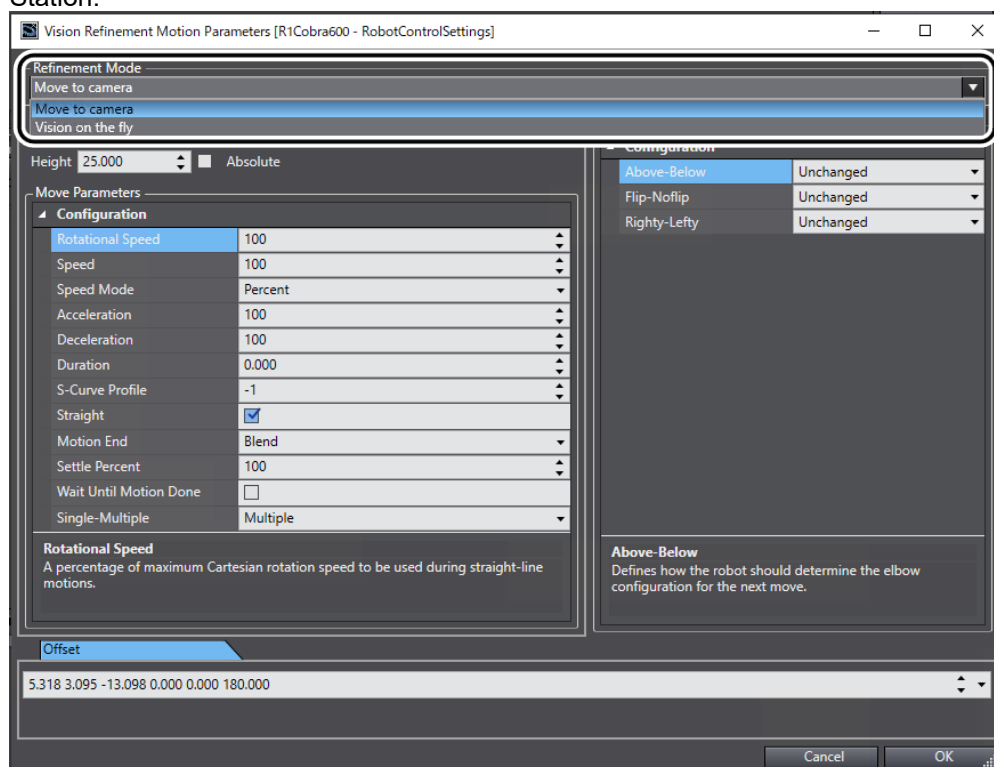
The distance value is useful when accessing pallets with multiple layers and an absolute approach height has been specified.

Use Custom Program Option (Motion Segment Tab)

The **Use Custom Program** check box allows you to specify a custom V+ program that controls the Motion segment. To use this option, select the check box and then click the **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 you to edit the gripper (tool) offset.

### 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 allow you to edit the start and end points for the robot path through the camera's field of view.

The **Trigger Percent** tab allows you to edit where the picture request is triggered as a percentage of the robot motion, from Start Location to End Location.



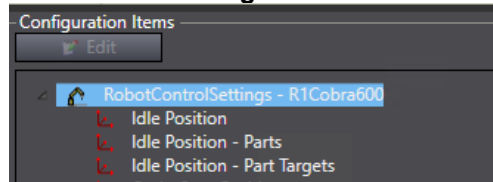
### Precautions for Correct Use

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 the **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.



The Wait Mode setting can be found in the Process Strategy area. Refer to Process Strategy for details.

### Idle Position

The location the robot uses when no Process is selected or when the Process Manager is aborted or stopped.

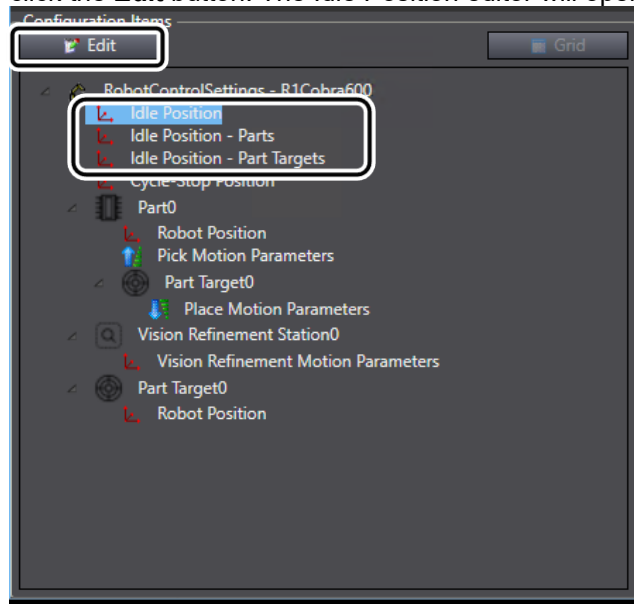
### Idle Position - Parts

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 you have 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 - Part Targets

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 you have 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

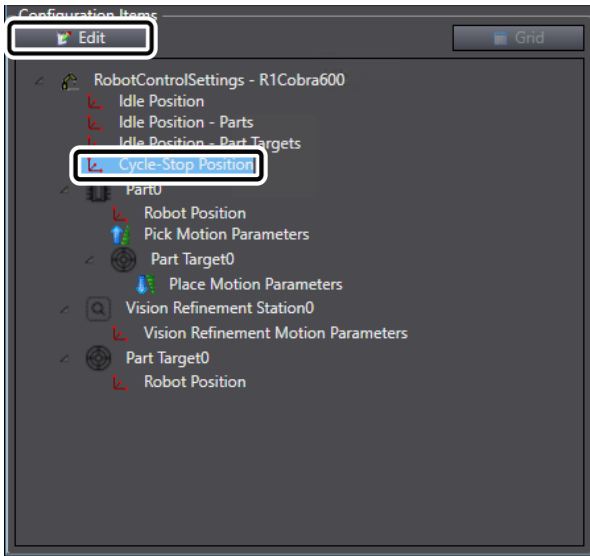
C# Program

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 a **Cycle-Stop** configuration item, or select an item and click the **Edit** button. The Cycle-Stop Position editor will open.



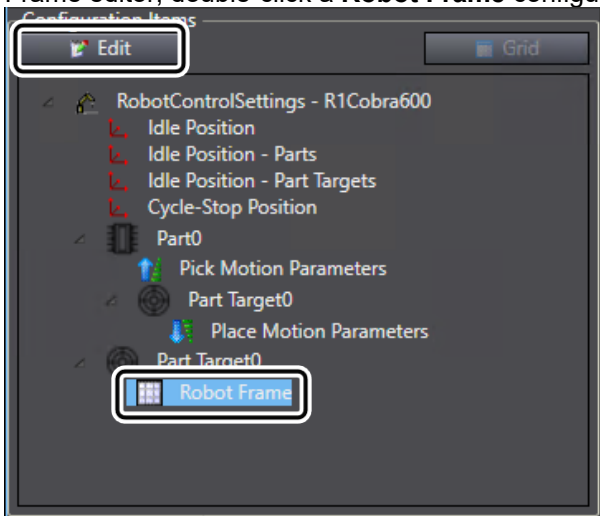


## ● Robot Frames

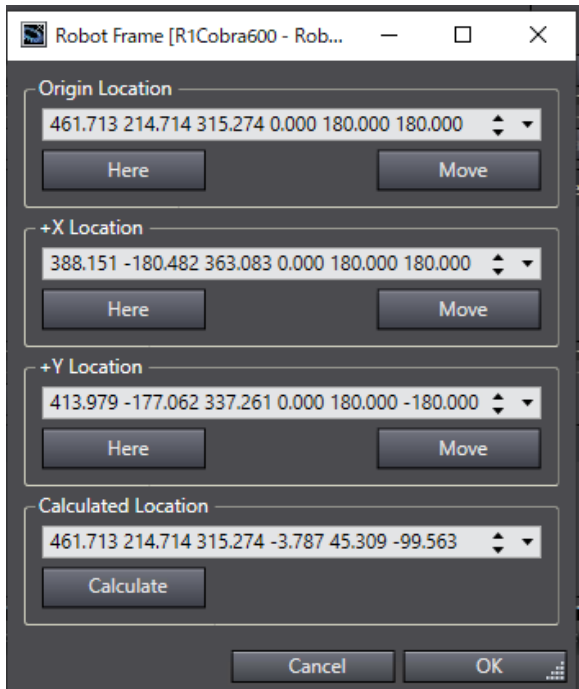
Robot frames (also known as reference frames) are useful because they allow you to teach locations relative to the frame. If the location of the frame changes in the workcell, you simply have to update the frame information to reflect its new location. Then you can use any locations created relative to that frame without further modifications.

A process pallet is typically taught relative to a reference frame. This avoids the problem of teaching many individual pallet positions and then having to re-teach 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 a **Robot Frame** configuration item or select the item and click the **Edit** button.



The Robot Frame editor will open.



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.

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 you to access and edit the common motion parameters used to optimize cycle time. You can change multiple speed/accel/decel parameters at the same time by clicking and dragging over the boxes before entering a value.

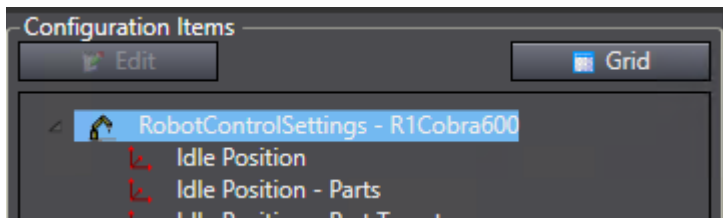
- Pick motion parameters
- Place motion parameters
- Vision refinement parameters
- Offset location parameters



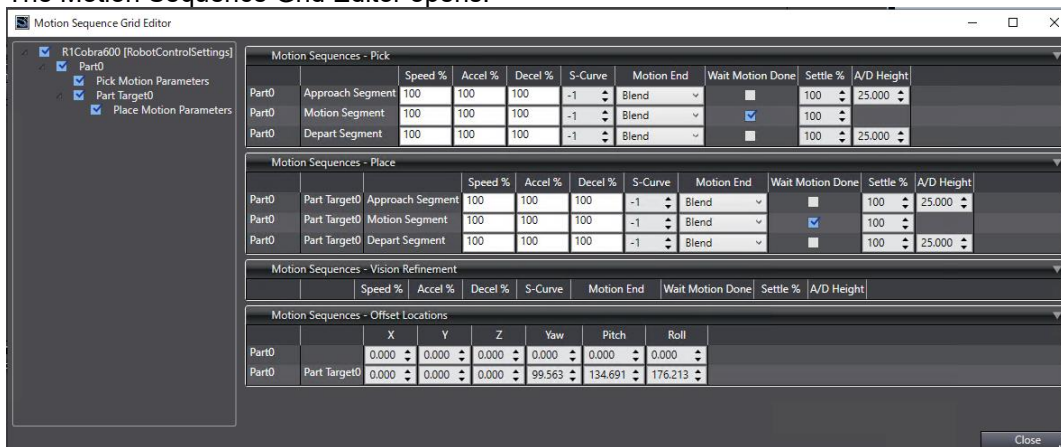
### Precautions for Correct Use

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, select the robot object in the Configuration Items group and then click the **Grid** button.



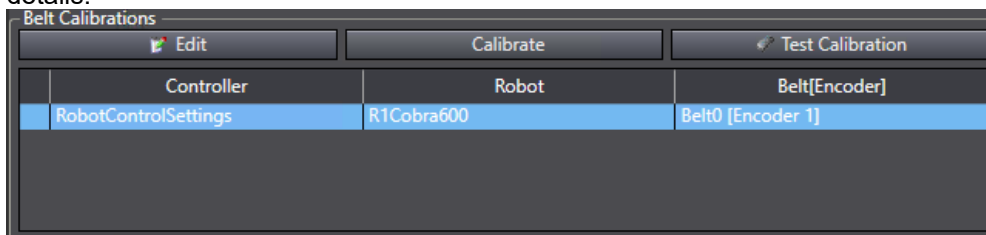
The Motion Sequence Grid Editor opens.



The left pane is used to select the items you want to display for editing. The right pane contains the editing parameters by group.

## 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 *Process List* for details.



Item	Description	Reference
<b>Edit</b> button	Click to edit the selected belt calibration.	<i>Editing the Belt Calibration Parameters</i>
<b>Calibrate</b> button	Click to begin the belt calibration procedure.	<i>Belt Calibration Wizard</i>
<b>Test Calibration</b> button	Click to test the current belt calibration. This button is not available until the belt has been calibrated.	<i>Testing the Belt Calibration</i>
Robot	Specifies the robot for the belt calibration. Double-click this item or click the Edit button to display the Belt Calibration Editor.	<i>Editing the Belt Calibration Parameters</i>
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.	<i>Editing the Belt Calibration Parameters</i>

### ● 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.

### ● Belt Calibration Wizard

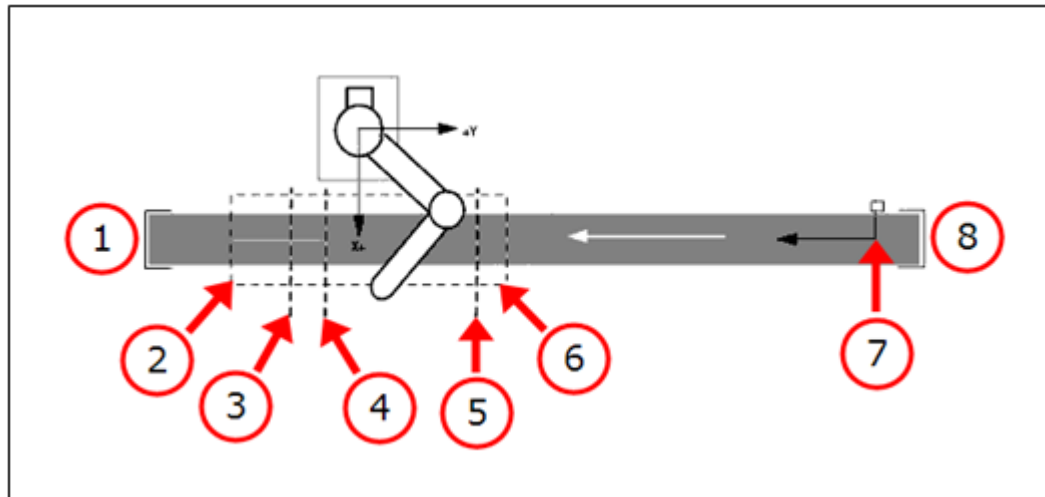
The Robot-to-Belt Calibration wizard provides an interactive series of steps that guide you through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and

used to populate the fields in the Belt Calibration Editor.  
Click the **Calibrate** button to begin the Belt Calibration wizard.

### ● Editing the Belt Calibration Parameters

After the belt has been calibrated using the Belt Calibration wizard, you can manually edit the stored belt calibration parameters and allocation limits, 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.

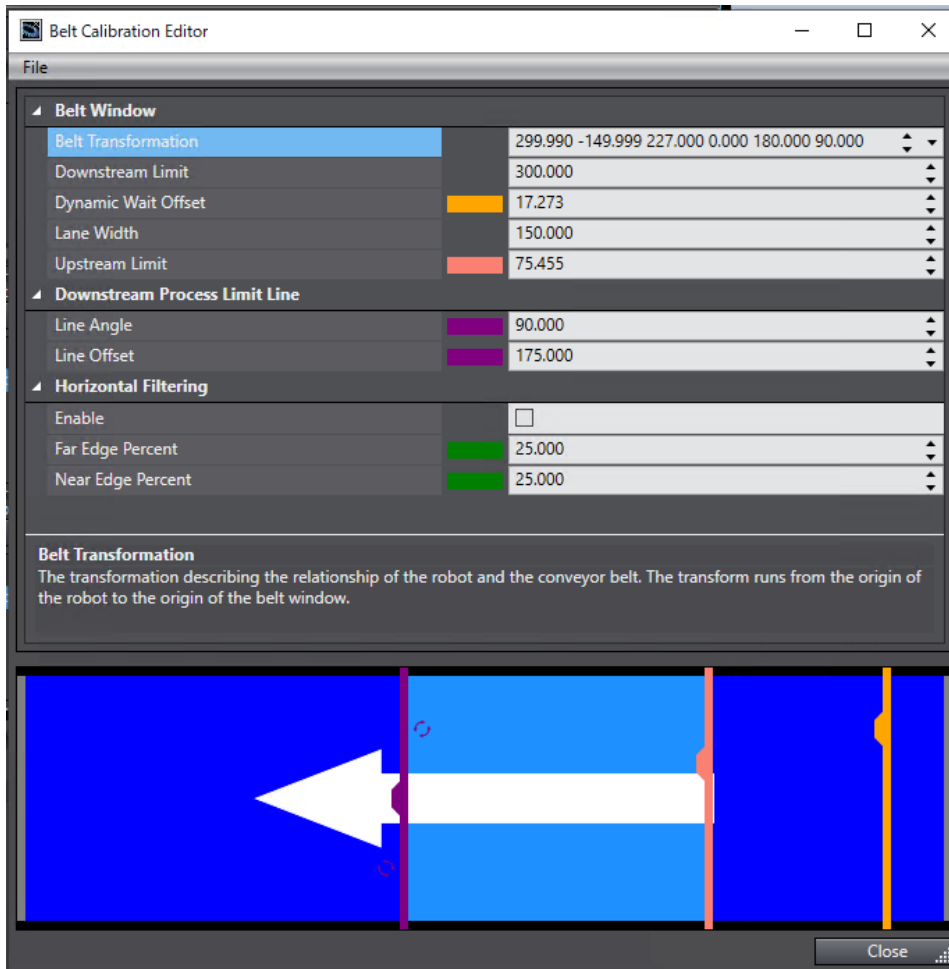
The belt must be calibrated using the wizard before the values can be manually edited.



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.

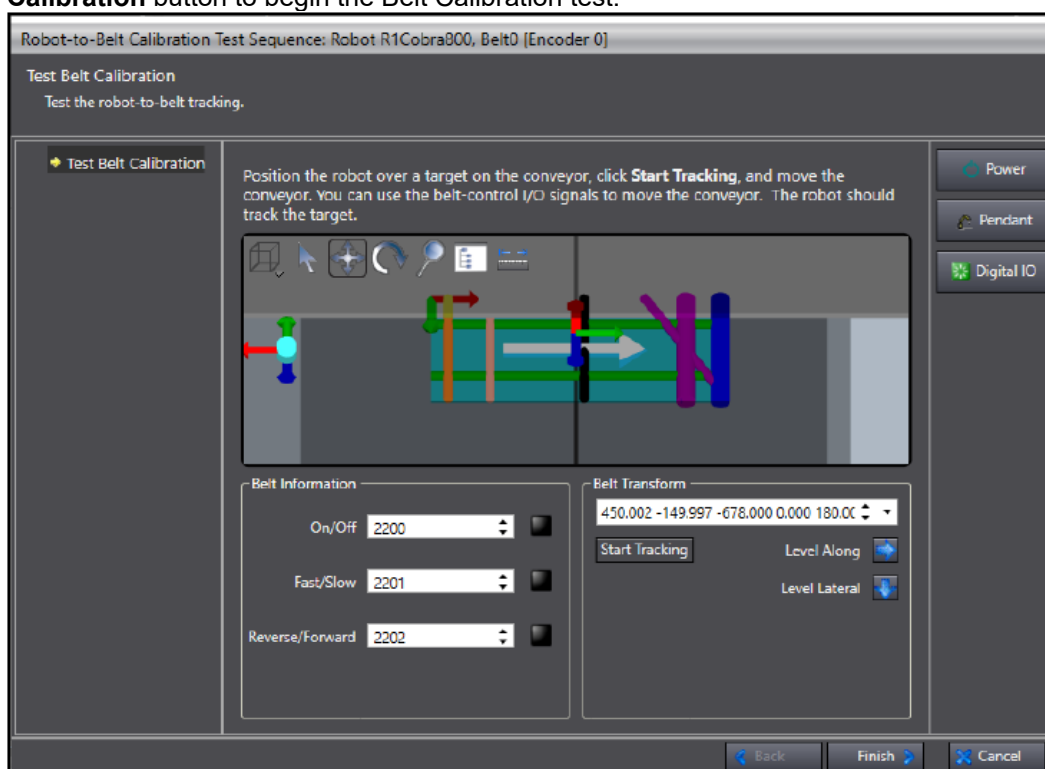


Group	Item	Description
Belt Window	Belt Transformation	The transformation describing the location of the belt window relative to the origin of the robot. This location also defines the upstream tracking limit of the belt window. The X-axis of the belt transformation is often referred to as the belt vector.
	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. It 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).
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

Group	Item	Description
		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, you 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. You can force different robots to pick in different horizontal regions (lanes) of the belt. For example, if you think of the conveyor belt as a three-lane highway (as shown in the previous figure), you may have robot one 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.

## ● Testing the Belt Calibration

The **Test Belt Calibration** page allows you 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 Belt 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 When you have confirmed correct tracking, 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.

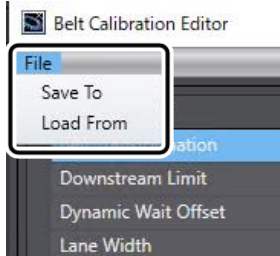


### Precautions for Correct Use

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.

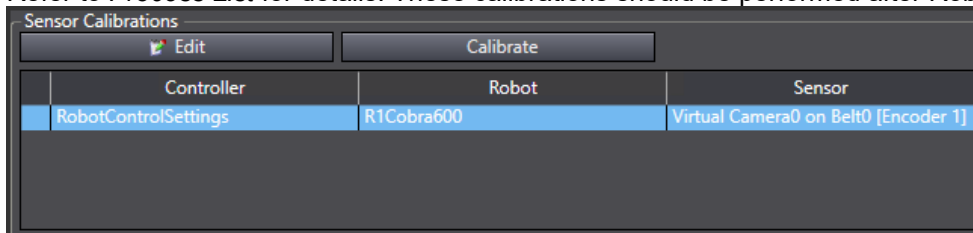
### ● Saving and Loading a Belt Calibration

After a calibration has been completed, the data can be saved by selecting **File - Save To** on the calibration editor menu. You can load a previously-saved calibration file by selecting **File - Load From** on the calibration editor menu.



## Sensor Calibrations

The Sensor Calibrations area defines the robot-to-sensor calibrations for the selected workcell process. Refer to *Process List* for details. These calibrations should be performed after Robot-to-Belt calibrations.



Item	Description	Reference
Robot	The robot specified for the belt calibration. Double-click this item or click the <b>Edit</b> button to display the Sensor Calibration Editor.	<i>Editing the Calibration Parameters</i>
Sensor	The sensor specified for the process. Double-click this item or click the <b>Edit</b> 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)	
<b>Edit</b> button	Click to edit the selected belt calibration.	
<b>Calibrate</b> button	Click to teach the selected process.	

### ● 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.

For details on the Vision Windows and image-editing controls in the wizards, refer to the *ACE Reference Guide*.

## ● Editing the Calibration Parameters

After the sensor is calibrated through the Sensor Calibration wizard, you can manually edit the stored sensor-calibration parameters, such as the robot-to-sensor offset. These parameters are edited using the Sensor Calibration Editor. To access the Sensor Calibration editor, click the **Edit** button in the Sensor Calibrations group. The Sensor Calibration Editor opens.

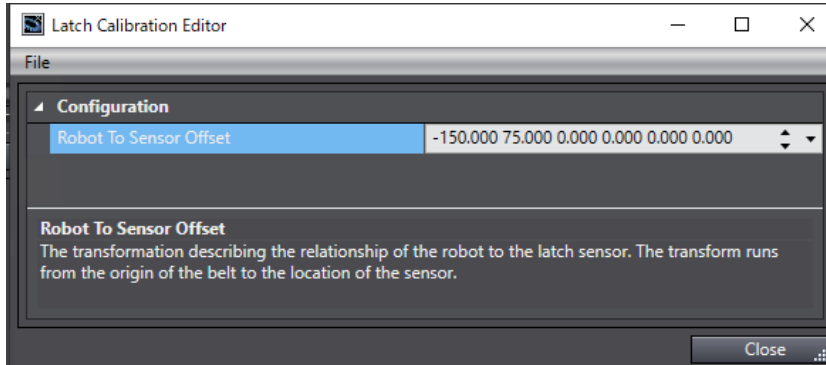


### Additional Information

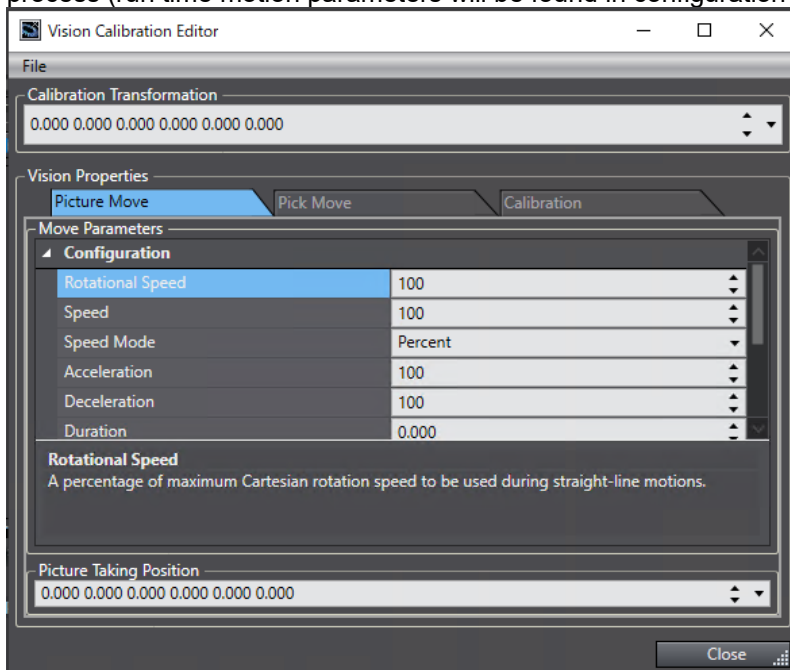
The sensor must be calibrated, or loaded from a previously-saved calibration data file, before the values can be manually edited.

The Sensor Calibration Editor contains the sensor properties configuration parameters. These parameters 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. They are not used during run time when the robot is performing the process (run time motion parameters will be found in configuration items).

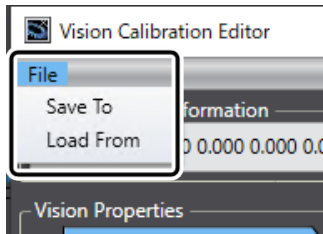


## ● Saving and Loading a Sensor Calibration

After a calibration has been completed, the data can be saved by selecting **File - Save To** on the calibration editor menu.

You can load a previously-saved calibration file by selecting **File - Load From** on the calibration editor menu.





## 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, you teach the initial locations and then the wizard automatically performs the robot movements to acquire enough data points to calibrate the system. In the manual procedure, you have to move the robot through each step of the process until enough data points have been acquired. The manual method is provided for cases where obstructions in the workcell do not allow for automated movement of the robot during the calibration process.

It is recommended that you use the calibration wizard, in order to obtain the optimum performance from your system.

The manual calibration procedure is available for the Fixed Camera and Refinement Camera calibrations.

### ● Using the Sensor Calibration Wizard

The Sensor Calibration wizard provides an interactive series of steps that guide you through the calibration process. Each step of the wizard contains graphical aids and text that describe the particular step being performed. As each step of the wizard is completed, the information from that step is collected and used to populate the fields in the Sensor Calibration Editor, which is described in the previous section.

### ● Calibration Types

The Sensor Calibrations area defines all of the calibration types that are used in the project. These calibration types are described below.

Type	Description
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 <i>3-3-3Robot-to-Camera Calibration</i> for details on the calibration process.
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 <i>3-3-3Robot-to-Camera Calibration</i> for details on the calibration process.
Belt Latch Calibration	The Belt Latch Calibration wizard configures the positioning of a latch sensor with respect to a belt. Similar to the Belt Camera Calibration wizard, it includes belt controls, indicators, and 3D visualization. However, instead of a vision window, this wizard has an additional indicator showing that the latch has been triggered. Refer to <i>3-3-4Robot-to-Latch Calibration</i> for details on the calibration process.
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.
Spacing Reference	The Spacing Reference Calibration wizard configures the positioning of

Calibration	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.
-------------	--

## Using Multiple Process manager Objects

The Sysmac Studio project can share robot-to-hardware calibration information between multiple Process Manager objects. If you use the same robot and hardware for each process, you can create multiple Process Manager objects in the Sysmac Studio project without having to repeat the calibrations. For example, assume you are setting up a project to handle the packaging of various fruits. You have three fruits that you want to pack: apples, oranges, and peaches. All fruits will use the same robot, sensor, and in-feed 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, you do not 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.

## Process Components Dealt by Process Manager

This section describes the Process components, which are accessed from the Process Manager object.

### ● 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 the **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.	<p>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.</p> <p>For this item, additional information is required based on the options below.</p> <p><b>Belt Mode: Vision</b> Vision properties are required. Spacing is not used.</p> <p><b>Belt Mode: Latch</b> The latch information is taken from the Belt object linked to the Belt Properties. Vision properties are not used. Spacing is not used.</p> <p><b>Belt Mode: Spacing</b> Vision properties are not used. Spacing is required.</p>
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.	<p>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.</p> <p>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.</p>

### ● 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.

### ● 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. It 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.

## ● 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.

## ● 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 allows you to modify certain parameters associated with the Source. As an example, the Vision and Static Source objects can be configured to interface with a feeder.

## ● Part and Part Target Instances

When an individual Part is located, it is represented by a Part Instance. When an individual Part Target point is identified, it is represented by a Part Target Instance. These objects identify the transformation and Part/Part Target information so the complete location can be resolved. If an individual Part Instance must be placed at a specific Part Target Instance, the Part Instance will have a link to the appropriate Part Target Instance.

Part Instance and Part Target Instance objects get allocated to a controller for processing by a robot. The Process Manager uses the Process Strategy to identify that allocation.



### Additional Information

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 is available to you and may 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 allows you to define your own Process Strategies using C# within the application.

### ● **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, and
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 Control**

The created Process Manager is executed as a task in the Task Status Control.

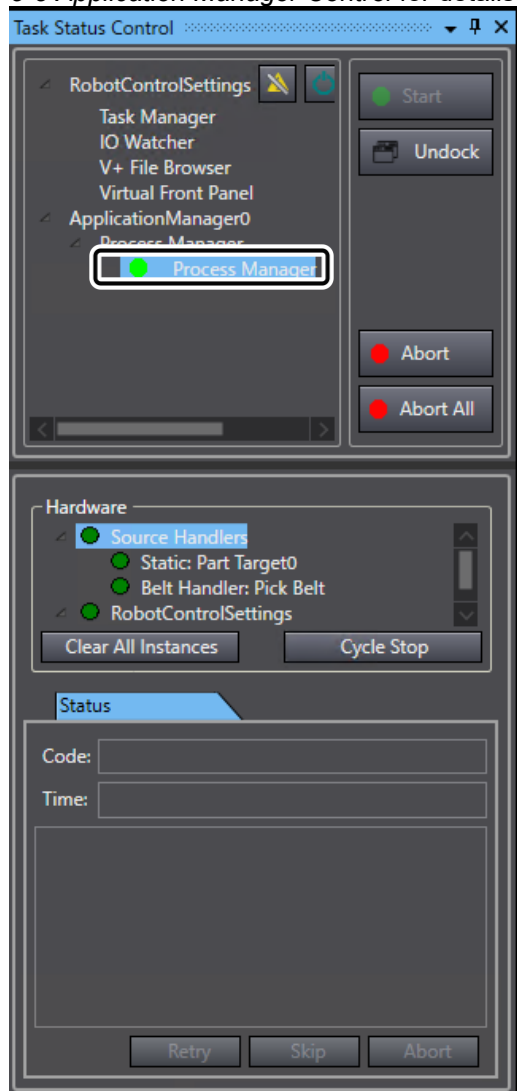
The Process Manager Control is used to start and stop a process-managed application, such as a Pack Manager packaging application.

Refer to 7-5 Task Status Control for details.

### ● Using the Process Manager Control

The Task Status Control interface is used to monitor and control Process Manager objects in the Sysmac Studio 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 7-5-8 Application Manager Control for details.






### ● 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 you select a task, the following functions become available.

Double-clicking a Process Manager task in the Task Status Control interface will open the item in the Edit Pane.

Function	Description
Start (Start)	Executes the selected task.
Undock (Undock)	Undocks the selected task's hardware and application information area. Close the undocked window to restore the view.
Abort (Abort)	Stops execution of the selected task.
Abort All (Abort All)	Stops execution for all tasks.

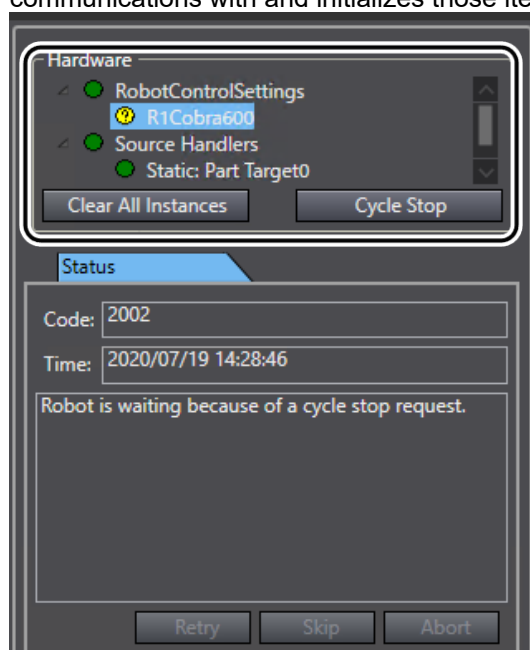
Tasks are marked with icons to indicate operational status.





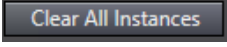
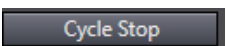
Icon	Description
	The task is running
	The task is stopped/aborted
	The task is idle

## ● 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.

Some items on the **Hardware** list are in Error and Warning states until the Process Manager establishes communications with and initializes those items.

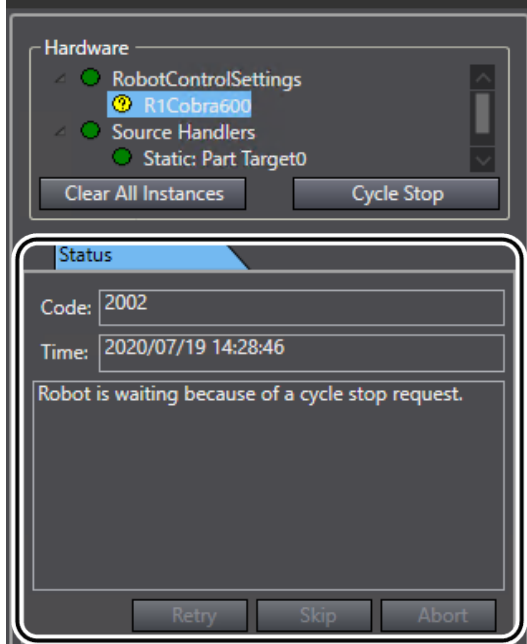


Item	Icon/Button	Description
Idle State icon		The item is idle
Operating State icon		The item is operating
Warning State icon		A warning condition is present for the item The specific warning message is displayed on the <b>Status</b> tab. Refer to Status Tab for details.
Error State icon		An error condition is present for the item The specific warning message is displayed on the <b>Status</b> tab. Refer to Status Tab for details.
<b>Clear All Instances</b> button		Clears all Part and Part Target instances from the system.
<b>Cycle Stop</b> 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. You can resume the robot(s) and process operations by clicking the <b>Cycle Stop</b> button.

		<p>The <b>Cycle Stop</b> button provides the capability to implement a variety of situations. For example, you could click the <b>Cycle Stop</b> button and leave the system running. When the system is in this state, all tracking is enabled. Therefore, you could either click the <b>Abort</b> button which stops everything, or you could simply resume the current cycle by clicking the <b>Cycle Stop</b> button again.</p>
--	--	---

## ● 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 the **Status** and **Instances** tabs 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.

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 are enabled

when in error state and return the corresponding response. See *Custom Error Programs* in the *ACE Reference Guide* 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.

## 5-9-6 Allocation Script

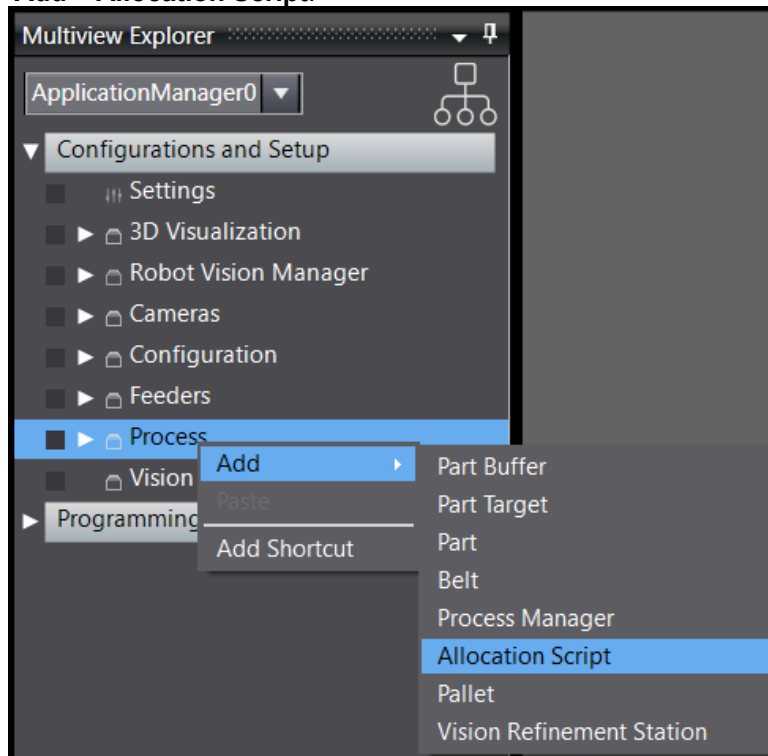
Create custom allocation script object which includes simple C# program editor. It is used when Process Strategy is edited with Process Manager.

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.

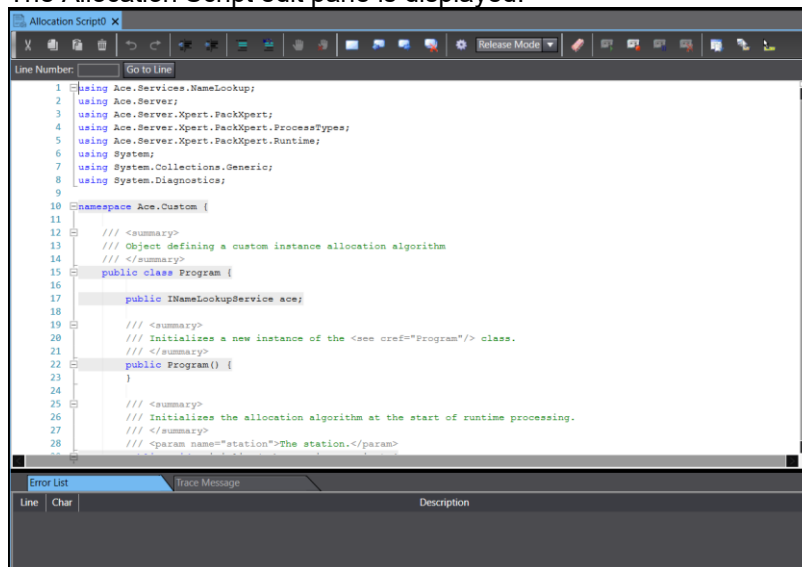


- 1 Right-click **Process** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Allocation Script**.



An **Allocation Script** is added under the **Process**.

- 2 Right-click the **Allocation Script**, and then select **Edit**. The Allocation Script edit pane is displayed.



Allocation Script objects are edited with the C# program editor. Refer to ACE Reference Guide for more information.

Refer to 5-11-1C# Program Editor for the details of the C# program editor.

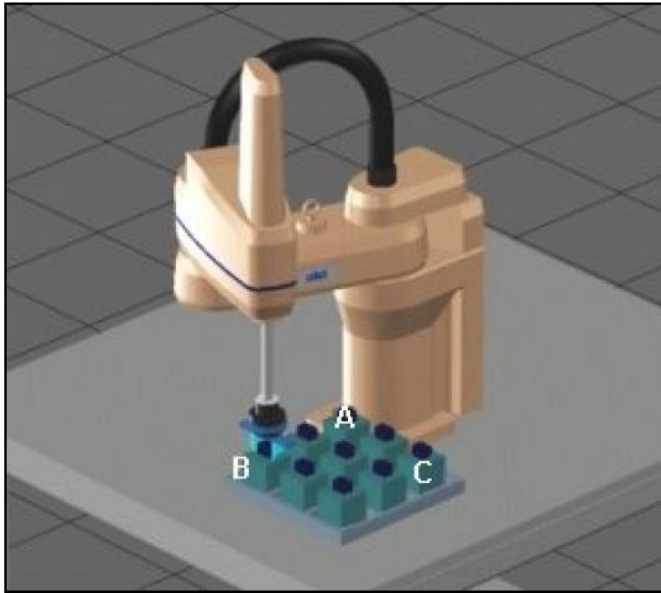
## 5-9-7 Pallet

Create the Pallet object to define the layout and settings of a pallet which can be used to pick parts from or place parts to.

### Defining a Pallet Layout

When defining a pallet layout, you are 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.

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



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 (B)	300.0	220.0	54.0	0.0	180.0	0.0
Position Along Y-axis (C)	220.0	300.0	54.0	0.0	180.0	0.0

You can 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 Sysmac Studio 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.

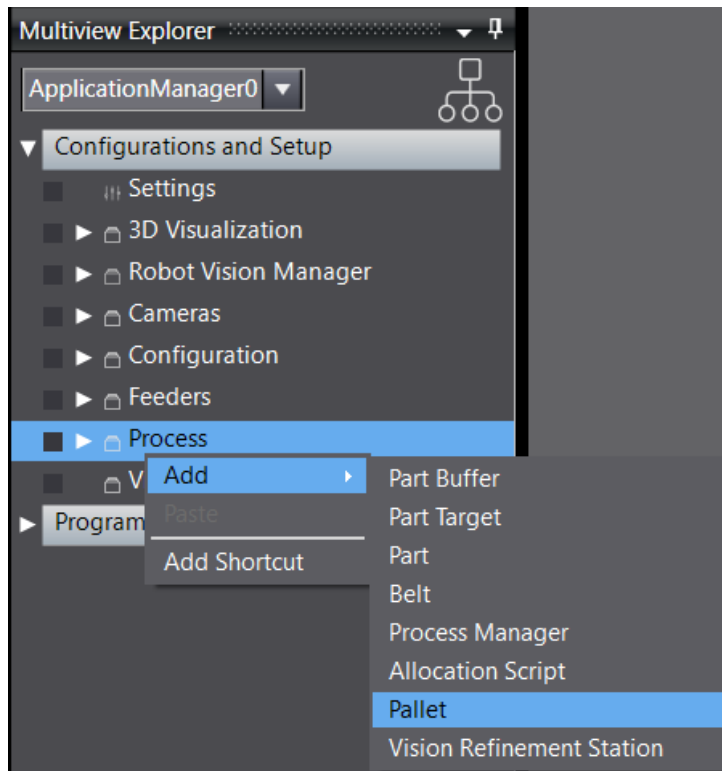
The initial pallet teaching process occurs in the Process Manager object configuration during calibration or process teaching (depending on the application needs). You can change the values obtained during the teaching process. Refer to 5-9-5 *Process Manager* for more information.

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

## Adding a Pallet

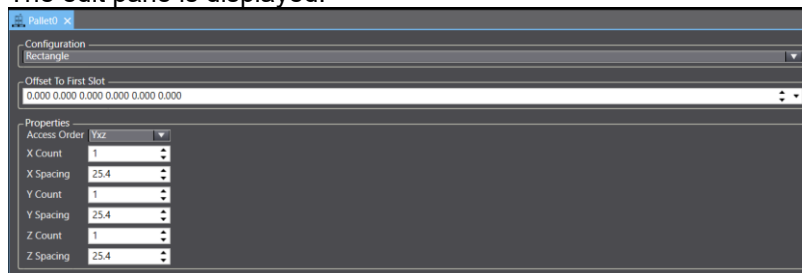
A Pallet object is added as follows.

- 1 Right-click **Process** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Process Pallet**.



A Process Pallet is added under the Process.

- 2 Right-click the **Process Pallet**, and then select **Edit**.  
The edit pane is displayed.



The setting items on the edit pane are listed in the following table.

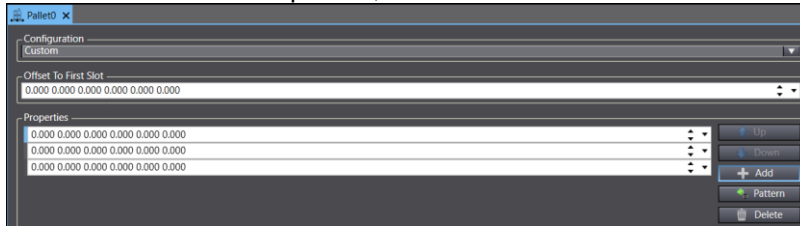
Item		Description
Configuration		Select the type of pallet being used from the followings. Rectangle: parts are arranged on the pallet in rows and columns. Custom: parts are arranged in rectangular or radial pattern. For a rectangular pallet, you specify the offsets, spacing, and part counts for X, Y, and Z. For radial pallets, you specify the start angle, angle spacing, part count, and radius.
Offset To First Slot		Specify the offset from the corner of the pallet to the first slot in the pallet.
Properties (When a rectangular pallet is selected)	Access Order	Specify the order how the robot accesses the pallet.
	X Count	Specify the number of slots on the X-axis of the pallet.
	X Spacing	Specify the slot spacing for the X-axis of the pallet in mm.
	Y Count	Specify the number of slots on the Y-axis of the pallet.
	Y Spacing	Specify the slot spacing for the Y-axis of the pallet in mm.
	Z Count	Specify the number of slots on the Z-axis of the pallet.
Properties (When a custom pallet is selected)		Set the slot to any position. Refer to <i>Custom Pallet Configuration Settings</i> for more information.

### ● Custom Pallet Configuration Settings

A custom Pallet is typically used for irregular slot arrangements. The custom Pallet configuration allows you to define each slot position. For example, if your pallet is 3 x 3 x 2, you will have 18 slot position items defined in the Properties area of the custom Pallet object as shown below.

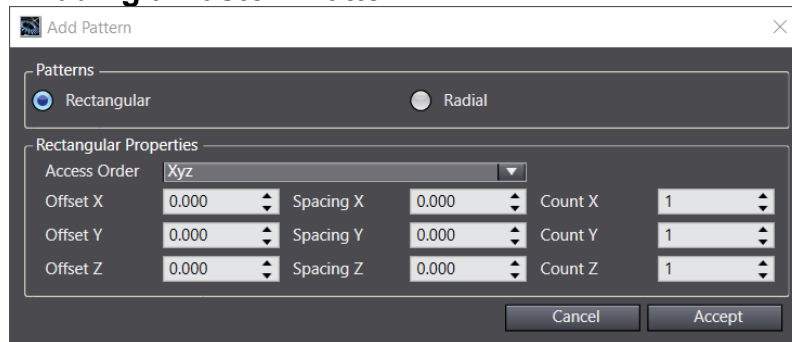
You can define individual slot positions manually using the **Add** button or automatically using the **Pattern** button as described below.

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 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 **Accept** button and the Custom Properties list will be populated accordingly.

## || Pallet Visualization

You can select a shape to represent the pallet in the 3D Visualizer. The shape is specified on the Part or Part Target object editor. The shape can be selected from a box, cylinder, or STL (STL mesh file). Refer to 5-9-3 Part and 5-9-2 Part Target for more information.

### 5-9-8 Vision Refinement Station

The Vision Refinement Station object defines a location with an upward-mounted camera that is used to improve the part-to-gripper orientation for more accurate placement of the part.

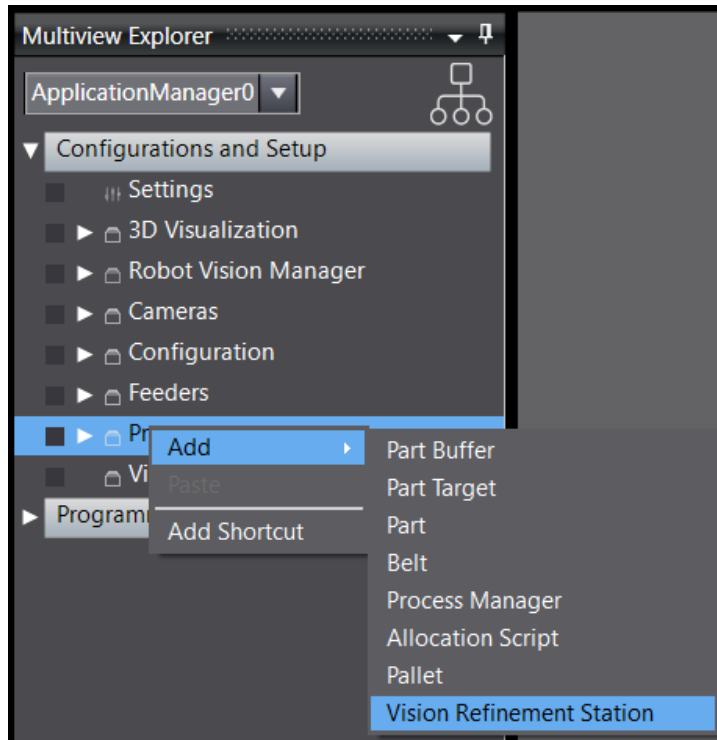


#### Precautions for Correct Use

The following information assumes you have already installed a physical camera, created a virtual camera, calibrated the camera, and created a vision tool and model.

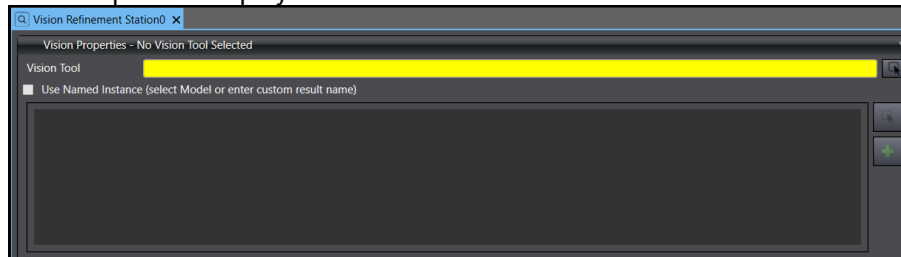
## || Adding a Vision Refinement Station Object

- 1 Right-click **Process** under **Configuration and Setup** in the Multiview Explorer, and then select **Add – Vision Refinement Station**.



A **Vision Refinement Station** is added under the **Process**.

- 2 Right-click **Vision Refinement Station**, and then select **Edit**. The edit pane is displayed.



The setting items on the edit pane are listed in the following table.

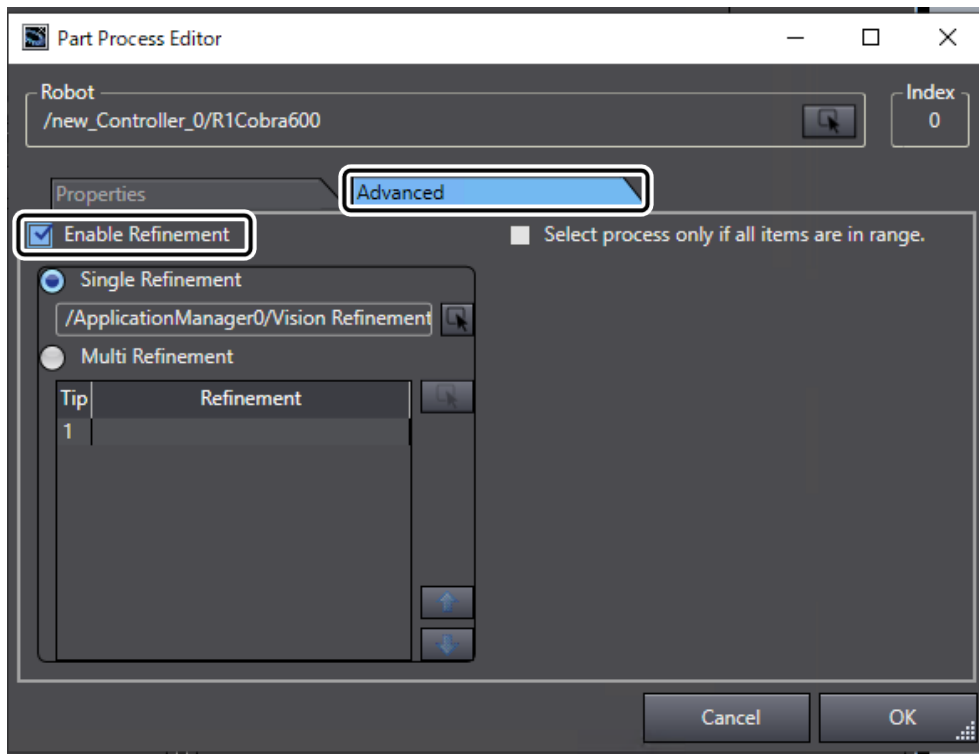
Item		Description
Vision Properties	Vision Tool	Specify the vision tool that will be used to locate the part in the gripper.
	Use Named Instance	Select the check box to display the named instance when the part is detected. It usually has the same name as the model name.

### ● 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 use the **Select** button to reference an existing Locator Model or use 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 you create the Vision Refinement Station, it must be added to a pick-place process. This is done using the **Advanced** tab of the Part Process Editor, shown in the following figure. Refer to *Part Process Editor* for more information.



## Editing Motion Parameters

After you add the Vision Refinement Station to the pick-place process, you can optionally edit the motion parameters for the station. This is done using Vision Refinement Motion Parameters that are accessed from the **Configuration Items** group. Refer to *Vision Refinement Motion Parameters* for more information.

# 5-10 Vision Tools

Add the tool to use for image processing with Robot Vision Manager.

The vision tool corrects the Images acquired from the camera or emulation camera to inspect easier, and then performs inspection.

These tools can perform the following functions.

- Image processing and filtering
- Coarse and fine location
- Position refinement
- Inspection
- Text and 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. Sysmac Studio 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.

Tool	Description
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 Sysmac Studio, their respective categories, and a brief summary of their functions. Refer to *Automation Control Environment (ACE) Version 4.0 User's Guide (I633)* for details on the functions and setting items of each tool.

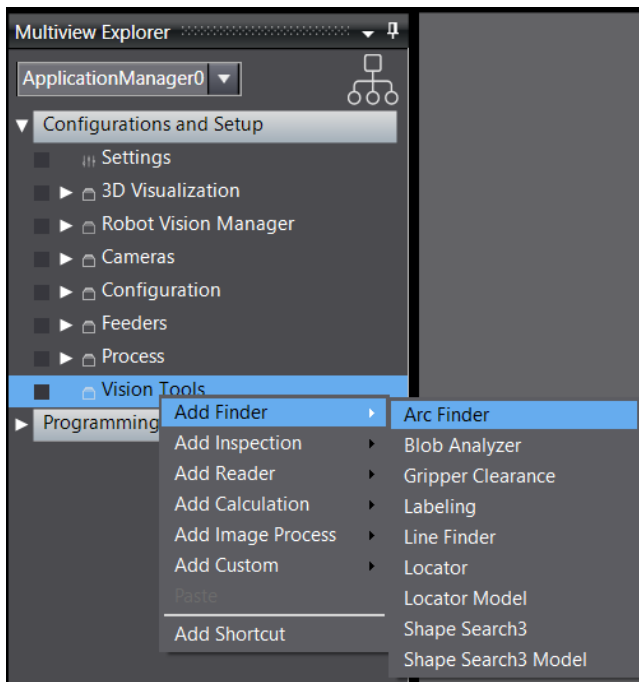
Category	Tool	Description
Finder Tools	Arc Finder	Identify circular features on objects and return the coordinates of the arc center, the angle of separation between the two ends, and the radius.
	Blob Analyzer	Process information within a region of interest to identify groups of similarly-colored pixels, called blobs, and return the position and size of each.
	Gripper Clearance	Use histograms to identify regions of clearance around detected objects.
	Labeling	Search the image for areas of a user-defined color or color range and return the results. Multiple colors or ranges can be searched.
	Line Finder	Identify linear features on and return the endpoint coordinates and line angle.
	Locator	Identify instances of objects defined by a Locator Model within a region of interest.
	Locator Model	This tool is customizable and accurate, but slower than Shape Search3.
	Shape Search3	Define the geometry of a model used by a Locator tool to identify instances.
	Shape Search3	Identify instances of objects defined by a Shape Search3

	Model	Model within regions of interest.
Inspection Tools	Arc Caliper	Identify one or more edge pairs in an arc-shaped or circular area and measure distance between the edges of each pair.
	Arc Edge Locator	Identify an edge or set of edges that meet user-defined criteria within an arc-shaped or circular area.
	Caliper	Identify one or more parallel edge pairs and measure distance between the edges of each pair.
	Color Data	Find the average color within a region and analyze its color variation and deviation from a specified reference color.
	Edge Locator	Identify an edge or set of edges that meet user-defined criteria within a rectangular region.
	Feeder Histogram	Calculate product density in specified regions of interest.
	Image Histogram	This tool is often used with the AnyFeeder object in regions associated with the dispense, pick, and front zones.
	Inspection	Compute gray-level statistics within a specified region of interest.
	Precise Defect Path	Judge the results of a tool and determine if they fall within a series of user-defined categories and filters.
	Precise Defect Region	Perform differential processing on an image to detect defects and contamination on a line or along an edge.
Reader Tools	Barcode	Read barcodes in an image and acquire text string data.
	Data Matrix	Read data matrices in an image and acquire text string data.
	OCR	Identify text characters in image and return them as text data.
	OCR Dictionary	Custom characters can also be registered to an OCR Dictionary for future identification.
	QR Code	Store dictionary data that OCR can use to identify text characters.
Calculation Tools	Calculated Arc	Calculate an arc based on a specified calculation mode and return the encompassing circle.
	Calculated Frame	Calculate a frame based on a specified calculation mode and return coordinates and orientation.
	Calculated Line	Calculate a line based on a specified calculation mode and return coordinates and dimensions.
	Calculated Point	Calculate a point based on a specified calculation mode and return coordinates.
Image Process Tools	Advanced Filter	Filter or alter an image using one of a variety of filter libraries, such as Background Suppression, Erosion / Dilation, and Color Gray Filter.
	Color Matching	This tool can be combined with other tools (including other Advanced Filters) to modify an image to the necessary specifications.
	Image Processing	Filter and analyze areas of a specified color or color range in images.
	Image Sampling	Filter or alter a gray scale image using one of a variety of filters, including logical and arithmetic calculations.
	Position Compensation	This tool can be combined with other tools (including other Image Processing tools) to modify an image to the necessary specifications.
Custom Tools	Custom Vision Tool	Extract an area of an image and outputs it as a new image source.
	Remote Vision Tool	Reserved for future use.

### 5-10-1 Adding Vision Tools

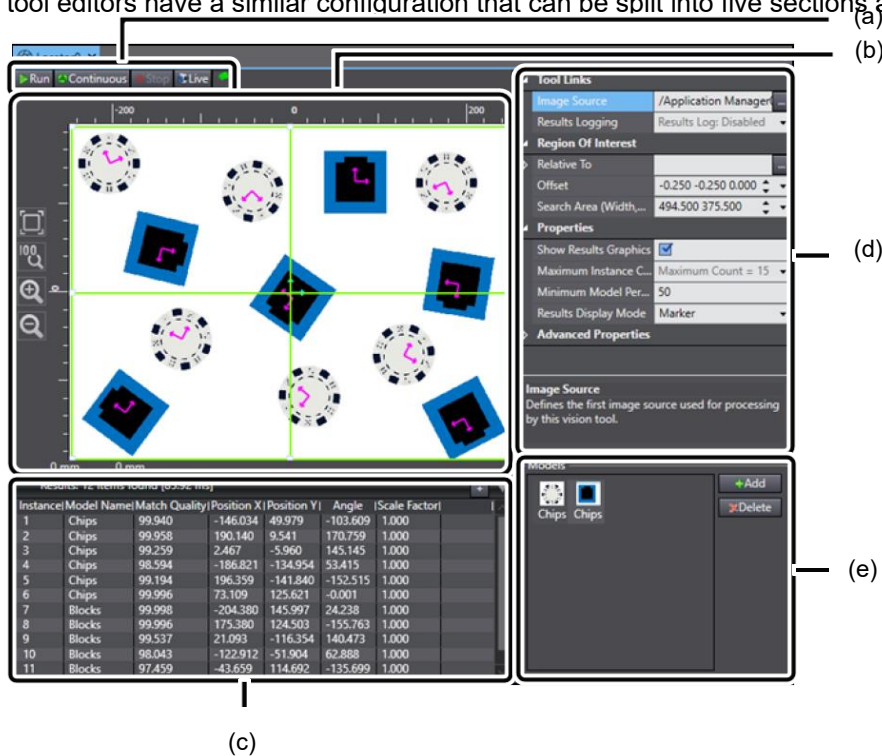
To add a new vision tool, right-click on the Vision Tools object in the Multiview Explorer, expand one of the categories and then select one of the tools, as shown in the figure below.



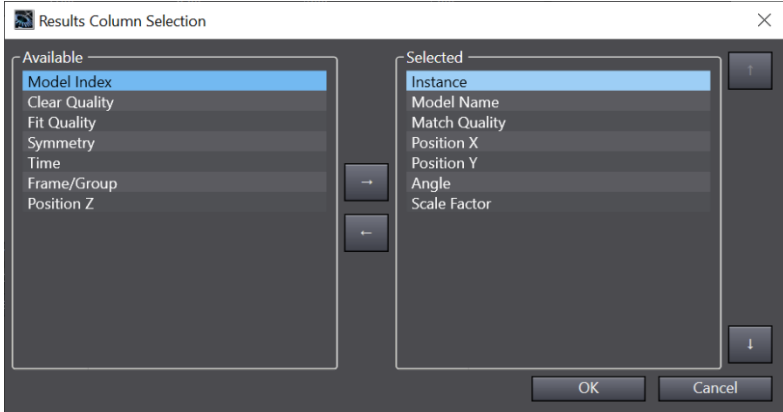


## 5-10-2 Vision Tool Editor

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	Description
(a)	Execution Buttons	<p>This area provides direct control of the tool with the following buttons.</p> <p><b>Run:</b> Run the tool once. This is available with most vision tools.</p> <p><b>Continuous:</b> Run the tool continuously. This is available with most vision tools.</p> <p><b>Stop:</b> Stop the tool or live operation. Can only be selected if Live or Continuous has been activated. This is available with most vision tools.</p> <p><b>Live:</b> Send images without running the tool. This is available with most vision tools.</p> <p><b>Auto Tuning:</b> Automatically set 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.</p> <p><b>New Image</b> button: Load a new image into the tool. The model will need to be retrained. This replaces the Run button in the respective tools.</p>

		This is available with the Locator Model and Shape Search 3 Model tools. <b>Register to OCR Dictionary</b> button: Register a character to the OCR Dictionary. This is available with the OCR Dictionary tool.
(b)	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 shown in the figure above displays markers for each identified instance within the green region of interest, which can be modified as needed.
(c)	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 <b>Plus</b> button which will display the <b>Results Column Selection</b> Dialog Box. Use this dialog box to select and arrange the data displayed in this area. 
(d)	Properties	This area displays the tool parameters. You can modify these parameters to change tool operation. Details of these parameters are found in the following sections.
(e)	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.

### 5-10-3 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.

The region orientation of some tools can only be controlled with the **Offset** property.

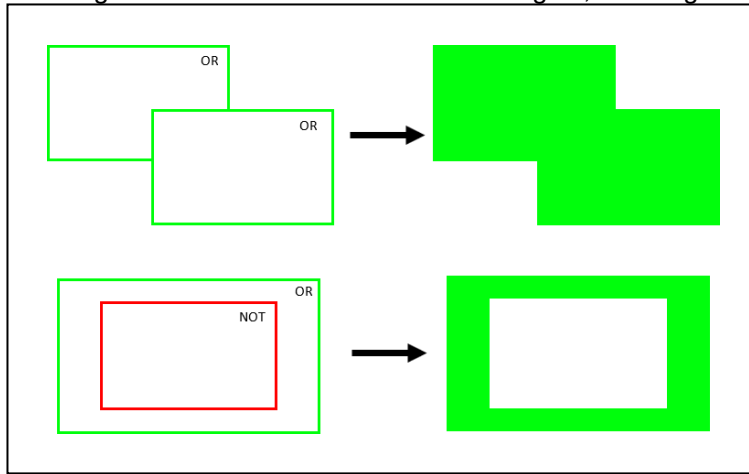
### Tools with Multiple Regions of Interest

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

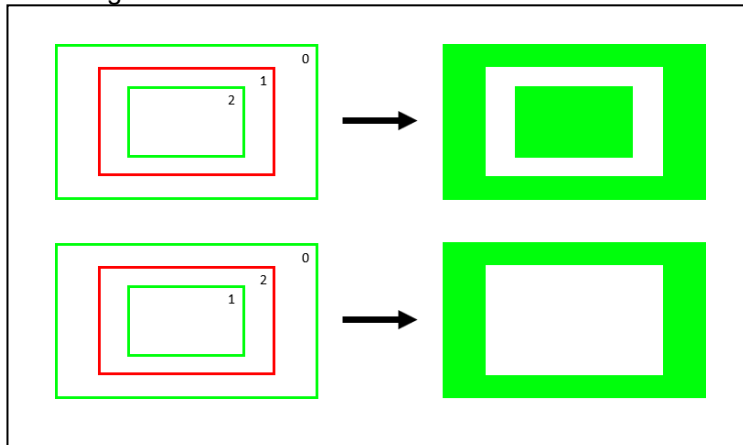
This defines the region as either inclusive or exclusive by setting it to **OR** or **NOT**, respectively. An inclusive region returns all detected contours or instances within its boundaries. An exclusive region hides all detected contours or instances within its boundaries. The figure below shows how 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 the following figure where 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.

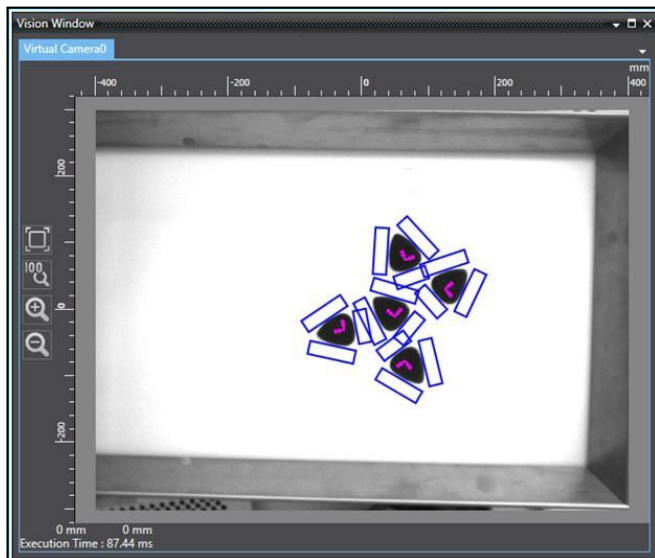
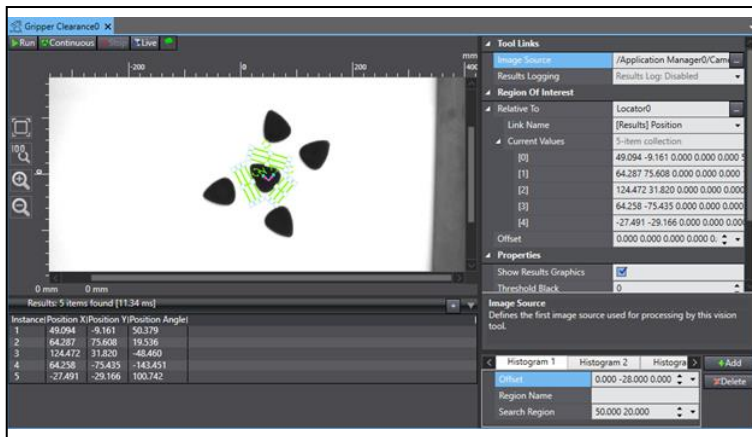


## Relative To Parameter Details

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 Vision Window as long as the property **Show Results Graphics** is enabled. Refer to the Vision Window below.



A tool set relative to another tool can be used to create a Robot Vision Manager Vision Sequence. Refer to 5-5-6 *Vision Sequence* for more information.

## 5-10-4 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
- Red, Green, and Blue (RGB) Color Space

#### ● Hue, Saturation, and Brightness (HSB) Color Space

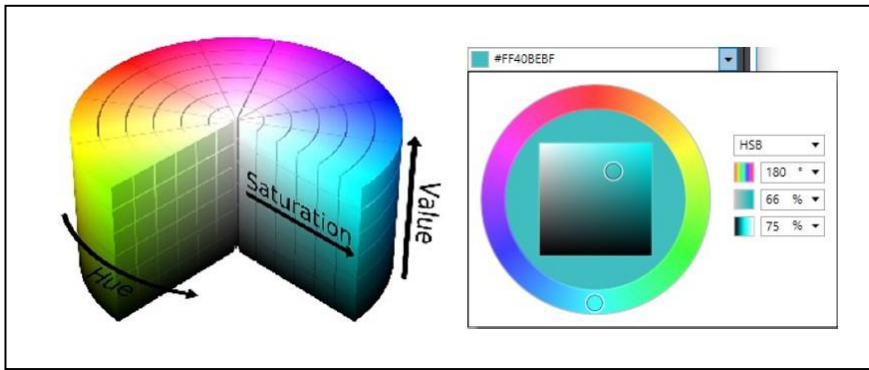
The HSB color space arranges colors of each hue radially around a central axis of basic colors, from white at the top to black at the bottom. Hue values are set in degrees from 0 to 360. Saturation and brightness are set in percentages from 0 to 100%.

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.

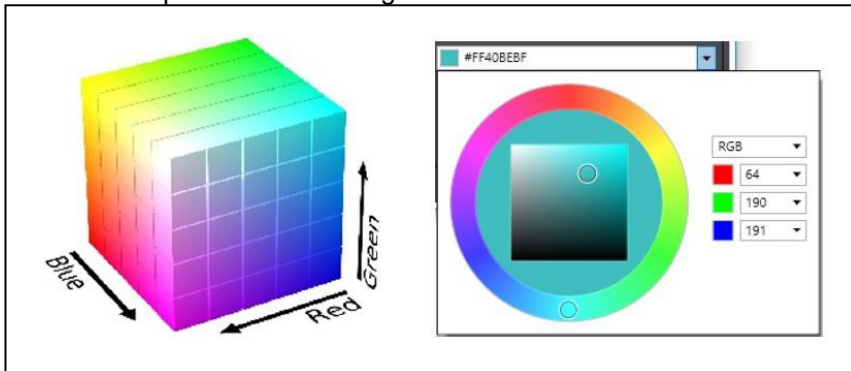
Brightness is the amount of white contained in the color. As the value increases, the color becomes lighter and becomes more white. As the luminance value decreases the color is darker and becomes more black.

HSB is also referenced as HSL (Hue, Saturation, Luminance) and HSV (Hue, Saturation, Value) in the Application Manager.



● **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.



Application Manager 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.

# 5-11 C# Program

You can program some applications of the Robot Integrated System such as the Application Manager and dynamic pick and place system in C# language.

As for the language specifications of C#, the editor uses Microsoft's C# version 6.0. Its variables, data types, constructs, functions, operators, and so on conform to the language specifications of the C# language.

For the language specifications of the C# language, refer to the references and guides provided by Microsoft, or commercially available technical books for the C# language. Refer to the following documents for API specifications of the Application Manager and samples of C# programs.

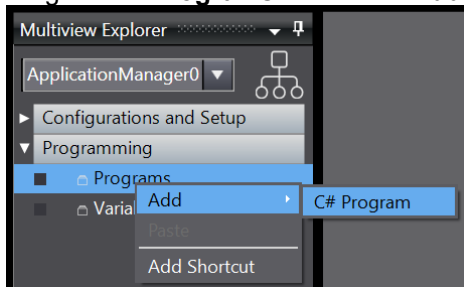
- ACE Reference Guide

## 5-11-1 C# Program Editor

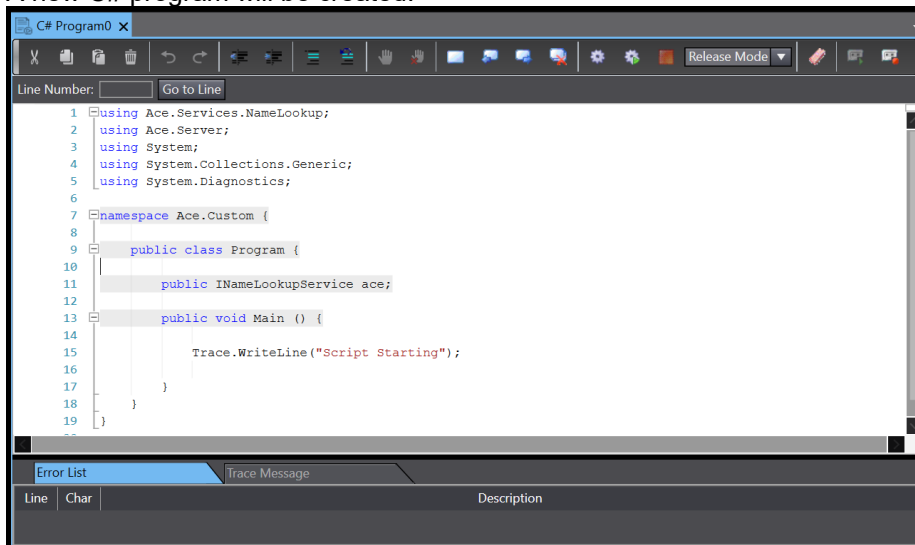
This section describes how to use the C# program editor to create and debug C# programs.

### Creating a New C# Program

- 1 Right-click **Programs** and select **Add - C# Program**.



A new C# program will be created.



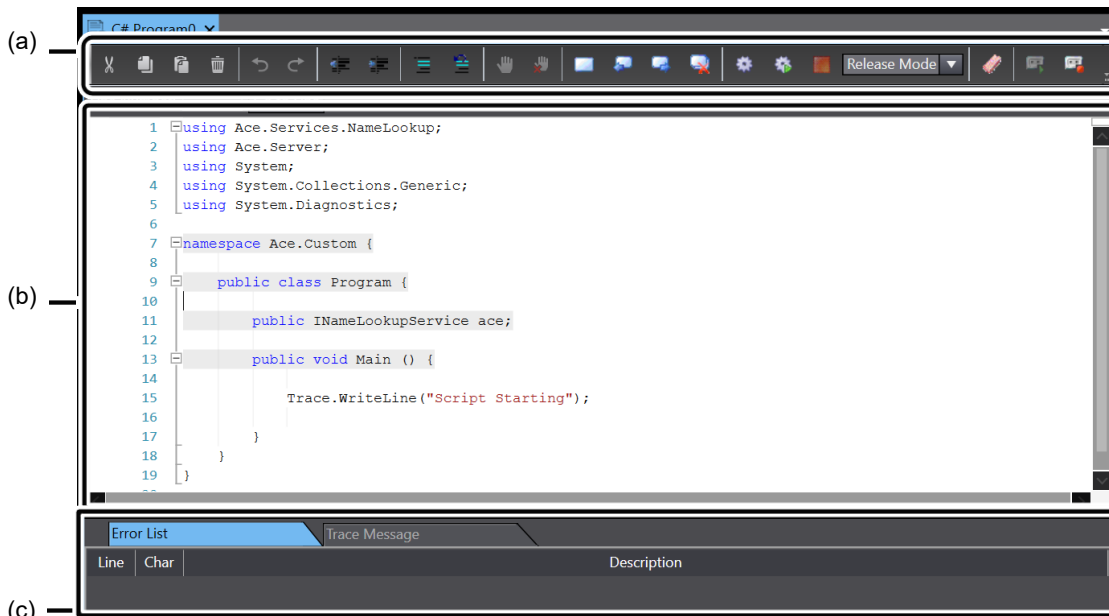
The new C# program is created with minimum program structure in place as well as a basic trace message to indicate when a script has started.

- **C# Program Names**

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

### C# Program Editor Tab Page

The functions of the C# Program Editor Tab Page are given below.



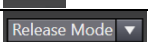















	Item	Description
(a)	Toolbar	The function buttons used for script creation are displayed. Refer to <i>C# Program Editor Toolbar</i> for details.
(b)	Editor	Use this editor to create scripts. Refer to <i>C# Program Programming</i> for details.
(c)	Tab page	Errors or trace messages are displayed. Refer to <i>Tab Pages of the C# Program Editor</i> for details.

## C# Program Editor Toolbar

You can perform the following operations with the buttons in the toolbar.

Button	Operation (shortcut)	Function
	Cut (Ctrl + X)	Cuts and saves the selected text to the clipboard.
	Copy (Ctrl + C)	Copies and saves the selected text to the clipboard.
	Paste (Ctrl + V)	Pastes the text saved in the clipboard to the cursor position.
	Delete (Del)	Deletes the selected text.
	Undo (Ctrl + Z)	Undoes the previous edit.
	Redo (Ctrl + Y)	Redoes the previous undo.
	Outdent (Shift + Tab)	Outdents the selected line.
	Indent (Tab)	Indents the selected line.
	Comment Selection	Comments out the selected line.
	Uncomment Selection	Uncomments the selected line.
	Toggle Breakpoint	Sets or clears a breakpoint, which is a stop point of the script, at the current cursor position only in DEBUG mode.
	Clear Breakpoints	Clears all breakpoints only in DEBUG mode.
	Toggle Bookmark	Sets or clears bookmarks on the selected line.
	Previous Bookmark	Jumps to the previous bookmark.
	Next Bookmark	Jumps to the next bookmark.
	Clear Bookmarks	Clears all bookmarks.
	Compile	Compiles the C# program to check the program for errors. Error information will be displayed on the Error List tab page.

	Compile And Run	Compiles and runs the C# program.
	Stop	Stops the C# program which is running.
	Mode Selection	Selects <b>Debug/Release</b> Mode. Use Debug mode to debug programs.
	Erase Trace Messages	Erases the information displayed on the Trace Messages tab page.
	Run Recorded Macro	Runs the recorded macro. Each time you click this button, a sequence of keystrokes recorded in the macro is executed.
	Record Macro	A Macro is a record of key input operations in the C# Program Editor. Click the button to start recording, and then click the button again to stop recording.
	Pause Recording	Pauses the macro recording. To resume recording the macro, click the button again.
	Cancel Recording	Cancels the macro recording.
	Display an Object Member List	Displays a candidate list of C# object members. Place the cursor after an entered C# object, and then click this button.
	Display Parameter Info	Displays an explanation of the C# object parameter at the cursor position.
	Display Quick Info	Displays information on an error at the cursor position as a tooltip.
	Display Word Completion	Displays the candidates of C# objects from characters that are in the middle of entry.
	Go to Line button	Jumps to the entered line number.
	Step Over	Displayed only in debug mode. Executes C# program step by step, not stopping at a code in a function.
	Step Into	Displayed only in debug mode. Executes C# program step by step in a function.
	START	Displayed only in debug mode. Restarts the stopped C# program.

## Tab Pages of the C# Program Editor

The following tab pages are available.

Item	Description
Error List	Displays a list of compile errors. The line number, error location, and error message are displayed for each error.
Trace Message	Displays trace messages (text included in Trace.WriteLine() calls) and warning information.

## C# Program Programming

The operating procedures for programming with the C# Program Editor are described below.

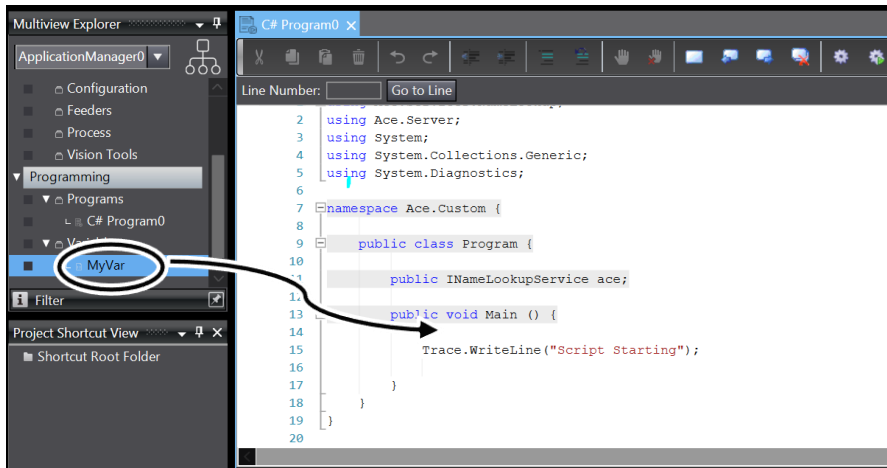
You can use the buttons in the toolbar to enter programs efficiently.

You can also create a variable declaration by dragging and dropping an object from the Multiview Explorer, to the C# Program Editor.

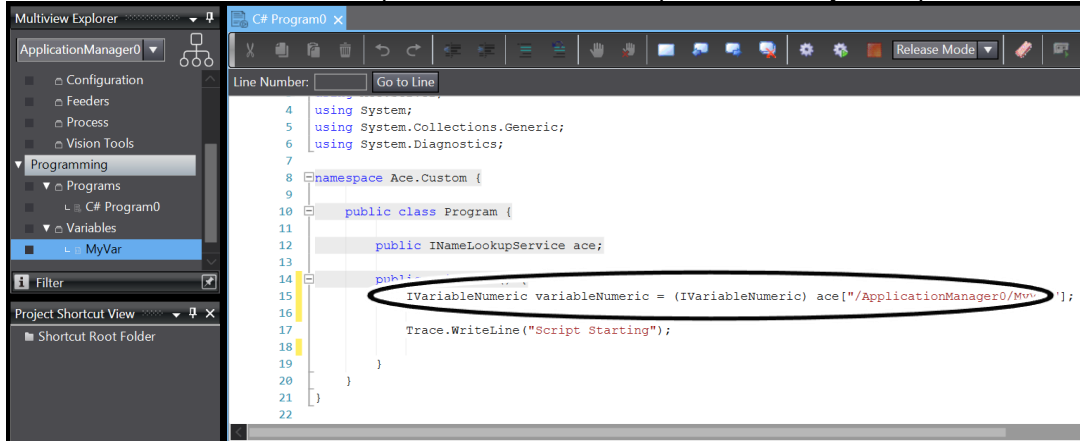
Here, the operation method is explained by using a variable declaration for using a C# variable object in the C# program as an example.

- 1 Drag and drop **MyVar**, which is under **Variables** under **Programming** in the Multiview Explorer, to the point of insertion.





The variable declaration for the part is inserted into the point at which you drop it.

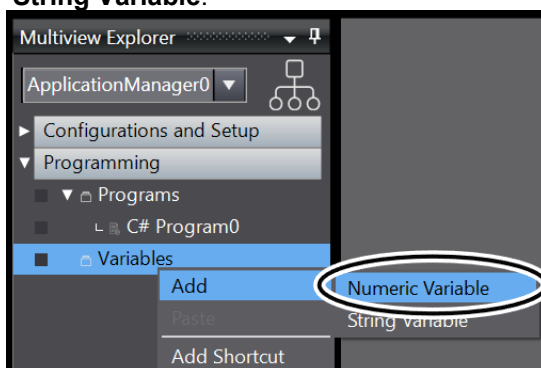


## 5-11-2 C# Variable Objects

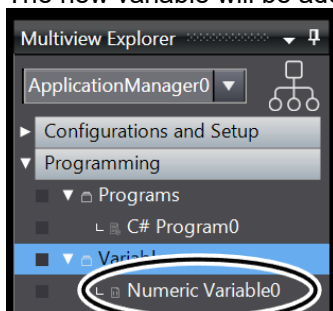
Numeric and string C# Variable objects can be used to share data between C# programs that run separately. Numeric types hold numerical values and string types hold text-based values used to represent characters. Valid string characters include 128 8-bit ASCII characters.

### Creating a New C# Variable Object

- 1 Right-click **Variables** in the Multiview Explorer and select **Add** and then **Numeric Variable** or **String Variable**.

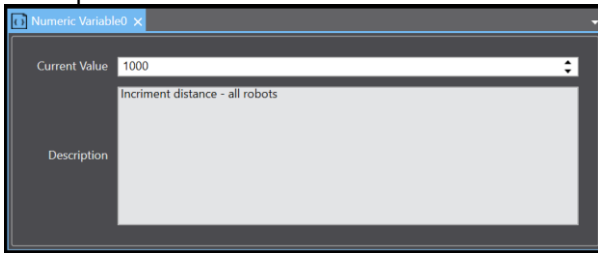


The new variable will be added to the Variables list.



## 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. This editor allows you to assign a current value and provide a description.



## Other Functions

Several functions are available for C# Variables objects. You can right-click a C# Variable object in the Multiview Explorer to do the following operations.

- Cut
- Copy
- Delete

Pasting will prepend *Copy\_1\_of\_* to the name of the new C# Variable object.

- Add To Watch

Selecting **Add to Watch** will place the C# Variable object in the V+ Watch window. Refer to *7-7-1 Adding C# Variable Objects to the V+ Watch Window* for details.

# 6

## 6 Online connection with the IPC Application Controller

This section describes the online connections, with the Robot Integrated CPU Unit and the IPC Application Controller, and data transfer procedures.

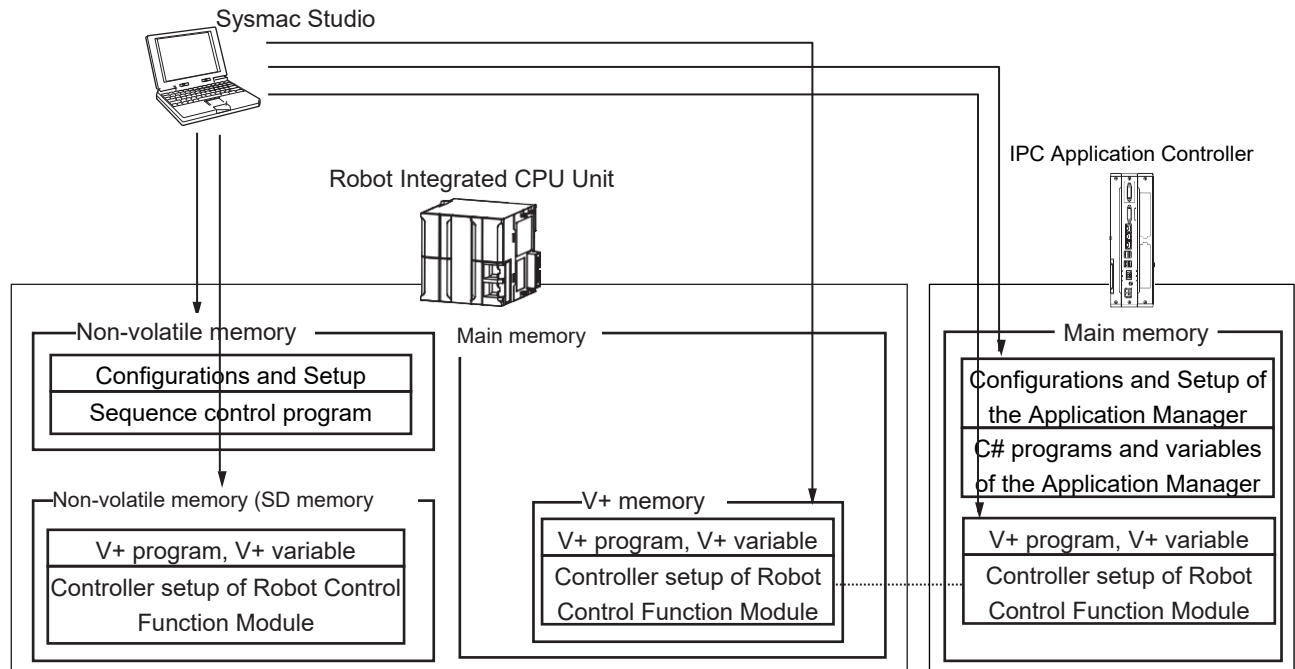
# 6-1 Data Arrangement of the Robot Integrated System Using the IPC Application Controller, and Device Relationship in the Sysmac Studio

You must transfer programs and settings of the Application Manager, that operates in the IPC Application Controller, in addition to programs and settings of the Robot Integrated CPU Unit, in the Robot Integrated System that uses the IPC Application Controller.

Each device of the Robot Integrated System, using the IPC Application Controller, has the following data.

Device	Data	
	Device in the Project	Description
Robot Integrated CPU Unit	Controller	Configurations and setup of the Controller, sequence control program
	RobotControlSettings	Controller setup for the Robot Control Function Module, V+ programs, V+ variables
IPC Application Controller	Application Manager	Configuration and setup of the Application Manager, C# programs

The IPC Application Controller manages the Part Processes and sends instructions, to operate the robot, to the Robot Integrated CPU Unit. It also processes the vision acquired from a camera connected to the IPC Application Controller, processes it as the vision sensor, and sends a signal to the Robot Integrated CPU unit. In order to achieve this, it is necessary to transfer the Controller setting of the Robot Control Function Module of the Robot Integrated CPU Unit and V+ programs and V+ variables to the IPC Application Controller.



In order to achieve such data placement, setting the Robot Integrated System that uses the IPC Application Controller and transferring programs, must be performed online to each device.

Use the following procedures for connecting and transferring.

Procedure	Device	Description	Reference
1	Robot Integrated CPU Unit	Online connection	<i>Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)</i>
2		Setup and program transfer	
3	IPC Application Controller	Online connection	<i>6-2 Online connection</i>
4		Setup and program transfer	<i>6-3 Project Synchronisation</i>

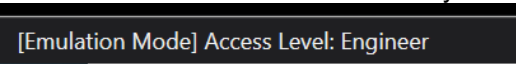
## 6-2 Online connection

Make an online connection to connect to the IPC Application Controller and use the online function.

### Precautions for Correct Use

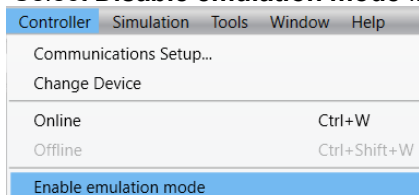
Match the Application Manager versions in the Sysmac Studio and in the IPC Application Controller. If the versions do not match, the IPC Application Controller cannot be connected.

- 1 Check the Status Bar to confirm that you are not in emulation mode.

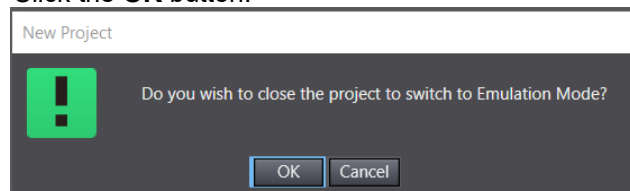


[Emulation Mode] Access Level: Engineer

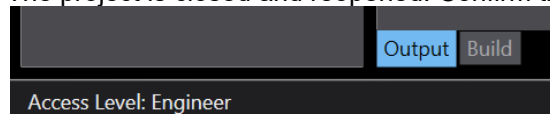
- 2 Select **Disable emulation mode** from the **Controller** menu when in emulation mode.



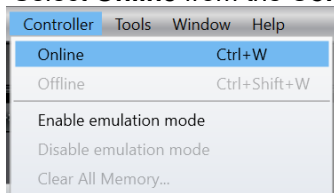
- 3 Click the **OK** button.



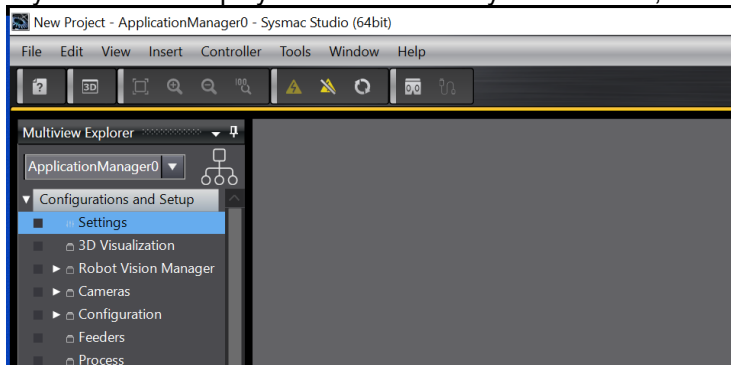
The project is closed and reopened. Confirm that you are not in emulation mode.



- 4 Select **Online** from the **Controller** menu. Or, click the **Online** button.



A yellow bar is displayed to indicate that you are online, and the **Synchronize** button is enabled.



## 6-3 Project Synchronisation

Compare the following data, of the Application Manager in the IPC Application Controller, with the project of the Sysmac Studio. And display the difference in the list.

- Configurations and Setup of the Application Manager, programming (C# programs, variables) data
  - Robot Control Module's controller settings of RobotControlSettings, V+ programs, V+ variables
- In addition, you can specify the transfer target and direction and download and upload the project data to the application manager in a batch.

⚠ **WARNING**

Check operations of the created user programs, data, and setting values carefully before proceeding to normal operation.

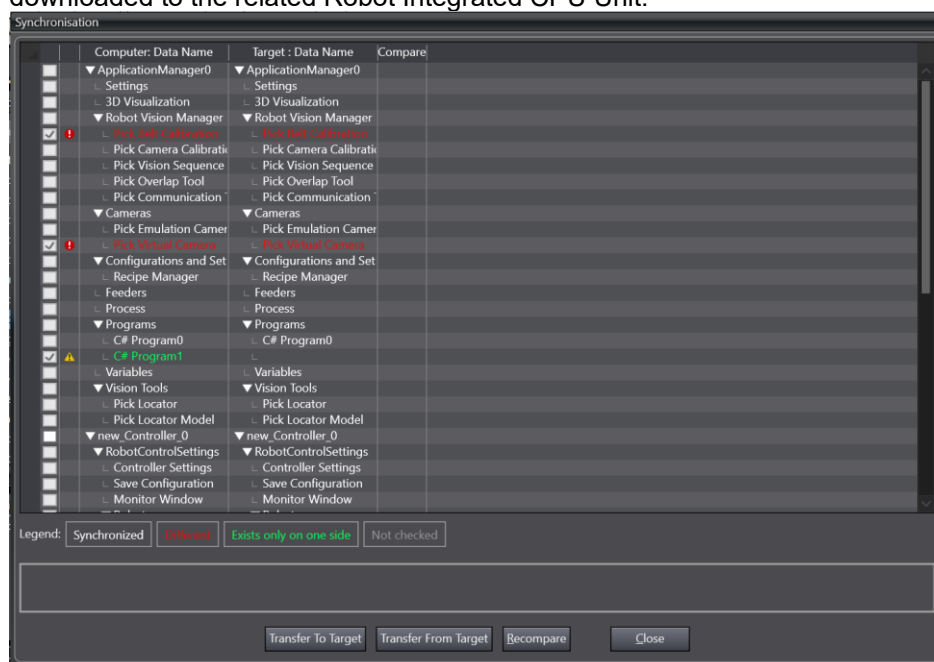
---

Ensure the enough safety before making any changes that may affect the operation of the robot.

- 1 Click the **Synchronize** button in the toolbar.



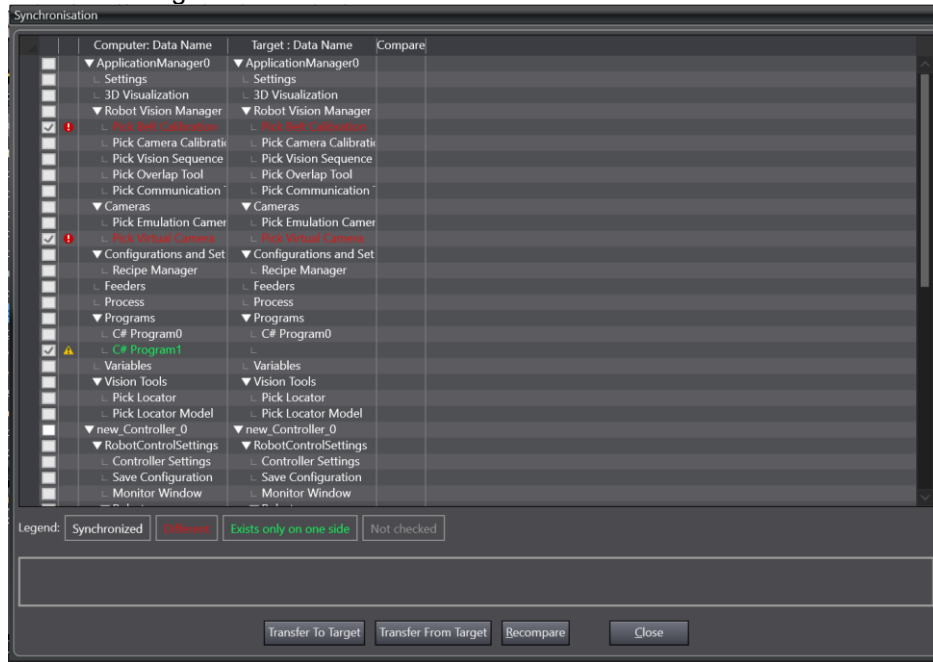
The following is listed: Configurations and Setup of the Application Manager device, programming, in addition, the robot subdevice settings, V+ programs, and V+ variables to be downloaded to the related Robot Integrated CPU Unit.



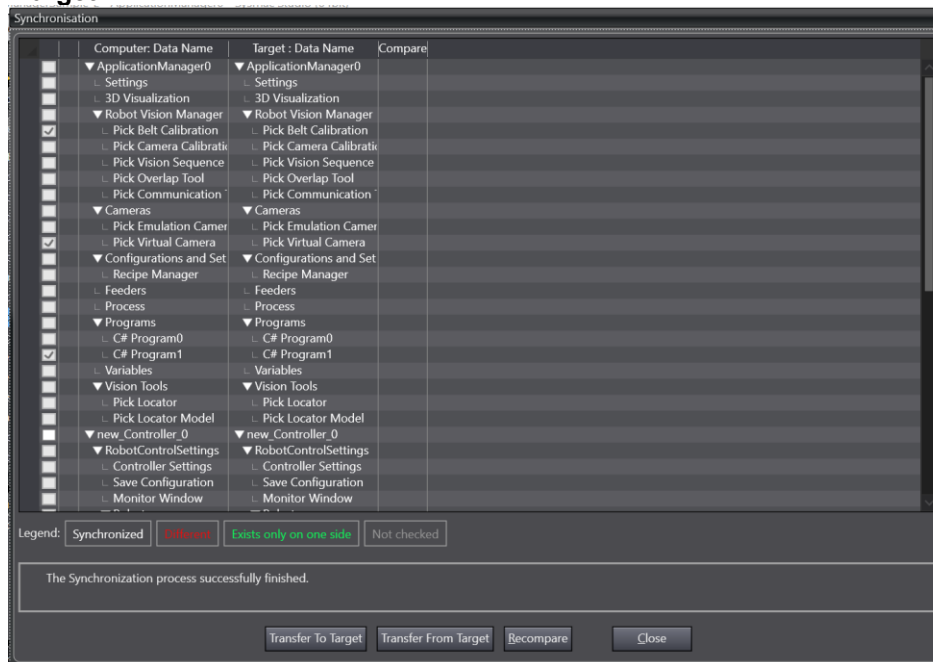
The results of the comparison are displayed as shown below. Detailed comparison results are displayed for V+ programs.

Legend	Description	Detailed comparison
Synchronized (no difference)	The project and Application Manager match.	—
Different	There are differences between the project and Application Manager settings, programs, and variables.	Detailed comparison results are not displayed.
Exists only on one side	Data exists only in either the project or the Application Manager.	Detailed comparison results are not displayed.

## 2 Select the target to transfer.



## 3 Click the **Transfer To Target** or the **Transfer From Target** button. Here, click the **Transfer To Target** button.



The data of the selected project is transferred to the Application Manager, and the synchronization process is completed.

## 4 Click the **Close** button.





# 7 Debugging of the Robot Integrated System

This section describes the functions used for debugging the Robot Integrated System including the IPC Application Controller and how to use them.

# 7-1 Introduction for Debugging of the Robot Integrated System Including the IPC Application Controller

For debugging the Robot Integrated System, there are offline debugging that uses the 3D simulation function of Sysmac Studio and online debugging that is performed by connecting to the physical device. Regarding the offline/online debugging functions of the Sysmac Studio, this section describes the operating procedures for their common and specific functions.

Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the debugging functions of the sequence control program.

⚠ **CAUTION**

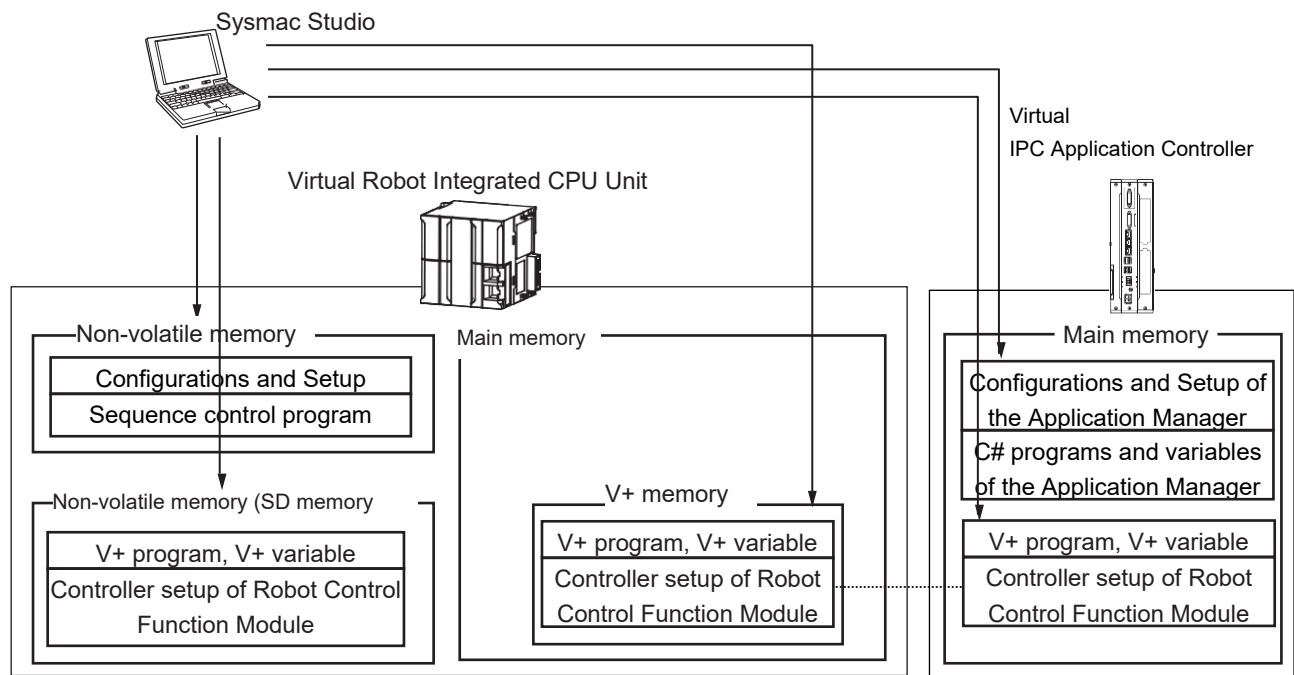
Although the Simulator in 3D Visualizer simulates the PLC and robot operations, there are differences in the operations and timings from the actual PLC and robot. After you debug the user program on the Simulator, always check operations on the physical Controller before you use the user program to operate the controlled system.

Accidents may occur if the controlled system performs unexpected operation.

## 7-1-1 Mechanism and Procedure of Offline Debugging

This section describes the offline debugging from the simulation operation.

Offline debugging of the Robot Integrated System including the IPC Application Controller is achieved by starting the virtual IPC Application Controller and virtual Robot Integrated CPU Unit on the Sysmac Studio.



Follow the procedure below to start the virtual IPC Application Controller and virtual Robot Integrated CPU Unit.

Procedure	Sysmac Studio Procedure	Reference
Start the virtual IPC Application Controller and virtual Robot Integrated CPU Unit.	Open a project in emulation mode.	4-6 <i>Offline and Online Modes of the Sysmac Studio and How to Use them Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)</i>
	Start the Simulator.	

Procedure	Sysmac Studio Procedure		Reference
Teaching in Simulation	Perform teaching by using 3D Visualizer and V+ Jog Control.		<i>7-23D Visualizer 7-4V+ Jog Control</i>
Offline Debugging	Use the Pack Manager.	Start the process.	<i>7-5-8Application Manager Control</i>
	Use the Robot Vision Manager .	Execute the V+ program.	<i>7-5-4 Task Manager</i>

## 7-2 3D Visualizer

The 3D Visualizer allows you to see simulated and real 3D motion for robots and other Process Manager items such as belts and parts the following information is displayed in the 3D Visualizer.

- 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

### 7-2-1 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 Offline Visualizer
- Collision detection in Emulation Mode
- Measurement between 3D objects
- Robot jogging
- Location visualization and editing
- Integration of workcell objects such as belts, parts, end effectors, and cameras

Use the following sections to understand the 3D Visualizer operation and functions.

### 7-2-2 Creating a 3D Workspace

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

- 1** Add OMRON robots in the EtherCAT Tab Page. Refer to *4-2-1 Creating a New Project* for how to add.
- 2** Add the 3D shape data 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.).



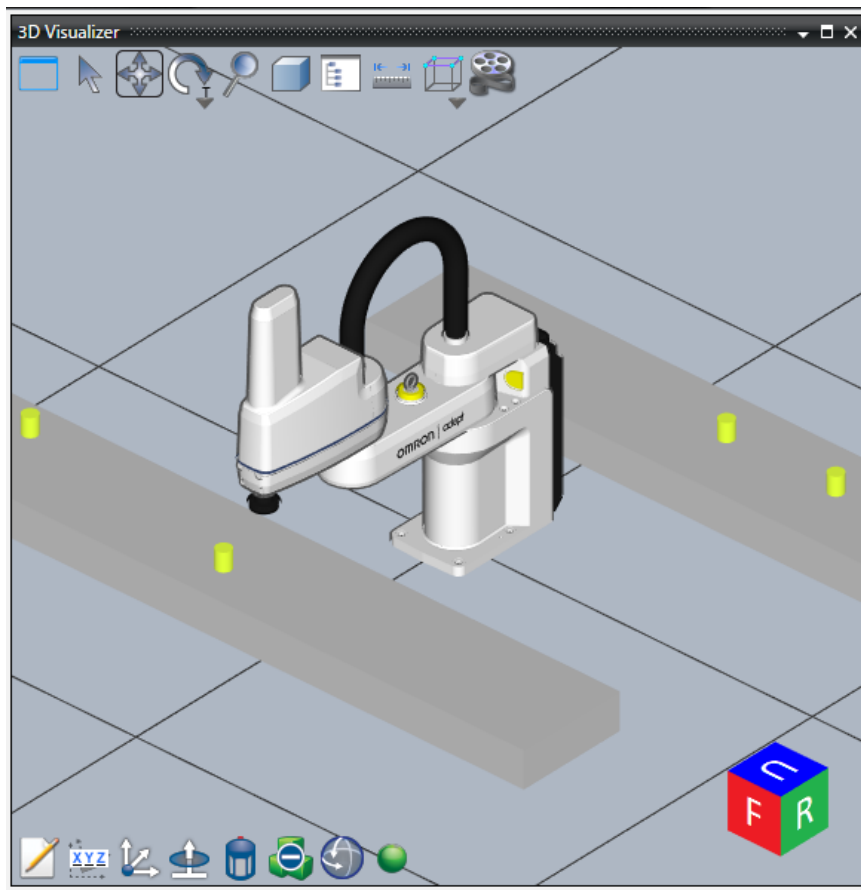
#### Additional Information

Register the boxes, cylinders, and imported CAD shapes in the Application Manager. Refer to the *Sysmac Studio 3D Simulation Function Operation Manual* (Cat. No. W618) for registration method.









### 7-2-3 Description of Icons













Use the main toolbar icon, or select **3D Visualizer** from the **View** menu to access the 3D Visualizer.

The 3D Visualizer has the following control icons.



The icons displayed at the lower left of the 3D Visualizer are associated with the objects that you selected in the 3D Visualizer. If no object is selected, these icons are not displayed. Some icons are associated with specific objects as described in the following table.

Icon	Name	Description
	Split Window	Splits the 3D Visualizer. Refer to <i>Split Window</i> for details.
	Selection	Selects the 3D shape data to edit.
	Translate	Translates the point of view in the 3D Visualizer. Click this icon or press the W key, press and hold the left mouse button, and then drag the mouse in the direction to translate.
	Rotate	Rotates the point of view in the 3D Visualizer. Click this icon or press the E key. Refer to <i>Rotate</i> for details.
	Zoom	Zooms in or out the 3D Visualizer. Click this icon or press the R key, press and hold the left mouse button, and then drag the mouse.
	Projection Mode	Changes the projection modes in the 3D Visualizer: parallel projection or perspective projection. Refer to <i>Projection Mode</i> for details.
	Scene Graph	Opens the Scene Graph dialog box to configure the visibility and collision detection settings for 3D shape data. Refer to <i>Scene Graph</i> for details.
	Measurement Ruler	Measures the distance between 3D shape data in the 3D Visualizer. Click this icon to display a ruler. Select the ruler and then use the <b>Select point 1</b> and <b>Select point 2</b> icons to adjust the ruler position. At the lower left of the 3D Visualizer, distance information is displayed for the coordinate components X, Y, Z. It is the easiest to place the ruler based on the camera angle that is set in alignment with the global coordinate plane.

	Snap	Moves 3D shape data, or a mount point or link point of 3D shape data, to a specified point. Refer to <i>Snap</i> for details.
	Record	Captures a simulation executed in the 3D Visualizer on video. Refer to <i>Record</i> for details.
	Edit	Displays the settings for the selected 3D shape data.
	Direct Position Edit	Changes the position and orientation of the selected 3D shape data. Refer to <i>Direct Position Edit</i> for details.
	Edit Workspace Position	Moves the position of the selected 3D shape data. This coordinate icon is displayed to allow you to assign the X, Y, and Z positions of the selected 3D data. If you place the cursor over an axis on the coordinate icon, the cursor shape changes. Left-click 3D shape data and drag it to a new position. If you place the cursor over the white circle on the coordinate icon, you can move 3D shape data freely by left-clicking and dragging it. The new position is reflected in the <b>Offset From Parent</b> value for the 3D shape data.
	Edit Workspace Orientation	Changes the orientation of the selected 3D shape data. This coordinate icon is displayed to allow you to manipulate the roll angle of objects. If you place the cursor over a roll axis (green ring), the cursor shape changes. Left-click an object and rotate it to a new roll angle. If you place the cursor over an axis on the coordinate icon after you set a new roll direction, the cursor shape changes. Left-click an object and drag it to a new position. If you place the cursor over the white circle on the coordinate icon, you can move objects freely by left-clicking and dragging them. The new position is reflected in the <b>Offset From Parent</b> value for the object. This icon is displayed only when a robot is selected.
	Jog Mode	Controls the jogging of robots in the 3D Visualizer. This icon is displayed only when a robot is selected. Refer to <i>Jog Mode</i> for details.
	Show Obstacles	Shows or hides obstacles. Refer to <i>Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual</i> (Cat. No. W595) for details. This icon is displayed only when a robot is selected.
	Show Robot Work Envelope	Shows or hides the work envelope of the selected robot. The displayed work envelope is applied to the tool flange of the robot without consideration of the applicable Tip Offset. This icon is displayed only when a robot is selected.
	Teach Point	Adds the current position of the robot to the V+ Variables for RobotControlSettings as a new position variable. This icon is displayed only when a robot is selected.
	Show/Hide Mount Points	Shows or hides mount points. Refer to the <i>Sysmac Studio 3D Simulation Function Operation Manual</i> (Cat. No. W618) for details on operating procedure.
	3D View Switching Tool	Switches the display direction of 3D shape data in the 3D Visualizer. Refer to <i>3D View Switching Tool</i> for details.





## Split Window

Use this icon to split the 3D Visualizer.

Click the icon to open the following window. To close the window, click X.



The functions of icons that you can select in the window are as follows.

Icon	Name	Function
	Not split	Does not split the 3D Visualizer.
	Split into two (Vertical)	Splits the 3D Visualizer vertically into two sections.
	Split into two (Horizontal)	Splits the 3D Visualizer horizontally into two sections.
	Split into four	Splits the 3D Visualizer into four sections.

## Rotate

Use this icon to rotate the point of view in the 3D Visualizer.



Press and hold the left mouse button, and then drag the mouse in the direction to rotate.

There are two rotation modes as follows.

Mode	Description
Tumbler rotation	3D shape data can be viewed from any angle.
Turntable rotation	The point of view can be rotated clockwise or counterclockwise around the Z axis of the world coordinate system. 3D shape data can be viewed in the range of $\pm 90^\circ$ vertically.

## Projection Mode

Use this icon to change the projection mode in the 3D Visualizer between parallel projection and perspective projection.

Icon	Name	Description
	Parallel projection	In this projection mode, the projection lines are connected in parallel between every point on the object and the point of view. It has a characteristic that an object is displayed in its true size regardless of the distance from the viewer. This projection method is suitable when you compare the sizes of objects that are placed.
	Perspective projection	A method of projecting an object based on the law of perspective. It has a characteristic that farther away an object from the viewer, smaller it appears, and closer the object to the viewer, larger it appears. This projection method is suitable when you display objects approximately in their size in the real world.

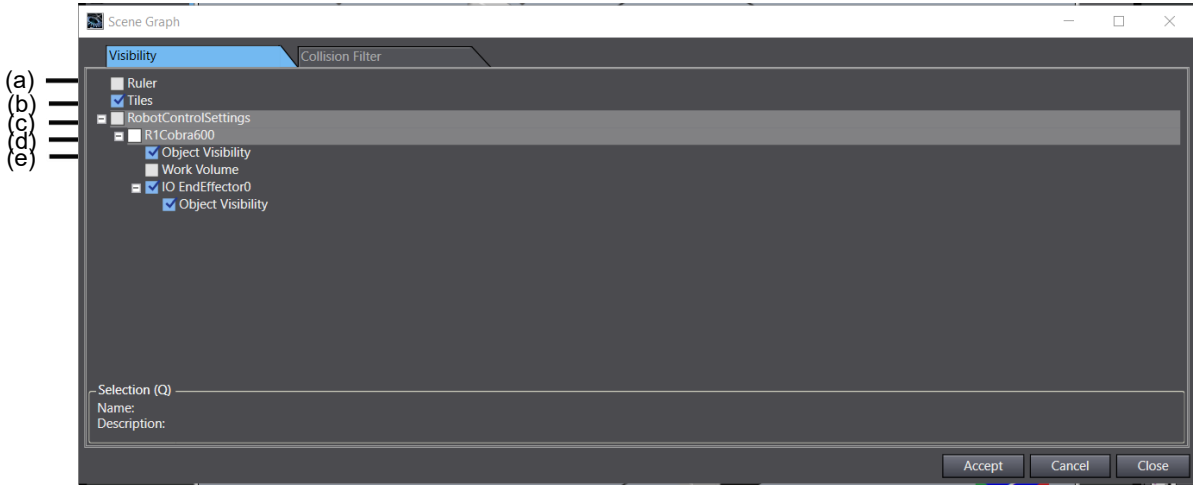
## Scene Graph

Use this icon to configure the visibility and collision detection settings for 3D shape data. The setting items are as follows.

### ● Visibility Tab Page

In the list displayed on the tab page, set whether to show or hide the tiles, ruler, and 3D shape data in the 3D Visualizer.

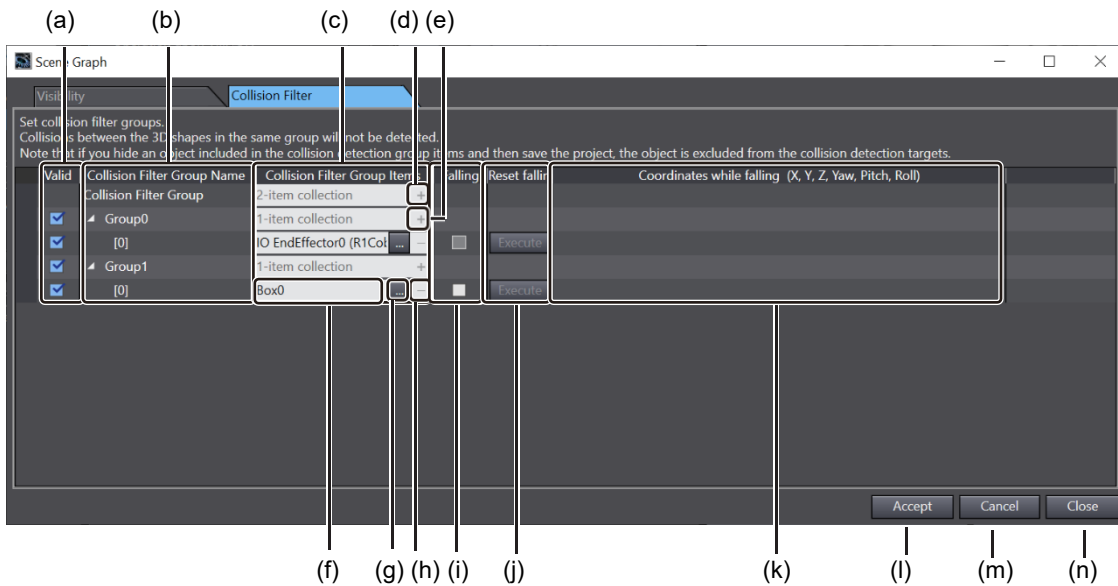
To display an item, select the corresponding check box.



	Setting item	Description	Set value	Initial value
(a)	Tiles	Select whether to display the XY plane with a 1,000-mm mesh when Z=0 in the 3D Visualizer.	Checked or unchecked	Checked
(b)	Ruler	Select whether to display a ruler when you measure the distance between 3D shape data in the 3D Visualizer. This is the same as the function of the Measurement Ruler icon.	Checked or unchecked	Unchecked
(c)	Devices	Select whether to show or hide in the 3D Visualizer the 3D shape data registered in the project by device.	Checked or unchecked	Checked
(d)	3D shape data	Select whether to show or hide in the 3D Visualizer the 3D shape data registered in the project by 3D shape data.	Checked or unchecked	Checked
(e)	Object Visibility			

### ● Collision Filter Tab Page

The Collision Filter tab is displayed only in the 3D Visualizer.



	Setting item	Description	Set value	Initial value
(a)	Valid check box	Select whether to enable the item in each collision filter group as a collision detection target. Select the check box to enable the item	Checked or unchecked	Checked



	Setting item	Description	Set value	Initial value
		as a collision detection target.	ked	
(b)	Collision Filter Group Name list	Manage the items in the Collision Filter Group list. The text box displays <i>*-itemcollection (* is the collision filter group number)</i> . Click the <b>Add Collision Filter Group</b> button to add a collision filter group. The group name to be added is <i>Group*</i> ( <i>* is the number of collision filter groups</i> ).	Text string	0-item collection (No collision filter group in the initial status)
(c)	Collision Filter Group Items list	Manage the items in the Collision Filter Group Items list. <i>*-itemcollection (* is the number of collision filter group items)</i> is displayed in the list. Click the <b>Add Collision Filter Group Items</b> button to add a collision filter group item.	Text string	0-item collection (No collision filter group item in the initial status)
(d)	Add Collision Filter Group button	Adds a collision filter group. Clicking this button adds a collision filter group to the list.	---	---
(e)	Add Collision Filter Group Item button	Adds a collision filter group item. Clicking this button adds a collision filter group item setting row to the list.	---	---
(f)	Collision Filter Group Item name	Set the target to add to the collision filter group. Refer to 7-2-4 <i>Collision Detection Settings</i> for the targets that you can set.	Text string	---
(g)	Collision Filter Group Item Selection button	Displays a dialog box in which you can select the target to add to the collision filter group.	---	---
(h)	Delete Collision Filter Group Item button	Deletes a collision filter group item from the collision filter group. Clicking this button deletes the item.	---	---
(i)	Falling	Clicking the <b>Accept</b> button with this check box selected causes the target 3D shape data to fall in the negative direction of the Z axis in the 3D Visualizer. This allows you to check how the falling 3D shape data collides with other 3D shape data.	Checked or unchecked	Unchecked
(j)	Reset falling	Resets the fallen target 3D shape data to the original position. When the <b>Falling</b> check box is selected, the target 3D shape data falls from the original position each time you click the <b>Execute</b> button.	---	---
(k)	Coordinates while falling (X, Y, Z, Yaw, Pitch, Roll)	The coordinate value indicating the position of a falling 3D shape data is displayed.	---	---
(l)	<b>Accept</b> button	Accepts the changes.	---	---
(m)	<b>Cancel</b> button	Cancels the changes and then, closes the Scene Graph dialog box.	---	---
(n)	<b>Close</b> button	Closes the Scene Graph dialog box.	---	---




Refer to 7-2-4 *Collision Detection Settings* for details on how to configure the collision detection settings.

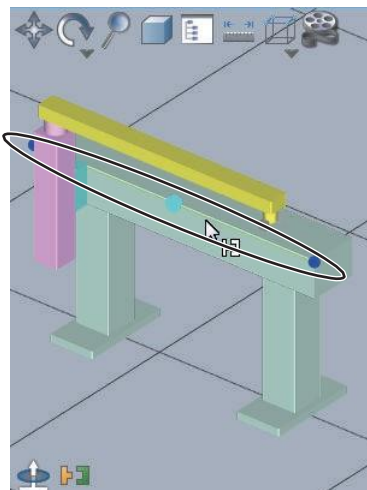
## Snap

Use this icon to set an offset for a mount point or a link point of 3D shape data. Select a mount point or a link point, and then click the icon to open the following window. To close the window, click **X**.

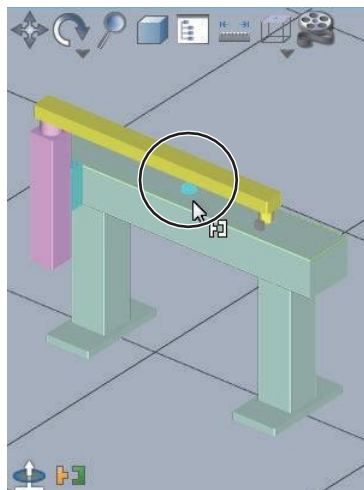


The functions of icons that you can select in the window are as follows.

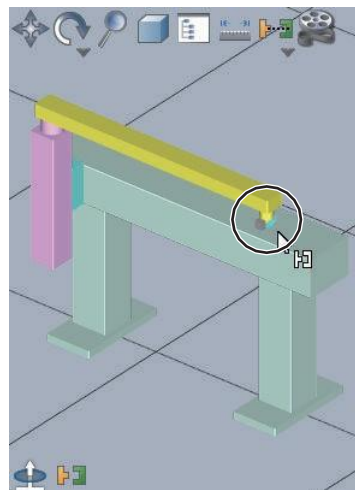
Icon	Name	Function
	Snap To Edge	Dragging a mount point or a link point near 3D shape data causes its edge nearest to the mouse cursor to be highlighted. Either both ends or the center of the highlighted edge is emphasized, and the mount point or the link point can be snapped to the position.
	Snap To Face	Dragging a mount point or a link point near 3D shape data causes its face nearest to the mouse cursor to be highlighted. The center of gravity of the highlighted face is emphasized, and the mount point or the link point can be snapped to the position.
	Snap To Link	Dragging a mount point to near 3D shape data causes its face nearest to the mouse cursor to be highlighted. Any link point that is present on the highlighted face is displayed, and the mount point can be snapped to the position.



Snap To Edge



Snap To Face



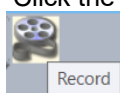
Snap To Link

## Record

Use this icon to capture a simulation executed in the 3D Visualizer on video.

### ● Recording Video

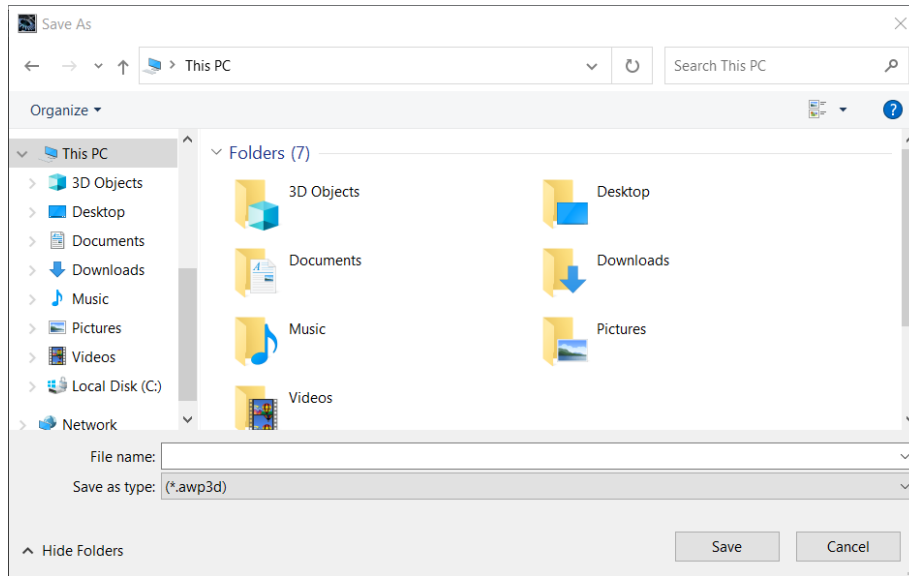
- 1 Click the **Record** icon during the execution of a 3D simulation.



The icon changes and starts flashing. This indicates that video recording is in progress.



- 2 To stop the recording, click the **Record** icon.  
The **Save As** dialog box is displayed.

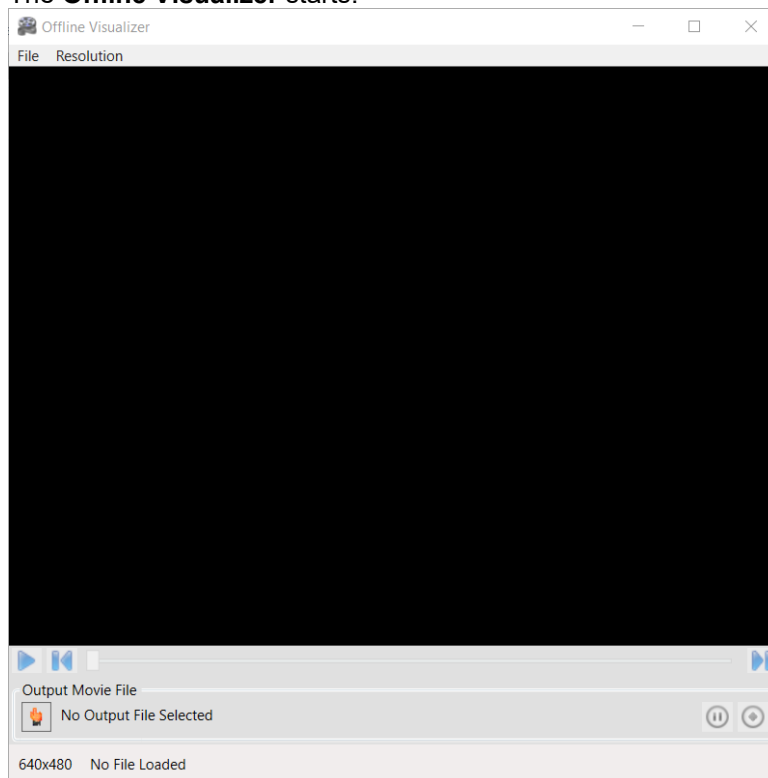


- 3 Enter the file name, and then click the **Save** button.  
The video is saved to a file.

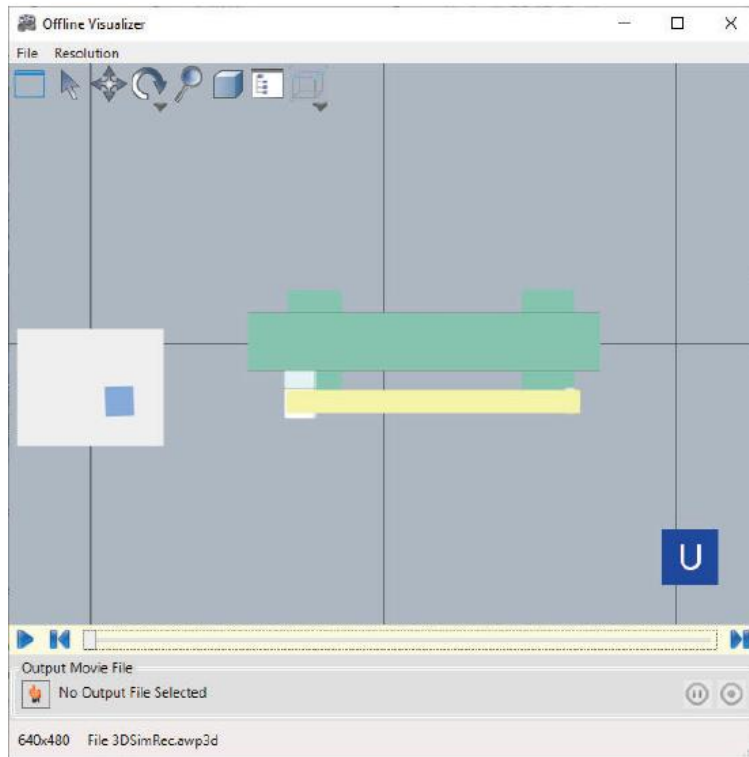
## ● **Playing Back Video**

Use the Offline Visualizer to play back video.

- 1 Select **All Programs – OMRON – Sysmac Studio – Tools – Offline Visualizer** from the Windows Start menu.  
The **Offline Visualizer** starts.



- 2 Select **Open** from the **File** menu.  
The **Open File** dialog box is displayed.
- 3 Select the record file (with a .awp3d extension) to play back and then click the **Open** button.  
The selected record file is opened.

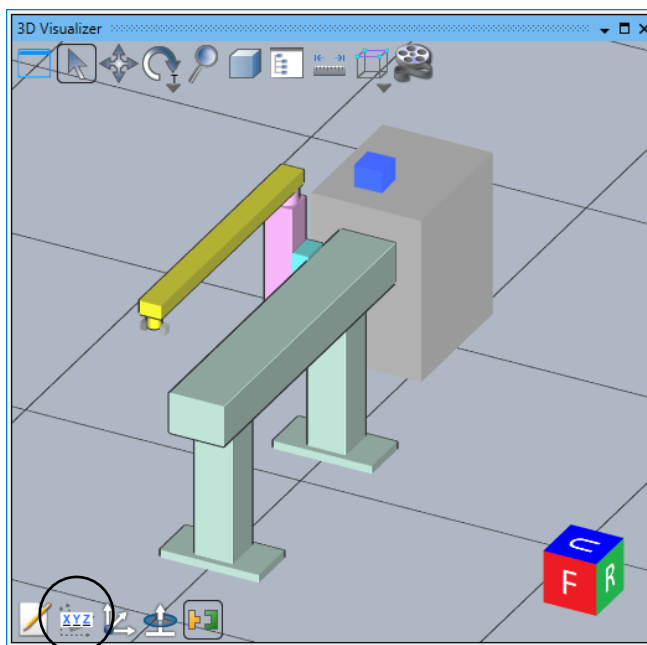


- 4 Click the **Play** button.  
The video is played back.

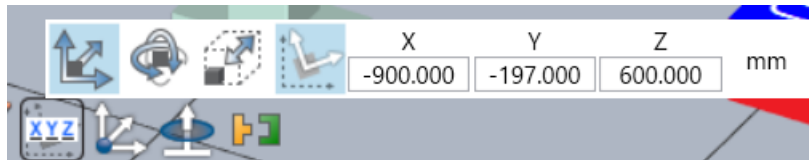
## Direct Position Edit

The 3D Visualizer allows you to edit 3D shape data by directly entering values while you are checking the positional relationship between 3D shape data.

- 1 In the 3D Visualizer, select the 3D shape data to edit with the mouse cursor.  
The **Direct Position Edit** icon is displayed in the 3D Visualizer.



- 2 Click the **Direct Position Edit** icon.  
The value input fields for the 3D shape data are displayed in the 3D Visualizer.



**3** When the **Direct Position Edit** icon is selected, you can enter numeric values for the 3D shape data.


Icon	Name	Function
	Edit Workspace Position	Enter the X, Y, and Z values to specify the position of the 3D shape data. <div style="text-align: right;"> <input type="text" value="X"/> <input type="text" value="Y"/> <input type="text" value="Z"/> mm  <input type="text" value="-900.000"/> <input type="text" value="-197.000"/> <input type="text" value="600.000"/> mm </div>
	Edit Workspace Orientation	Enter the Yaw, Pitch, and Roll values to specify the orientation of the 3D shape data. <div style="text-align: right;"> <input type="text" value="Yaw"/> <input type="text" value="Pitch"/> <input type="text" value="Roll"/> degree  <input type="text" value="0.000"/> <input type="text" value="0.000"/> <input type="text" value="1.944"/> degree </div>
	Edit Size	Enter the values to specify the size of the 3D shape data. The values to enter are changed depending on the 3D shape data. <b>Box</b> <div style="text-align: right;"> <input type="text" value="DX"/> <input type="text" value="DY"/> <input type="text" value="DZ"/> mm  <input type="text" value="100.000"/> <input type="text" value="100.000"/> <input type="text" value="80.000"/> mm </div> <b>Cylinder</b> <div style="text-align: right;"> <input type="text" value="Radius"/> <input type="text" value="Height"/> mm  <input type="text" value="100.000"/> <input type="text" value="100.000"/> mm </div> <b>Virtual Part Detection Sensor</b> <div style="text-align: right;"> <input type="text" value="Length"/> mm  <input type="text" value="250.000"/> mm </div> <b>Belt</b> <div style="text-align: right;"> <input type="text" value="Width"/> <input type="text" value="Length"/> mm  <input type="text" value="250.000"/> <input type="text" value="3000.000"/> mm </div>
	Object Coordinates	Select the coordinate system to which the <b>Edit Workspace Position</b> or <b>Edit Workspace Orientation</b> icon is applied. The coordinate system toggles every time you click the icon. The initial value is Object Coordinates.
	World Coordinates	

## Jog Mode

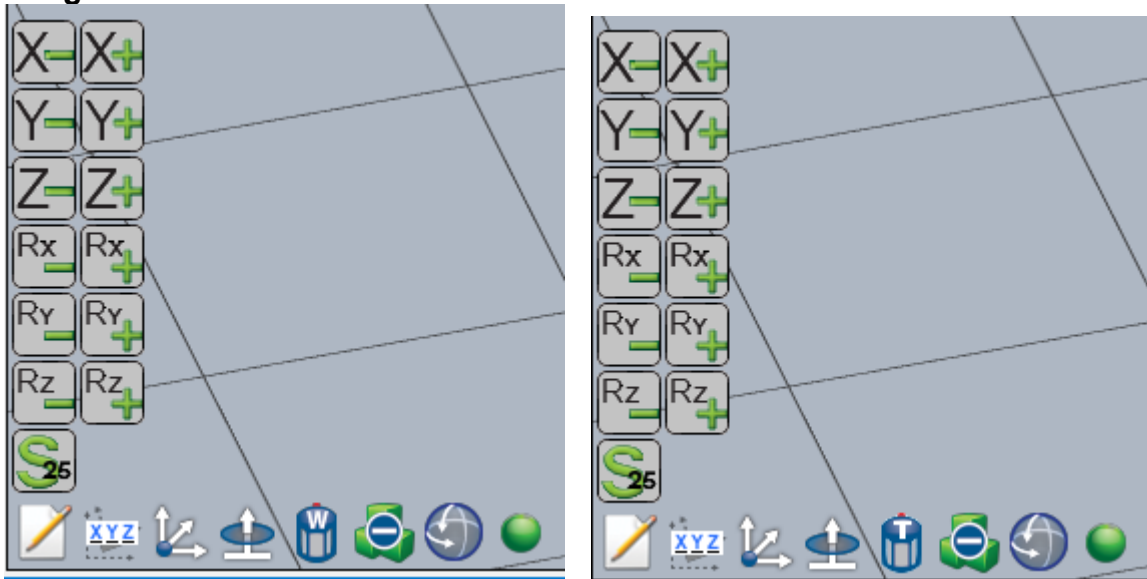
Use this icon to control the jogging of robots in the 3D Visualizer.

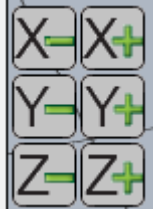
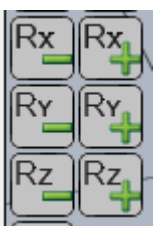

Clicking this icon displays Jog icons at the left of the 3D Visualizer. Use the Jog icons to manually control the position of the selected robot. The Jog Mode cycles through World, Tool, and Joint every time you click this icon.

Icon	Mode
	World
	Tool

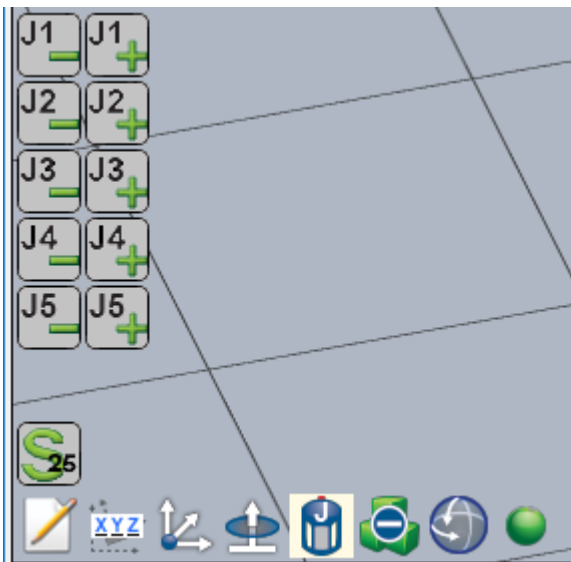
Icon	Mode
	Joint



● **Jog Mode: World or Tool**



Icon	Name	Description
	Jog icons for axial travel	Click one of these buttons repeatedly to move the robot along the X, Y, or Z axis in the positive or negative direction.
	Jog icons for axial rotation	Click one of these buttons repeatedly to rotate the robot around the X, Y, or Z axis in the positive or negative direction.
	Jog Speed	The Jog Speed cycles through 100%, 50%, 25%, and 5% every time you click the icon.

● **Jog Mode: Joint**






Icons	Name	Description
	Jog icons for Joint axial rotation	Click one of these buttons repeatedly to rotate each joint of the robot in the positive or negative direction.
	Jog Speed	The Jog Speed cycles through 100%, 50%, 25%, and 5% every time you click the icon.

### 3D View Switching Tool

Use this icon to switch the display direction of 3D shape data in the 3D Visualizer.

The 3D View Switching Tool is made up of three elements, i.e., Face, Corner, and Edge. Place the mouse cursor over an element and, when it turns black, click it. Then, the view is switched so that the portion that you clicked is the front face. Accordingly, the display direction of 3D shape data in the 3D Visualizer is switched.

Configurati on element	Name	Description
	Face	Represents a face. A face is indicated with one of the following symbols. F (Front): The front face when 3D shape data faces the zx plane R (Right): The right side face to the F face L (Left): The left side face to the F face B (Back): The face parallel to the F face U (Up): The upper orthogonal face to the F face D (Down): The lower orthogonal face to the F face
	Corner	Represents a corner.

Configuration element	Name	Description
	Edge	Represents an edge.



#### Additional Information

To reset the scale and display position of 3D visualization to the initial status, place the mouse cursor in the 3D Visualizer, and then press the Ctrl + 8 keys.

### 7-2-4 Collision Detection Settings

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 is typically used while testing an application with the 3D Visualizer in Emulation Mode and will not prevent physical hardware collisions.



#### Precautions for Correct Use

Be sure not to be confused with obstacles configured in Robot Settings. Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual* (Cat. No. W595) for the information on obstacles.

### Configuring Collisions in the 3D Visualizer

Use the Scene Graph icon in the 3D Visualizer to configure collisions. **Scene Graph** dialog box is opened. Refer to *Scene Graph* for details of items displayed.

Click the Collision Filter tab to register an object you want to detect its collision to the different Collision Group. Refer to *Sysmac Studio 3D Simulation Function Operation Manual* (W618) for details.



# 7-3 Event logs

The event logs display event logs that have occurred since the Robot Integrated System was started. The 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.

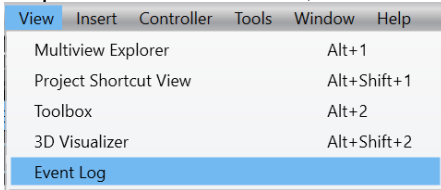


### Precautions for Correct Use

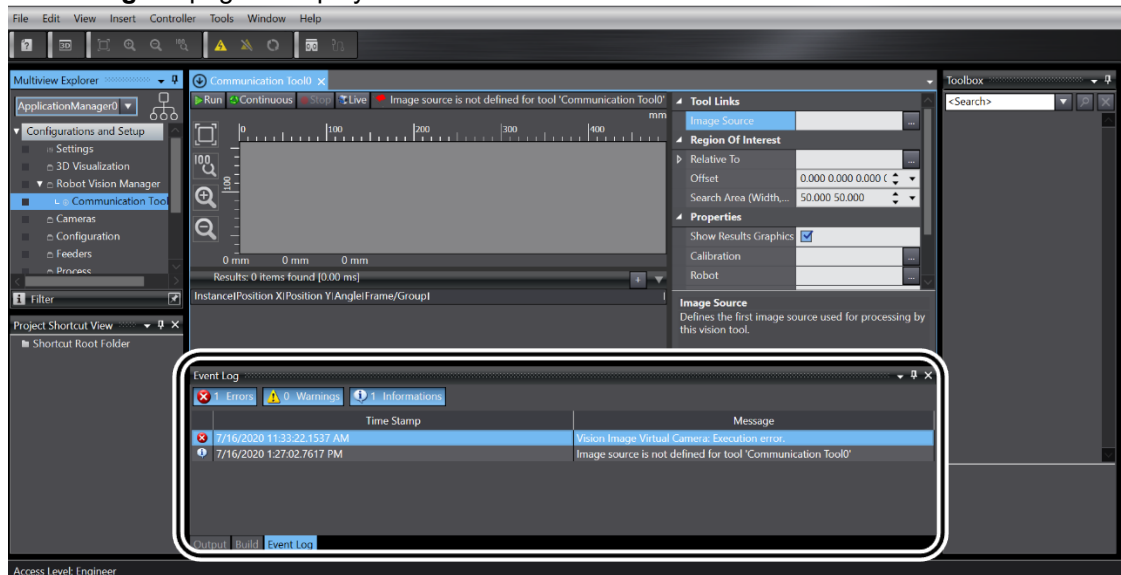
Past information that is displayed in the logs may not reflect the current status of the controller or programs.

## Starting Event logs

- 1 Select the **RobotControlSettings** or **Application Manager** from the device list in the Multiview Explorer. With selected it, select **Event Log** from the **View** menu.



Event Log tab page is displayed.



## Displayed Items of Event Logs

Event Log tab page is displayed the following information.

Icon	Name	Function
0 Errors	Error	Click the Error button to view all error events.
0 Warnings	Warning	Click the Warning button to view all warning events.
0 Informations	Information	Click the Information button to view all information events.

## 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.

## 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.

## 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 events as an alternative.

# 7-4 V+ Jog Control

This section describes how to operate the robot in the JOG mode through Sysmac Studio. V+ Jog Control is a function used to determine the selected robot(s) position and monitor it. Generally, it is used to teach robots their positions. You can see the robot's movements on the 3D Visualizer when using V+ Jog Control in the EMULATION mode.

 <b>WARNING</b>	
Ensure the enough safety before making any changes that may affect the operation of the robot.	
Make sure that there are no hazards caused by robot's movements before operating the robot using the V+ Jog Control function.	
Take a particular attention to the robot speed setting when you operate the robot using the V+ Jog Control function. Get ready to bring the robot to an emergency stop at an emergency. Make sure that there are no hazards caused by robot's movements before operating the robot.	
Confirm that you are operating the right robot before conducting a jog operation using V+ Jog Control function.	



### Additional Information

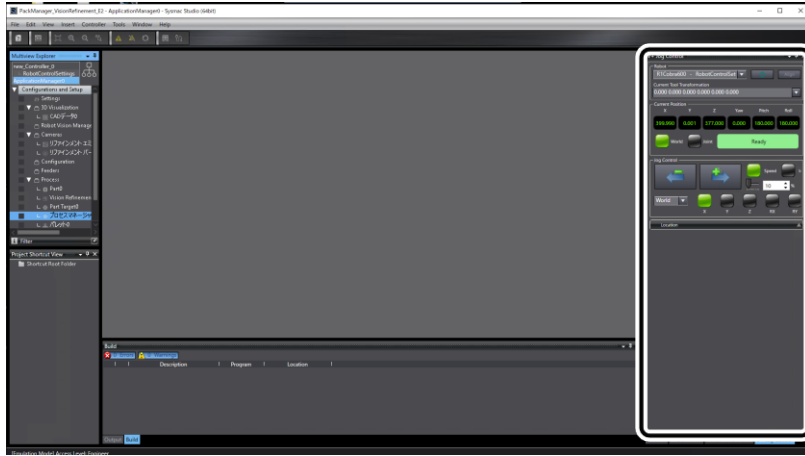
V+ Jog Control is effective on both the robot in the EMULATION mode and the actual robot. Most of jog commands and settings are disabled when the robot is controlled by Program Control. Refer to *Current Position Section* for details.

## 7-4-1 Starting V+ Jog Control

- 1 Select **RobotControlSettings** from the device list in the Multiview Explorer or **Application Manager**, and click **View - V+ Jog Control** in the main menu.

View	Insert	Controller	Tools	Window	Help
Multiview Explorer					Alt+1
Project Shortcut View					Alt+Shift+1
Toolbox					Alt+2
3D Visualizer					Alt+Shift+2
Output Tab Page					Alt+3
Watch Tab Page					Alt+4
Watch Tab Page(Table)					Alt+Shift+4
Cross Reference Tab Page					Alt+5
Build Tab Page					Alt+6
Event Log					
Search and Replace Results Tab Page					Alt+7
Simulation Pane					Alt+8
Differential Monitor					Alt+9
Smart Project Search					Ctrl+Shift+F
Recently Closed Windows					Ctrl+Shift+H
Clear Recently Closed Windows History					
V+ Jog Control					Alt+Shift+J

**V+ Jog Control** Pane is displayed.



## 7-4-2 V+ Jog Control Setting Items

The following section describes items to be set for V+ Jog Control.



## Robot Section

The Robot section provides the following functions.

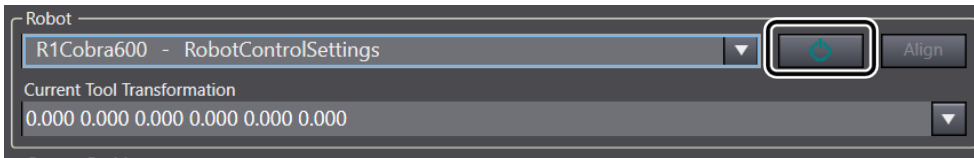
- **Robot**

Select a robot to control with the V+ Jog Control. You can access to all robots in the project from the drop-down.



- **Robot Power**

The **Power** button toggles the robot high power ON and OFF and calibrates the selected robot. If gone online with the Robot Integrated CPU Unit, pressing the **Power** button has the Robot High Power button connected to a robot blinking for the period that set in the Safety Timeout. The power will be provided to the robot ready for operation if the Robot High Power button is pressed during this period. Robot power must be ON to allow jog control.

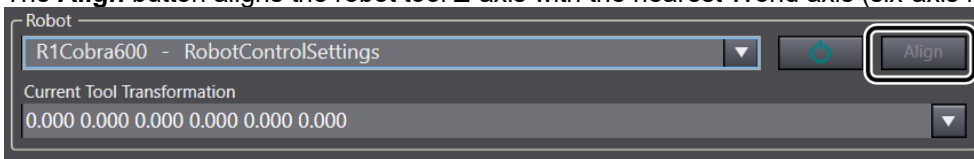


### Additional Information

- Refer to *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual* (Cat. No. W595) for how to configure Safety Timeout.
- Turning the robot high power ON for the first time after system power up executes the CALIBRATE() function to load joint calibration offsets into memory. It does not mean that a full robot hardware calibration will be performed.

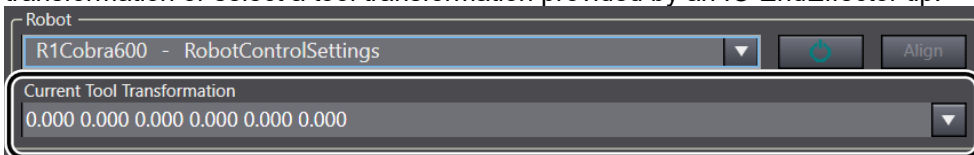
### Align

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



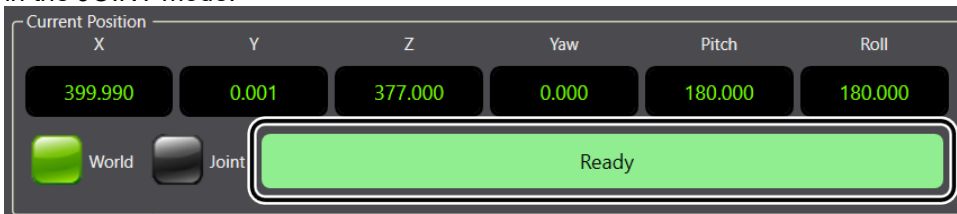
### Current Tool Transformation

The current tool transformation applied to the robot is displayed. The dropdown can be used to clear the tool transformation or select a tool transformation provided by an IO EndEffector tip.



### Current Position Section

In this section, the status and current position of the robot in world or joint coordinates are displayed. Click the **World** button to display coordinates in the WORLD mode. Click the **Joint** button to display coordinates in the JOINT mode.



### Additional Information

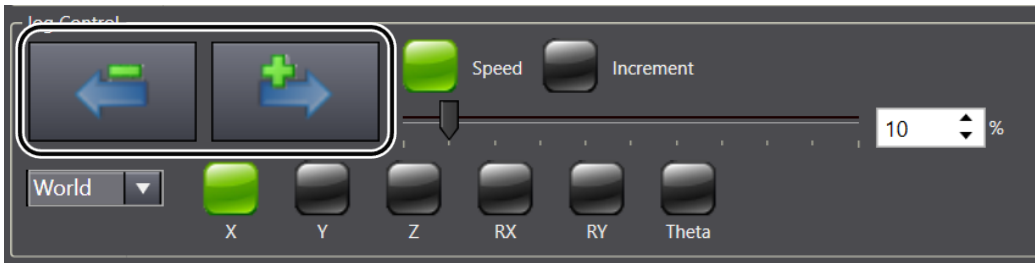
Jogging is only possible when **Ready** is displayed in the status area. If a robot is under program control, "*Robot under program control*" will be displayed in this area and jogging is not possible. The task controlling the robot must be stopped before jogging functions are enabled.

## Jog Control Section

You can manually position a robot in the Jog Control section.

### Move Axis Buttons

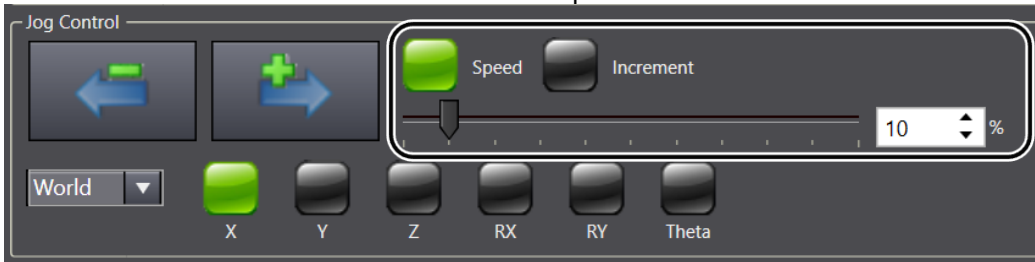
After all jog control settings are made, use the move axis buttons to move the selected axis in the positive or negative direction.



● **Speed/Increment Selection Buttons**

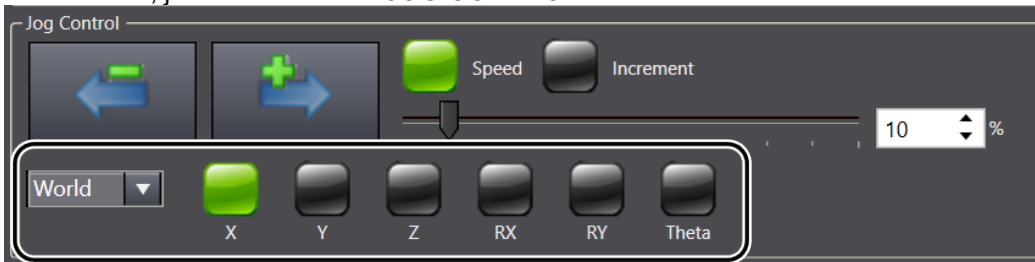
Jogging is possible at a preset speed or in incremental distances, to allow greater positioning precision. When the **Speed** button is active, use the slider or input a value between 0 and 100% to set the jog speed when a move axis button is pressed.

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.



● **World, Joint, Tool Selection**

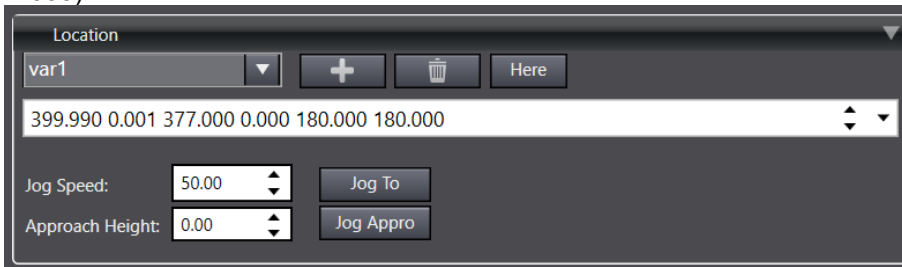
Select world, joint or tool for the JOG CONTROL mode.



Coordinate Type	Description
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.

**Location Section**

The Location section is used to view, teach, remove, and jog to robot locations Refer to *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual* (Cat. No. W595) for details.



● **Jog to a Robot Location**

Implement the following procedure to jog to a robot location.

- 1 Select a location with the drop-down menu.
- 2 Select values for the **Jog Speed** and **Approach Height** fields.

- 3 Click and hold the **Jog To** button to make the robot jog to the specified location.
- 4 Click and hold the **Jog Appro** button to make the robot jog to the specified location at the approach height specified.

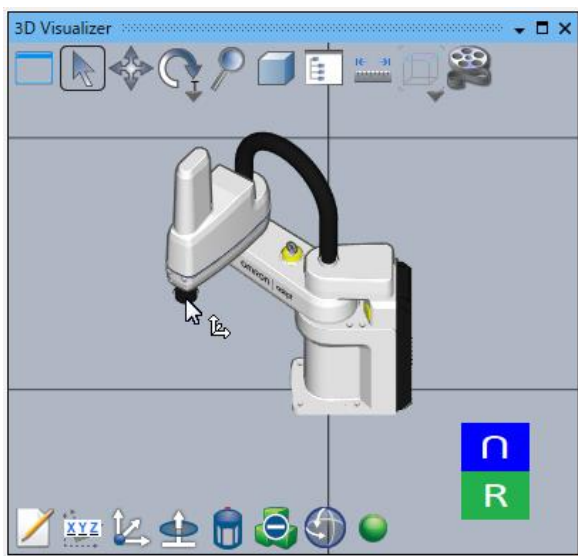


### Precautions for Correct Use

- A robot location must exist and be selected to use this function.
- 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. Clicking the **Here** button will put the robot's current axis positions into the display field for use in the following teach procedure.



### Additional Information

In the EMULATION mode, you can change the robot tool tip position in the 3D Visualizer with the mouse cursor. Hover over the tool tip until the mouse pointer changes, and then, left-click and drag to the new position.

Refer to *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual* (Cat. No. W595) for other robot position teach functions.

- 1 Click the **+** button. The **Add a new variable** dialog box appears.

- 2 Select a variable type (location or precision point), provide a new name, and verify the value. If you change a selection from the robot drop-down, click the **Record** button to update the value for that robot accordingly.

- 3** Select a display mode, category, and provide a description, if necessary.
- 4** Select values for Array, Starting and Ending Index, if necessary.
- 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 drop-down menu and then click the **Delete** button. A confirmation dialog box is displayed. Click **Yes** to remove the robot location variable.



# 7-5 Task Status Control

The Task Status Control provides a multi-device monitoring interface for all robot related activity in the project. This allows you to quickly view and access all controller connection and power statuses, monitor speeds, and execution status of Vision Sequences, C# programs, Process Managers, and more.

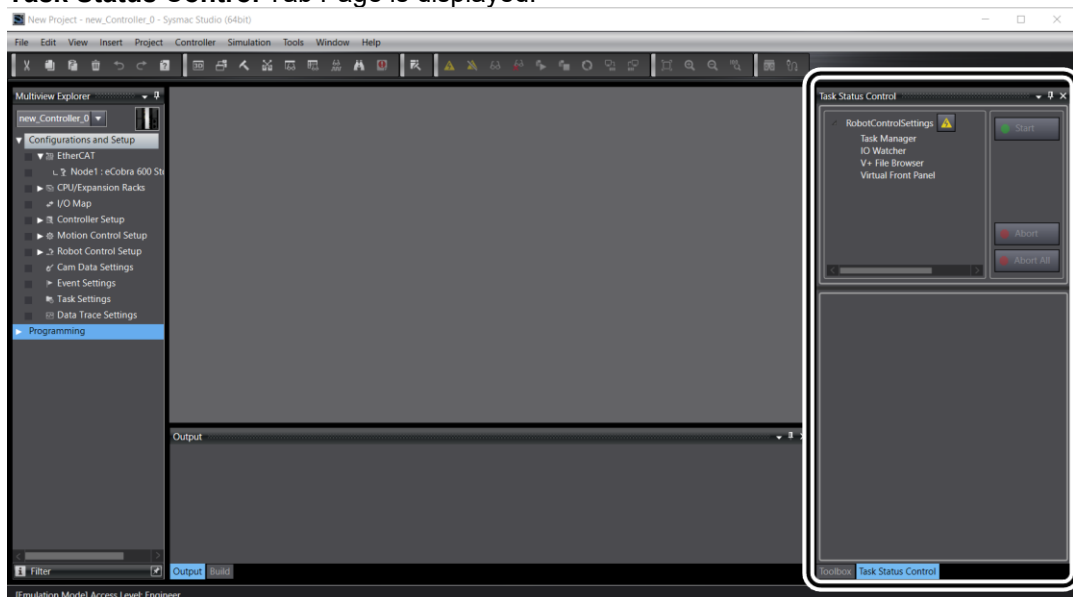
All controllers in a project will be displayed in the Task Status Control interface. If an Application Manager Device is present in the project, items such as Robot Vision Manager sequences, C# programs, Process Managers, and Recipe Managers may also be displayed in the Task Status Control interface.

## 7-5-1 Starting Task Status Control

- 1 Select **RobotControlSettings** from the device list in the Multiview Explorer or **Application Manager**, and click **View - Task Status Control** in the main menu.

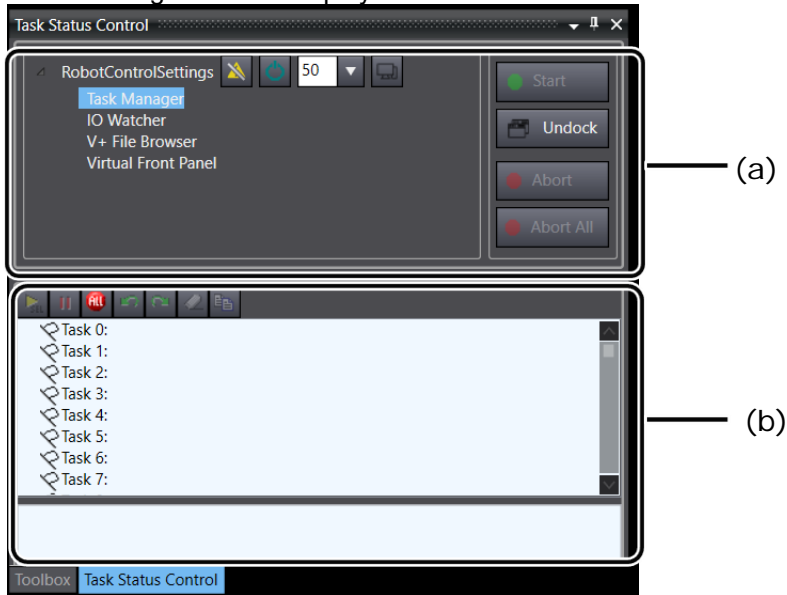
View	Insert	Controller	Tools	Window	Help
Multiview Explorer					Alt+1
Project Shortcut View					Alt+Shift+1
Toolbox					Alt+2
3D Visualizer					Alt+Shift+2
Output Tab Page					Alt+3
Watch Tab Page					Alt+4
Watch Tab Page(Table)					Alt+Shift+4
Cross Reference Tab Page					Alt+5
Build Tab Page					Alt+6
Event Log					
Search and Replace Results Tab Page					Alt+7
Simulation Pane					Alt+8
Differential Monitor					Alt+9
Smart Project Search					Ctrl+Shift+F
Recently Closed Windows					Ctrl+Shift+H
Clear Recently Closed Windows History					
V+ Jog Control					Alt+Shift+J
<b>Task Status Control</b>					<b>Alt+Shift+T</b>
Vision Window					Alt+Shift+V

**Task Status Control** Tab Page is displayed.



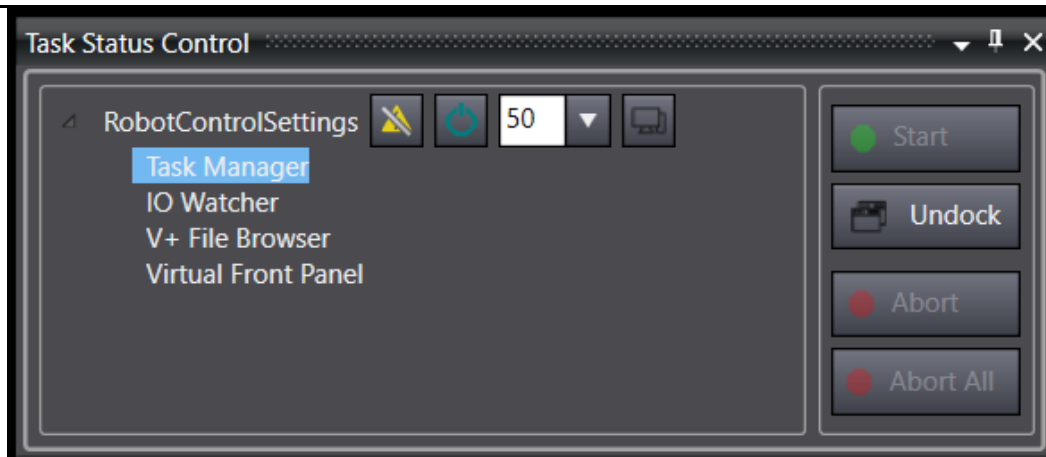
## 7-5-2 Displayed Items in Task Status Control



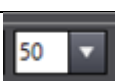

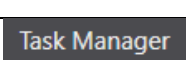
The following items are displayed in Task Status Control.

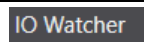
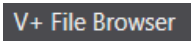


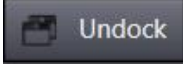
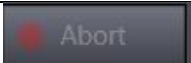



	Item	Descriptions
(a)	Item list area	Items controlled with Task Status Control Tab Page are displayed.
(b)	Description display area	Details on the Item selected in the item list area is displayed. Press the <b>Undock</b> button to show contents in this area in a different window.

### Item List



Item	Description	Reference
	Online button Control the connection status to the Robot Integrated CPU Unit	---
	Robot High Power button Toggle the robot power state This button is only available while online with the controller.	---
	Monitor speed setting drop-down Configure robot's motion speed (min. 0, max. 100)	<i>Monitor Speed Setting</i>
	Open the Monitor Window button Monitor Window Tab Page is displayed	<i>Open the Monitor Window</i>
	Task Manager Task Manager is displayed at the bottom	<i>Task Manager</i>

Item		Description	Reference
	IO Watcher	IO Watcher is displayed at the bottom	<i>IO Watcher</i>
	V+ File Browser	V+ File Browser is displayed at the bottom	<i>V+ File Browser</i>
	Virtual Front Panel	Virtual Front Panel is displayed at the bottom	<i>Virtual Front Panel</i>
	Start	Execute the selected Application Manager function.	---
	Undock	Show the currently displayed view in a different window.	---
	Abort	Abort the selected running Application Manager function.	---
	Abort All	Abort all Application Manager functions.	---

### 7-5-3 Robot Integrated CPU Unit Settings Items

You can establish a connection with the Robot Integrated CPU Unit or toggle ON and OFF of robot's high power.

#### Monitor Speed Setting

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

#### Open the Monitor Window

The **Monitor Window** button opens the Monitor Window in the Edit Pane. Refer to *10-2 System Monitor* for details.

### 7-5-4 Task Manager

The Task Manager displays and controls activities of user tasks 0 to 27. Robot Control Function Module in Robot Integrated CPU unit uses two tasks and 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.



#### Additional Information

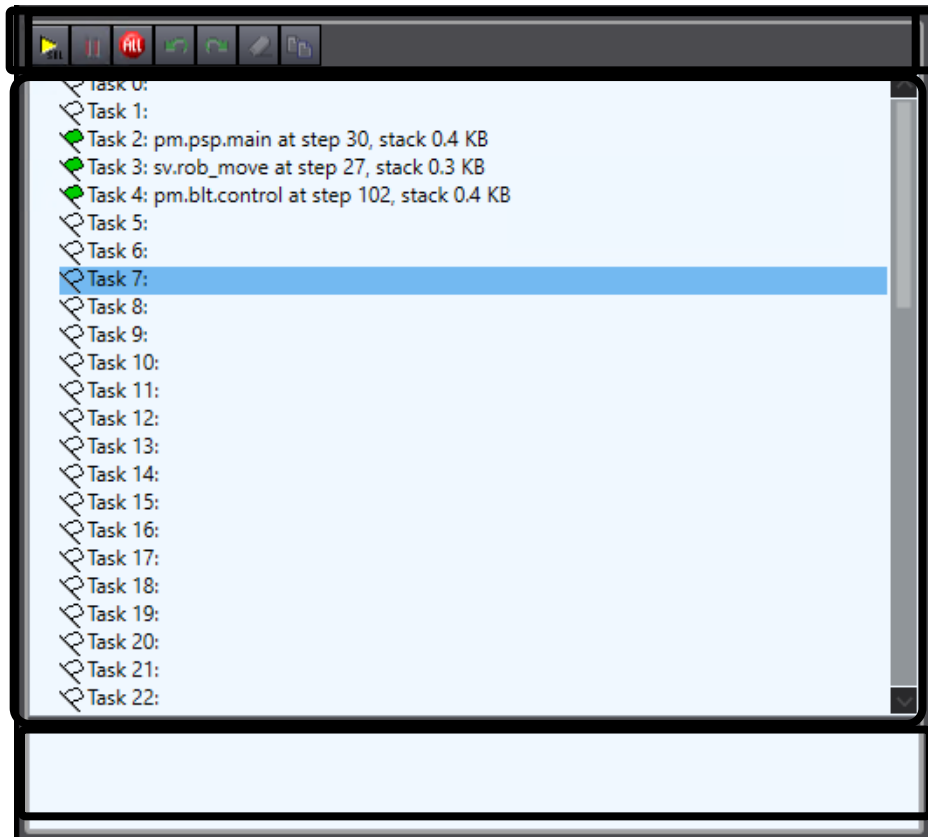
If a program is paused the task can be expanded to view the current program stack.

## WARNING

When building a robot system that includes this CPU Unit or an Omron robot, be sure to ensure compliance with the laws and regulations on the safety of industrial robots in the country or region where the robot is operating in design and operation of the system. Refer to Robot SAFETY GUIDE (I590) for details.



The following describes Task Manager controls.




## Task Manager Toolbar Items

The following functional icons are provided in the toolbar. You can select a function from the right-click menu of a task.

Icon	Item	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 <b>OK</b> will execute the program on the selected task.
	Pause Task	The selected task execution is paused at the next instruction.
	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 allows you to copy and paste the stack contents for troubleshooting. Robot IDs are also recorded in this operation.

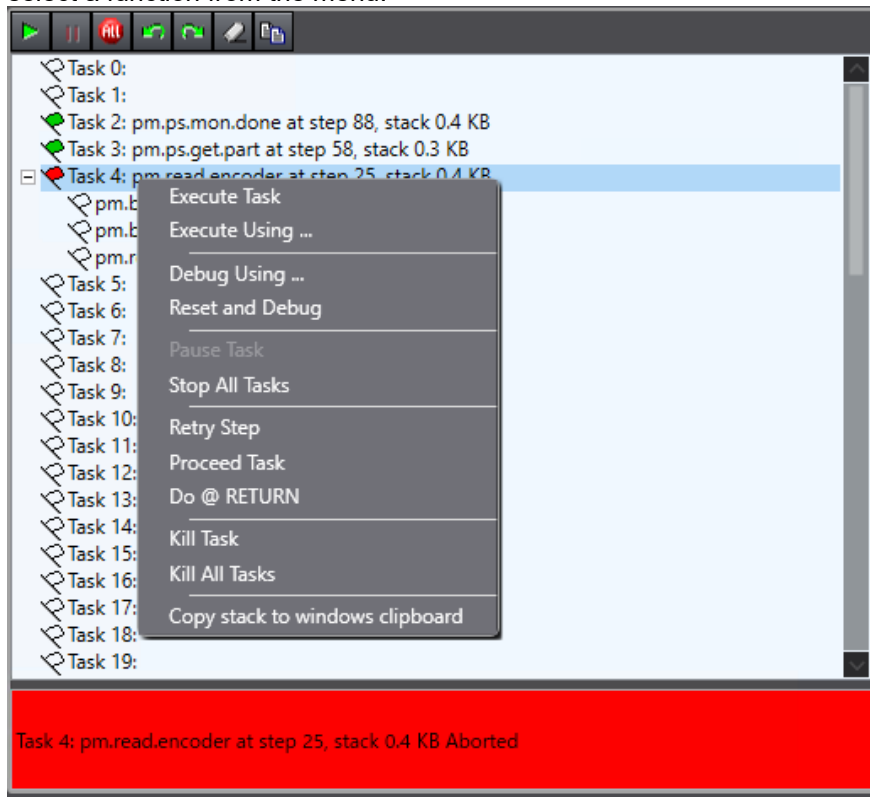
The flag icon next to each task in the list area represents the task state.

Task Flag Icon	Description
	Task is idle or primed.
	Task is being executed.
	Task is paused or at a breakpoint. A program's task flag icon will turn yellow if you drag it onto a task to prime it.
	Task has an execution error or program execution was manually aborted.

Task Flag Icon	Description
	Task execution has completed.



## Other Functions

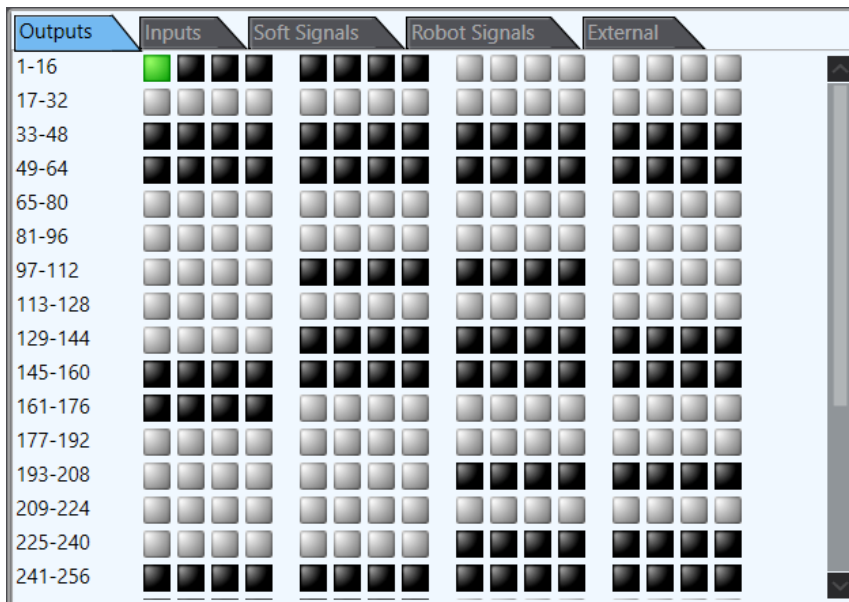
This section provides the information about other functions regarding to task execution. Right-click a task to select a function from the menu.



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 <b>Return</b> key is pressed.
Kill All Tasks	Clears all tasks that do not have running programs.

### 7-5-5 IO Watcher

Select IO 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 the signal button(/ ).

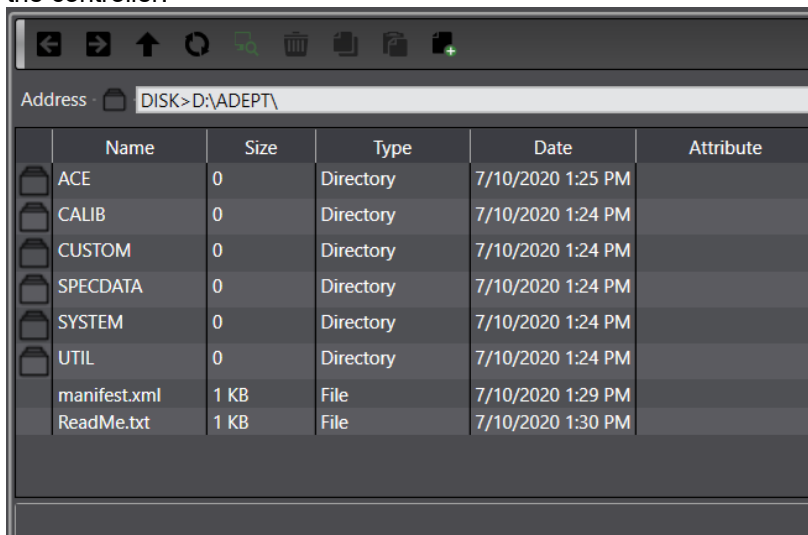


### Additional Information

When the EMULATION mode is enabled, digital input signals can be manipulated.

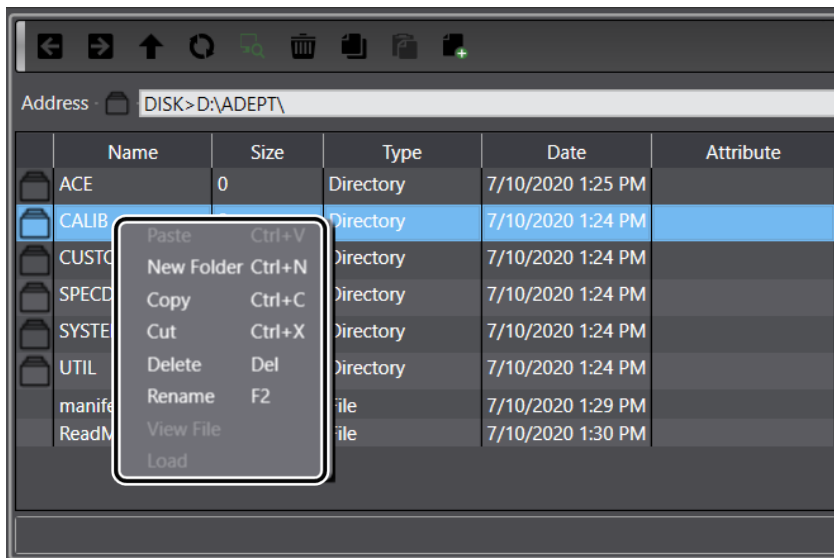
## 7-5-6 V+ File Browser

The V+ File Browser allows you to browse files and folders on the controller. Only available while online with the controller.



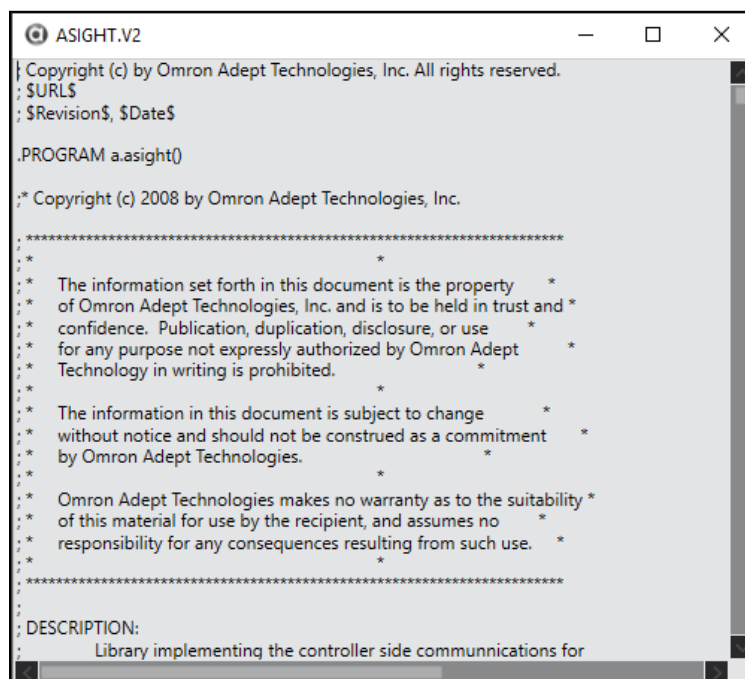
The V+ File Browser works with the Windows clipboard to enable easy transferring of files to and from controllers.

Icons in the V+ File Browser toolbar enable you 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

Selecting **View File** will open the file in a quick-view window without the need for transferring the file to the PC. Available for program, variable, and text files.



## Load

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

### 7-5-7 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 the 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.

## WARNING

When building a robot system that includes this CPU Unit or an Omron robot, be sure to ensure compliance with the laws and regulations on the safety of industrial robots in the country or region where the robot is operating in design and operation of the system. Refer to Robot SAFETY GUIDE (I590) for details.





Icon	Item
	<b>Mode Selection</b> button
	MANUAL mode
	AUTOMATIC mode
	<b>Robot Power</b> button
	<b>E-Stop</b> button

## Mode Selection

Switches between the MANUAL and AUTOMATIC mode. In the AUTOMATIC mode, executing programs control the robot, and the robot can run at full speed. In the MANUAL mode, the system limits robot speed and torque so that an operator can safely work in the cell. The MANUAL mode initiates software restrictions on robot speed, commanding no more than 250 mm/sec. In the MANUAL mode, no high speed mode is available. Refer to the user's manual of each robot for details.

## Robot Power

The **Robot Power** button enables power to the robot motors. This button has an indicator to show the following robot power states.

- OFF: Power is not supplied to the robot
- ON: Power is supplied to the robot
- Blinking: Power will be supplied if the **Robot Power** button is pressed within the specified Safety Timeout period. Refer to *Sysmac Studio Robot Integrated System Building Function with Robot*



Integrated CPU Unit Operation Manual (Cat. No. W595) for details. Robot power must be active to operate the robot.

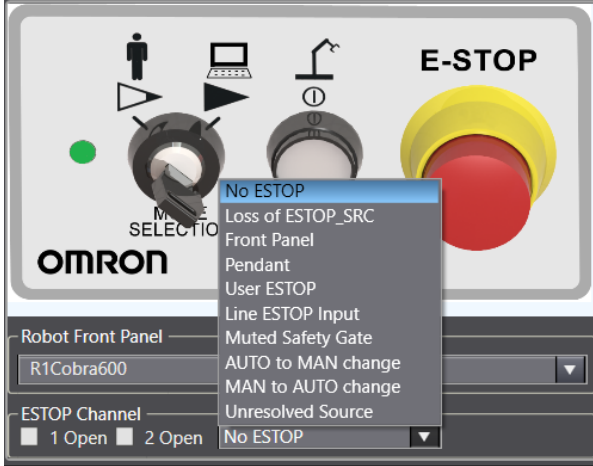


### Precautions for Correct Use

Enabling power to the robot motors is not possible while in an E-Stop state.

## E-Stop

Behaviors at an emergency stop 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 *eV+3 Keyword Reference Manual (I652)* for details.

## 7-5-8 Application Manager Control

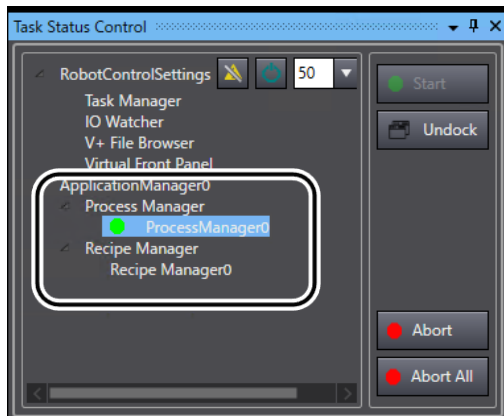
Use the Application Manager Control area to examine information about, and control activity associated with the Application Manager. Application Manager objects such as Robot Vision Manager sequences, C# programs, Process Managers, and Recipe Managers that require user interaction for execution or run time monitoring are displayed in this area.

### WARNING

When building a robot system that includes this CPU Unit or an Omron robot, be sure to ensure compliance with the laws and regulations on the safety of industrial robots in the country or region where the robot is operating in design and operation of the system. Refer to Robot SAFETY GUIDE (I590) for details.



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.



# 7-6 Vision Window

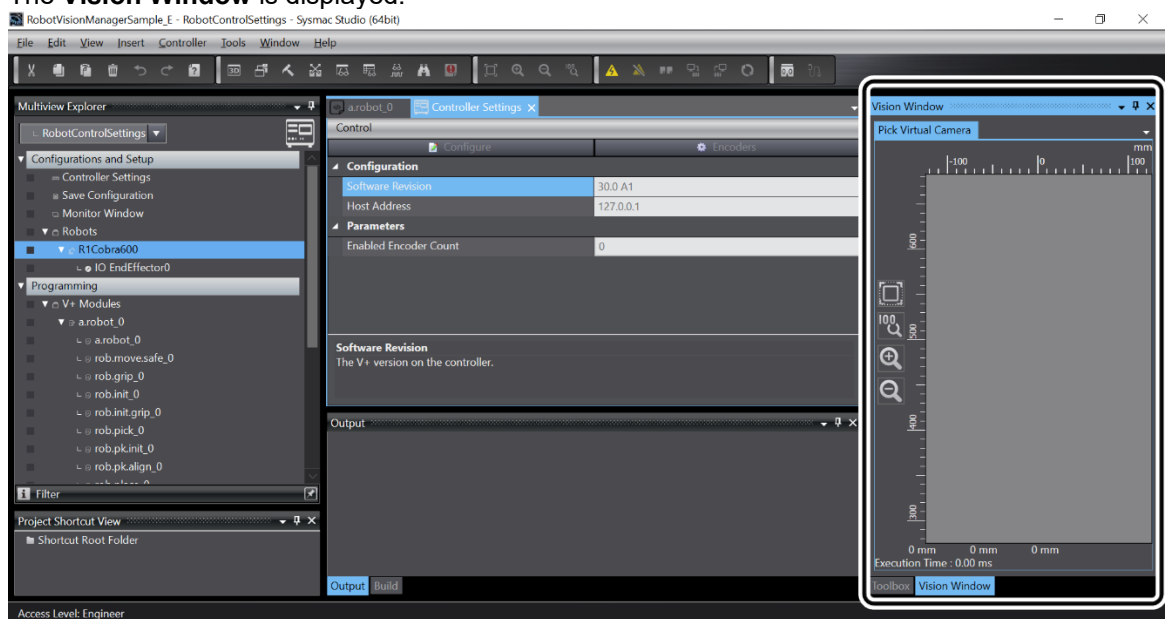
The Vision Window displays the input for each image source defined in the system. Using a tab for each source, you can view various cameras and tools configured in the project. Most vision tools can display their results in this window, allowing you to troubleshoot, test, or monitor the system performance during run time. In addition, like the Edit Pane in the main 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 5-10 *Vision Tools* for more information.

## 7-6-1 Vision Window Display

- 1 With **RobotControlSettings** or Application Manager selected from the device list in the Multiview Explorer, select **View - Vision Window** from the Main Menu.

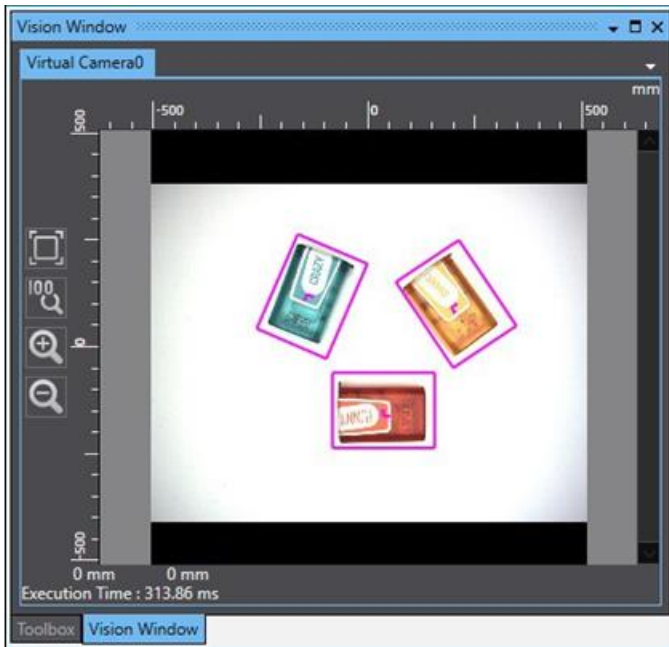
View	Insert	Controller	Tools	Window	Help
Multiview Explorer				Alt+1	
Project Shortcut View				Alt+Shift+1	
Toolbox				Alt+2	
3D Visualizer				Alt+Shift+2	
Event Log					
Smart Project Search				Ctrl+Shift+F	
Recently Closed Windows				Ctrl+Shift+H	
Clear Recently Closed Windows History					
V+ Jog Control				Alt+Shift+J	
Task Status Control				Alt+Shift+T	
Vision Window				Alt+Shift+V	

The **Vision Window** is displayed.






## 7-6-2 Vision Windows Functions

The functions of the Vision window are shown below.



## Zoom Level

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

	Item	Description
	Fit to Screen	Click the icon to change the zoom level so the entire image fits in the Vision Window
	Zoom to 100%	Click the icon to change the zoom level to the default size of the acquired image.
	Zoom in/out	Click the icons to manually adjust the zoom level.

## Calibration Scale

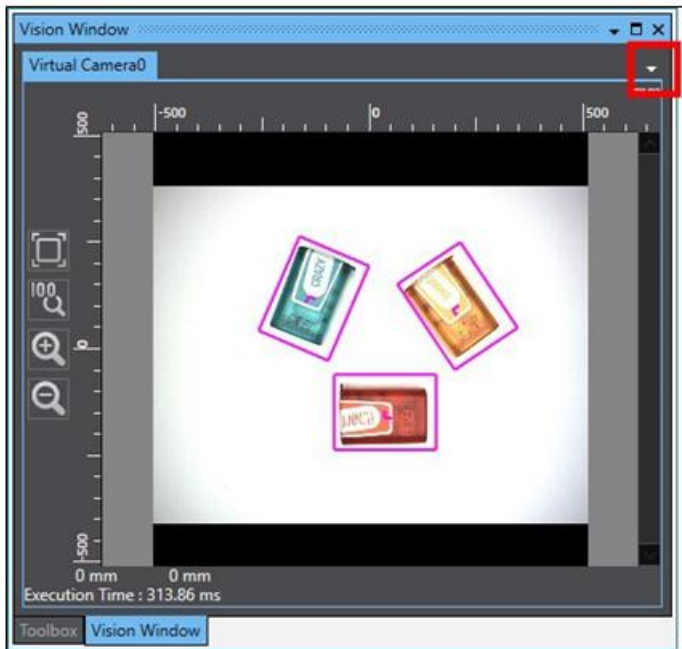
The left and top axis values represent the calibration scale setting in mm/pixel that is present in the Virtual Camera settings. Refer to *Virtual Camera Calibration* for more information.

## 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.

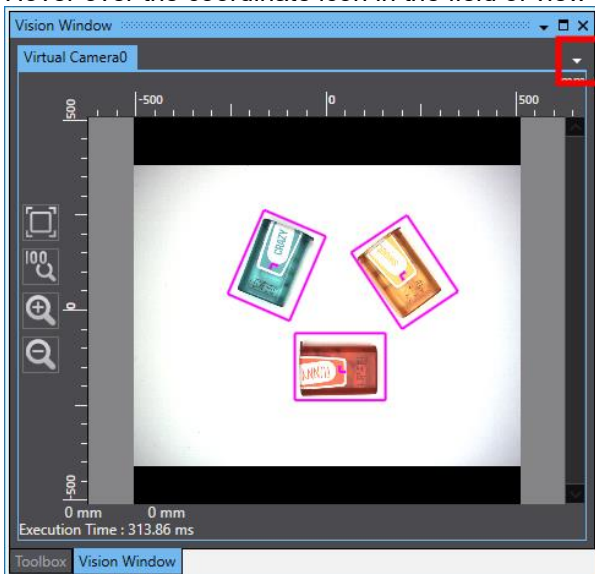


## Cursor Information

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.




# 7-7 Monitoring Variables

This section describes the monitor function for C# Variable objects. 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.

Variable/Expression	Value	Source	Task	Program	Type
Robot_pos1	123.45	/ApplicationManager0/			Numeric Variable
Robot_pos2	-5671.567	/ApplicationManager0/			Numeric Variable
Robot_pos3	1043.4	/ApplicationManager0/			Numeric Variable

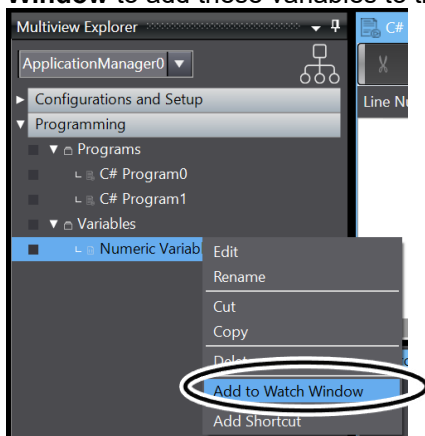
## ⚠ WARNING

Check operations of the created user programs, data, and setting values carefully before proceeding to normal operation.



## 7-7-1 Adding C# Variable Objects to the V+ Watch Window

Use one of the following methods to add variables to the V+ Watch window. Select and right-click the C# Variable objects to add in the Multiview Explorer. Select **Add to Watch Window** to add these variables to the V+ Watch window.







## 8 Application Sample Wizard

---

This section describes the procedure for creating the Robot Integrated System using the IPC Application Controller with the application sample wizard.

# 8-1 Overview of Application Sample Wizards

---

Application Sample Wizards will help you to create a template for a picking and placing robot integrated system using the Robot Integrated CPU Unit and IPC Application Controller. You can create a template easily just following the wizard: selecting peripherals for the system and configuring settings.

There are two types of Application Sample Wizard.

Type	Description
Pack Manager Sample	Create a picking and placing template for packaging machine with Pack Manager running on IPC Application Manager. You do not have to know V+ because the created sample consists of Pack Manager object settings only, not program code. Pack Manager on the IPC Application Controller and the Robot Integrated CPU Unit control a robot coordinately.
Robot Vision Manager Sample	Create a picking and placing template for packaging machine with Robot Vision Manager running on IPC Application Manager and V+ programs running on the Robot Integrated CPU Unit. The Robot Integrated CPU Unit controls a robot.



# 8-2 Application Sample Wizards

## 8-2-1 Application Samples

You can create a system template using various part detection methods and picking and placing methods with Application Sample Wizards for Pack Manager Sample or Robot Vision Manager Sample. Application sample consists of a combination of system elements listed below:

- Part detection method
- Part position adjustment in the gripper (after picking a part)
- Part placement position determination method

For example, Pack Manager Application Sample Wizard items are provided. Robot Vision Manager Application Sample Wizard items are the same, though details are different.

### Part Detection Methods

Part detection methods are as shown below.

Type	Description
At a fixed position	Always detects a part at a fixed position.
At a fixed pallet	Always detects a part on a pallet at a fixed position.
On a belt located with a camera	Detects a part on the conveyor belt with a fixed camera.
With an arm- or fixed- mounted camera	Detects a part with a camera mounted on the robot arm.
With a belt latch sensor	Detects a part on a conveyor belt with the latch sensor mounted on the conveyor belt.
At a pallet located by a belt latch sensor	Detects a pallet carrying a part, which on the conveyor belt, with the latch sensor mounted on the conveyor belt.

### Part Placement Position Determination Methods

Part placement position determination methods are as shown below.

Type	Description
At a fixed position	Always places a part at a fixed position.
At a fixed pallet	Always places a part on a pallet at a fixed position.
On a belt located with a camera	With a fixed camera, detects a position on the conveyor belt where a part is placed.
With an arm- or fixed- mounted camera	With a camera mounted on the robot arm, detects a position where a part is placed.
With a belt latch sensor	With the latch sensor mounted on the conveyor belt, detect a position on the conveyor belt where a part is placed.
At a pallet located by a belt latch sensor	With the latch sensor mounted on the conveyor belt, detect a position on the conveyor belt where a pallet carrying a part is placed.

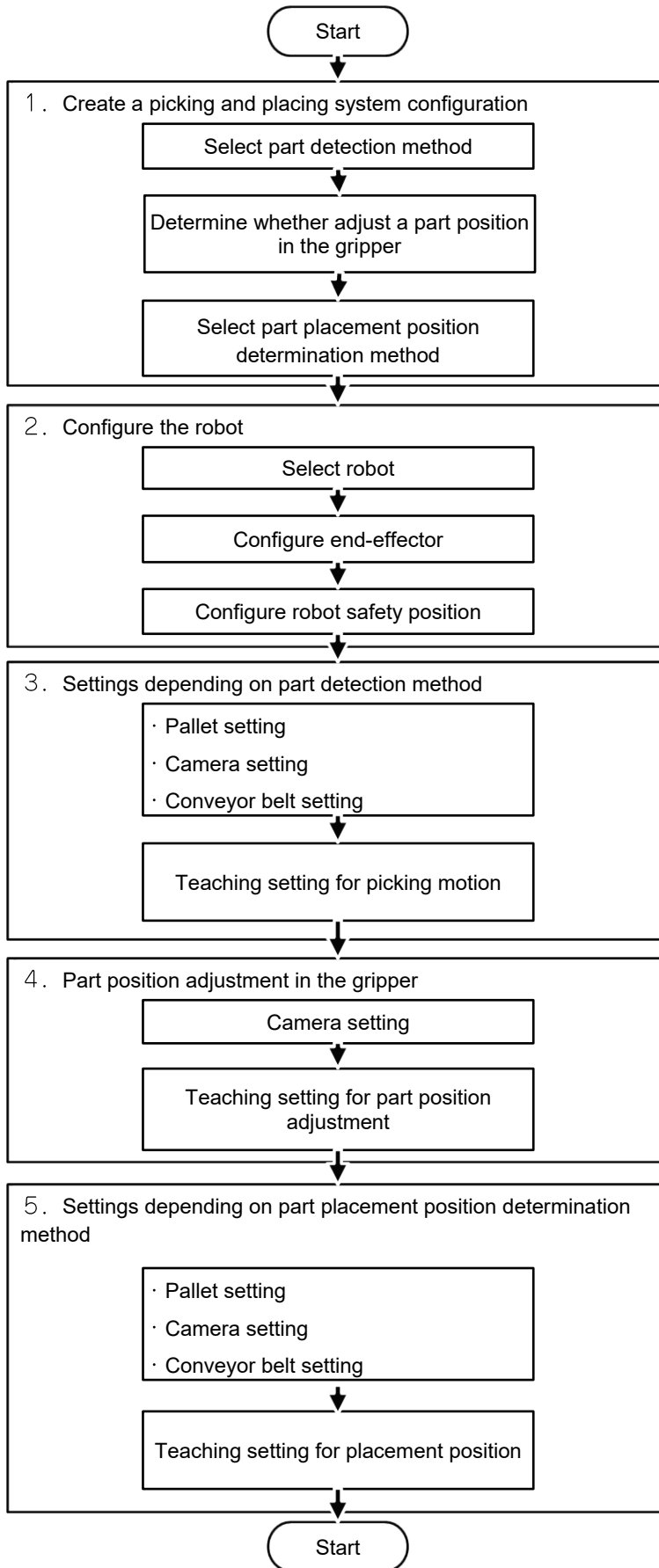
### Part Position Adjustment in the Gripper

Using this item, you can determine whether you adjust a picked part's position in the gripper.

Type	Description
Not refine the part position	Maintain a part as it was picked up.
Refine the part position using a camera	Shoot a picked-up part with a camera from underneath to adjust the parts position with respect to the tip of the robot arm.

## 8-2-2 Application Sample Wizard Flowchart

This section describes a procedure using Pack Manager Application Sample Wizard as an example. Robot Vision Manager Application Sample Wizard has the same procedure.



- 1** Create a picking and placing system configuration  
Select system's elements to create a picking and placing system configuration.

What to do	Description	Reference
Select part detection method	Select a method to detect a part when the robot picks up the part.	Part Detection Methods
Determine whether adjust a part position in the gripper	Determine whether adjust the position in the gripper of the picked part.	Part Position Adjustment in the Gropper
Select part placement position determination method	Select a method to determine a position where the robot place a part.	Part Placement PositionDetermination Methods

- 2** Configure the robot  
Select a robot you use, then set a safety position for the robot.

What to do	Description	Reference
Select robot	Select a robot you use for an application sample.	Select Robot
Configure end-effector	Configure properties of an end-effector's tip.	Configure End-effector
Configure robot safety position	Set a safety position for the robot.	Configure Robot Safety Position

- 3** Settings depending on part detection method  
Teach the robot to pick up a part.

What to do	Description	Reference
Pallet setting Camera setting Conveyor belt setting	Configure settings of Pallet, Camera, and Conveyor Belt according to the selected part detection method.	Teaching Setting for Picking Motion
Teaching setting for picking motion	Teach the robot a picking motion to clamp the part.	

- 4** Part position adjustment in the gripper  
If you have selected to adjust the position of the picked part in the gripper, teach the robot to adjust the part position.

What to do	Description	Reference
Camera setting	Configure camera setting	Teaching setting for part position adjustment
Teaching setting for part position adjustment	Teach the robot to adjust part position.	

- 5** Settings depending on part placement position determination method  
Teach the robot where to place a part.

What to do	Description	Reference
Pallet setting Camera setting Conveyor belt setting	Configure settings of Pallet, Camera, and Conveyor Belt according to the selected part placement determination method.	Teaching Setting for Placement Position
Teaching setting for placement position	Teach the robot a placing position where it put the part on.	

### 8-2-3 Startup Application Sample Wizard

Using an example of part picking with a belt camera system including a scalar robot (Cobra 600), this section describes how to create a Pack Manager sample application.

Sysmac Studio must be online with the simulator or the actual robot Integrated CPU Unit in order to use Application Sample Wizard. The following shows a procedure for an example of online-connected simulator.



### Additional Information

- Register a robot to use to the EtherCAT network of the Robot Integrated CPU Unit in advance. Refer to *4-2-1 Creating a New Project* for details.
- If you want to create a Robot Vision Manager sample application, select **Application Sample - Robot Vision Manager Sample** from the **Insert** menu.

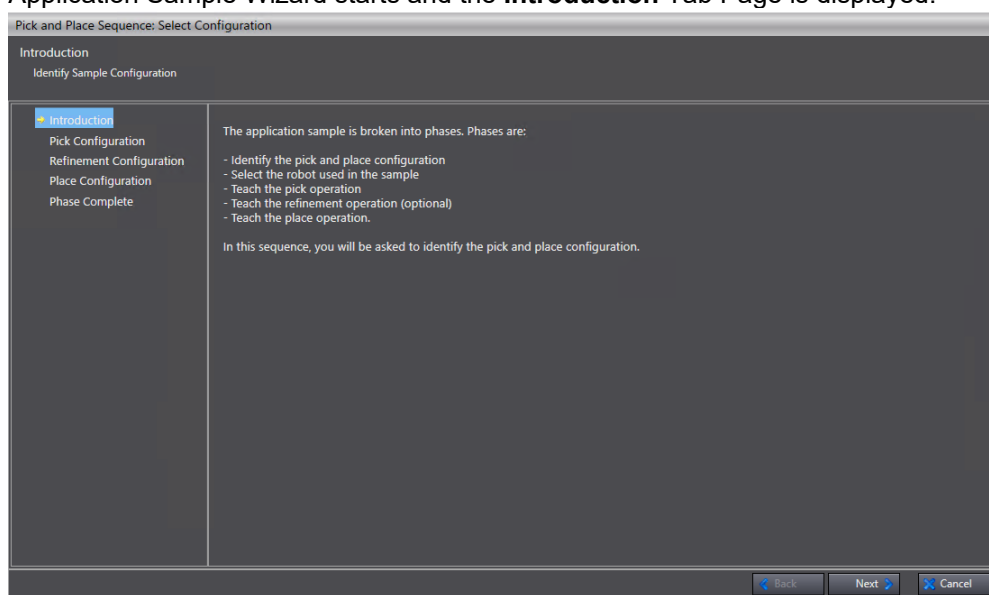
## Create a picking and placing system configuration

This section provides the information about the procedure to create a picking and placing system configuration by selecting system's elements, such as part detection method, picked part position adjustment in the gripper, or part placement position determination method.

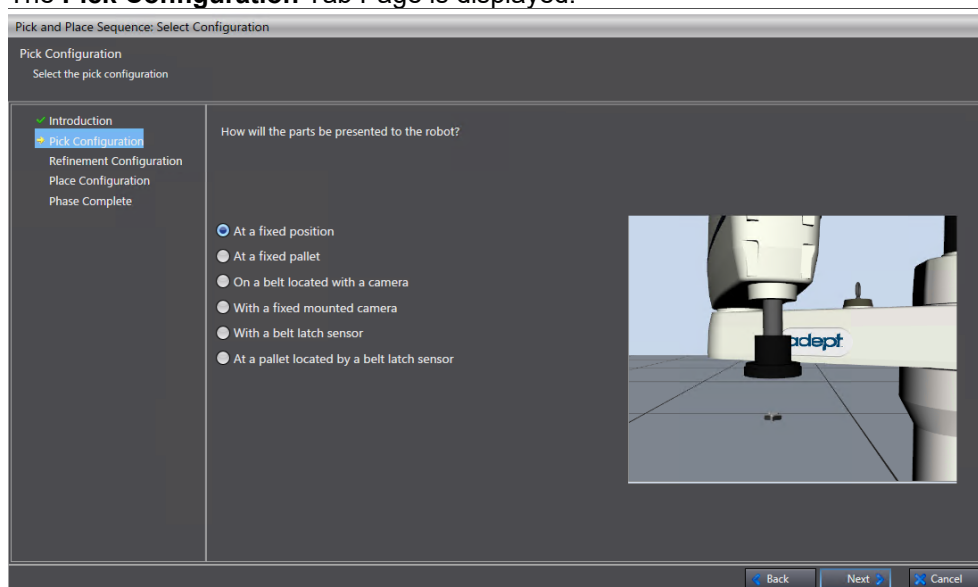
### ● Select part detection method

Select how the robot detects the part when it grabs.

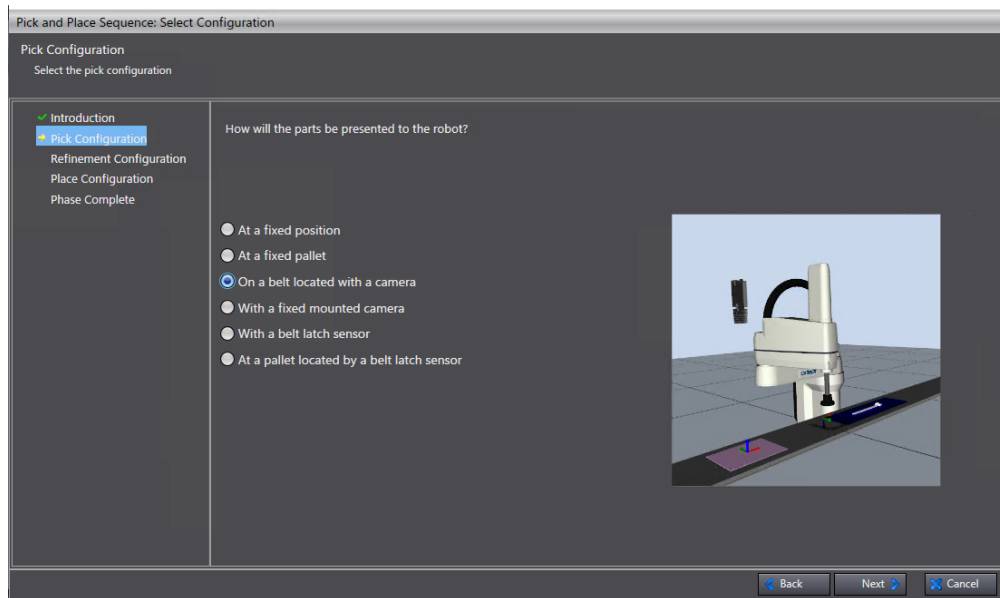
- 1 Select **Application Sample – Pack Manager Sample** from the **Insert** menu. Application Sample Wizard starts and the **Introduction** Tab Page is displayed.



- 2 Click the **Next** button. The **Pick Configuration** Tab Page is displayed.



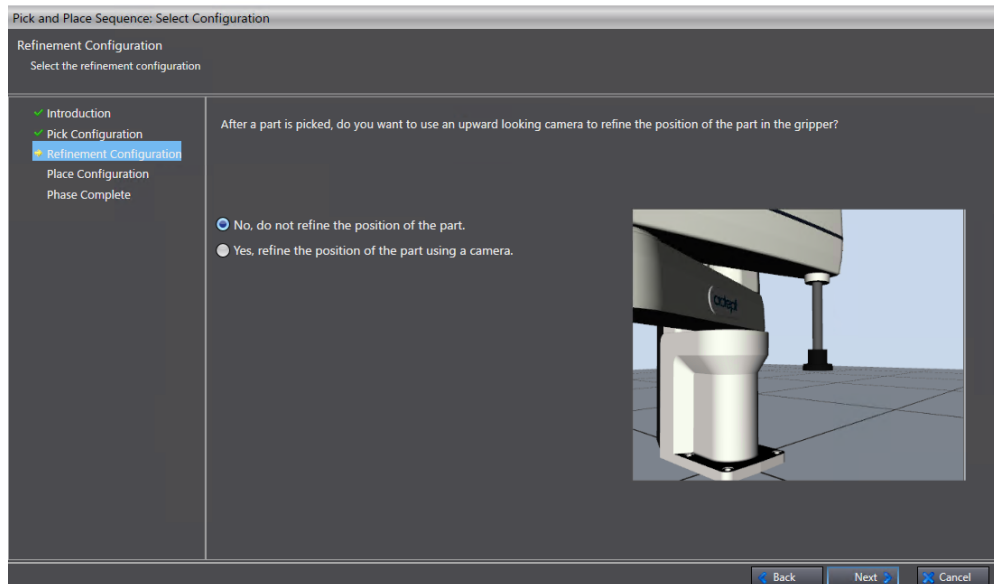
- 3 Select **On a belt located with a camera** to detect a part on the conveyor belt with a fixed camera.



Part detection methods are as shown below.

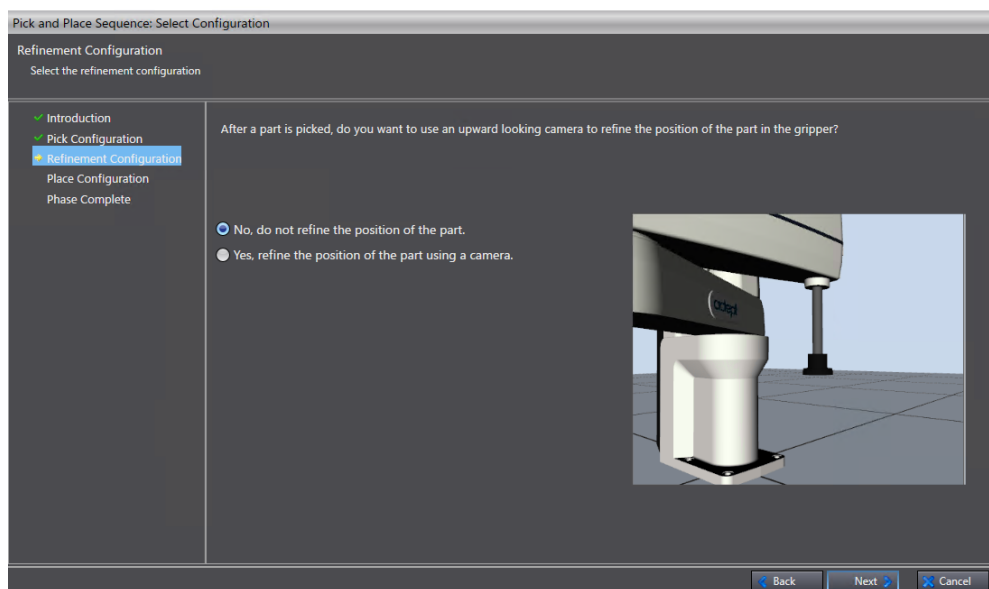
Type	Description
At a fixed position	Always detects a part at a fixed position.
At a fixed pallet	Always detects a part on a pallet at a fixed position.
On a belt located with a camera	Detects a part on the conveyor belt with a fixed camera.
With a fixed mounted camera	Detects a part with a camera mounted on the robot arm.
With a belt latch sensor	Detects a part on a conveyor belt with the latch sensor mounted on the conveyor belt.
At a pallet located by a belt latch sensor	Detects a pallet carrying a part, which on the conveyor belt, with the latch sensor mounted on the conveyor belt.

- 4** Click the **Next** button.  
The **Refinement Configuration** Tab Page is displayed. Select if you adjust the picked part's position in the gripper. Refer to *Part Position Adjustment in the Gripper* for details.



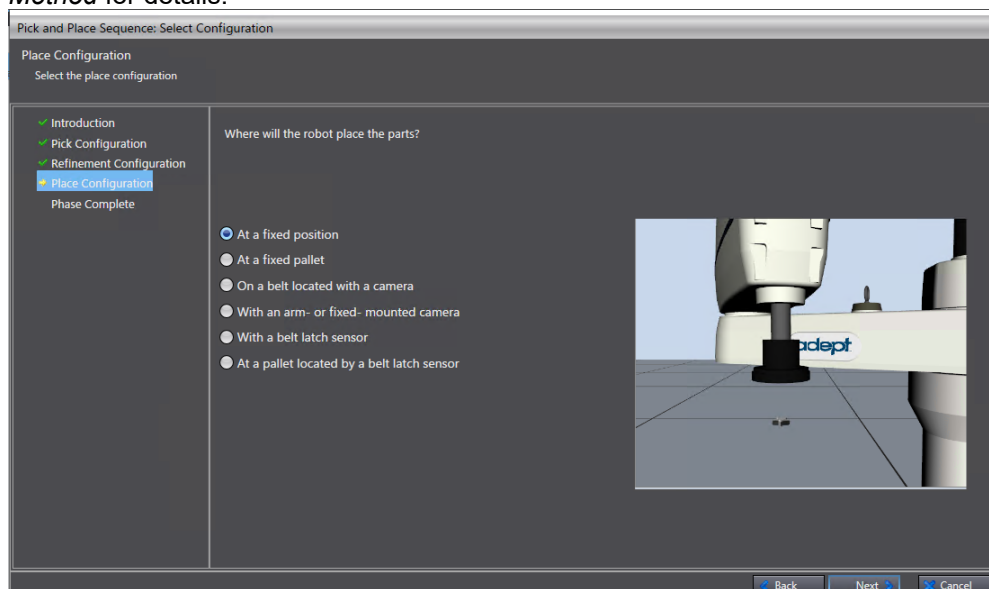
- **Part Position Adjustment in the Gripper**  
Select if you adjust the picked part's position in the gripper.

- 1** Select **No, do not refine the position of the part.**



If you want to adjust the part position using a camera, select **Yes, refine the position of the part using a camera**.

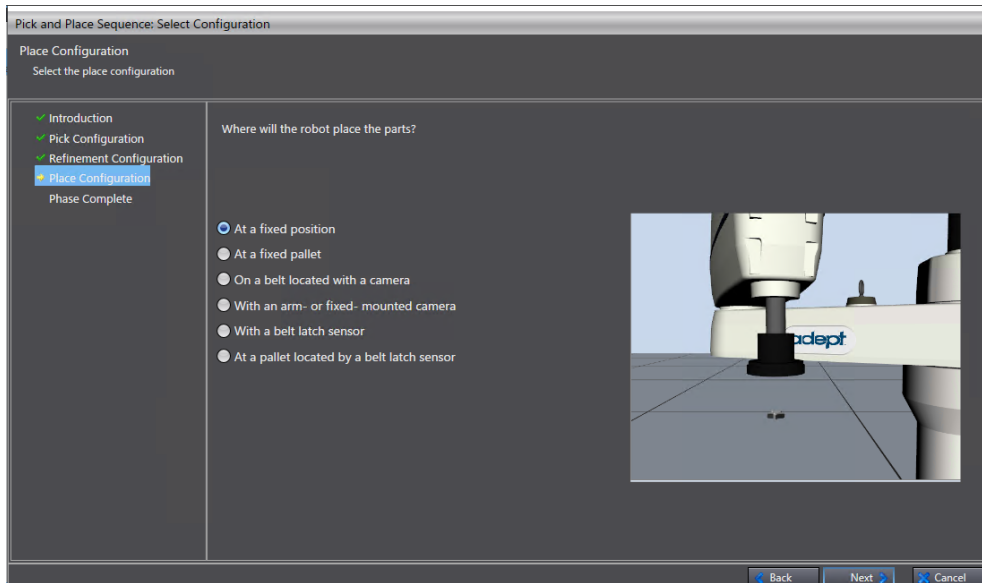
- 2 Click the **Next** button. The **Place Configuration** Tab Page is displayed, where you can select a position determination method with which the robot places a part. Refer to *Part Placement Position Determination Method* for details.



### ● Part Placement Position Determination Method

Select a method to determine where the robot places a part.

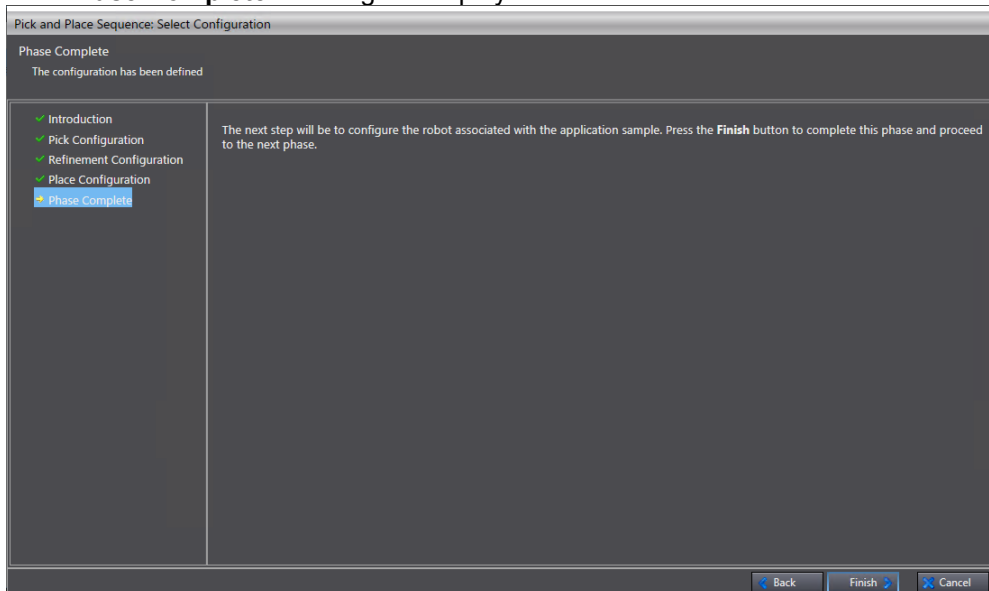
- 1 Select **At a fixed position**, which is the method to place a part at a fixed position.



Part placement position determination methods are as shown below.

Type	Description
At a fixed position	Always places a part at a fixed position.
At a fixed pallet	Always places a part on a pallet at a fixed position.
On a belt located with a camera	With a fixed camera, detects a position on the conveyor belt where a part is placed.
With an arm- or fixed- mounted camera	With a camera mounted on the robot arm, detects a position where a part is placed.
With a belt latch sensor	With the latch sensor mounted on the conveyor belt, detect a position on the conveyor belt where a part is placed.
At a pallet located by a belt latch sensor	With the latch sensor mounted on the conveyor belt, detect a position on the conveyor belt where a pallet carrying a part is placed.

- 2 Click the **Next** button.  
The **Phase Complete** Tab Page is displayed.



Now you have completed the picking and placing system configuration. Click the **Finish** button to proceed to the robot configuration. Refer to *Configure the Robot* for details.

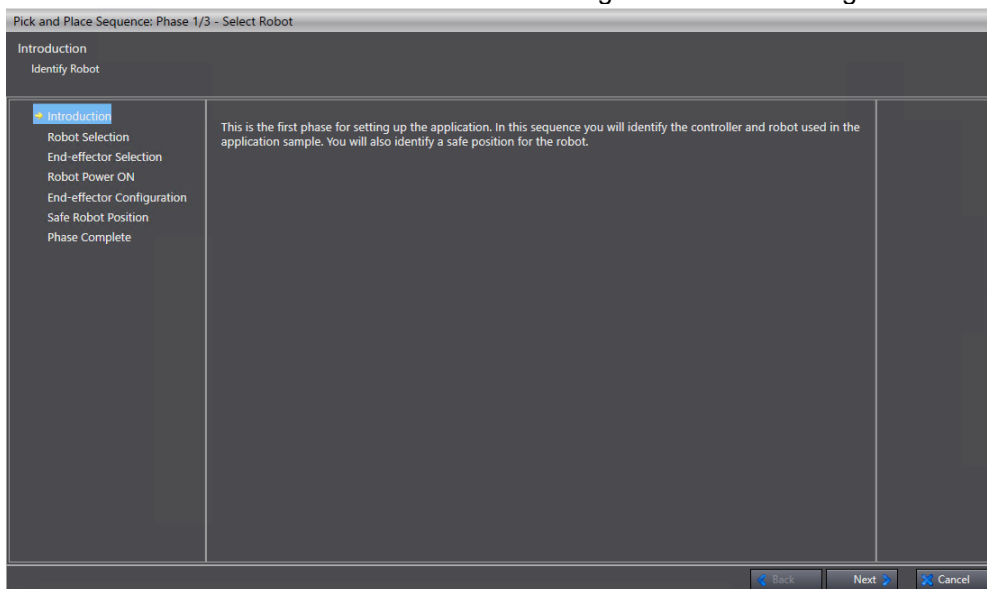
## Configure the Robot

This section provides the information about the procedure to select the robot to be used and setting the safe position.

- **Select Robot**

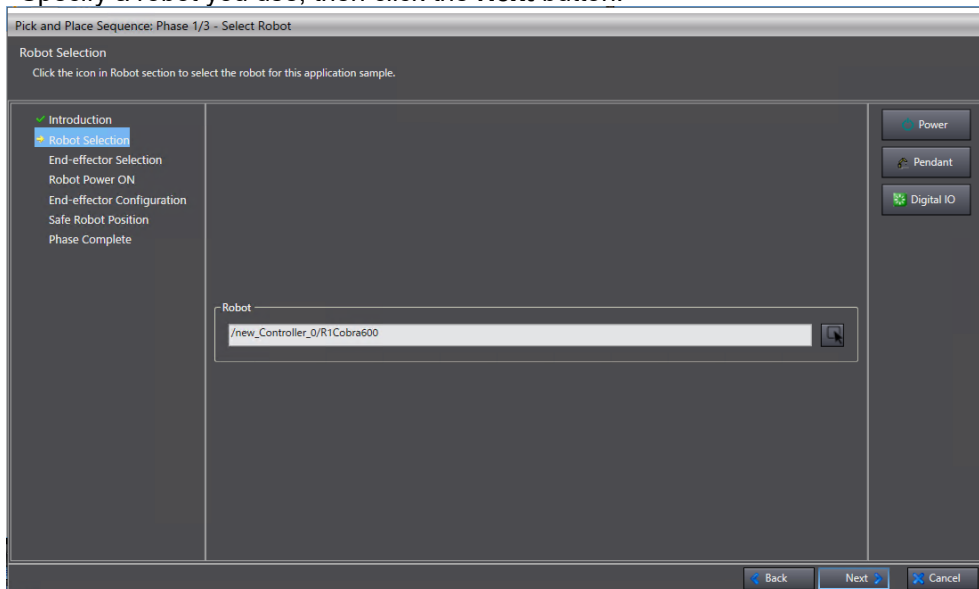
Select a robot you use for the application sample.

- 1 Click the **Next** button in the **Introduction** Tab Page for the robot configuration.

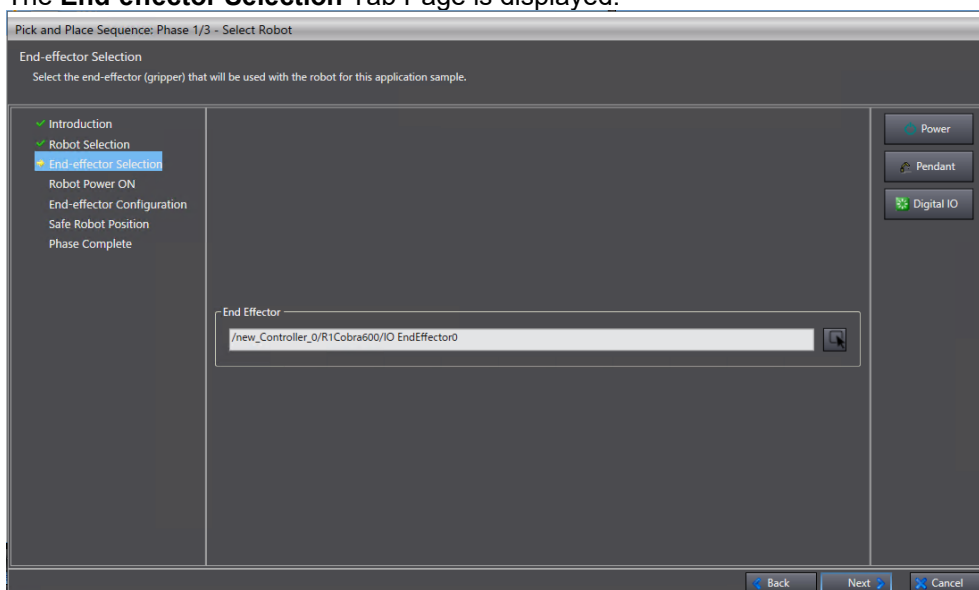


The **Robot Selection** Tab Page is displayed.

- 2 Specify a robot you use, then click the **Next** button.



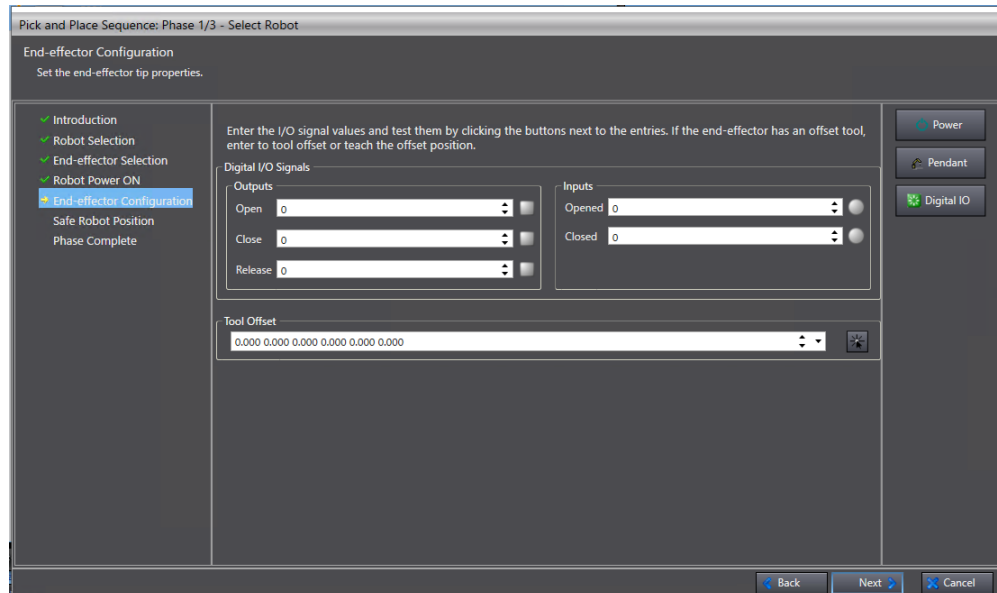
The **End-effector Selection** Tab Page is displayed.



- 3 Select an end-effector, then click the **Next** button.



The **End-effector Configuration** Tab Page is displayed. You can configure end-effector setting there.

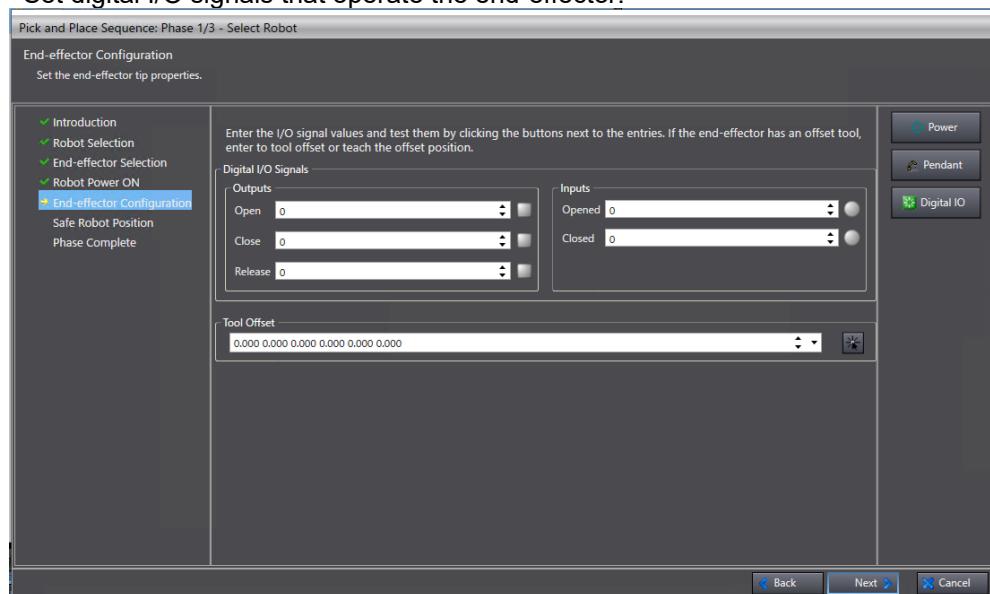


Refer to Configure End-effector for details.

## ● Configure End-effector

Configure properties of an end-effector tip.

- 1 Set digital I/O signals that operate the end-effector.



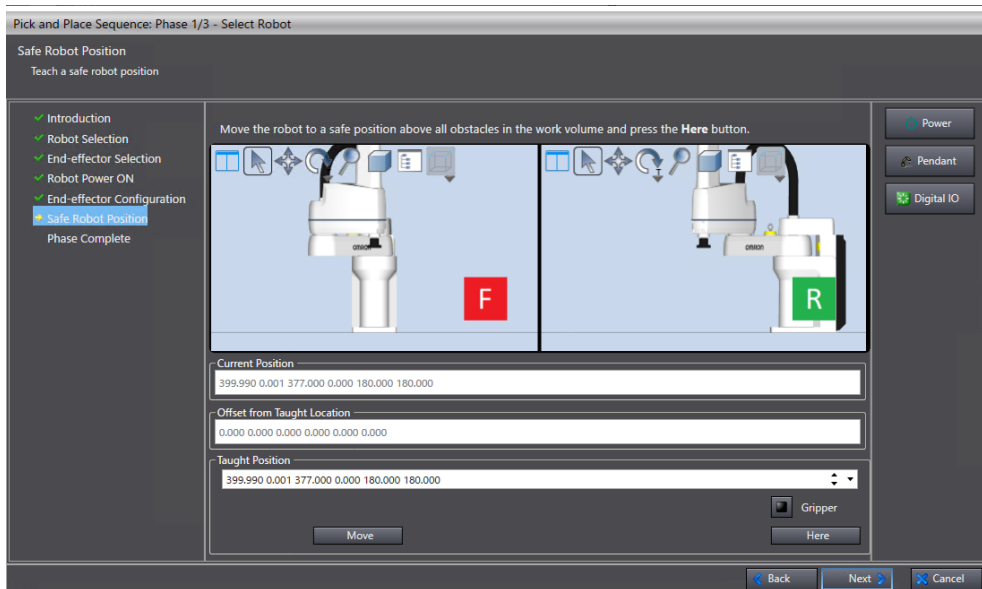
End-effector is not configured in this case. Leave them default.

- 2 Click the **Next** button.  
The **Safe Robot Position** Tab Page is displayed where you can configure a safety position of the robot arm. Refer to *Configure Robot Safety Position* for details.

## ● Configure Robot Safety Position

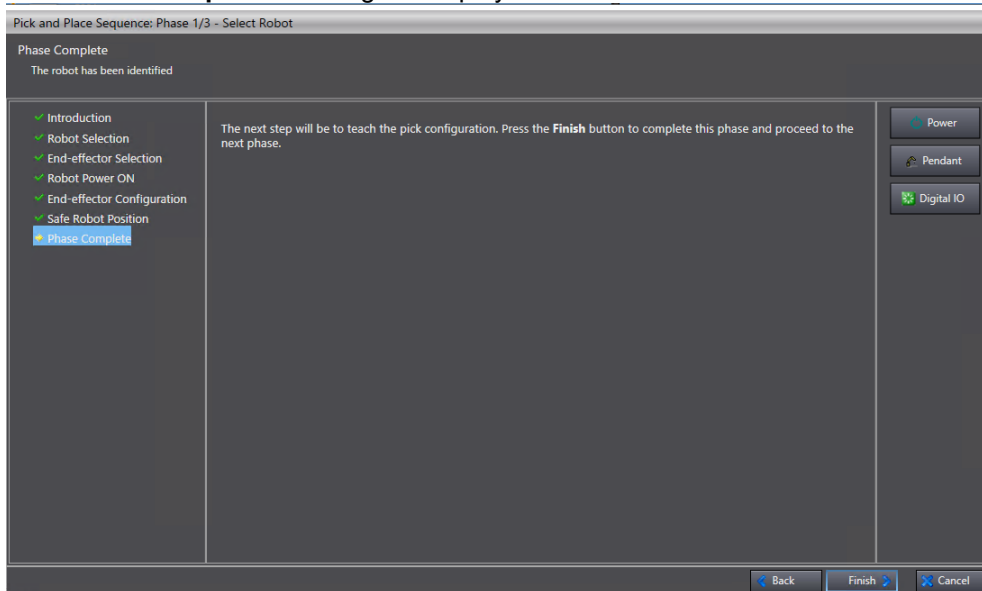
Configure a safe position for the robot.

- 1 Configure a safe position for the robot arm tip.



Click and drag the robot in the Robot View area to change the robot position.

- 2 Move the robot to the safe position, then click the **Here** button. The robot position is determined. Click the **Next** button. The **Phase Complete** Tab Page is displayed.

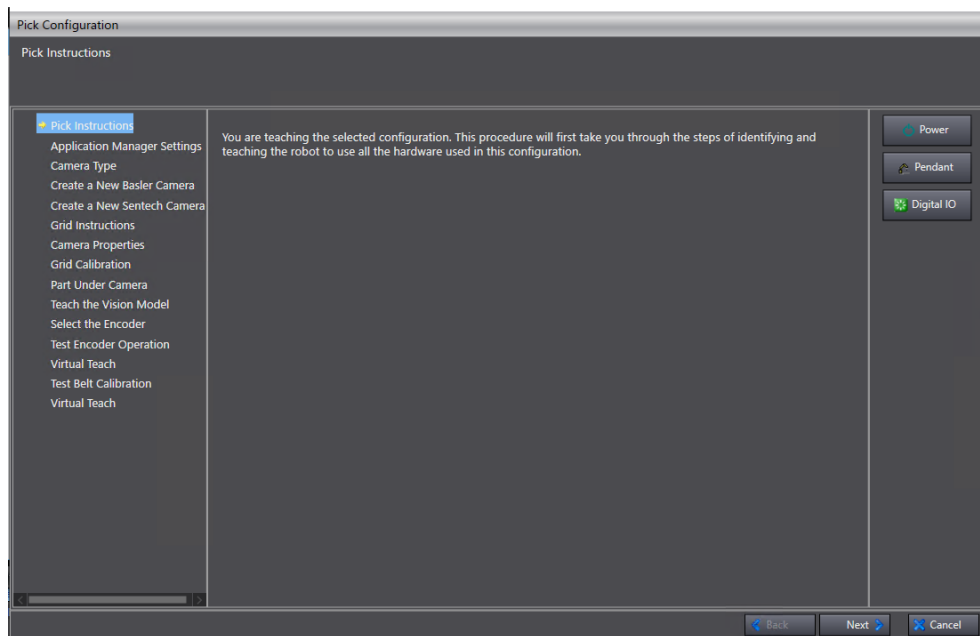


Now you have finished basic settings for the robot. Click the **Finish** button, then teach the robot to pick up a part. Refer to *Teaching Setting for Picking Motion* for details.

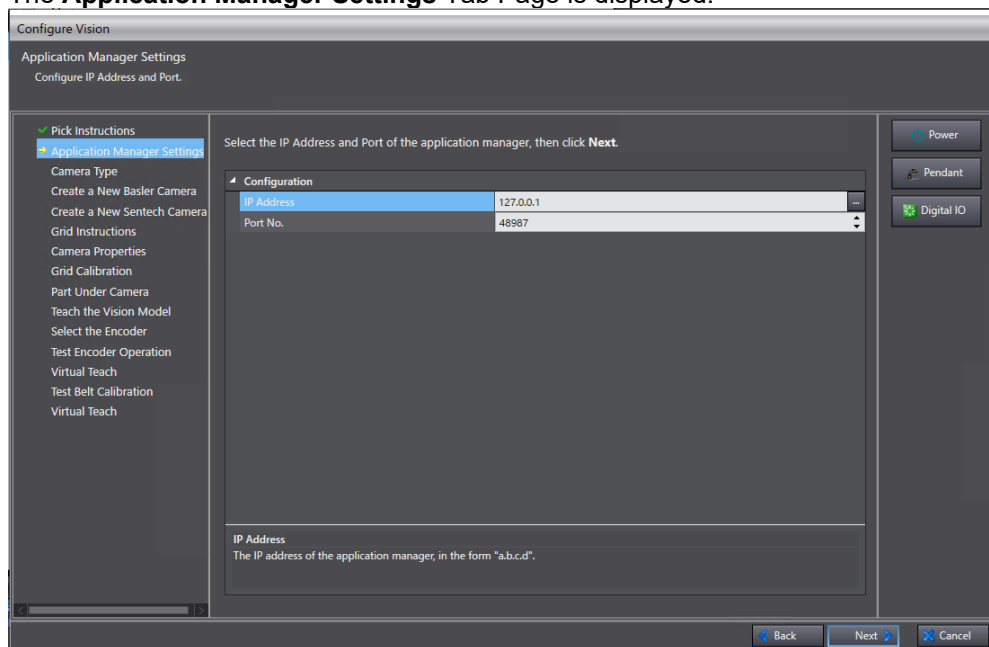
## Teaching Setting for Picking Motion

Teach the robot to pick up a part.

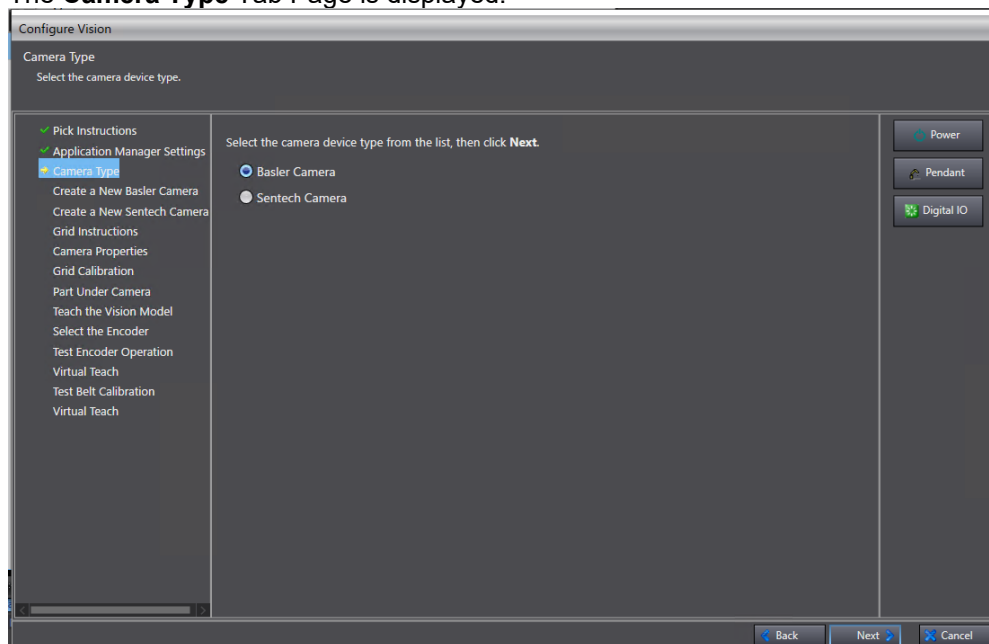
- 1 Click the **Next** button in the **Pick Instruction** Tab Page of Pick Configuration.



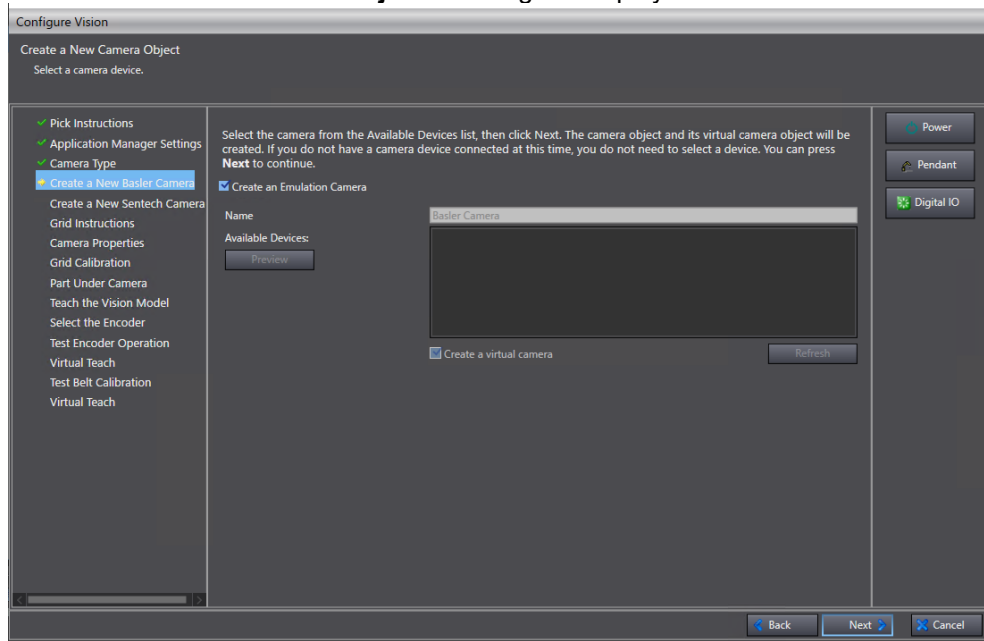
The **Application Manager Settings** Tab Page is displayed.



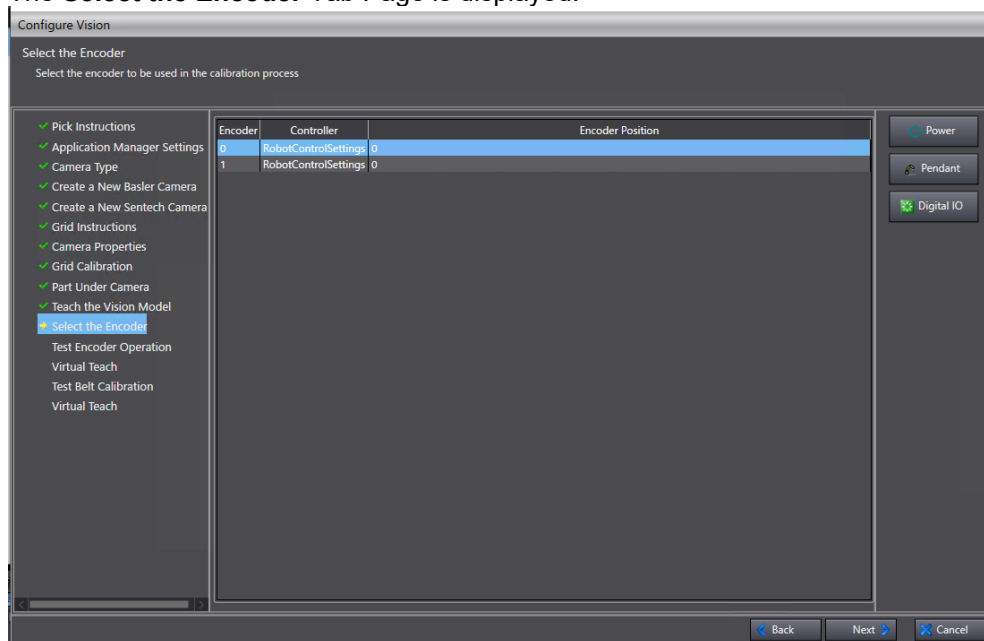
**2** Click the **Next** button.  
The **Camera Type** Tab Page is displayed.



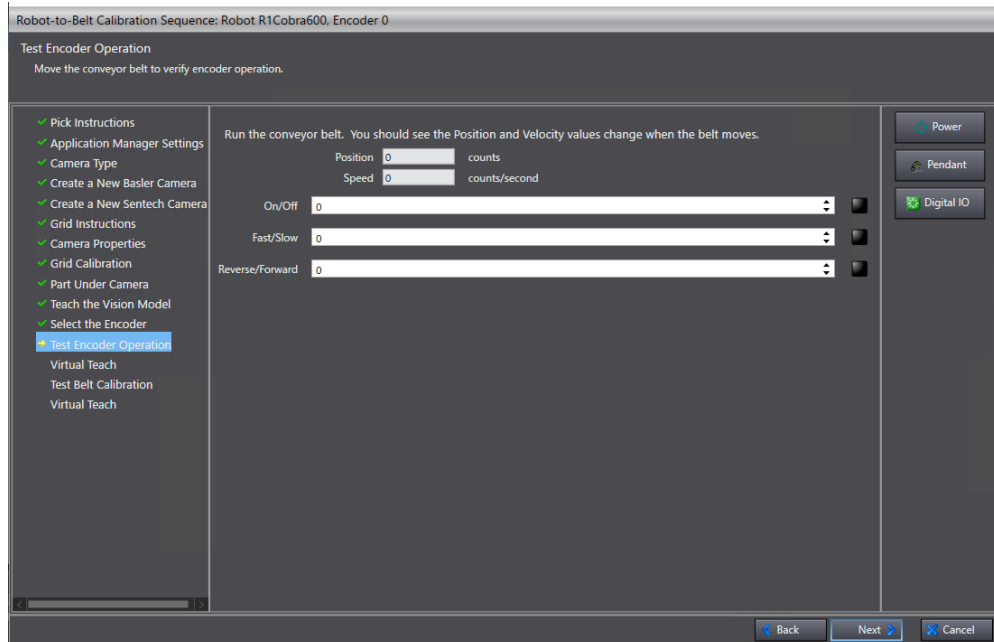
- 3 Click the radio button for the camera to use, then click the **Next** button. The **Create a New Camera Object** Tab Page is displayed.



- 4 Confirm that **Create an Emulation Camera** is checked. Click the **Next** button. The **Select the Encoder** Tab Page is displayed.

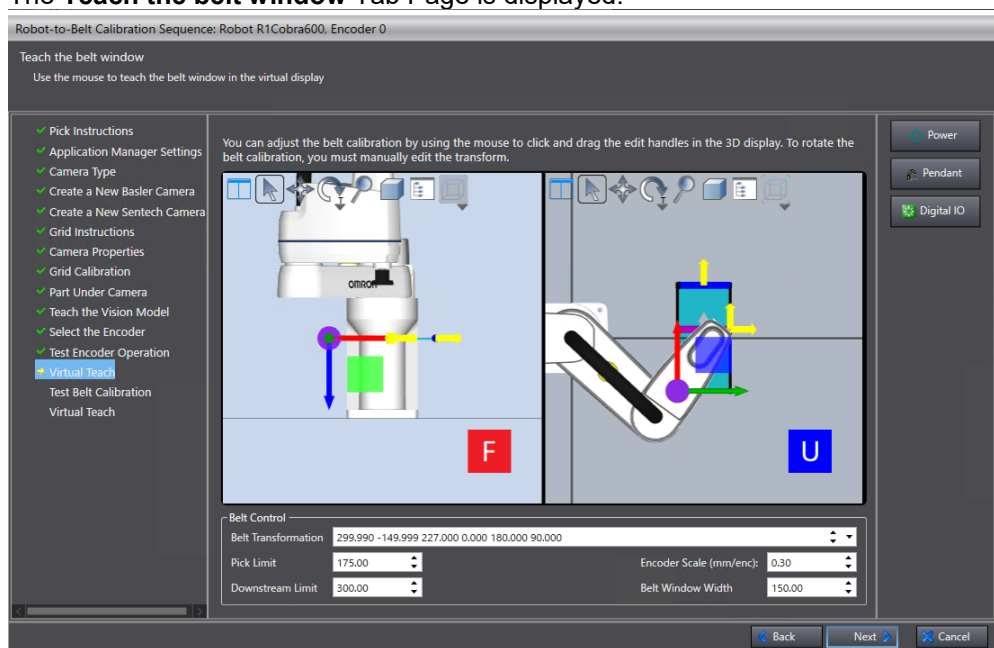


- 5 Select the encoder to use, then click the **Next** button. The **Test Encoder Operation** Tab Page is displayed.



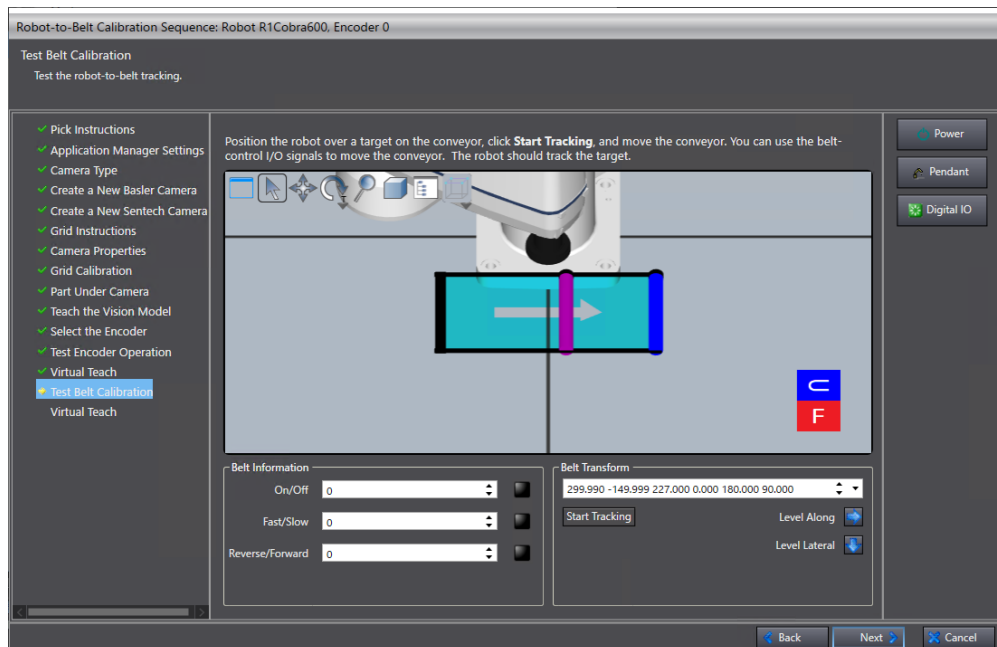
**6** Clicking the **ON/OFF**, **Fast/Slow**, or **Reverse/Forward** button moves the virtual conveyor belt. You can check the encoder's operation.

**7** After checking the encoder, click the **Next** button. The **Teach the belt window** Tab Page is displayed.



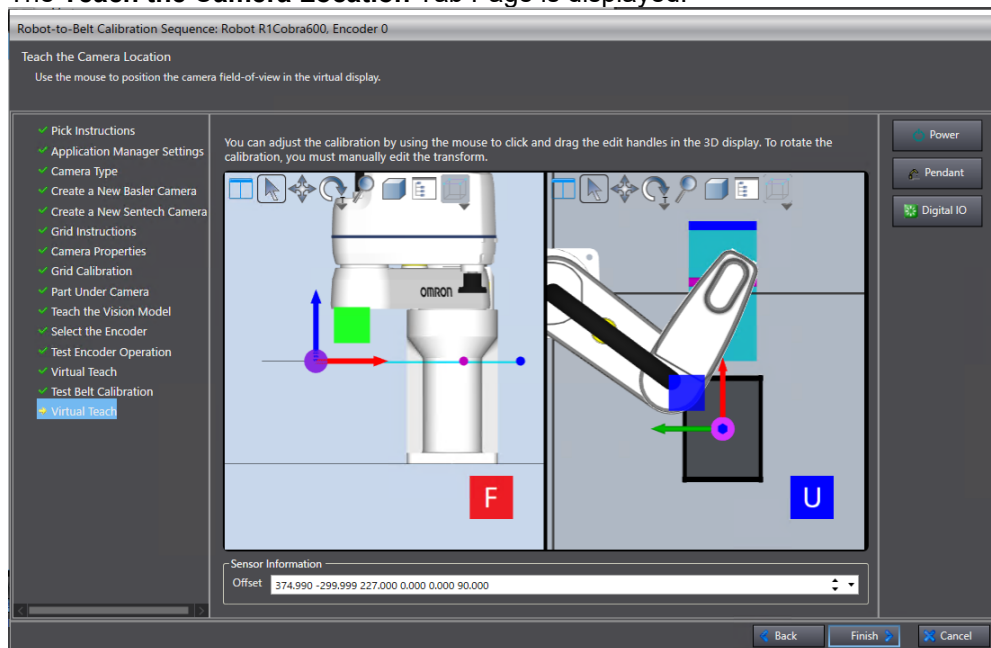
**8** Relate the belt's moving direction to the robot's work space. Click and drag the edit handle in the left view to calibrate the belt. Refer to *5-5-3 Belt Calibration* for detailed information on belt calibration.

**9** Click the **Next** if you have finished relating the belt's moving direction to the robot's work space. The **Test Belt Calibration** Tab Page is displayed.



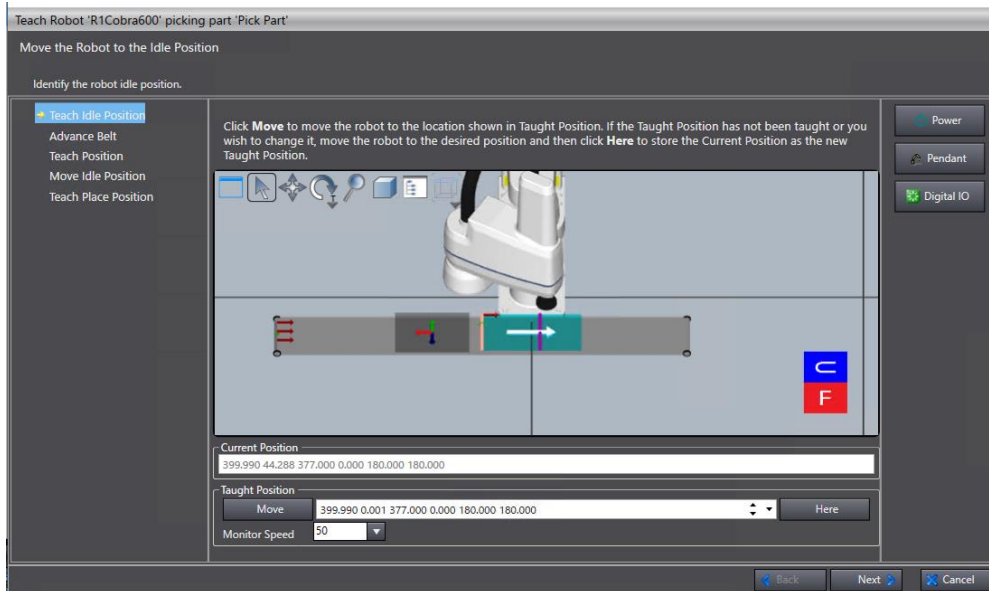
- 10** Test belt calibration.  
 Click and drag the robot tip to the center of the part on the conveyor belt. Then Click the **Start Tracking** button.  
 Click the **ON/OFF**, **Fast/Slow**, or **Reverse/Forward** button to move the virtual conveyor belt.  
 You can confirm the robot is tracing a target.  
 Refer to *Testing the Calibration* for details.

- 11** Click the **Next** button after finishing belt calibration test.  
 The **Teach the Camera Location** Tab Page is displayed.



- 12** Specify the camera position.  
 Clicking and dragging the edit handle in the left view moves the camera view. Refer to *5-5-4 Camera Calibration* for details.

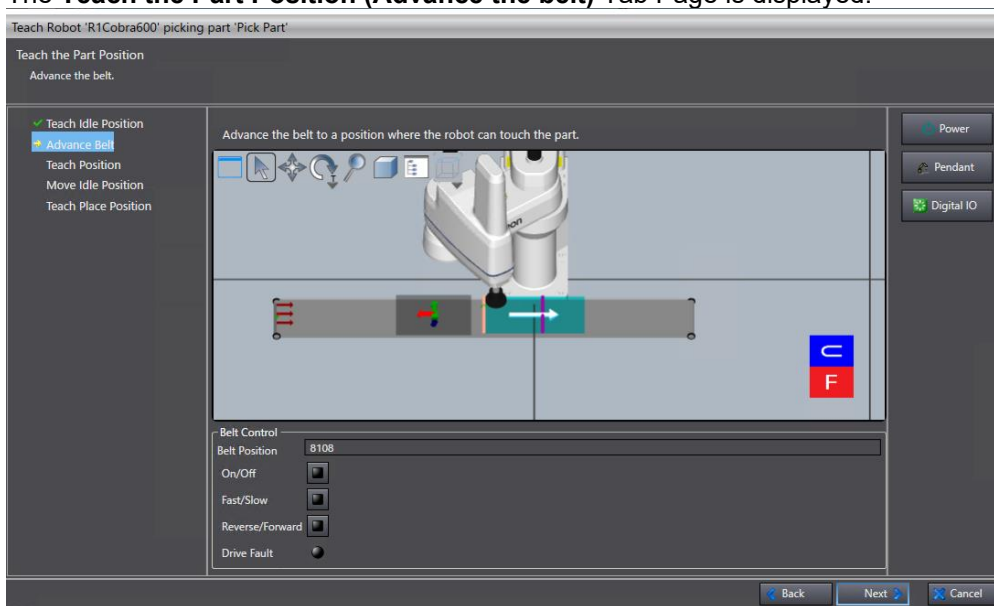
- 13** Click the **Finish** button after specifying the camera position.  
 The **Move the Robot to the Idle Position** Tab Page is displayed.



**14** Specify the position where the robot stands-by until a processing of the part or the target begins.

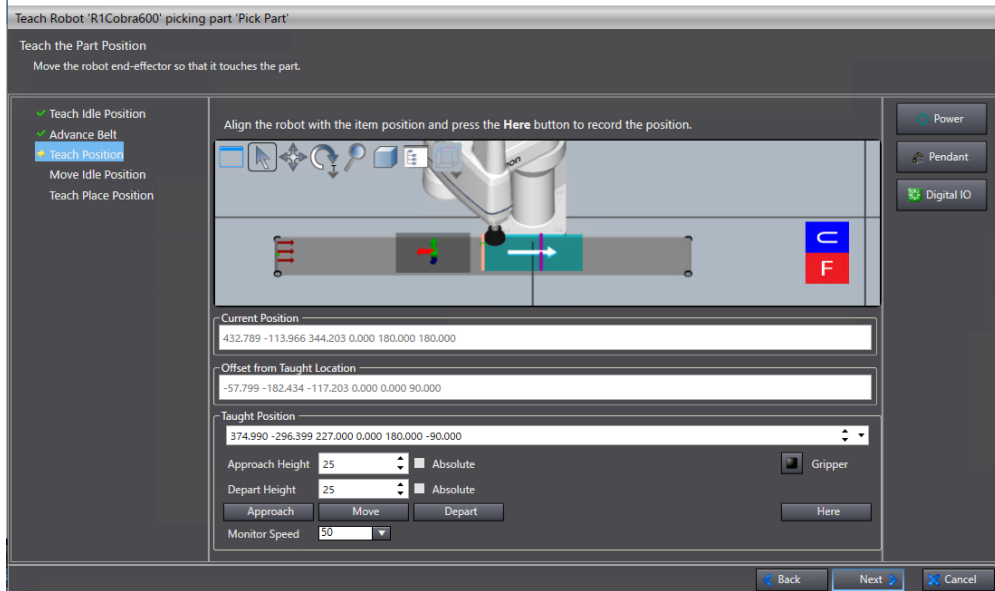
Clicking and dragging the robot in the robot view area changes robot position. Specify the position around the upper place between the picking and placing belts.

**15** Click the **Here** button to determine the position. Then click the **Next** button. The **Teach the Part Position (Advance the belt)** Tab Page is displayed.



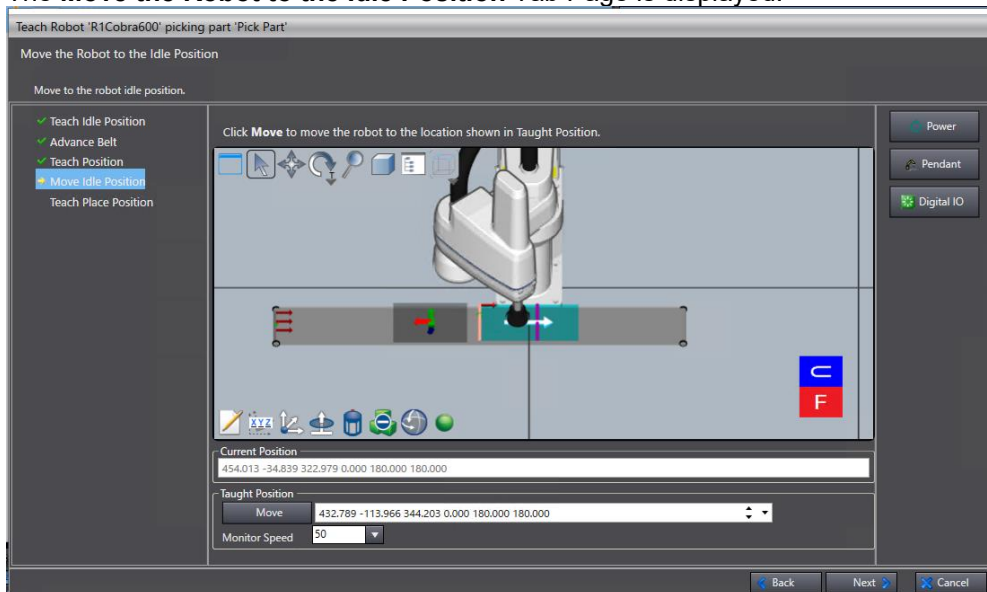
Click the **ON/OFF**, **Fast/Slow**, or **Reverse/Forward** button to move the belt so that the part is within the robot's range of movement.

**16** Click the **Next** button if you can confirm that the part is in the robot's moving range. The **Teach the Part Position (Move the robot end-effector so that it touches the part)** Tab Page is displayed.



- 17** The robot's end-effector moves so that it touches the part.  
 Drag the end-effector so that it touches the part. Then click the **Here** button.  
 Clicking the **Approach** button moves the robot above the part position measured by the camera.  
 Clicking the **Move** button makes the end-effector touch the part.  
 Clicking the **Depart** button moves the robot upward.

- 18** Click the **Next** button if you have finished teaching the part position.  
 The **Move the Robot to the Idle Position** Tab Page is displayed.



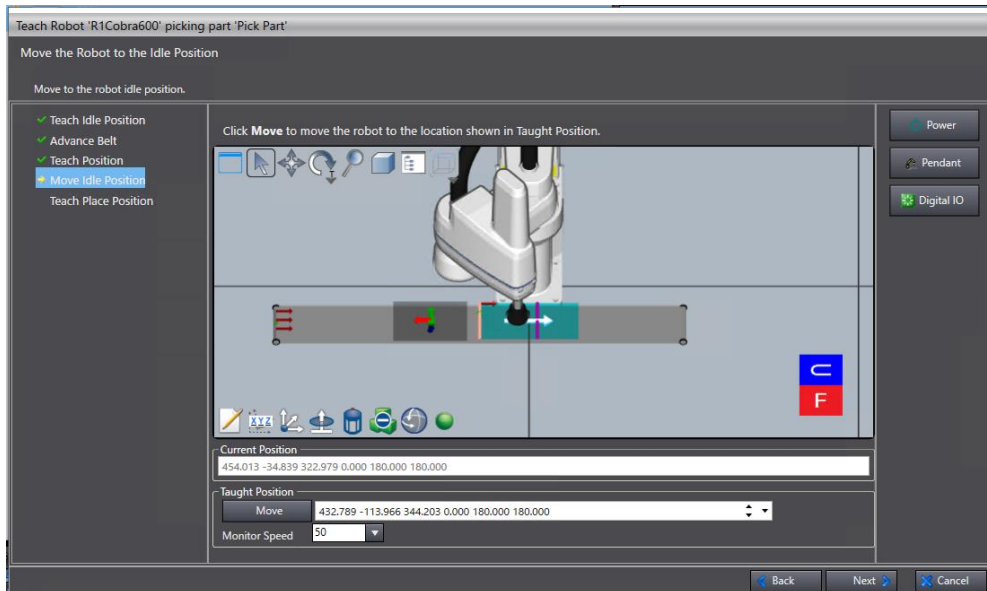
Now you have completed teaching the picking motion.  
 Next, teach the robot where to place the part. Refer to *Teaching Setting for Placement Position* for details.

## Teaching Setting for Placement Position

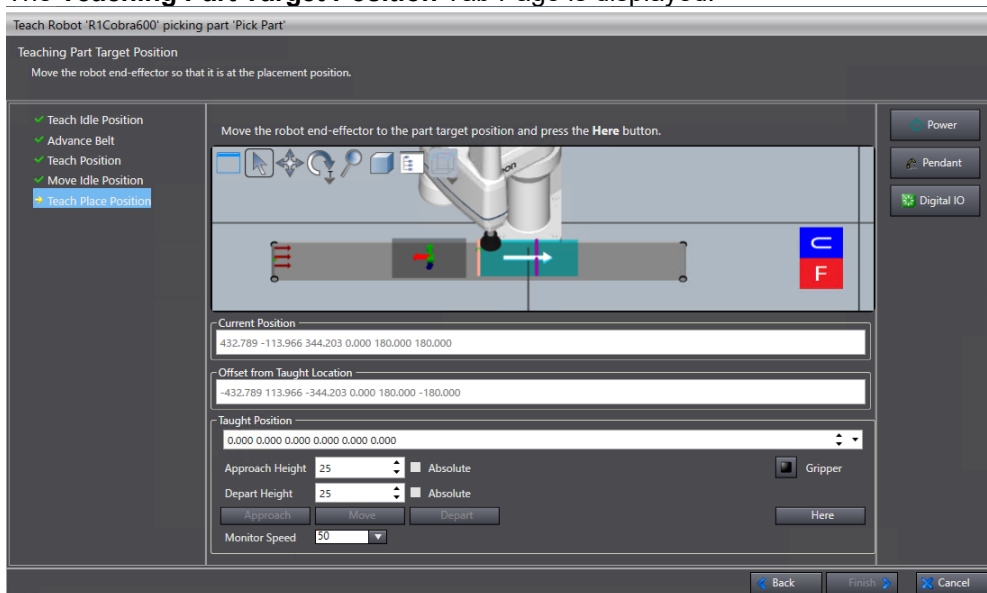
Teach the robot where to place the part.

- 1** Click the **Move** button to move the robot to the registered stand-by position.



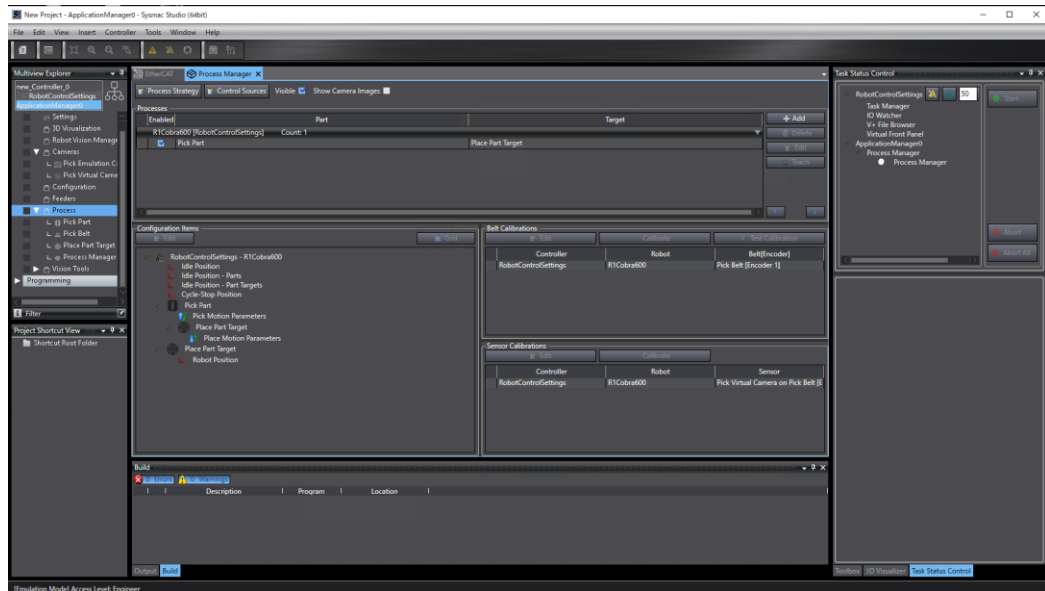


- 2 Click the **Next** button.  
The **Teaching Part Target Position** Tab Page is displayed.



Move the end-effector to where the part is to be placed.  
You can change the robot position by clicking and dragging the end-effector in the robot view area.

- 3 Click the **Here** button. Then, click the **Finish** button.  
Application Sample Wizard exits.  
Now you have completed creating the Pack Manager sample, which contains picking and placing movements.  
**Application Manager** is shown in the device list of Multiview explorer, and the **Process Manager** tab is displayed in the Edit Pane.



# 8-3 Pack Manager

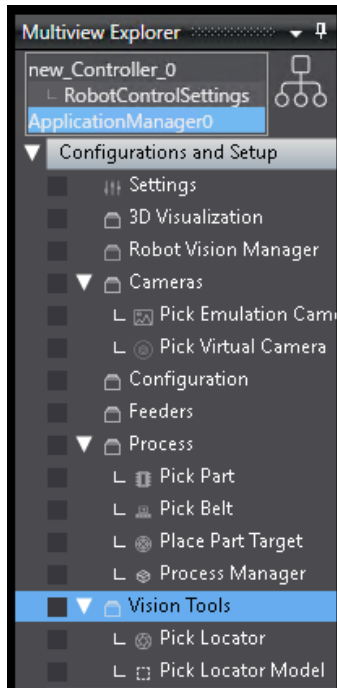
This section describes settings and offline debugging of Pack Manager.

## 8-3-1 Pack Manager Settings

Pack Manager items are automatically created after Pack Manager Application Sample Wizard exits. Modify settings for the items according to the system you want to build.

### Pack Manager Application Sample Items

The following items are registered when Pack Manager Application Sample wizard exits.

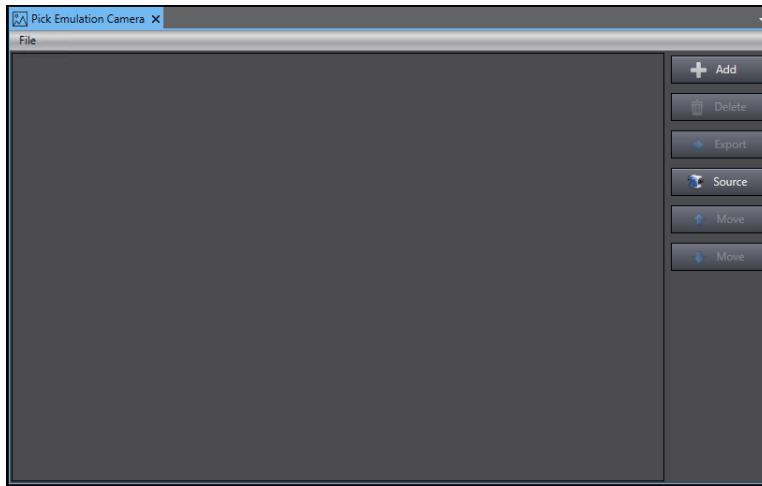


Item	Reference
Process manager	5-9-5 <i>Process Manager</i>
Recipe Manager	5-7-6 <i>Recipe Manager</i>
Data Mapper	5-7-2 <i>Data Mapper</i>
Program System Startup	5-7-5 <i>Program System Startup</i>
Pick Part	5-9-3 <i>Part</i>
Pick Belt	5-9-4 <i>Belt</i>
Pick Emulation Camera	5-6-2 <i>Emulation Camera</i>
Pick Locator	5-10 <i>Vision Tools</i>
Pick Locator Model	5-10 <i>Vision Tools</i>
Pick Part Target	5-9-2 <i>Part Target</i>
Pick Virtual Camera	5-6-1 <i>Virtual Camera</i>

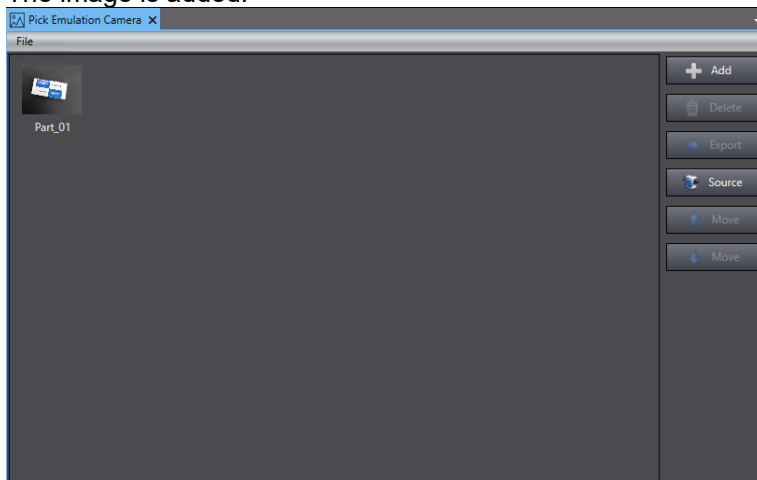
### Registering Part Images

If you want to perform an offline simulation on image identification processing, add an image of the part dealt in your robot integrated system, if necessary. Then register the added image to Locator Model and configure locator detection setting.

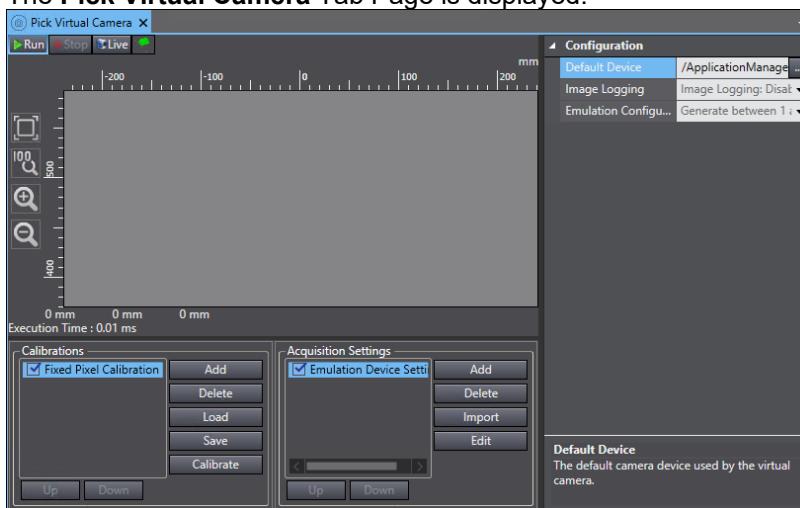
- 1 Double-click **Cameras – Pick Emulation Camera** under **Configurations and Setup** in Multiview Explorer.  
The **Pick Emulation Camera** Tab Page is displayed.



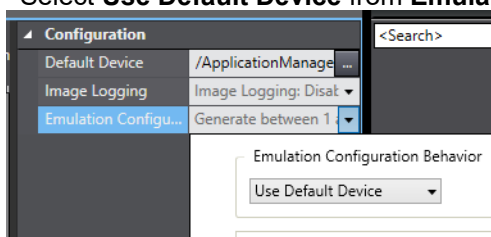
- 2 Click the **Add** button to select an image of the part. Then Click the **Finish** button. The image is added.



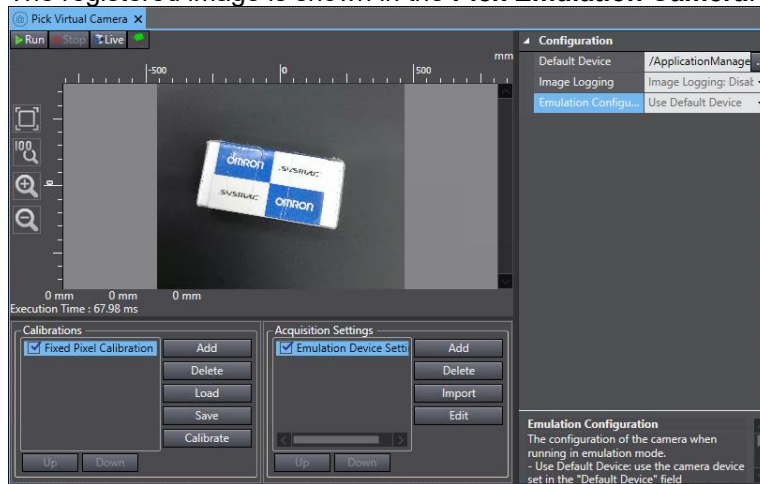
- 3 Double-click **Cameras – Pick Virtual Camera** under **Configurations and Setup** in Multiview Explorer. The **Pick Virtual Camera** Tab Page is displayed.



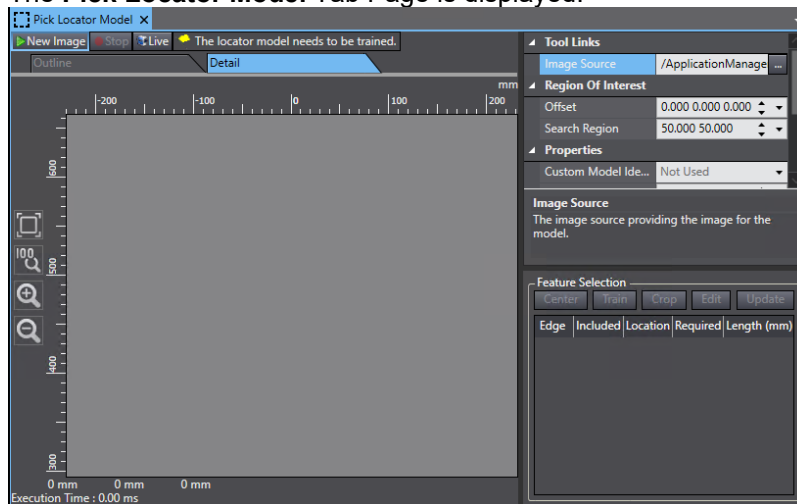
- 4 Select **Use Default Device** from **Emulation Configuration** under **Configuration**.



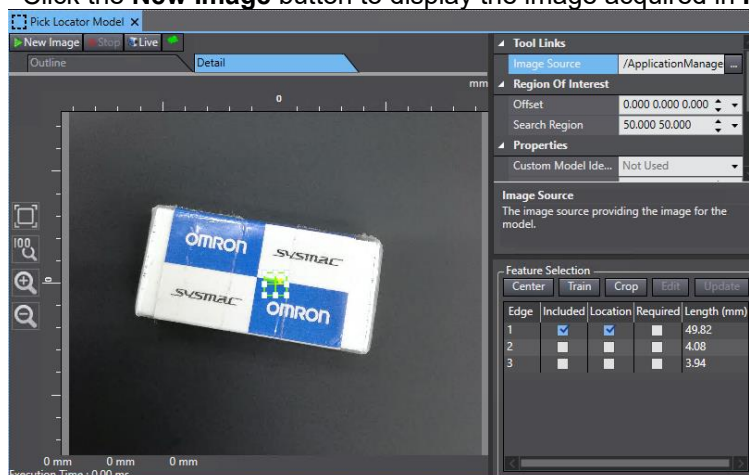
- 5 Click the **Run** button.  
The registered image is shown in the **Pick Emulation Camera**.



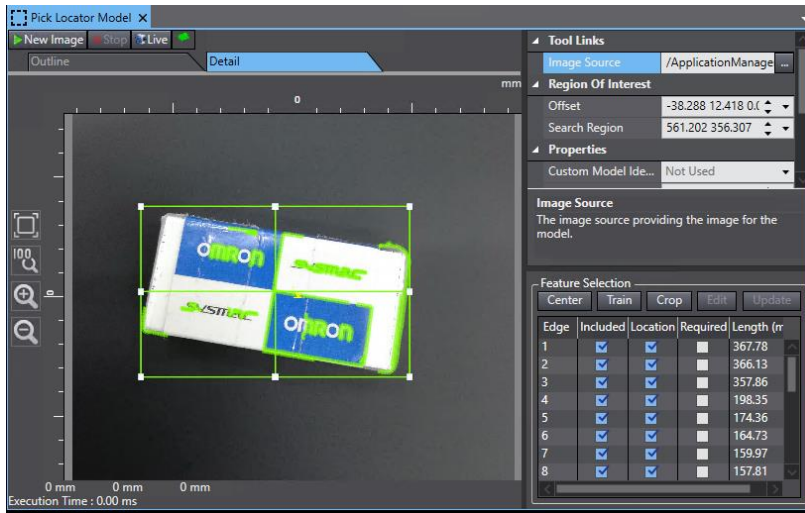
- 6 Double-click **Pick Locator Model – Vision Tools** under **Configurations and Setup** in Multiview Explorer.  
The **Pick Locator Model** Tab Page is displayed.



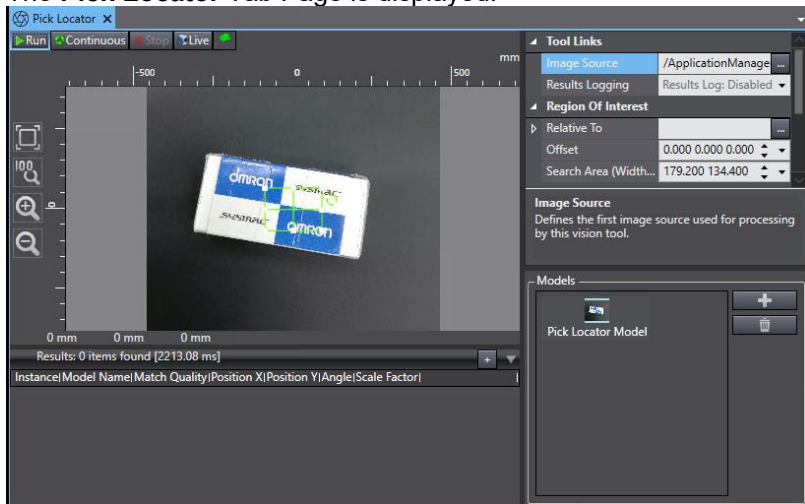
- 7 Click the **New Image** button to display the image acquired in **Pick Virtual Camera**.



- 8 Set region and center point. Then click the **Train** button in the **Feature Selection** field. Check the features used for detection.

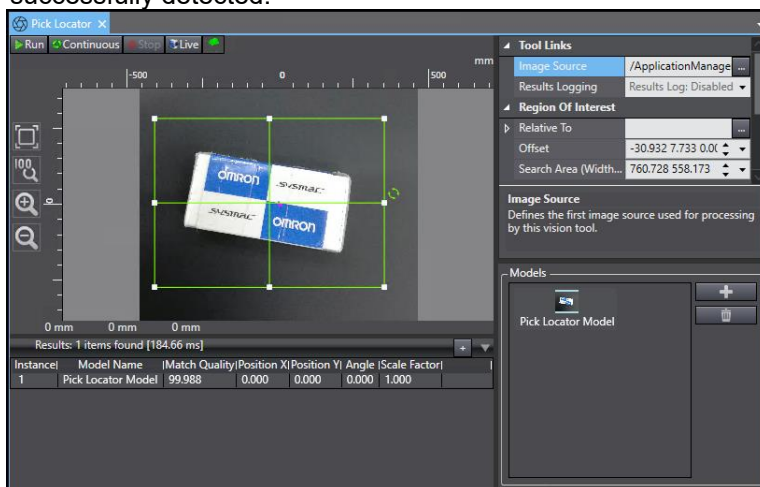


- 9 Double-click **Pick Locator** under **Configurations and Setup** in Multiview Explorer. The **Pick Locator Tab Page** is displayed.



If a locator model is not shown, click the **[+]** button in the **Models** field. Then select **Pick Locator Model**.

- 10 Confirm that the locator model registered to **Models** in **Configuration**. Move and adjust the green frame to specify the region in which the locator is detected. Then click the **Run** button. If the reading “**1 items found**” is displayed in the **Results** field, the locator has been successfully detected.

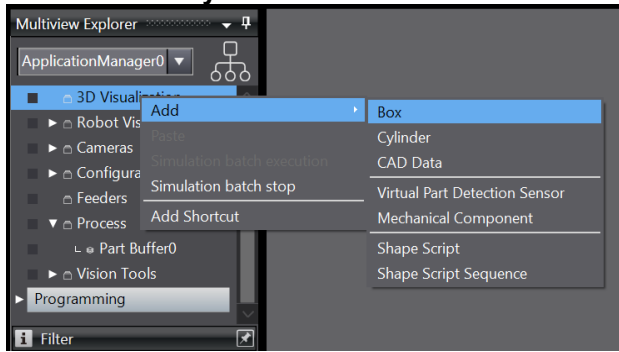


Detect locators for other regions or images to test detections in the same way.

## Adding Peripheral Objects

If you need to check whether the robot interferes with peripheral devices, add 3D shape data to your robot integrated system.

- 1 Select **Box** or **Cylinder** from the **3D Visualization** menu to add 3D shape data.



3D shape data is placed where it could contact the robot arm, e.g., beside the conveyor belt. Refer to *5-4 3D Visualization* for details.

### 8-3-2 Pack Manager Offline Debugging

Offline debugging of Pack Manager is executed by operating Application Manager and Robot Integrated CPU Unit on the virtual controller. Refer to *7-1-1 Mechanism and Procedure of Offline Debugging* for details.

# 8-4 Robot Vision Manager

This section describes the Robot Vision Manager configuration and offline debugging.

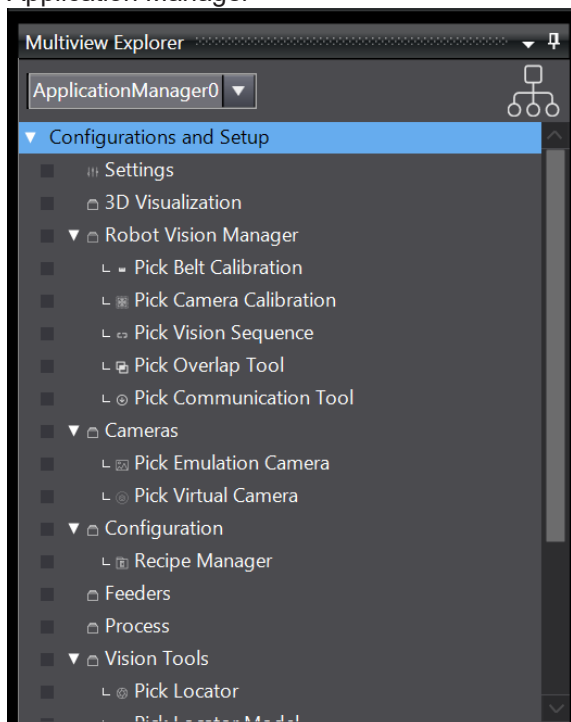
## 8-4-1 Robot Vision Manager Configuration

When the application sample wizard for the Robot Vision Manager is completed, the items in the Robot Vision Manager configuration are automatically created. Modify the configuration elements according to the system you are creating.

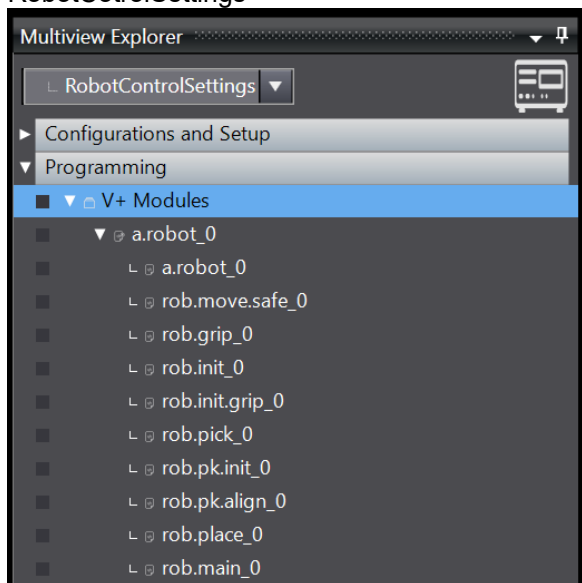
### Application sample configuration elements of the Robot Vision Manager

When you complete the Robot Vision Manager application sample wizard, the following items are registered.

Application Manager



RobotControlSettings



Item	Reference
Pick vision sequence	5-5-6 Vision Sequence
Pick belt calibration	5-5-3 Belt Calibration
Pick camera calibration	5-5-4 Camera Calibration



Item	Reference
Pick overlap tool	<i>5-5-7 Overlap Tool</i>
Pick communication tool	<i>5-5-8 Communication Tool</i>
Pick emulation camera	<i>5-6-2 Emulation Camera</i>
Pick virtual camera	<i>5-6-1 Virtual Camera</i>
Pick locator	<i>5-10 Vision Tools</i>
Pick locator model	<i>5-10 Vision Tools</i>
V+ module	<i>Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)</i>
V+ variable	<i>Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)</i>

## Registration of part images

To simulate image identification processes offline, add part images handled by the Robot Integration System as required.

Refer to *Registering Part Images* in *8-3-1 Pack Manager Settings*.

## Adding peripheral objects

To check for interference between robots and peripheral devices, add the 3D shape data to the Robot Integrated system as required.

Refer to *Adding Peripheral Objects* in *8-3-1 Pack Manager Settings*.

### 8-4-2 Offline Debugging of the Robot Vision Manager

The Robot Vision Manager offline debugging is performed by operating the Application Manager and the Robot Integrated CPU Unit with the virtual Controller. Refer to *7-1-1 Mechanism and Procedure of Offline Debugging*.



# 9

## 9 Maintenance

---

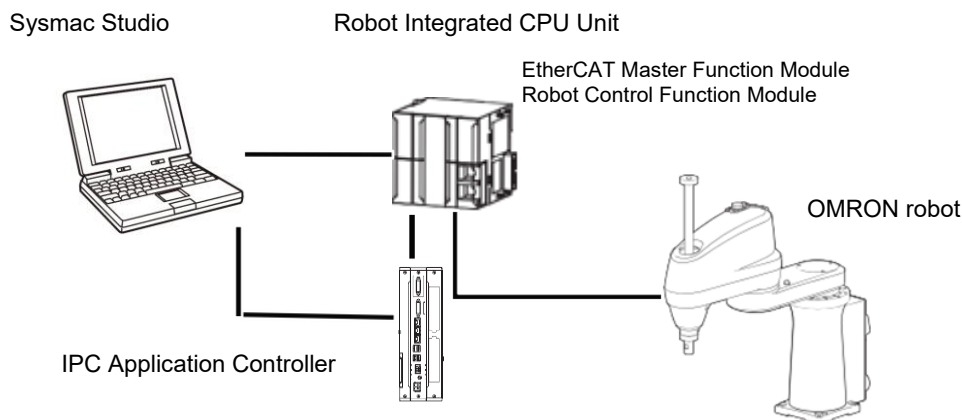
This section describes how to check errors and events that occur in the Robot Integrated System.

# 9-1 Troubleshooting

This section describes the functions of the Sysmac Studio that can be used for system troubleshooting and maintenance.

## 9-1-1 Introduction for functions of troubleshooting and maintenance

This section introduces the functions for troubleshooting and maintenance of the Robot Integrated System that consists of the IPC Application Controller, the Robot Integrated CPU Unit, and the OMRON robot.



Application	Function	Object	Introduction	Reference
Checking errors at the time of troubleshooting	Troubleshooting	EtherCAT Master Function Module Robot Control Function Module	You can use troubleshooting to check the errors that occurred in the Controller, display corrections for the errors, and clear the errors. You can check for errors in the EtherCAT Master Function Module and in the Robot Control Function Module.	<i>Sysmac Studio Version 1 Operation Manual (Cat. No. W504)</i>
Trouble cause investigation and adjustment	System Monitor	Application Manager Robot Control Function Module	Monitors parameters of the Application Manager and the Robot Control Function Module in real time.	<i>10-2 System Monitor</i>
	View eV+ Log	Robot Control Function Module	Displays the processing history of the Robot Control Function Module.	<i>Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)</i>
	Hardware Diagnostics	OMRON robot	You can check robot motor status.	<i>10-3 Hardware Diagnostics</i>
	Robot data collection	OMRON robot	Views and stores the Robot Integrated System data.	<i>10-4 Data Collection</i>
	Robot motor tuning	OMRON robot	Sends a square wave positioning command to the specified motor and observe the response.	<i>10-5 Motor Tuning</i>

# 10

## 10 Other Functions

---

This section describes other functions as the printing and hardware diagnostics.

# 10-1 Printing

This section describes printing operations of the Application Manager. Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for basic functions and operation procedures of printing.

## 10-1-1 Items You Can Print

You can print the following items. You can select any of the items to print.

Main category	Middle category
Setting	IP Address
	Port number
Adding 3D object	3D CAD Data
	Cylinder
	Box
	Mechanical component
	Virtual part detection sensor
Robot vision manager	Belt latch calibration
	Belt calibration
	Camera calibration
	Gripper Offset Table
	Vision sequence
	Vision sequence
Setting cameras	Virtual camera
	Emulation camera
	Basler camera
	Sentech camera
	Custom device
Configuration	Controller Connection Startup
	Data Mapper
	Note
	OPC container
	Program System Startup
	Recipe Manager
	Recipe Manager Script
Feeder	AnyFeeder
	IO Feeder
Process	Part Buffer
	Part Target
	Part
	Belt
	Process Manager
	Allocation Script
	Pallet
	Vision Refinement Station
Vision Tool	Finder
	Inspection
	Reader
	Calculation
	Image Process
Programming	Custom
	C# program
	Variable

# 10-2 System Monitor

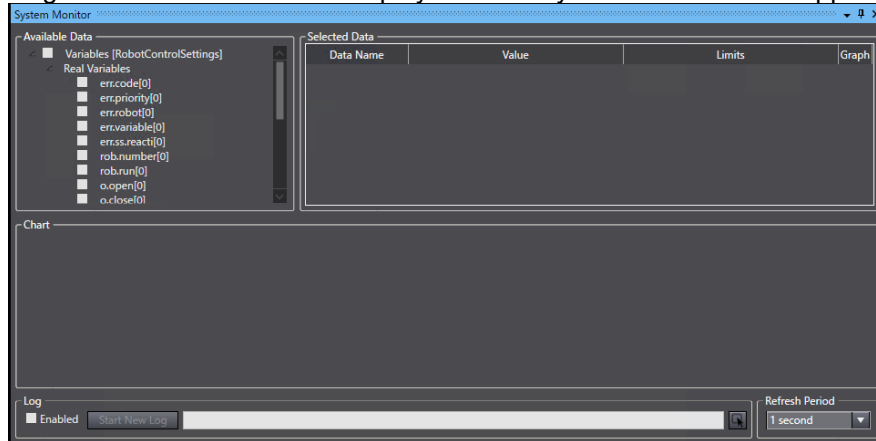
The System Monitor can display the status of process running in the Application Manager, the robot parameters. In addition, it can display the V+ variable values of the robot integrated CPU Unit in a graph and output them to a log file.

The procedure for displaying the V+ variables in the System Monitor during the simulation, is described below as an example.

**1** Select **RobotControlSettings** from the device list in the Multiview Explorer.

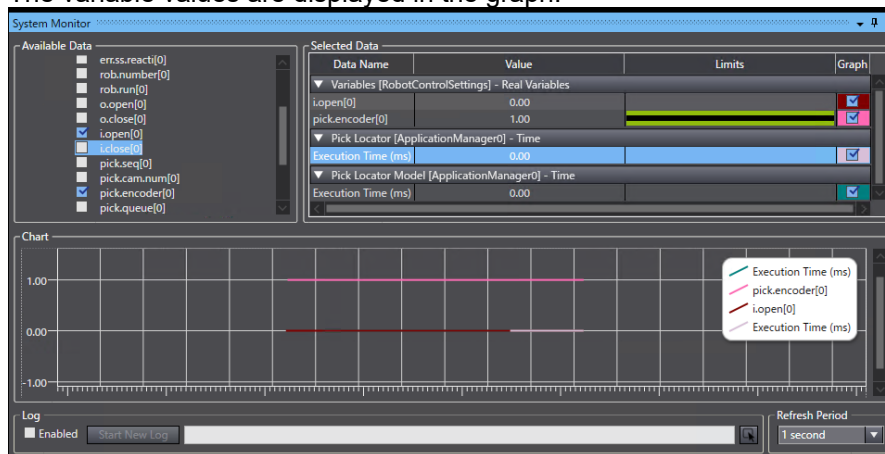
**2** Select **System Monitor** from the **View** Menu.

Registered V+ variables are displayed in the System Monitor Pane appeared.



**3** Check the variables to monitor in **Available Data**. And then check **Graph** that corresponding the variables in **Selected Data**.

The variable values are displayed in the graph.



**4** In **Log**, select the folder to save the log. And then check **Enabled** and click the **Start New Log** button.

A text file, that saves the variable values, is created in the specified folder.

## Monitor data

Use the following tables to understand System Monitor data.

### ● Robot Parameter

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

Parameter name	Description
	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

Parameter	Description
Instance Count	The number of instances that have been available since the last restart or reset.
Belt Velocity (mm/s)	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 500ms, it will tell you the instantaneous instances/minute.
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

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.)



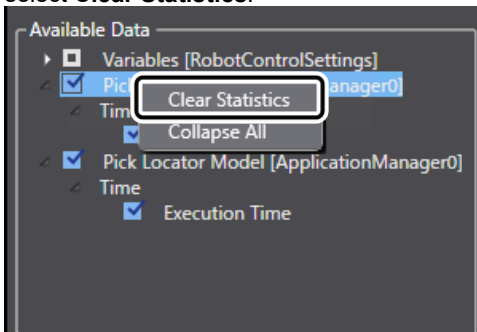
Parameter	Description
Processing Time (%)	The average processing time percentage of the total run time since the last restart or reset. (Processing is when the Process Manager is actively processing part or part target instances.)
Average Total Time (ms)	For the Process Manager group only. The average 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**

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

## Clear Statistics

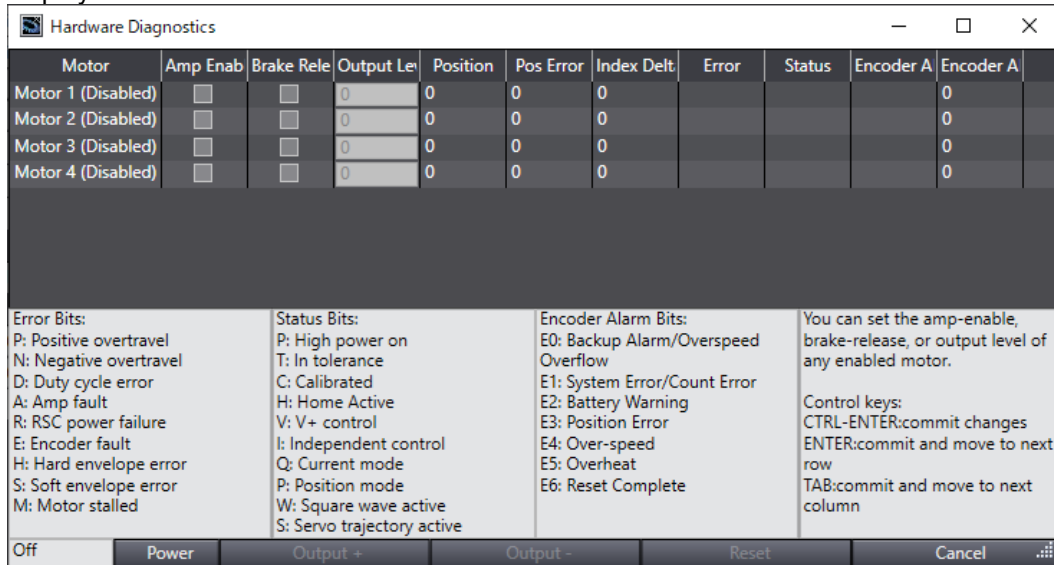
You can clear all Process Manager General Statistics. To clear statistics, right click the Process Manager item and select **Clear Statistics**.



# 10-3 Hardware Diagnostics

Hardware Diagnostics is used to check robot motor status when the Sysmac Studio goes online with the Robot Integrated CPU Unit. For example, when a robot's segmented display shows encoder error "E2", it 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.

Select **Control – Hardware Diagnostics** in the **Robot Settings**. The **Hardware Diagnostics** dialog box is displayed.



Item	Description
Amp Enable	Enables/disables an amplifier for the selected motor.
Brake Release	Enables/disables a brake release for the selected motor.
Output Level	Specifies a commanded torque, which is used to test the operation of the selected motor. The range is from -32767 to 32767.
Position	Displays the current position of the selected motor by encoder counts.
Pos Error	Displays the position error of the selected motor by encoder counts.
Index Delta	Displays a change from the previous latched zero index and the most recent latched zero index of the selected motor by encoder counts. Note that it 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 over-travel N: Negative over-travel 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: 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 current power state is shown in the status field.
Output +	DAC output controls.

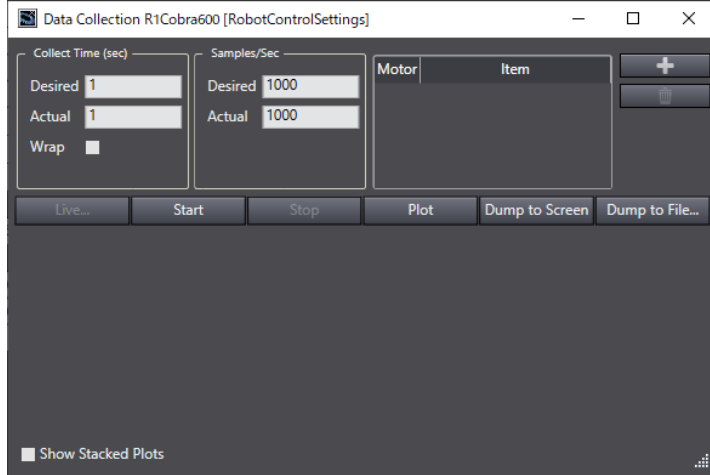
Item	Description
Output -	Click the <b>Output +</b> button to increase the DAC output to the selected motor. Click the <b>Output -</b> button to decrease the DAC output to the selected motor.

# 10-4 Data Collection

You can log the robot system data by using the data collection function.

Data Collection can be used to view, store, and plot various robot system data while online with the Robot Integrated CPU Unit. You can acquire up to 8 data items to the memory limit of the controller's data buffer, at the maximum of 8 kHz sampling rate.

Select **Control – Data Collection** in the **Robot Settings**. The window shown below is displayed. The Data Collection function is not available in the EMULATION mode.



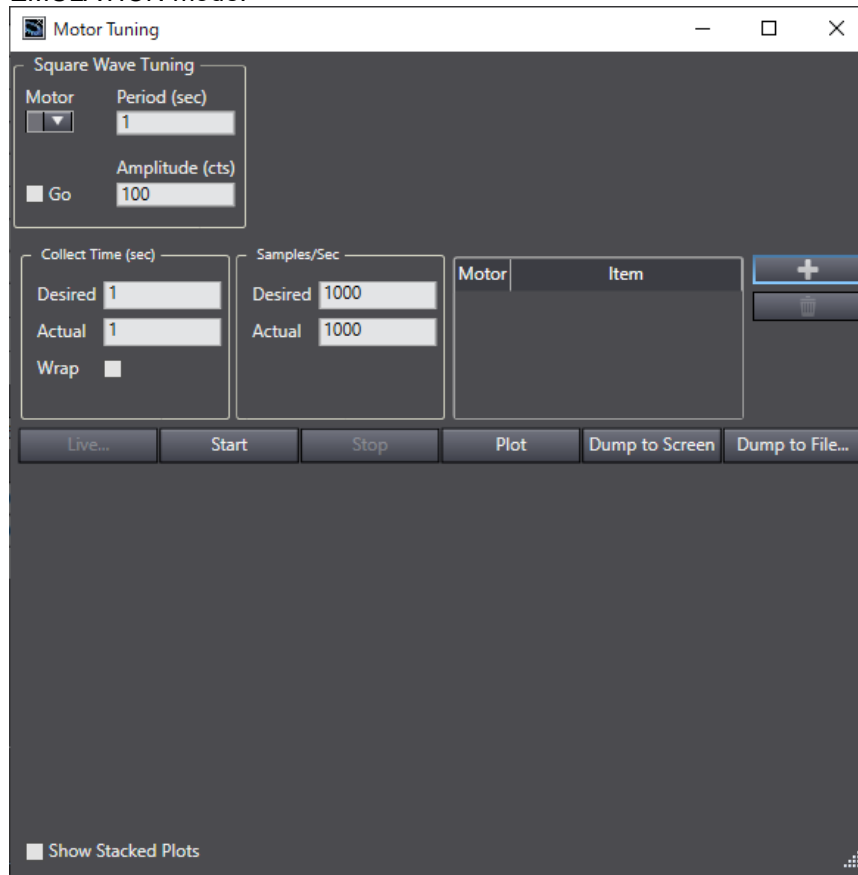
Item	Description
Collect Time (sec)	Specifies the data collection time in seconds. The default value is 1.
Samples/sec	Specifies the data collection frequency in samples per second. The default value is 1000.
Add.../Remove	<p>Add or remove motor items to examine with these buttons. Clicking the <b>Add</b> button will display the <b>Add Items to Collect</b> dialog box.</p> <p>Enter an Absolute Address or select <b>Opcode</b> and select an available data item from the drop-down list. Then select the motor(s) from which you want to collect data.</p>
Live	Displays a window that shows the real-time data being collected.
Start	Click this button to start a data collection. The data collection will continue until either the <b>Stop</b> button is clicked, or the specified data collecting time is up.
Stop	Click this button to stop a data collection. If the specified data collecting time has already passed, this button is disabled.
Plot	Click this button to plot the collected data. The progress bar appears while the data is being processed. After the processing has completed, the data is plotted on the graph located at the lower part of the <b>Data Collection</b> dialog box.

Item	Description
Dump to Screen	Displays the collected data in the Data Dump window in text-file format.
Dump to File...	Displays the <b>Save As</b> dialog box used for saving the collected data to a text file. You can display or edit the text file later.

# 10-5 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 responses works same as Data Collection. Refer to *10-4 Data Collection* for details.

Selecting **Motor Tuning** will display the following window. The Motor Tuning function is not available in the EMULATION mode.



Item	Description
Motor	Specifies the motor that will receive the square wave positioning command.
Period (sec)	Specifies the length of square wave in seconds.
Amplitude (cts)	Specifies the amplitude of square wave in servo counts.
Go	Turns ON/OFF the square wave positioning command to the specified motor.

# 10-6 Searching and Replacing

This section describes searching and replacing functions of the Application Manager. Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for basic functions and operation procedures of searching and replacing. Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual* (Cat. No. W595) for function operations of searching and replacing of *RobotControlSettings*, the subdevice of the Robot Integrated CPU Unit.

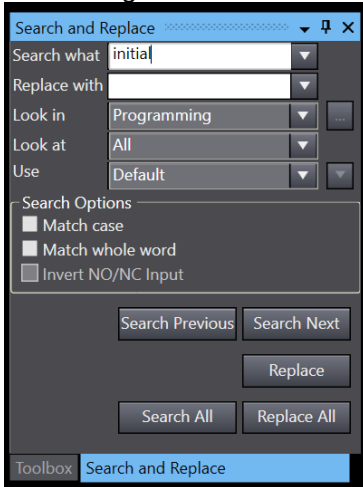
## 10-6-1 Search and Replace Targets and Setting Items

### Search and Replace Targets

You can search and replace character strings displayed in the C# program editor.

### Setting Items

The settings in the Search and Replace Pane are explained below.



Setting item	Description
Search what	<ul style="list-style-type: none"> <li>Enter a search string.</li> <li>You can select from previous search strings in the list.</li> </ul>
Replace with	<ul style="list-style-type: none"> <li>Enter the string to replace the search string with.</li> <li>You can select from previous search strings in the list. You cannot use wildcard characters. (If you try to use them, they are treated as normal text strings.)</li> </ul>
Look in	Specify the range to search. You can select from the following. <b>Programming:</b> C# program <b>Checked elements:</b> The item that is selected in the <b>Select search and replace scope</b> Dialog Box is searched. <b>Current View:</b> The current view is searched.
Look at	Specify the items to search. You can search for text strings in the following item of the Application Manager. <b>All:</b> Searches all text strings
Use	Specify if you want to use wildcard characters. <b>Default:</b> Do not use wildcard characters. <b>Wildcard:</b> Use wildcard characters.







# Appendices

---

The appendices provide the option settings, error messages lists, and other supplemental information for the body of this manual.

# A-1 Precautions on Use of Version Control

This section describes the precautions common to all devices and precautions for devices that require attention for safe use of project version control of the application manager and the Robot Integrated CPU Unit.

Refer to the *Sysmac Studio Project Version Control Function Operation Manual* (Cat. No. W589) for precautions and specifications of other devices.

## A-3-1 Precautions Common to All Devices

Precautions that are common to all devices are given below.

- You cannot import password-protected projects to the version control system. Disable the password protection before you import the project.
- If you develop a project with multiple developers, all involved should use the Sysmac Studio with the same language settings.
- You cannot merge changes to data in the Multiview Explorer, except for the following data. This data will be always overwritten by the contents of either the source or target of merge.
  - Controller's data in **Programming** and lower-level folders
  - HMI's data in Page and lower-level folders
  - **Program** and lower-level folders in **Programming** of the application manager
- User names and access levels and passwords and default user information of the robot system operation authority verifications are not version controlled. To use the robot system operation authority verifications, set the user names and access levels and passwords and default user information when using.

## A-3-2 Robot Integrated CPU Unit

Observe the following precautions when you control the versions of projects that include the Robot Integrated CPU Unit.

### Use

You cannot merge changes to the sub-device of the Robot Integrated CPU Unit, *RobotControlSettings*. This data will be always overwritten by the contents of either the source or target of merge.

If you develop Controller programs with multiple developers, allow one supervisor of the project development to edit the *RobotControlSettings*. You may not be able to merge the changes made to the *RobotControlSettings* as intended if the *RobotControlSettings* is edited by multiple developers.

## A-3-3 Application Manager

Observe the following precautions when you control the versions of projects that include the application manager.

### Use

The application manager can merge changes only in C# programs under **Program** in **Programming**. This data will be always overwritten by the contents of either the source or target of merge.

If you develop Controller programs with multiple developers, allow one supervisor of the project development to edit anything other than C# programs under **Program** in **Programming**. Otherwise, the changes may not be merged as intended.



**OMRON Corporation Industrial Automation Company**  
Kyoto, JAPAN

Contact: [www.ia.omron.com](http://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 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 ASIA PACIFIC PTE. LTD.**

No. 438A Alexandra Road # 05-05/08 (Lobby 2),  
Alexandra Technopark,  
Singapore 119967  
Tel: (65) 6835-3011/Fax: (65) 6835-2711

**OMRON (CHINA) CO., LTD.**

Room 2211, Bank of China Tower,  
200 Yin Cheng Zhong Road,  
PuDong New Area, Shanghai, 200120, China  
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

**Authorized Distributor:**

© OMRON Corporation 2020 All Rights Reserved.  
In the interest of product improvement,  
specifications are subject to change without notice.

**Cat. No. W621-E1-01**

0820