Automation Software

Sysmac Studio

Robot Integrated System Building Function with IPC Application Controller Operation Manual

SYSMAC-SE2□□□
SYSMAC-SE200D-64

always in control

W621-E1-07

NOTE

1. 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.
2. 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.
3. 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, Visual Basic, and Microsoft Edge 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.
- AMD Radeon™ is the 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.
- JavaScript® is a trademark or registered trademark of Oracle Corporation and its subsidiaries and affiliates in the United States and other countries.
- EPLAN and EPLAN Electric P8 are registered trademarks of EPLAN GmbH & Co.KG.

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

Microsoft product screen shots used with permission from Microsoft.

Introduction

Thank you for purchasing the Sysmac Studio Automation Software.
This manual contains information that is necessary to configure Robot Integrated System using Sysmac Studio Robot Integrated CPU Unit. 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.

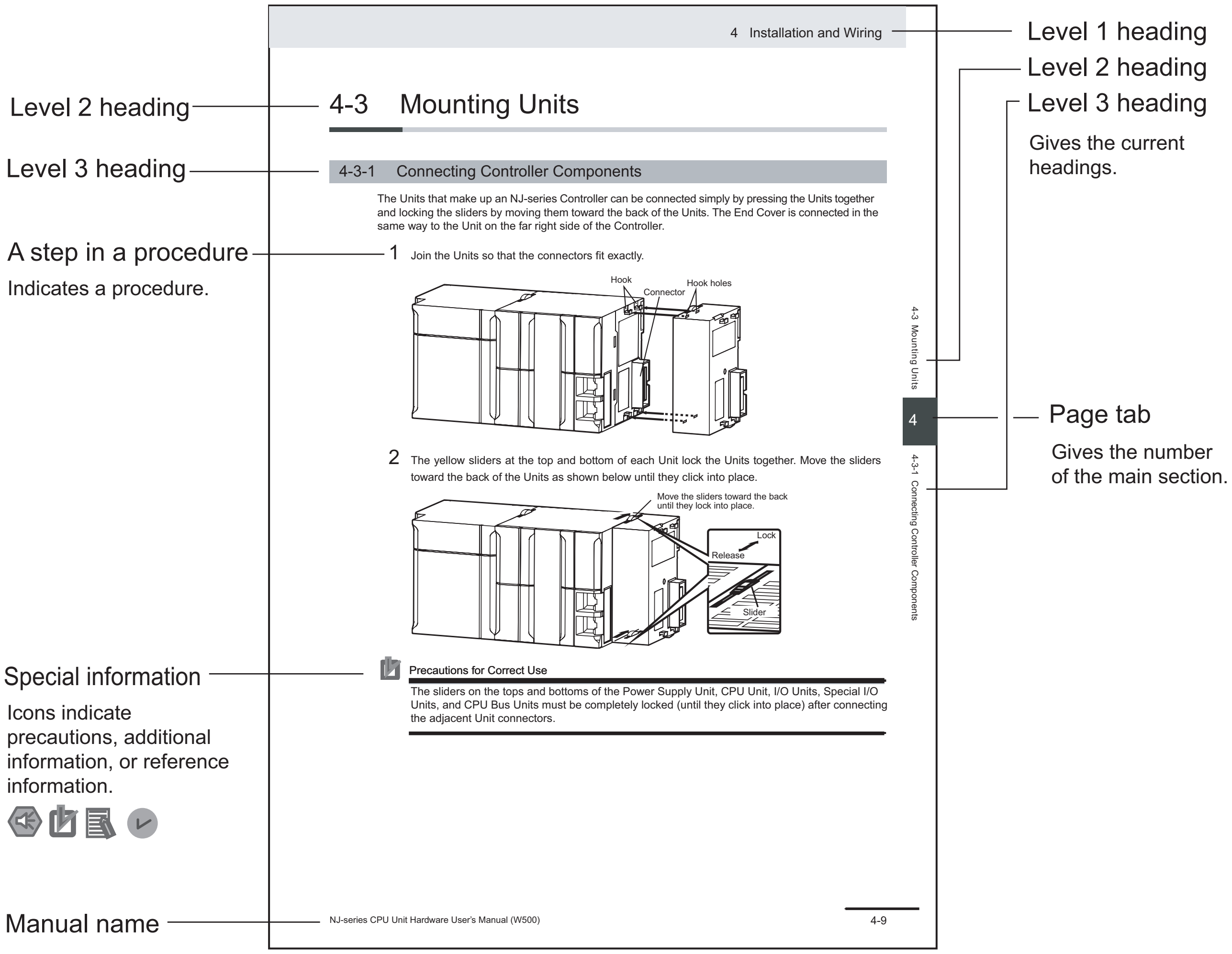
- Sysmac Studio Standard Edition
- NJ501-R□□□ CPU Unit

Part of the specifications and restrictions for the products are given in other manuals.
Refer to *Related Manuals* on page 18.

Manual Structure

Page Structure

The following page structure is used in this manual.



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

Special Information

Special information in this manual is classified as follows:

Precautions for Safe Use

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

Precautions for Correct Use

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

Additional Information

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

Version Information

Information on differences in specifications and functionality 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

CONTENTS

Section 7 Debugging the Robot Integrated System

Section 8 Application Sample Wizard

Section 9 Maintenance

Section 10 Other Functions

Appendices

Index

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.

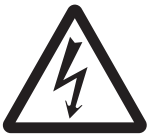
Safety Precautions

Definition of Precautionary Information

The following notation is used in this manual to provide precautions required to ensure safe usage of the Sysmac Studio. 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.

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

Symbols

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

WARNING

⚠ 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 (Cat. No. 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.

When more than one software application included in the Sysmac Studio or IPC Application Controller is simultaneously connected online to the CPU Unit that controls a robot integrated system, do not perform the following operations at the same time. The robots controlled by the CPU Unit may not operate as intended.

- Changing the settings of devices
- Online debug
- Teaching robots

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.

To prevent computer viruses, install antivirus software on a computer where you use this software. Make sure to keep the antivirus software updated.

Keep your computer's OS updated to avoid security risks caused by a vulnerability in the OS.

Always use the highest version of this software to add new features, increase operability, and enhance security.

Manage usernames and passwords for this software carefully to protect them from unauthorized uses.

Set up a firewall (E.g., disabling unused communication ports, limiting communication hosts, etc.) on a network for a control system and devices to separate them from other IT networks. Make sure to connect to the control system inside the firewall.

Use a virtual private network (VPN) for remote access to a control system and devices from this software.

Cautions

⚠ 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/.

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 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 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 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 about the Sysmac Studio project version control function and its operating procedures.	Provides an introduction to the Sysmac Studio project version control function along with its installation method, basic operations, execution method for the main functions, and other information.
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.

Manual name	Cat. No.	Model numbers	Application	Description
NJ/NX-series CPU Unit Software User's Manual	W501	NX701-□□□□ NX502-□□□□ 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-□□□□ NX502-□□□□ 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.
eV+3 User's Manual	I651	NJ501-R□□□	Operating the OMRON robot with the V+ program.	Describes the V+ language to control the OMRON robots.
eV+3 Keyword Reference Manual	I652	NJ501-R□□□	Operating the OMRON robot with the V+ program.	Describes V+ keywords that are used in the V+ language.
Robot Safety Guide	I590	RL4-□□□□□□□ RS4-□□□□□□□ RL6-□□□□□□□ RX3-□□□□□□□ RX4-□□□□□□□	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 OMRON robot with a teaching pendant.	Describes the setup, operation, and user maintenance for the Teaching Pendant T20.

Terminology

The following describes the terms used in this manual.

Term	Description
IPC Application Controller	A PC-based controller with the ACE (Automation Control Environment) software package installed to manage multiple OMRON robots and recipes controlled by the Robot Integrated CPU Unit. It can perform image processing by using an image sensor.
Robot Integrated CPU Unit	A CPU Unit that supports control function for the OMRON robot with the NJ-series CPU Unit.
Robot subdevice	Subdevice that manages robot control functions of the Robot Integrated CPU Unit. Displayed in the Multiview Explorer device list in Sysmac Studio. It includes controller settings, robot settings, V+ programs, and V+ variables of Robot Control Function Modules.
RobotControlSettings	A robot subdevice. Shown as RobotControlSettings in Sysmac Studio's Multiview Explorer.
V+ language	A programming language for OMRON robot control.
V+ program	A control program written in the V+ language.
V+ variable	Variable used in a V+ program.
sequence control program	A control program written in IEC 61131-3 language including the motion control.
Robot Control Function Module	Software to perform robot control that is installed in the Robot Integrated CPU Unit.
IO EndEffector	Device for picking, placing, or material applying, attached on a robot's tip.
Emulation mode	Mode for a robot simulation on Sysmac Studio using the Robot Integrated CPU Unit. Select **Emulation Mode** to open a project.
Pack Manager	An application that controls automated pick-and-place systems with cameras, conveyor belts, and robots based on the data configured in Sysmac Studio. It runs on the IPC Application Controller to control the process and recipe data.
Robot Vision Manager	An application that processes images captured by cameras based on the data configured in Sysmac Studio. It runs on the IPC Application Controller.
Application Manager	Device that manages settings and programs of an IPC Application Controller, which controls a robot integrated system, and necessary data and settings for the 3D simulation function. Refer to *Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618)* about Application Manager functions for 3D simulation.
Cameras	Camera settings for a robot integrated system. Following devices are included: Virtual Camera, an interface for accessing to images acquired by a camera from a vision tool; Emulation Camera, which handles the stored image data; Visualizer Capture Device, which handles image data that captured the 3D shape data in the 3D Visualizer; Basler Camera and Sentech Camera, the settings for physical cameras; Custom Device, which processes the image data from an external camera or a vision system with C# scripts.
Feeder	A device used in a robot integrated system to feed or dispense parts to/from a robot.

Term	Description
Process	Pack Manager elements. It includes the following objects: Part, Part Buffer, Part Target, Belt, Pallet, and Process Manager. The Part object defines an object to be picked and processed. The Part Buffer object defines an object that is an overflow area where parts can be temporarily stored. The Part Target object defines an object that is a possible destination for a part. The Belt object defines a physical conveyor belt used by the system. The Process Manager associates these elements to define pick-and-place motions. In addition, it has Allocation Script and Vision Refinement Station. The Allocation Script object is used to create and edit custom part-allocation programs. The Vision Refinement Station is used to refine the part to gripper orientation for improved placement accuracy.
Vision Tools	Settings to correct images from a physical camera, Emulation Camera, or Visualizer Capture Device, to inspect, and to judge the images. Vision Tools are categorized into Finder Tools, Inspection Tools, Reader Tools, Calculation Tools, Image Process Tools, and Custom Tools.
Virtual equipment model	An equipment model created on Sysmac Studio to execute a 3D simulation. A Virtual equipment model is made up of 3D shape data for a Mechanical Component, a Virtual Part Detection Sensor, and a part, and the settings and scripts that define the operations of their 3D shape data.
3D shape data	Data that represent a virtual equipment model's shape, size, and position in the 3D space, which created on Sysmac Studio. 3D shape data is made up of CAD data, boxes, and cylinders.
Box	A box-shaped object that becomes the basis of 3D shape data. This object is provided as standard in the Sysmac Studio, and has height, width, depth, and color settings.
Cylinder	A cylinder-shaped object to generate the 3D shape data. This object is provided as standard in the Sysmac Studio, and has radius, height, and color settings.
CAD data	3D CAD data for equipment or a part, which becomes the basis of 3D shape data. Use third party 3D CAD software to create CAD data. You can load CAD data files with a .stp, .step, .igs, .iges, .usd, .usda, .usdc, or .usdz extension.
C# program	A program written in C# and runs on IPC Application Controllers. It can call Application Manager functions.
C# variable object	An object that enables C# programs, which run separately, to share data.
Event Log	Displays a log of events that have occurred since the Robot Integrated System was started. Events are categorized as Error, Warning, and Information and can be used for troubleshooting and diagnostics.
V+ Jog Control	A function to operate an actual robot or a robot in a simulation in the jog mode.
Task Status Control	A group of functions to monitor and control all the robots in a robot integrated system. It allows you to display and change a connection to a robot controller, power status, monitoring speed, and execution statuses of Vision Sequences, C# programs, Process Managers, V+ programs,and more.
Application Sample Wizard	A function to create templates of pick-and-place type robot integrated systems including Robot Integrated CPU Units and IPC Application Controllers. You can easily create templates only following the wizard: select peripherals for the system and configure settings. Pack Manager Sample and Robot Vision Manager Sample are available.
Pack Manager Sample	You can create pick-and place system templates for packaging machines with a Pack Manager running on the IPC Application Controller.
Robot Vision Manager Sample	You can create pick-and place system templates for packaging machines with a Robot Vision Manager on the IPC Application Controller and V+ program running in the Robot Integrated CPU Unit.

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-E1-07

↑ Revision code

Revision code	Date	Revised content
01	August 2020	Original production
02	April 2021	Revisions for an upgrade to Sysmac Studio version 1.45.
03	April 2022	Revisions for an upgrade to Sysmac Studio version 1.49.
04	October 2022	Revisions for an upgrade to Sysmac Studio version 1.52.
05	April 2023	Revisions for an upgrade to Sysmac Studio version 1.54.
06	October 2024	Revisions for addition of safety precautions for operation when more than one software application included in the Sysmac Studio or IPC Application Controller is simultaneously connected online.
07	April 2025	Revisions for addition of terminology.

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 Wizards

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

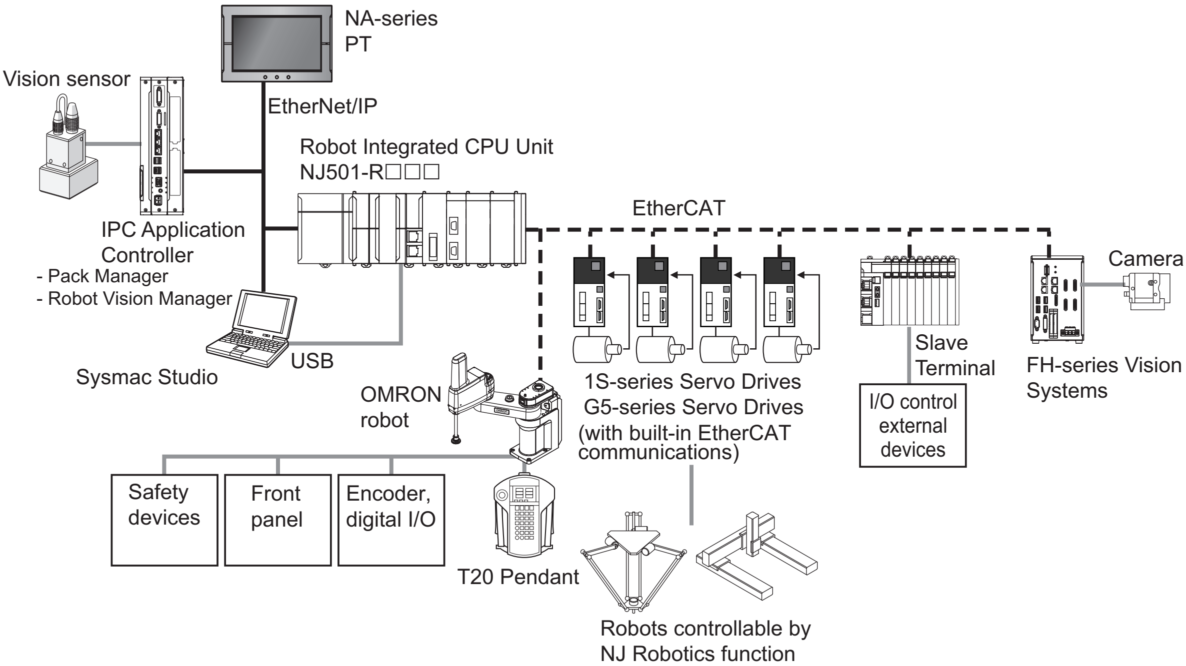
The Sysmac Studio can be used for setting up and programming not only the controller to control peripheral devices, but also peripheral devices as conveyors, robots, Feeders, sensors, and HMIs. You can use only the Sysmac Studio to set up system configuration devices, instead of using separate Support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

Component	Description
Vision sensor	Cameras that capture part 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 Sysmac 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 Option License

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 conform to the specifications of the Sysmac Studio. Refer to *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. For details of the installation procedure, refer to *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*.

3

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. Pack Manager application sample wizard to create a programless system.	IPC Application Controller	Refer to *3-6 Flow of System Basic Building Controlled with IPC Application Controller* on page 3-20 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 *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*

3-2 Coordinate Systems of Robot Integrated System

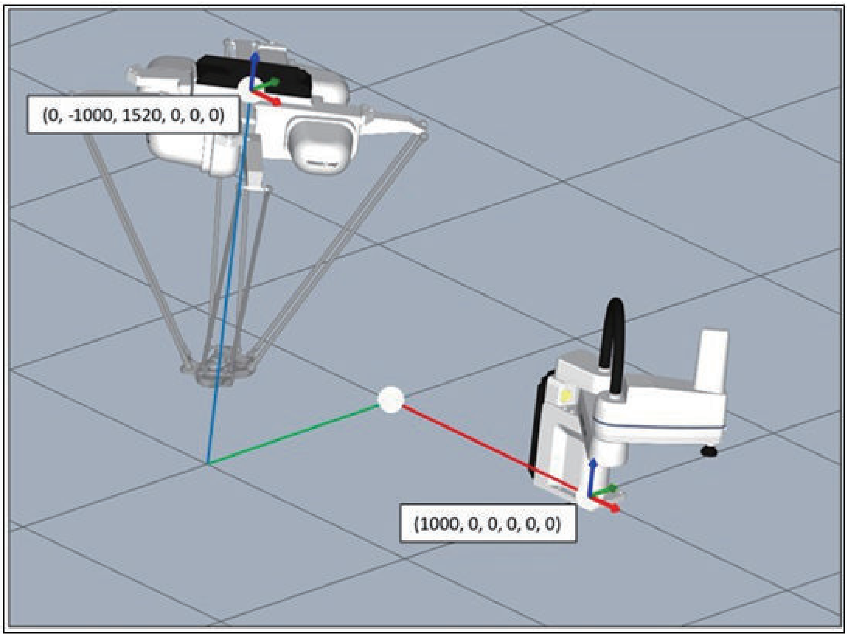
In Sysmac Studio, define the positions and directions using multiple coordinate systems for each motion element of the robot integrated system. The coordinate systems used in Sysmac Studio are as follows.

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.	*3-2-1 Workspace Coordinate System* on page 3-3
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.	*3-2-2 Robot - World Coordinate System* on page 3-4
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.	*3-2-3 Robot - Joint Coordinate System* on page 3-5
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.	*3-2-4 Robot - Tool Coordinate System* on page 3-6
Belt Coordinate System	This is the coordinate system describing the direction and orientation of a conveyor belt. Each robot may have a different reference to belt coordinates for the same physical conveyor belt.	*3-2-5 Belt Coordinate System* on page 3-6
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. The origin is located at the center of the field of view.	*3-2-6 Camera Coordinate System* on page 3-8

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.

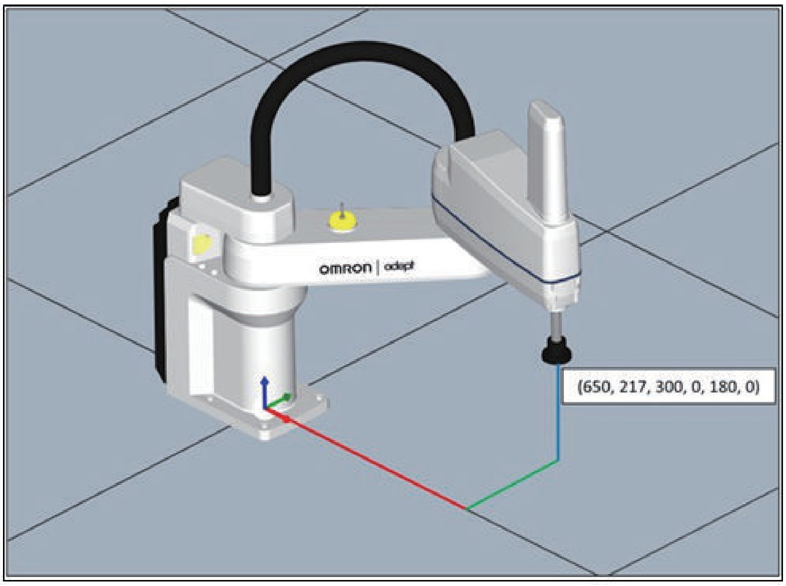
Workspace coordinates are primarily used for positioning robots and other features in the workspace. Allocation of belt-relative Part and Part Target instances during run time depends on the relative position of robots along a process belt object, therefore robot positions cannot be changed while a Process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

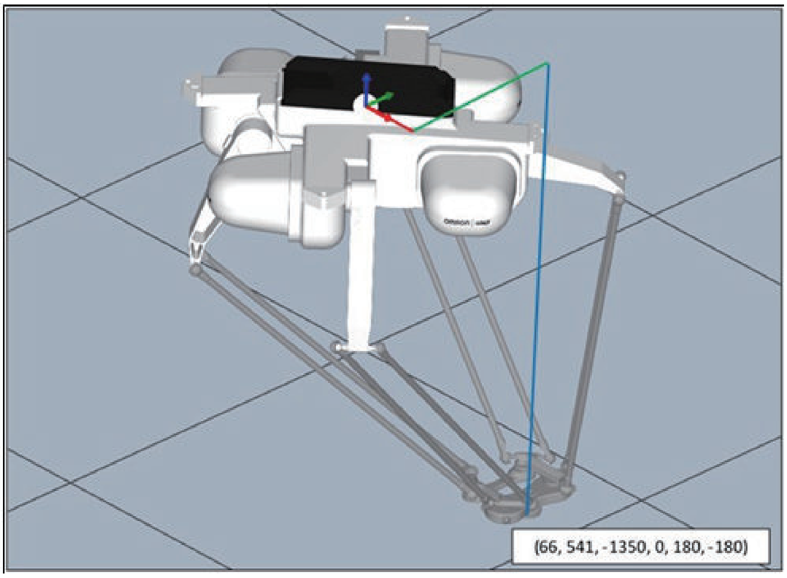


3-2-2 Robot - World Coordinate System

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.The tool coordinate system is used to define the position of tool tips. Whenever a position is taught or motion executed, it is usually done with respect to this coordinate system.



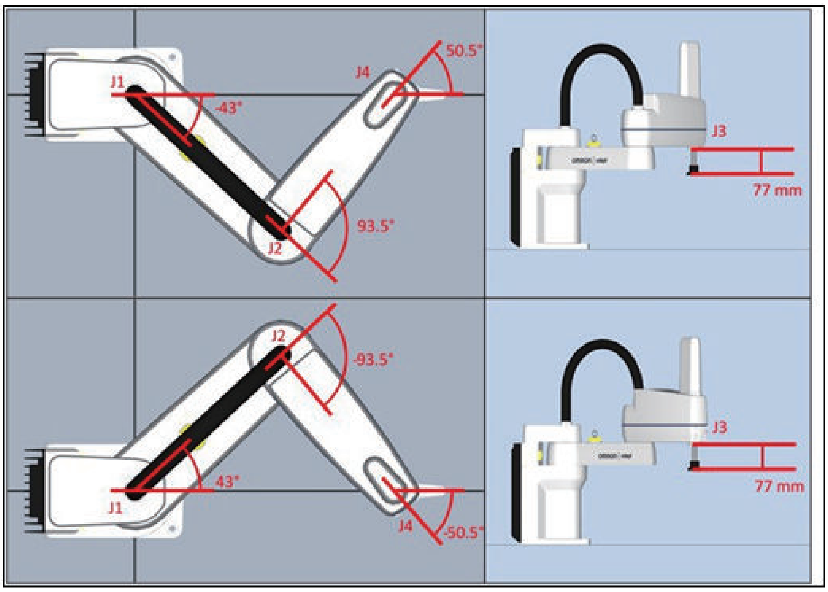


3-2-3 Robot - Joint Coordinate System

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

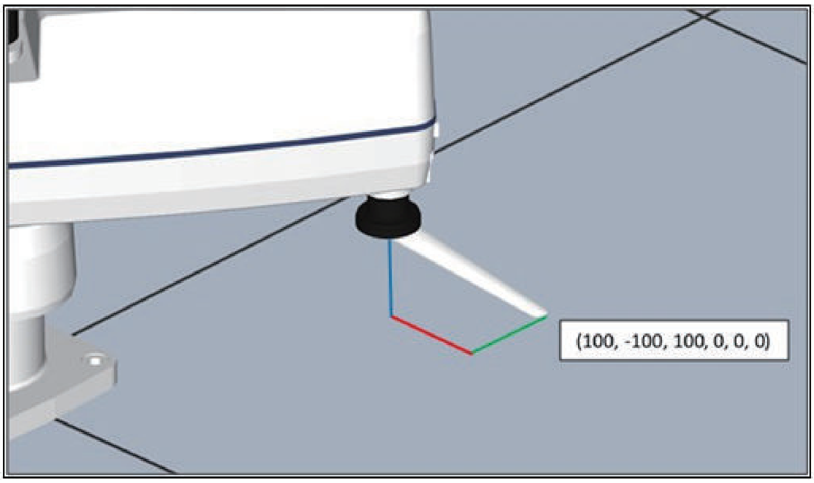
A location based on joint coordinates instead of world coordinates is called a precision point. It is useful in cases where one orientation would cause the robot to strike an obstacle.
A precision point guarantees that a robot will always move to the correct orientation. This is most commonly seen in 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.

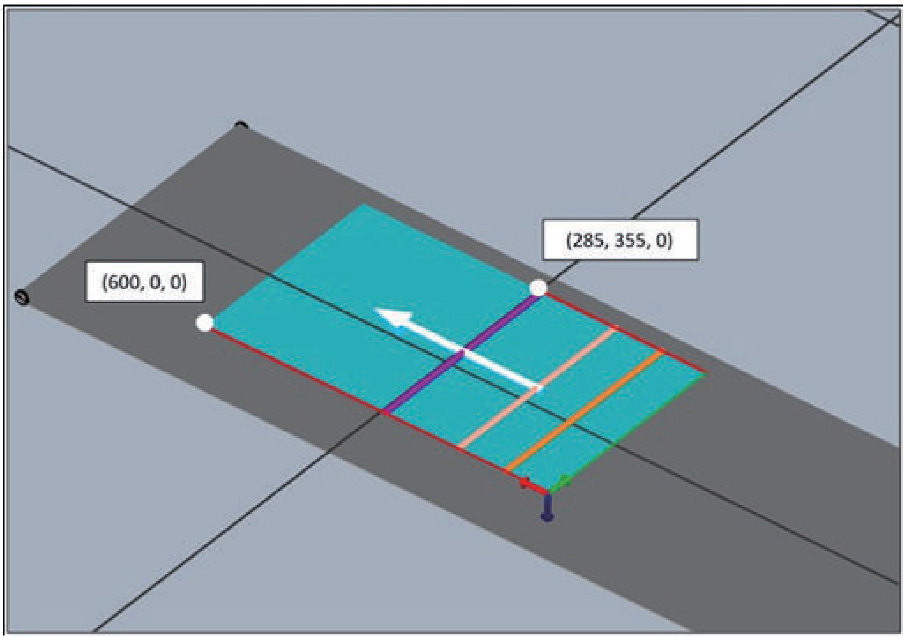
3-2-4 Robot - Tool Coordinate System

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.

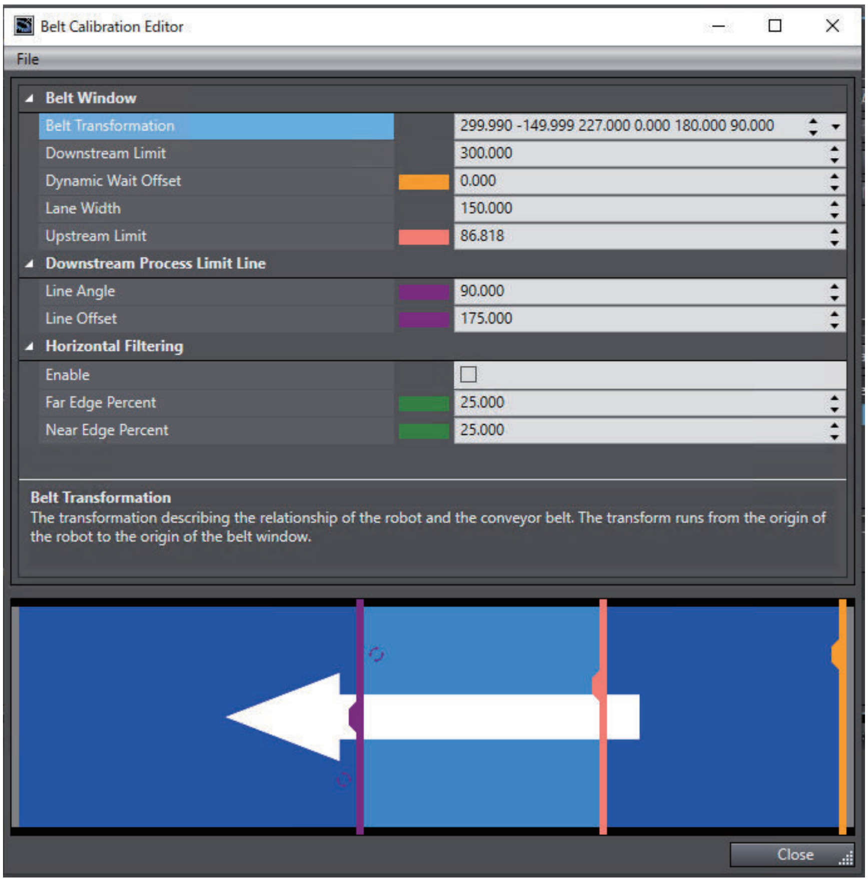


3-2-5 Belt Coordinate System

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* on page 5-21 for more information.



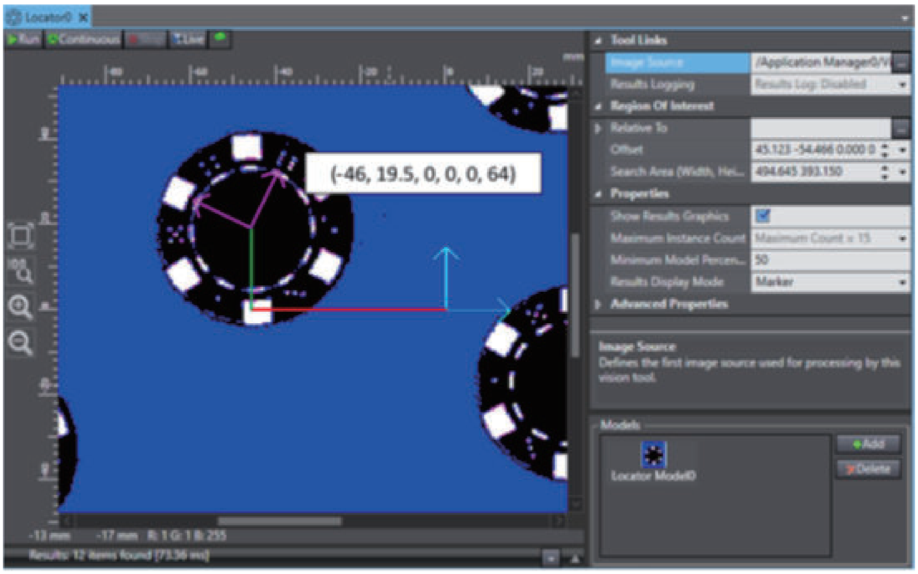
Additional Information

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 Types 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. It 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 Robot-to-Belt Calibration* on page 3-9
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 Robot-to-Camera Calibration* on page 3-10
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 Robot-to-Latch Calibration* on page 3-11

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.

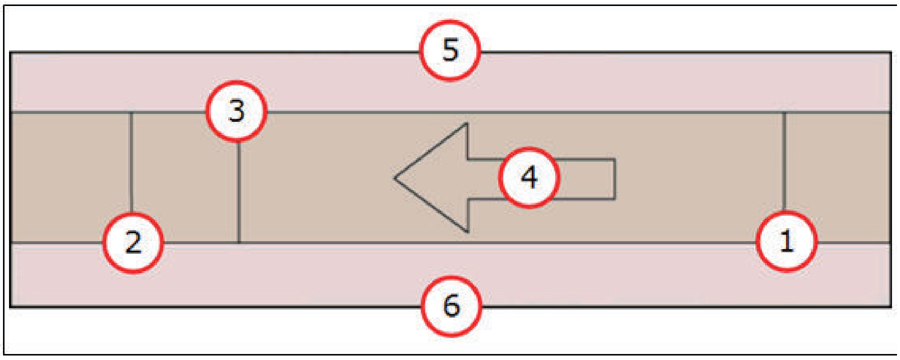
It is important to ensure the calibration target does not move relative to the belt while advancing the belt.

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* on page 5-21 for more information.



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

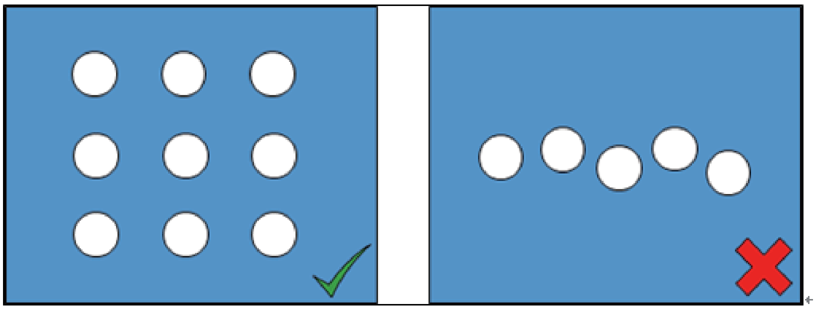
3-3-3 Robot-to-Camera Calibration

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* on page 5-49 for more information.
The process of calibrating the camera is dependent on the type of pick application in which it is involved. Generally, there are three categories with which the application could be associated:

- Fixed-mounted

- Arm-mounted
- Belt-relative

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



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

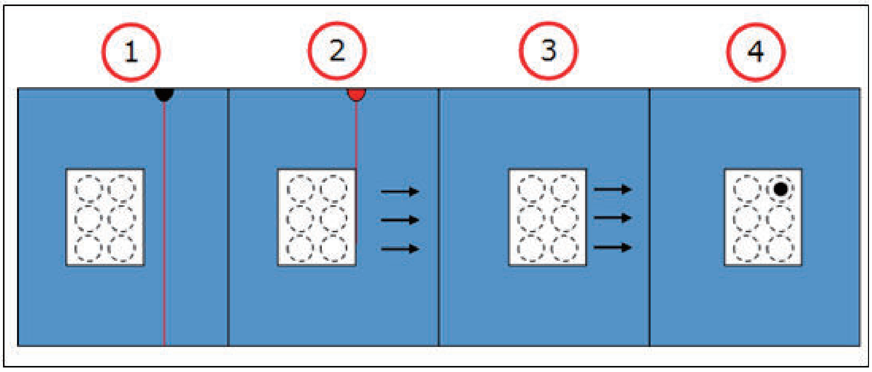
If the application has a belt, the calibration is effectively the same, but it must be executed in two phases since the robot cannot reach the camera field of view. In the first phase, the targets are placed on the belt underneath the camera and their positions are recorded with respect to the camera.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

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.

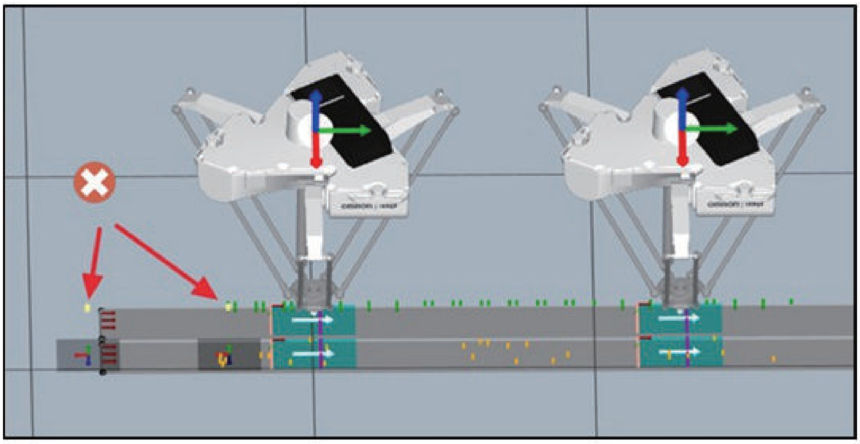
In the following figure, the blue field represents the belt with the arrows indicating the direction of belt travel. In the following figure, the blue field represents the belt with the arrows indicating the direction of belt travel.



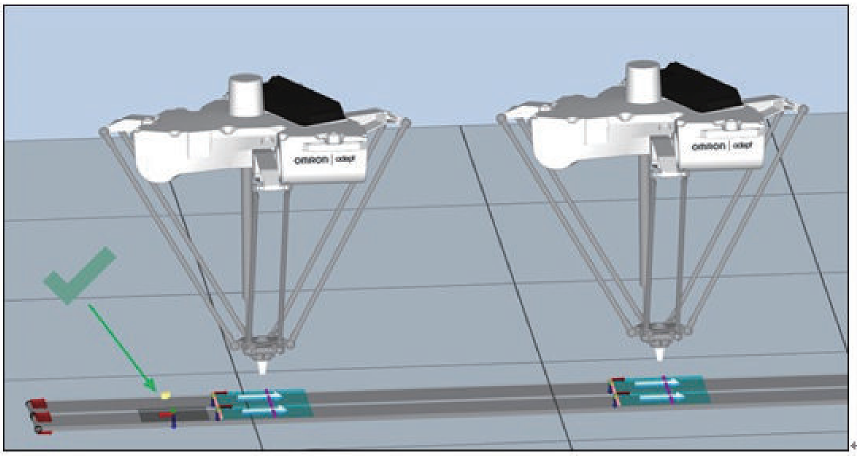
Item	Description
1	The pallet is positioned upstream of the latch sensor and the belt is not in motion.
2	The belt is advanced and the pallet is detected by the latch sensor. The belt encoder position at this position is recorded.
3	The belt is advanced to a position where the pallet is within the robot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

3-4-1 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 *3-4-6 Joint-Interpolated Motion VS. Straight-Line Motion* on page 3-17 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.

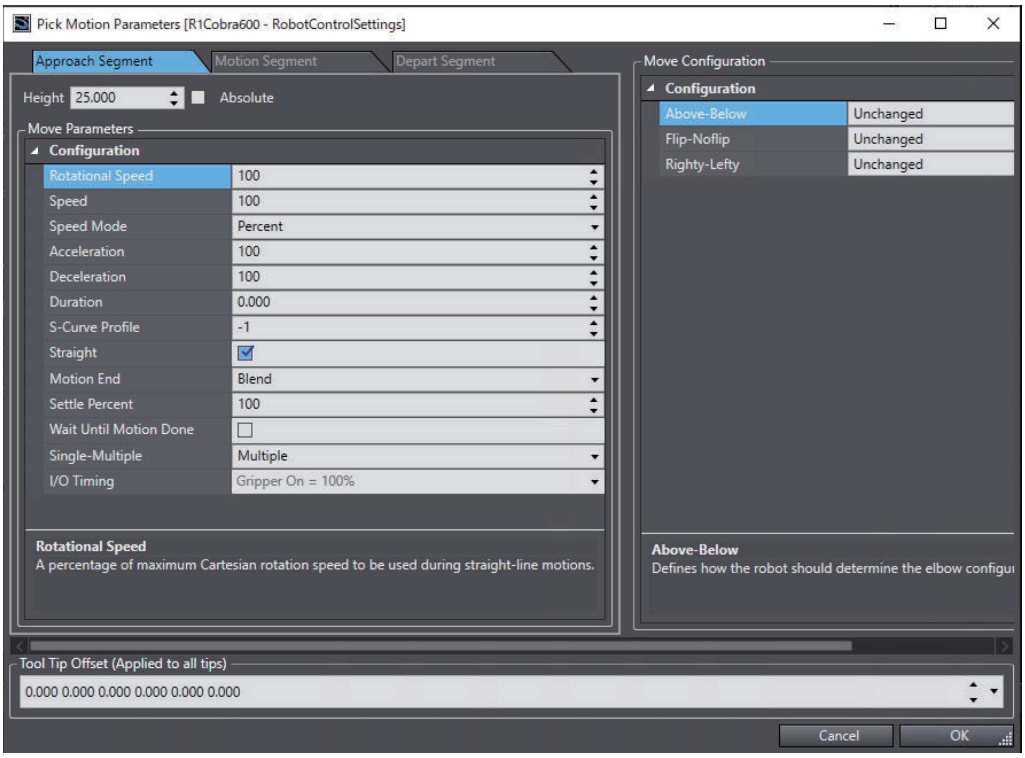
Another case would be a relatively high payload and inertia coupled with tight positioning tolerances. A high deceleration rate may cause overshoot and increase settling time.

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

3-4-2 Approach and Depart

Approach and depart heights are used to make sure that the robot approaches and departs from a location without running into any other objects or obstructions in the robot envelope. Approaches and departs are always parallel to the Z-axis of the tool coordinate system.

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



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

Notice that all the motion parameters that apply to a motion to a location also can be applied to approach and depart motions. This allows you to move at optimum speed to the approach height above a location, then move more slowly when actually acquiring or placing the part, and finally depart quickly if the application requires this.

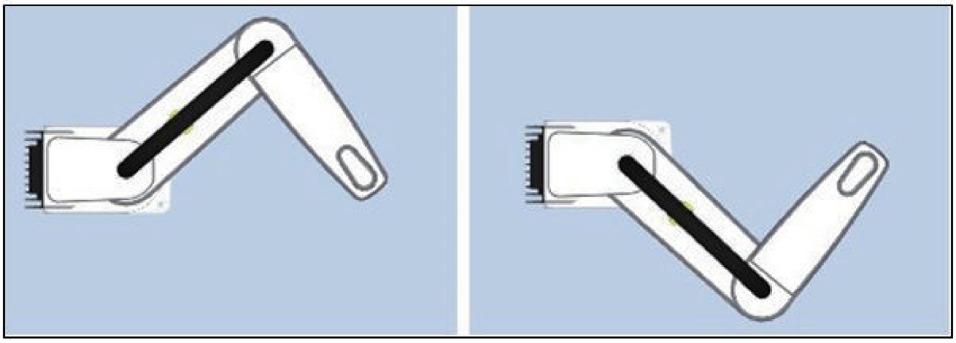
3-4-3 Arm Configuration

Another motion characteristic that you can control is the configuration of the robot arm when moving to a location. However, configuration options apply only to specific types of robots.

For example, the lefty/righty option applies to SCARA-type robots (such as the Cobra robots), but the above/below option does not apply to those robots.

The arm configuration can be specified in the **Configuration Items** of the Process Manager object or in a V+ program. Refer to *Configuration Items* on page 5-148 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.



3-4-4 Location Precision

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

3-4-5 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 *3-4-6 Joint-Interpolated Motion VS. Straight-Line Motion* on page 3-17 for more information.

3-4-6 Joint-Interpolated Motion VS. Straight-Line Motion

The path a robot takes when moving from one location to another can be either a joint-interpolated motion or a straight-line motion. A joint-interpolated motion moves each joint at a constant speed except during the acceleration/deceleration phases. Refer to *3-4-1 Speed, Acceleration, and Deceleration* on page 3-14 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.

3-4-7 Performance Considerations

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

Robot Mounting Considerations

The mounting surface should be smooth, flat and rigid. Vibration and flexing of the mounting surface will degrade performance. Therefore, it is recommended that you carefully follow the robot-mounting procedures described in your robot user's 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* on page 5-21 for more information.

3-5-1 Indexing VS. Tracking Conveyors

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

Indexing Conveyors

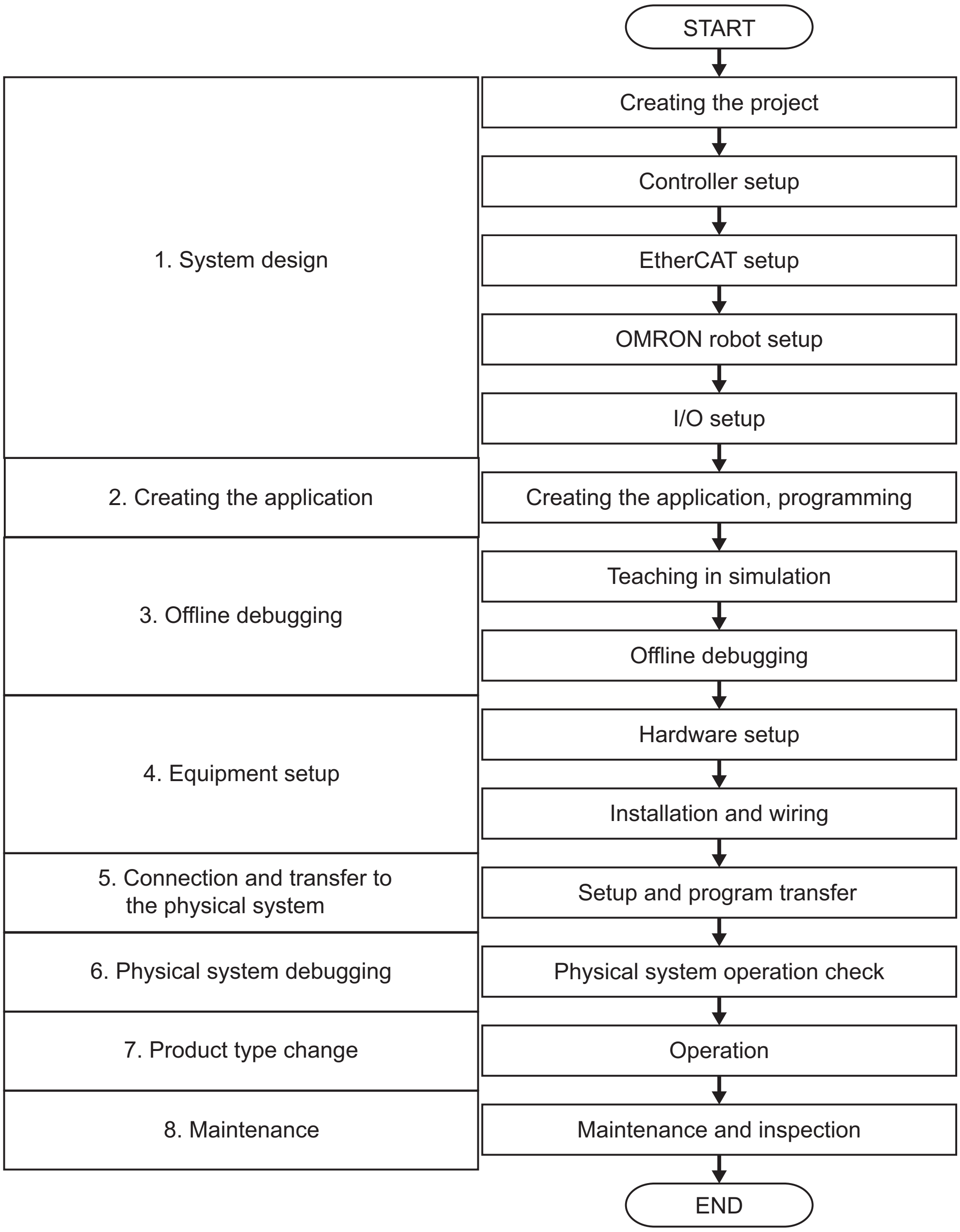
Indexing conveyor systems are configured as either non-vision or vision. With a non-vision indexing system, the part must be in the same location each time the belt stops. In a vision-equipped indexing system, a fixed-mount camera takes a picture when the belt stops and the robot accesses any objects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

Tracking Conveyors

In a tracking conveyor system, the belt moves continuously and the robot tracks parts until the speed and location of the robot gripper match those of a part on the belt. The robot then accesses the part.
Tracking conveyors must be equipped with an encoder that reports the movement of the belt and distance moved to the 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. 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 the 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	Create a project for the Robot Integrated CPU Unit and adds a robot to the EtherCAT network.	*4-2-1 Creating a New Project* on page 4-4
		The Robot Integrated CPU Unit setup	Sets up the Robot Integrated CPU Unit for the robot control.	*Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*
		EtherCAT setup	Adds the EtherCAT slaves including the OMRON robots on the EtherCAT, and makes the settings.	*Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*
		OMRON robot setup	Makes the robot control parameter settings.	*4-4 Robot Settings* on page 4-10
		I/O setup	Makes the setting to control I/O from a user program.	*Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*
2	Application creation	Application creation	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.	*Section 5 IPC Application Controller Settings* on page 5-1 *Section 8 Application Sample Wizard* on page 8-1
		Programming	Creates an application program by using C# language as required.	*5-11 C# Program* on page 5-192
3	Offline Debugging	Simulation of the Robot Integrated System	Simulates operations of the Robot Integrated System that includes robots and conveyor belts on the computer, and perform the debugging. Checks takt time and interference between a robot and peripheral equipment as required.	*Section 7 Debugging the Robot Integrated System* on page 7-1

Proce-dure	Item		Description	Reference
4	Equip-ment Set-up	Hardware set-ting, installation and wiring	Makes the settings for hardware switches on the equipment, installation and wiring for I/O and the network.	Refer to the *NJ-series Robot Integrated CPU Unit User's Manual (Cat. No. O037)*, *IPC Application Controller User's Manual (Cat. No. I632)*, and manuals for the OMRON robots and EtherCAT slaves you use.
5	Connec-tion and Transfer to the Physical System	Connection and transfer to the Robot Integrated CPU Unit	Connects to the Robot Integrated CPU Unit online, and transfer the Controller settings and programs.	*Section 6 On-line Connections to IPC Applica-tion Controllers* on page 6-1
		Connection and transfer to the IPC Application Controller	Connects to the IPC Application Controller online, and transfer the Application Manager setting.	
6	Physical System Debug-ging	Operation check on the physical system	Performs operations such as camera calibration, belt calibration with the robot, and robot teaching on the physical system to check the system opera-tion.	*Section 7 De-bugging the Ro-bot Integrated System* on page 7-1
7	Product Type Change	Setup for prod-uct type change	Sets up the IPC Application Controller and the Con-troller according to the product type to change to, create programs, and simulate the operation of the system.	*5-7-6 Recipe Manager* on page 5-81
8	Mainte-nance	Troubleshooting	If any device error occurs, checks the error infor-mation on the IPC Application Controller and takes an action according to the error.	*Section 9 Main-tenance* on page 9-1

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.

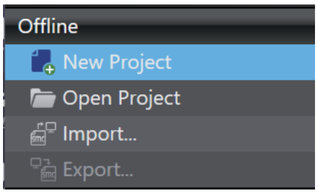
4-1 Starting and Exiting the Sysmac Studio

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)*.

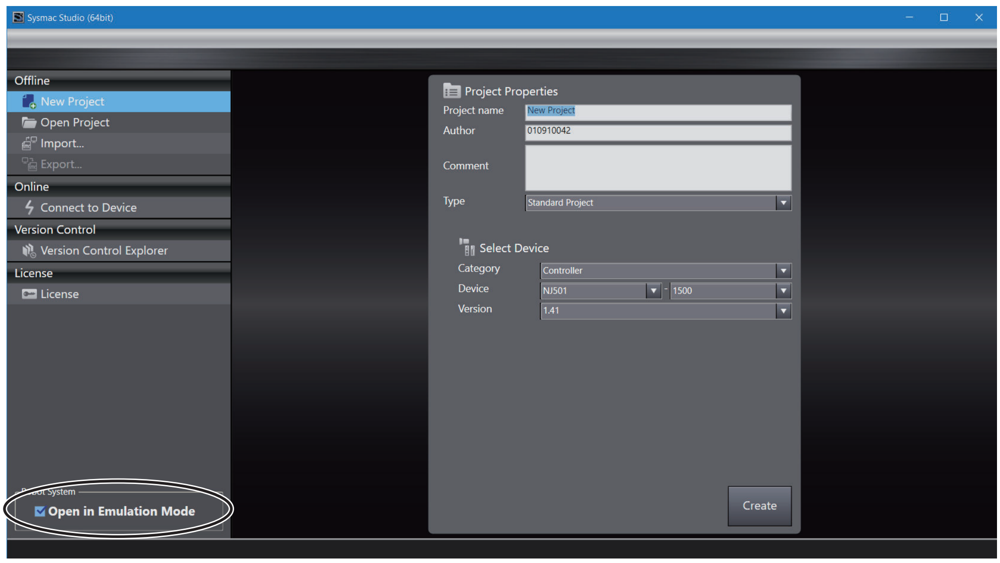
4-1-1 Starting the Sysmac Studio

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

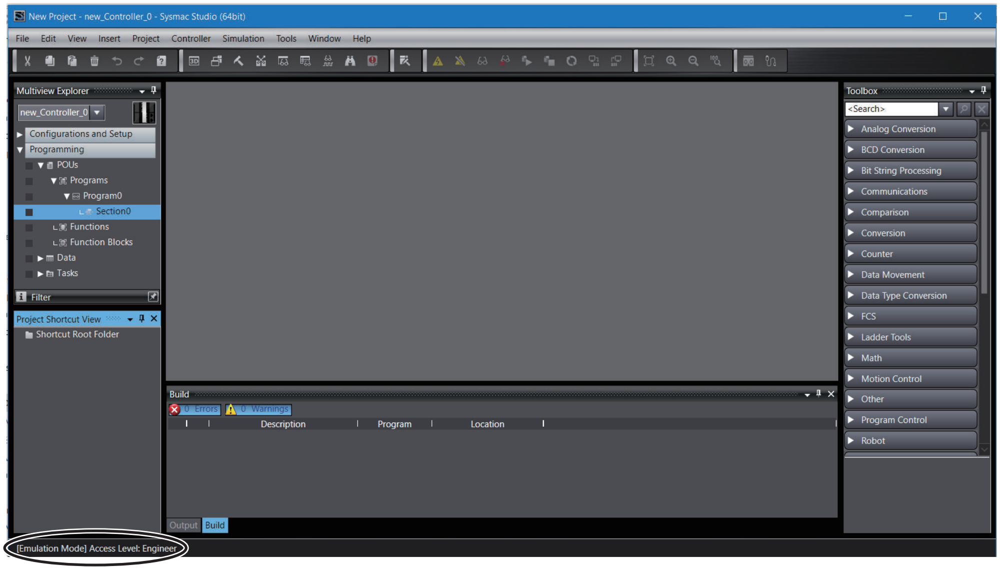
1 Start the Sysmac Studio and click **New Project** on the start page. If you open an existing project, click **Open Project** to select the target project.



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.

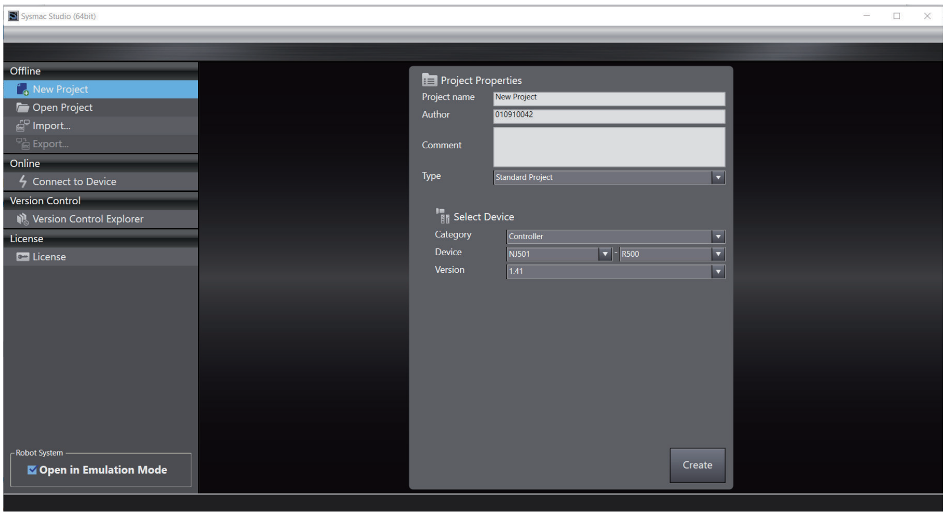
- Click the × button on the right end of the title bar.
- Select **File** – **Close** from the main menu.

4-2 Creating a Project

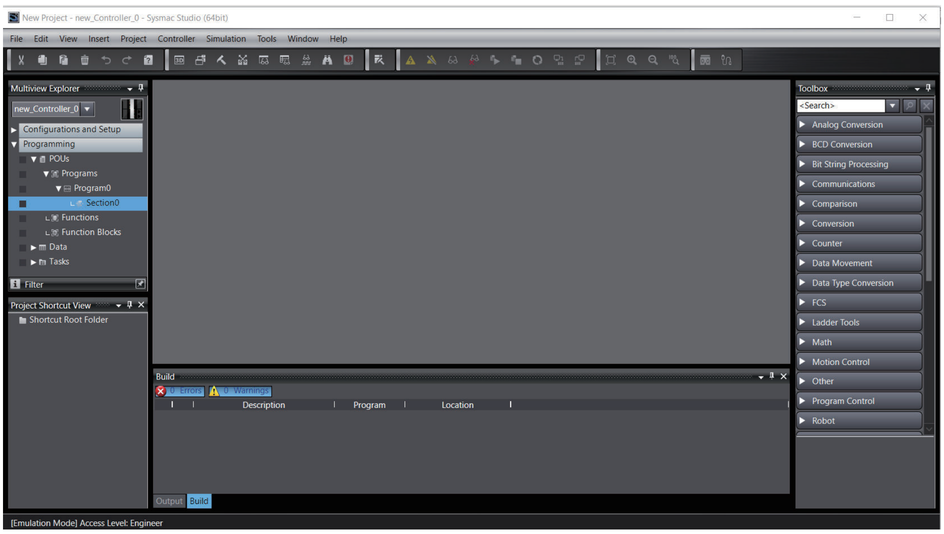
This section describes the procedure for creating a new project for a system that controls a robot with the IPC Application Controller. Create a new project for the Robot Integrated CPU Unit and register a robot to use.

4-2-1 Creating a New Project

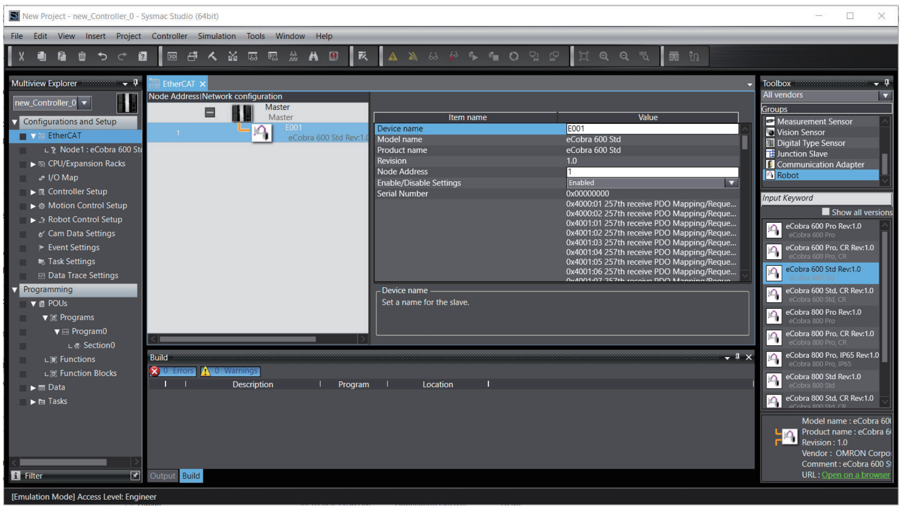
1 Select **New Project** on the Start menu of the Sysmac Studio. Then, select the Robot Integrated CPU Unit in **Select Device**.
In this example, select the Robot Integrated CPU Unit *NJ501-R500*.



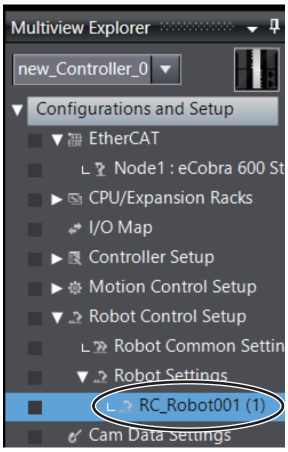
2 After entering data in the **Project Properties** dialog box, select **Open in Emulation Mode** check box and click the **Create** button.
A new project is now created.



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**.

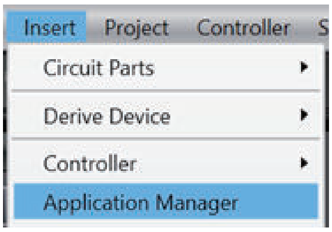


It is necessary to set the cycle of the primary periodic task to 2 ms or more in the task setting when you use a robot. Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for how to operate the task settings.

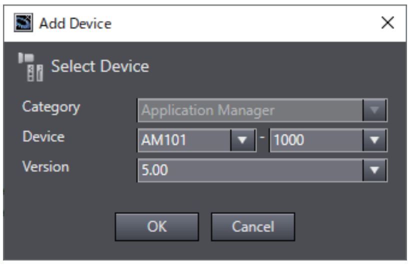
4 In the EtherCAT tab page, register the peripheral devices to connect to the EtherCAT network as required.
Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the registration procedure.

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 the Application Sample Wizard* on page 8-2 for using the application sample wizard.

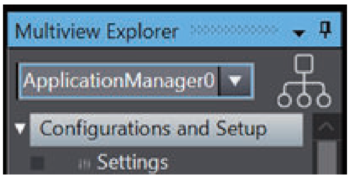
Select **Application Manager** from the **Insert** menu in order to register an empty Application Manager.



The **Add Device** dialog box is displayed.



Check the version of the ACE software package running on the IPC Application Controller, and set the Application Manager version of the Sysmac Studio in **Version**, in consultation with the following **Version Information**. You can check the version of ACE from the ACE toolbar or the Help menu.
The Application Manager is added.



Precautions for Correct Use

The Sysmac Studio's Application Manager project version must be the same as the Application Manager version of the ACE software package running on the IPC Application Controller. If not, you cannot go online with the IPC Application Controller.
Refer to the *Automation Control Environment (ACE) Version 4 User's Manual (I633)* for details on the connectable Application Manager versions.

Version Information

The Application Manager versions supported by ACE vary depending on the version of ACE. Refer to the following information for setting the Application Manager version of the Sysmac Studio.

		Application Manager versions supported by ACE				
ACE version	---	1.0	2.0	3.0	4.0	5.0
	4.0	○	×	×	×	×
	4.2	○	○	×	×	×
	4.3	○	○	○	×	×
	4.4	○	○	○	○	×
	4.5 to 4.7	○	○	○	○	○

○: Supported, ×: Not supported

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 Connection Setting for IPC Application Controller* on page 5-6
3D Visualization	Adds 3D Shape Data in the 3D Visualizer of the robot system.	*5-4 3D Visualization* on page 5-7
Robot Vision Manager	Adds the settings for using Robot Vision Manager.	*5-5 Robot Vision Manager* on page 5-15
Camera	Adds cameras used in Robot Systems. Virtual cameras are added and used instead of actual cameras in offline simulation.	*5-6 Camera* on page 5-47
Configuration	Customizes the application operations by settings of the Application Manager starting, Data Mapper, and recipe.	*5-7 Configuration* on page 5-72
Feeder	Adds feeders supplying parts, to the robot system.	*5-8 Feeder* on page 5-93
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 Process* on page 5-104
Vision Tool	Adds tools for image processing use in the Robot Vision Manager.	*5-10 Vision Tools* on page 5-182

● Programming

Item	Description	Reference
Program	Creates an application program by using C# language.	*5-11 C# Program* on page 5-192
Variable	Is defined when various objects of Application Manager and variables shared by multiple C# programs, are required.	*5-11-2 C# Variable Objects* on page 5-196

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.

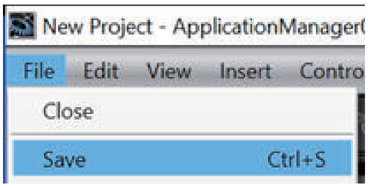
Setting Item	Setting Outline	Reference
Controller setting	Performs the Controller setting (operation settings and built-in EtherNet/IP settings) of the Robot Integrated CPU Unit.	*Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*
EtherCAT setting	Adds the EtherCAT slaves including the OMRON robots on the EtherCAT, and makes the settings.	*Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*
The Robot Integrated CPU Unit setup	Sets robot numbers of OMRON robots, robot names specified by user program, and the I/O control by the V+ program.	
Robot Subdevice Settings	Sets the Robot Control Function Module, the Controller Setup, and the Save Configuration.	
Robot Settings	Sets OMRON robots. You can set the option settings/parameters and obstacle settings for each axis that operates the robots.	*4-4-1 Setting Up IO EndEffectors* on page 4-10 *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*

Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* and the *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 *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for other robot settings.

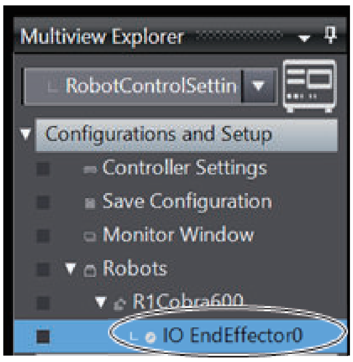
4-4-1 Setting Up IO EndEffectors

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.

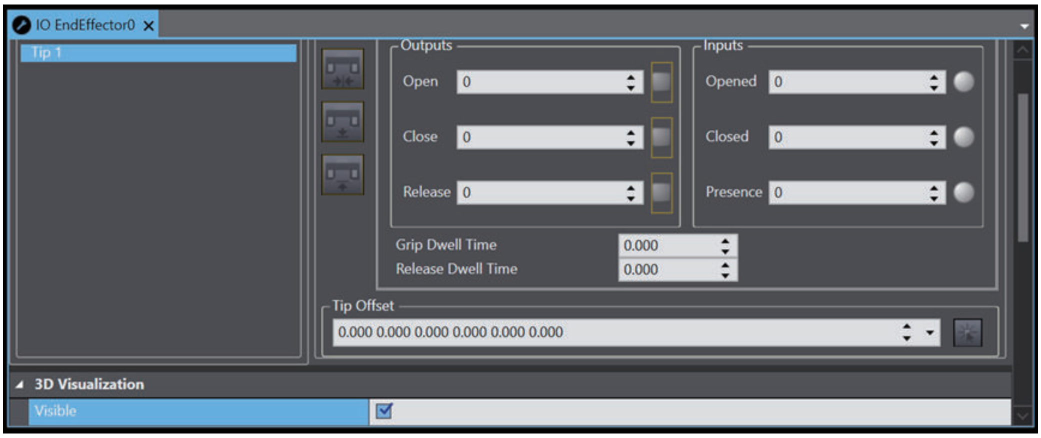
Displaying the End-Effector Tab Page

1 In the Multiview Explorer, select **RobotControlSettings** from the device list.

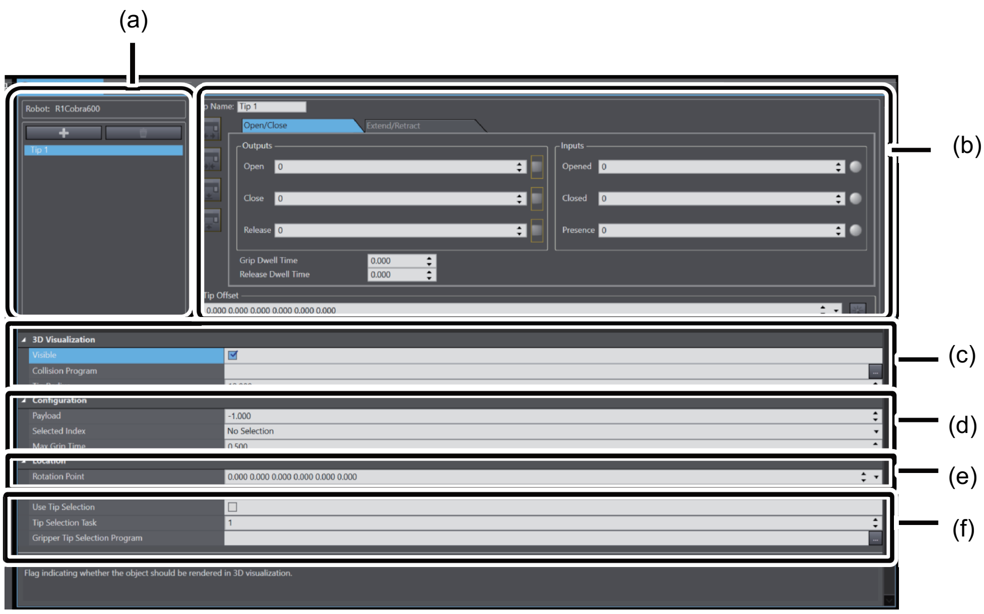
2 Double-click an End-Effector under **Configurations and Setup** − **Robot** in the Multiview Explorer. Or, right-click it and select **Edit** from the menu.
In this example, select **IO EndEffector0**.



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



Setup Tab Page for IO EndEffectors



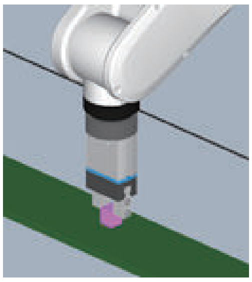
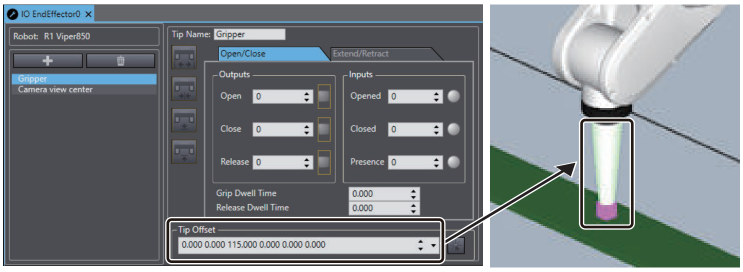
Item	Name	Description
(a)	**Add/Delete** buttons	Adds or deletes gripper tips.
	Tool tip list	Displays a list of defined gripper tips. For *Robot-to-Belt* or *Robot-to-Camera* calibrations, if a gripper has multiple tips defined, the entire tool (all tips) is used to pick up and release the part during the calibration procedure. The center line of the gripper is used as the reference tool in the calibration procedure. 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 *5-9-5 Process Manager* on page 5-127 for more information.

Item	Name	Description
(b)	**Outputs/Inputs**	Defines the **Open**/**Close** or **Extend**/**Retract** activation signals and **Opened**/**Closed** or **Extended**/**Retracted** 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.
	Open/Close tab	Click this tab to access the Open/Close signal settings for the selected tool tip.
	Extend/**Retract** tab	Click this tab to access the Extend/Retract signal settings for the selected tool tip.
	Open Tip/**Close Tip** buttons	Use the **Open Tip** and **Close Tip** buttons to send an open tip or close tip signal to the selected tool tip.
	Extend Tip/**Retract Tip** buttons	Use the **Extend Tip** and **Retract Tip** buttons to send an extend tip or retract tip 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 **Teach Tool Tip Wizard** button, which starts the **Tip Offset** wizard.
(c)	**Visible**	Enables the display of the IO EndEffector tool tip positions in the 3D Visualizer. Refer to *5-4 3D Visualization* on page 5-7 for more information.
	Collision Program	Select a program that is invoked when the 3D Visualizer detects a collision. Refer to *7-2-4 Collision Detection Settings* on page 7-19 for more information.
	Tip Radius	The radius of the tool tip when the IO EndEffector is drawn in the 3D Visualizer. Refer to *5-4 3D Visualization* on page 5-7 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.
	Max Grip Time	The maximum time (in seconds) allowed for grip verification and part presence check.
(e)	**Rotation Offset**	The offset of the axis that becomes the center of the rotation operation.

Item	Name	Description
(f)	**Use Tip Selection**	When it 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.

Additional Information

You can teach the center position of the gripper or field of view of the camera connected to the robot by using the tool offset for the tip of the end effector.
For example, when teaching the position where the gripper connected to the robot comes in contact with the target part, add a *Tip* to the robot's end effector and set the tip position of the gripper relative to the tip of the robot as the tool offset. In addition, you can perform teaching taking into consideration the coordinates of the gripper by selecting the *Tip* in offline teaching and snapping it to the part.

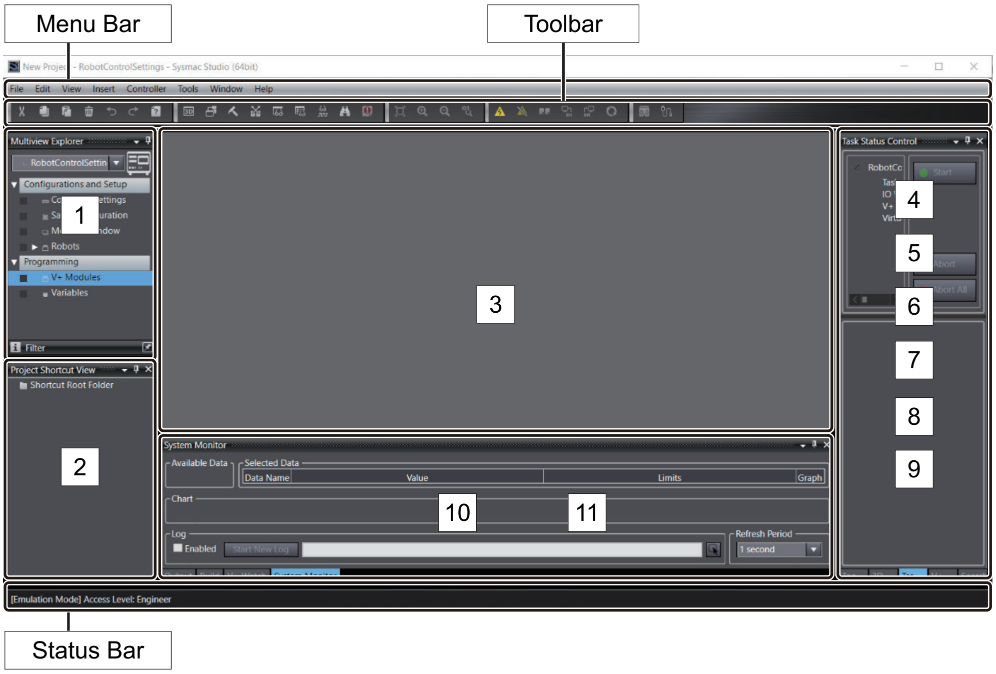
Image of an actual gripper	Image of the gripper when the tip position is set as the tool offset
	

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 window of Sysmac Studio.



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.

When opening the project, the on/off 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, which are displayed under Configurations and Setup and Programming in the Multiview Explorer, to access them easier.

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* on page 7-30 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* on page 7-4 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* on page 7-22 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* on page 7-43 for details.

4-5-10 V+ Watch Tab Page

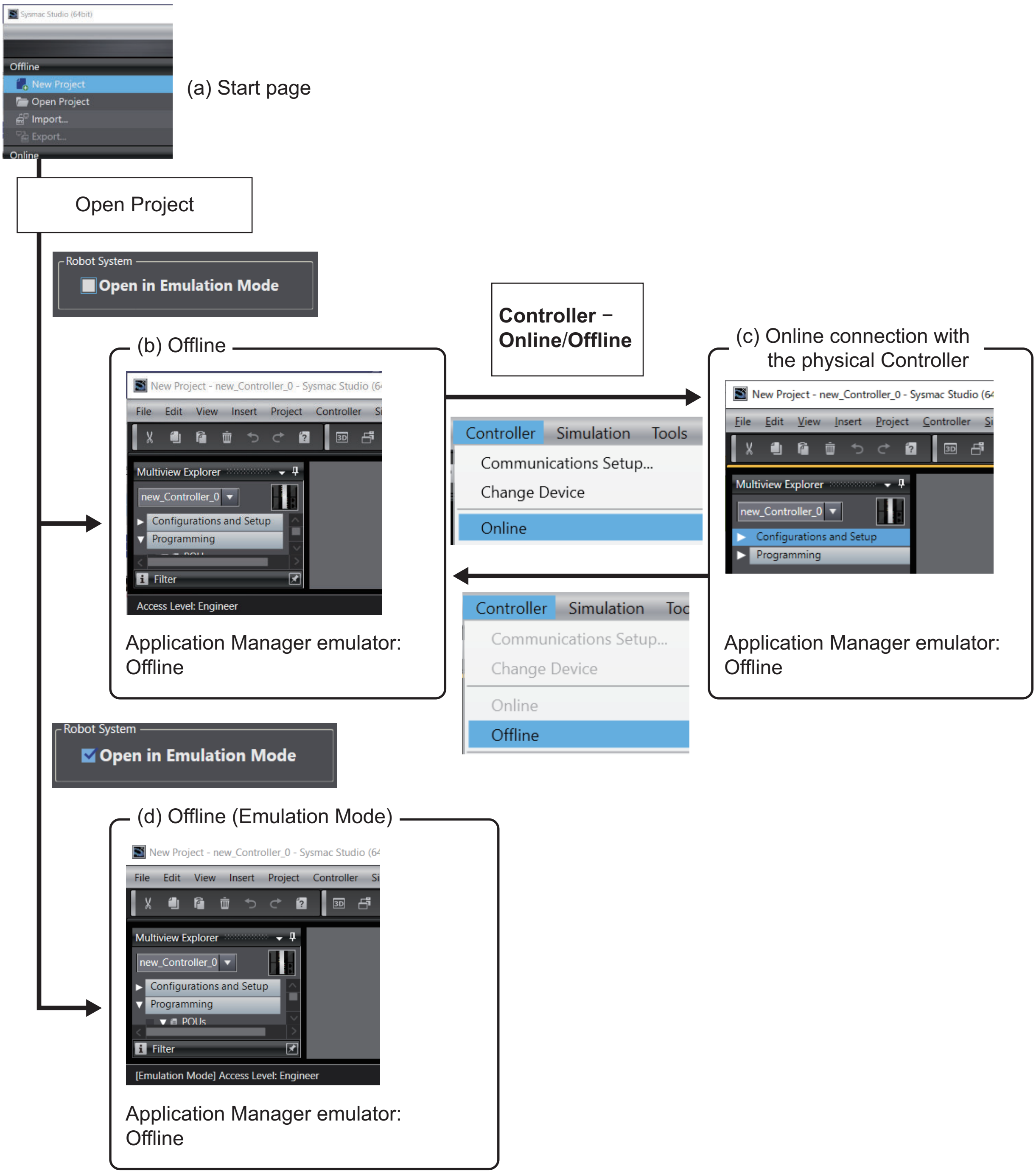
Current values of C# variable objects are displayed in this window. Refer to *7-7-1 Adding C# Variable Objects to V+ Watch Tab Page* on page 7-46 for details.

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* on page 10-4 for details.

4-6 Using Online and Offline Modes of the Sysmac Studio

The *Emulation mode* is available in a project of the Robot Integrated System which uses the IPC Application Controller and the Robot Integrated CPU Unit. In the mode, the Application Manager emulator and the Robot Control Function Module can be executed. In the *Emulation mode*, you can configure the Robot Control Function Module, edit and offline-debug V+ programs. Also you can configure and offline-debug the Application Manager.
The editable data and usable functions for the Robot Control Function Module and the Application Manager differ depending on the combination of the *mode* and the connection statuses (e.g. *Connection to the simulator* or *Online connection to the physical controller*).
Regarding the mode and the connection statuses, the state transition diagram of the Application Manager starting from the Start page is shown below.
Refer to *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 following table shows the operations and purposes available in each status.

	Status	Purpose	Editable data and available operation	Non-editable data	State transits to	State transited by
(a)	Start page (Project not opened)	Selecting a project	Selecting the project and choosing whether to open the project in the Emulation Mode	Project data	Offline	Opening the project with the **Open in Emulation Mode** check box cleared.
					Offline (Emulation mode)	Opening the project with the **Open in Emulation Mode** check box selected.

	Status	Purpose	Editable data and available operation	Non-editable data	State transits to	State transited by
(b)	Offline	Editing Configurations and Setup and Programming (#C programs, variables) of the Application Manager	Configurations and Setup and Programming (#C programs, variables) of the Application Manager	None	Online connection to the physical controller	Selecting **Controller** - **Online**, or pressing the Online button
(c)	Online connection to the physical controller	Transferring, executing, and monitoring an application Editing Configurations and Setup and Programming (#C programs, variables) of the Application Manager	Transferring, executing, and monitoring an application Configurations and Setup and Programming (#C programs, variables) of the Application Manager	None	Offline	Selecting **Controller** - **Offline**, or pressing the Offline button
(d)	Offline (Emulation mode)	Editing Configurations and Setup and Programming (#C programs, variables) of the Application Manager Offline-debugging*1	Editing Configurations and Setup and Programming (#C programs, variables) of the Application Manager Executing and monitoring an application	None	-	-

*1. The Robot Integrated CPU Unit must be connected to the simulator in advance to debug the Application Manager.

Additional Information

- A Robot Integrated System project can be debugged offline only in the Emulation mode.
- When the state is between *(b) Offline* and *(d) Offline (Emulation mode)*, you can change the state by selecting **Controller** - **Enable emulation mode**/ **Disable emulation mode**. It causes the project to once close and then open again.

5

IPC Application Controller Settings

This section describes the operation procedures of Sysmac Studio in the robot system which includes the IPC Application Controller. You can learn how to use functions of Application Manager, Pack Manager for robot system control, and Robot Vision Manager for camera image processing included in the IPC Application Controller.

5-1 Overview of Application Manager

Application Manager is the software runs on the IPC Application Controller.
It consists of *Pack Manager* which controls a pick-and place system, and *Robot Vision Manager* which processes images shot by the cameras connected to the IPC Application Controller. You can configure Application Manager on Sysmac Studio.
This section provides overviews of Pack Manager and Robot Vision Manager, and describes how to configure Application Manager setting items on Sysmac Studio.

5-1-1 Pack Manager

Overview

Pack Manager is the software that allows you to build a wide range of pick-and-place applications, from for basic pick-and-place cells to complex cells with multiple cameras, conveyors, and robots.
Pack Manager Application Sample Wizard enables you to build an application without writing program codes. You can create an application that requires more complex controls by customizing the default V + programs generated through the Application Sample Wizard.

Features

Pack Manager on the IPC Application Controller and the Robot Integrated CPU Unit control robots that connected to the Robot Integrated CPU Unit together. You do not have to write V+ programs to control robots.
The IPC Application Controller controls a cell, and the Robot Integrated CPU Unit controls a device for sequence control. Pack Manager can manage processes (e.g. Load control) of multiple Robot Integrated CPU Units and OMRON robots that connected to them.

5-1-2 Robot Vision Manager

Overview

Robot Vision Manager calculates the image data acquired by the sensor cameras connected to the IPC Application Controller and generates the outputs. It also configures and stores the calibration information of belts (conveyors), sensor cameras, etc. for pick-and-place applications, and automatically generates V+ programs run on the Robot Integrated CPU Unit.
The Robot Vision Manager Application Sample Wizard automatically generates the Robot Vision Manager settings and V+ programs to create an application. It makes a robot program development easier.

Features

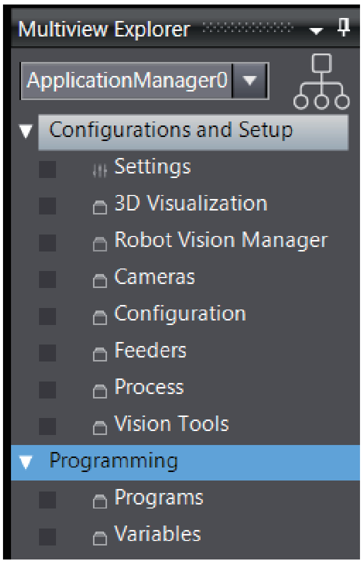
Robot Vision Manager controls robots connected to the Robot Integrated CPU Unit on it.
V+ programs to control cells and robots are necessary.

The Robot Integrated CPU Unit controls cells, robots, and devices for sequence controls.

5-2 Application Manager Settings

The following functions and setting items are available in an Application Manager.

Double-click an item in the Multiview Explorer to edit. To add detail items, select **Add** in the right-click menu.



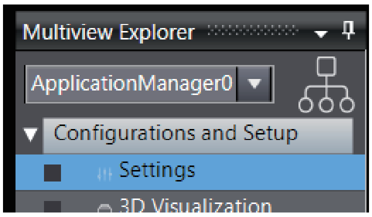
Item	Description	Reference
Settings	Sets the IP address and port number of the IPC Application Controller to be connected.	*5-3 Connection Setting for IPC Application Controller* on page 5-6
3D Visualiza-tion	Adds 3D Shape Data in the 3D Visualizer of the robot system.	*5-4 3D Visualization* on page 5-7
Robot Vision Manager	Adds the settings for using Robot Vision Manager.	*5-5 Robot Vision Manager* on page 5-15
Cameras	Adds cameras used in Robot Systems. Virtual cameras are added and used instead of actual cameras in offline simula-tion.	*5-6 Camera* on page 5-47
Configuration	Customizes the application operations by settings of the Ap-plication Manager starting, Data Mapper, and recipe.	*5-7 Configuration* on page 5-72
Feeders	Adds feeders supplying parts, to the robot system.	*5-8 Feeder* on page 5-93
Process	Makes various settings, related to the process, to pick a part and place it at the target position based on the specified exe-cution order.	*5-9 Process* on page 5-104
Vision Tools	Adds tools for image processing use in the Robot Vision Man-ager.	*5-10 Vision Tools* on page 5-182

5-3 Connection Setting for IPC Application Controller

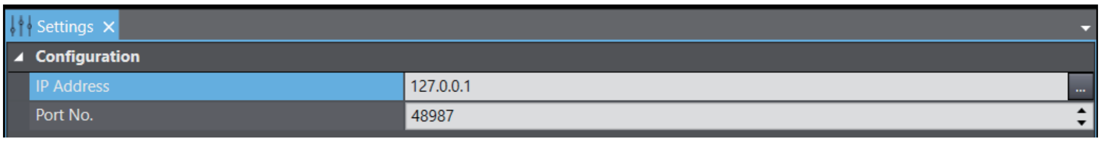
Configure settings to connect online with the IPC Application Controller.
Enter the IP address and port number of the IPC Application Controller to be connected.

5-3-1 Displaying the Settings Tab Page

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



The **Settings** tab page is displayed.



5-3-2 Setting Items

The setting items are as follows.

Item	Description
IP Address	Enter the IP address of the IPC Application Controller to be connected.
Port No.	Select the port number for connection.

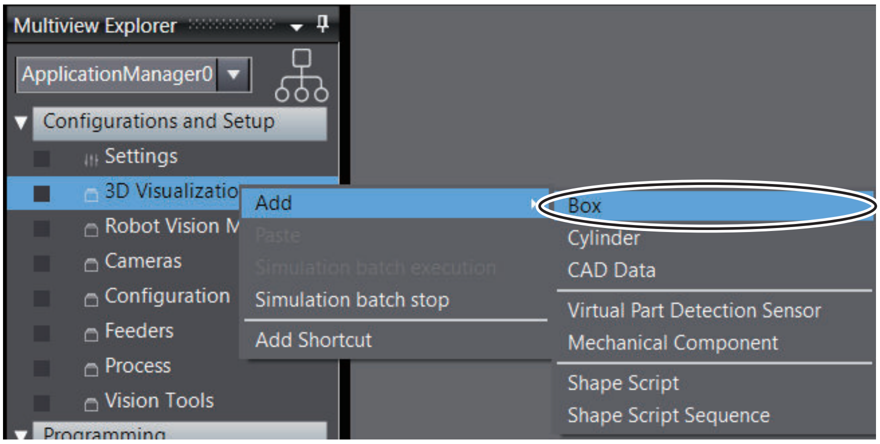
5-4 3D Visualization

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

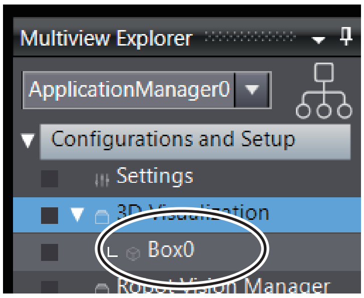
5-4-1 Box and Cylinder

Create a box and a cylinder that used as 3D shape data for parts.

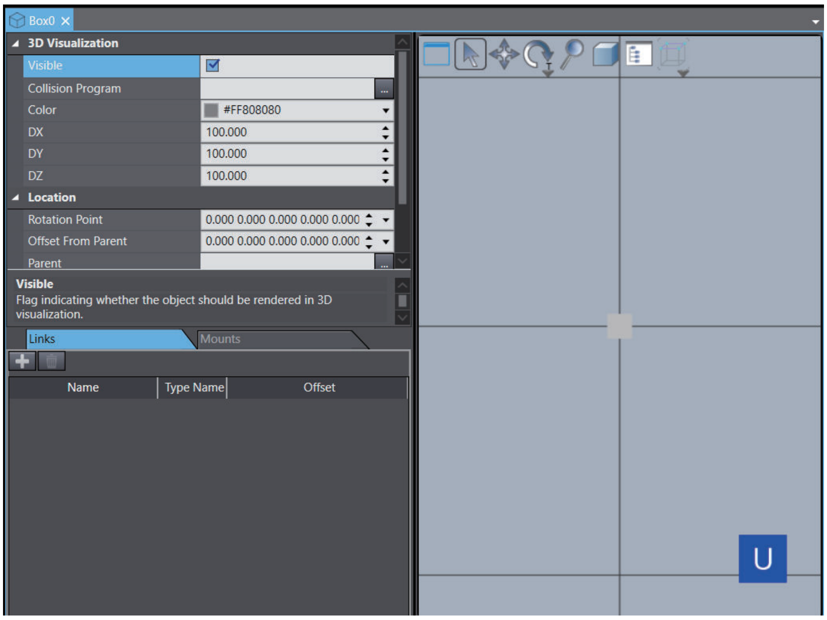
1 Right-click **3D Visualization** under **Configurations and Setup** in the Multiview Explorer. Select **Add** and then, **Box** or **Cylinder**.



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**.
The editor window will be displayed.

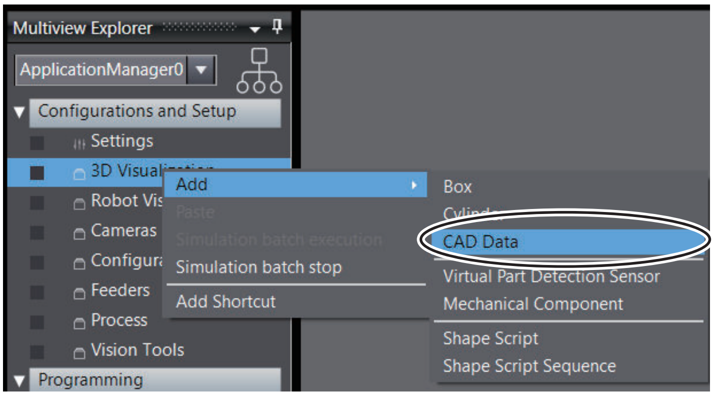


3 Configure setting items for the box or cylinder.

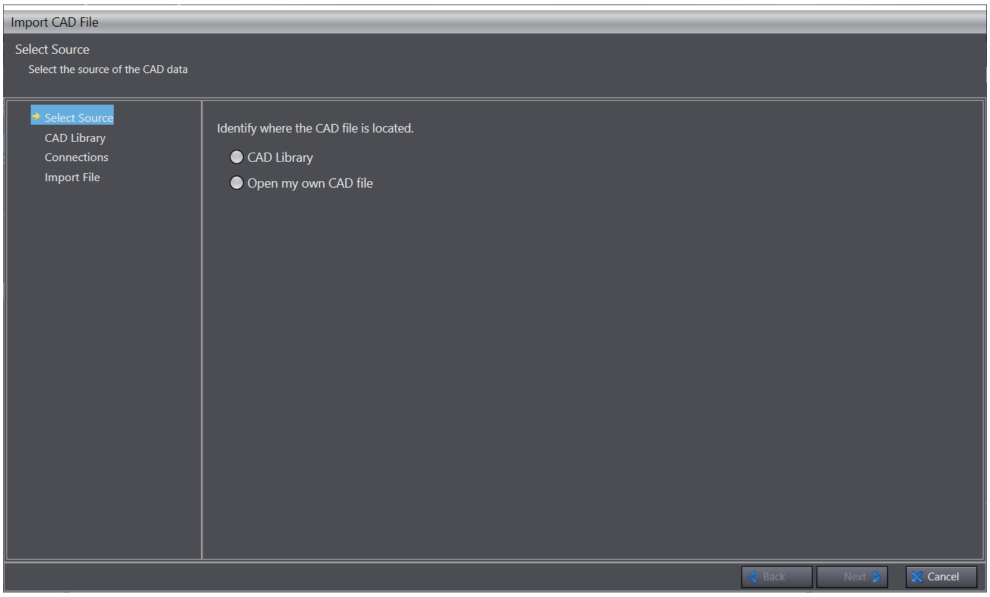
5-4-2 CAD Data

Load the CAD data which used as 3D shape data that represent a gripper, table, rack, part, peripherals and others from a 3D CAD data file.

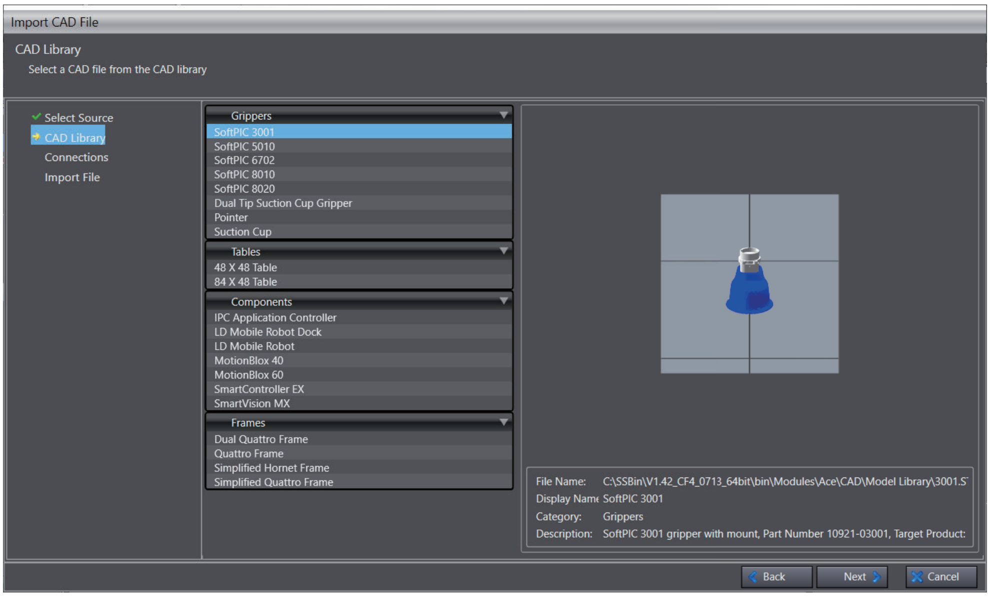
1 Right-click **3D Visualization** under **Configurations and Setup** in the Multiview Explorer. Select **Add** and then, **CAD Data**.



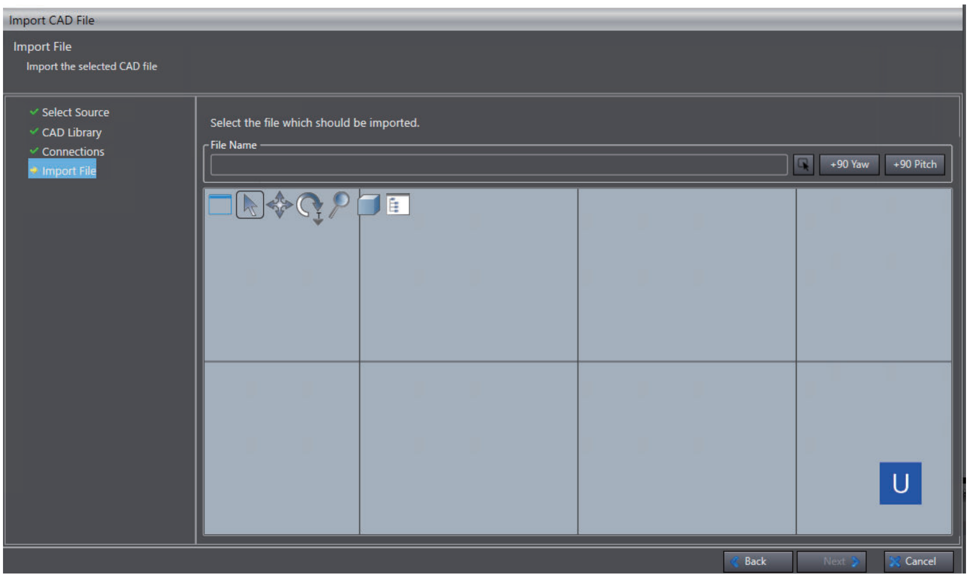
The **Import CAD File** wizard is displayed.



2 Check the CAD file you want to import. Then click the **Next** button.
When you select **CAD Library**, the **CAD Library** window is displayed. Go to Step 3.

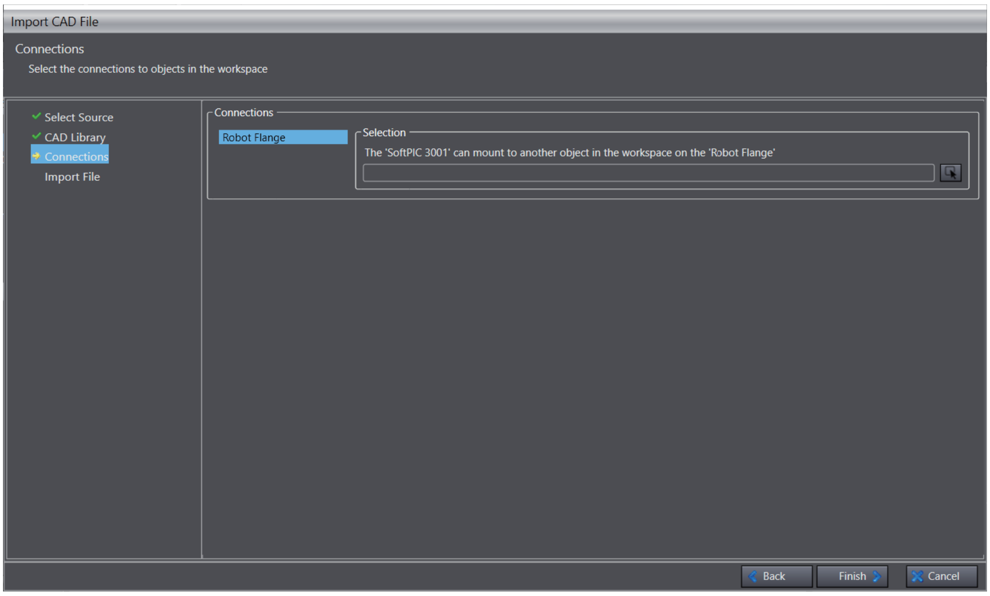


Selecting **Open my own CAD File** displays the **Import CAD File** wizard.

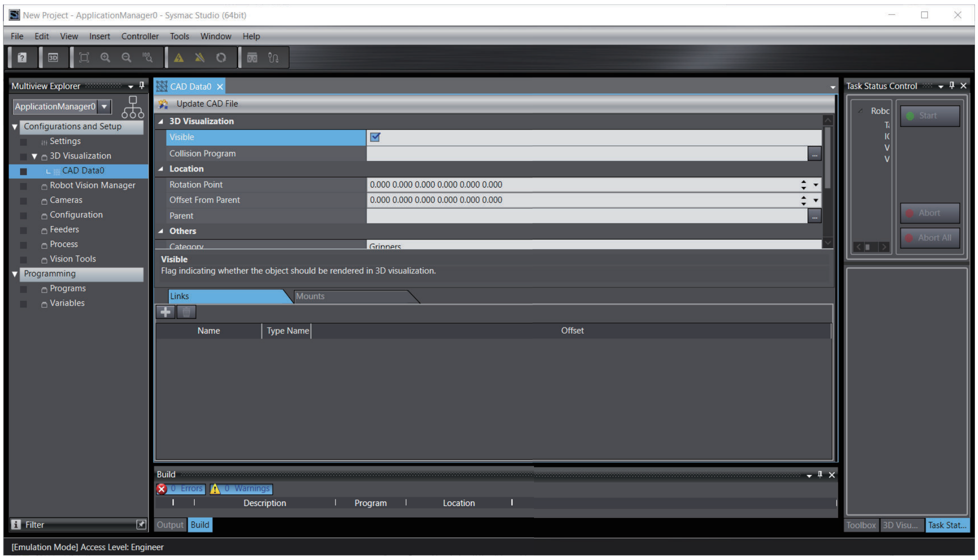


Specify a CAD file you want to import. Then click the **Next** button. The CAD file will be imported. Now you have imported the CAD file.
Refer to *Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618)* for details of operations when selecting **Open my own CAD File**.

3 Specify a CAD file you want to import from the library. Then click the **Next** button.
The **Connections** dialog is displayed.



4 To connect a CAD file 3D shape data to an existing 3D shape data, select a target 3D shape data, then click the **Finish** button.
CAD Data is added under **3D Visualization**.
Setting values are registered to **CAD Data**. You can edit those values.
Right-click the **CAD Data** you want to edit. Then click **Edit** to display the edit window.



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

5-4-3 Exporting/Importing 3D Shape Data

You can export 3D shape data from another project, and import it to the target project. In the project that includes no 3D shape data, you will be able to perform a simulation and a collision judgment based on the 3D shape data.

Additional Information

At the time of project synchronization, if it is transferred to the Controller with the synchronization check item **Transfer 3D shape data to the target when performing Transfer To Target** unchecked, it will be the project that includes no 3D shape data at the transfer destination, so a simulation and a collision judgment based on the 3D shape data cannot be performed. In that case, use this function to import the 3D shape data to the project, and you can perform a simulation and a collision judgment.

3D Visualization Objects Ready for Export/Import 3D Shape Data

The 3D Visualization objects that can be exported and imported are as follows.

- CAD Data
- Mechanical Component
- Custom Mechanics
- Parallel Link Model

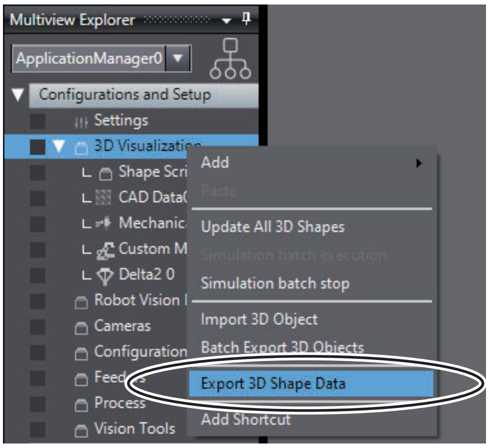
However, the mechanical component conveyor to which no 3D shape data is assigned, the parallel link model to which no 3D shape data is assigned, and the custom mechanics to which no movable parts are added are not available.

Additional Information

Refer to the *Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618)* for the meanings of the 3D shape data and the 3D Visualization object.

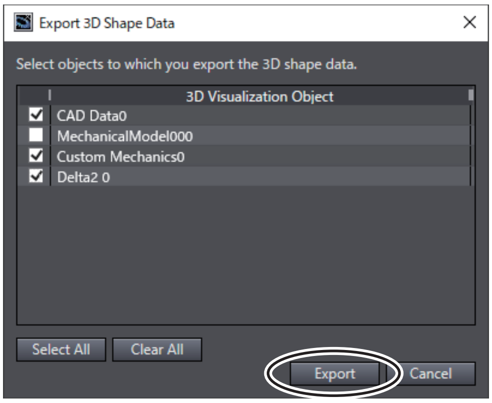
Exporting 3D Shape Data

1 Right-click **3D Visualization** under **Configurations and Setup** in the Multiview Explorer and click **Export 3D Shape Data** from the menu.



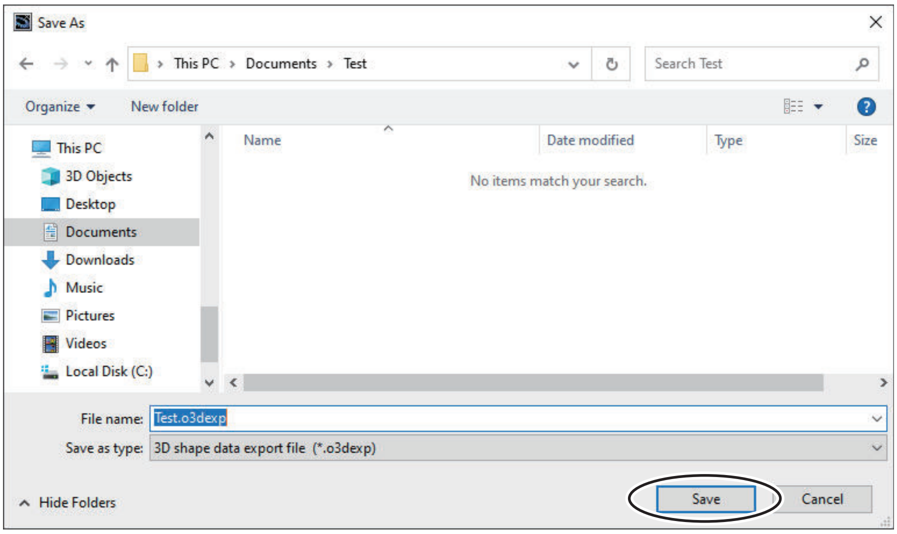
The **Export 3D Shape Data** dialog box is displayed.

2 Select 3D shape data to export, and then click the **Export** button.



The save file dialog box is displayed.

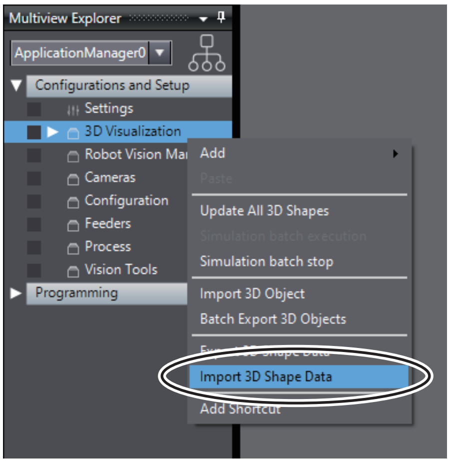
3 Enter the file name, and then click the **Save** button.



The 3D shape data to export are saved in an .o3dexp format file.

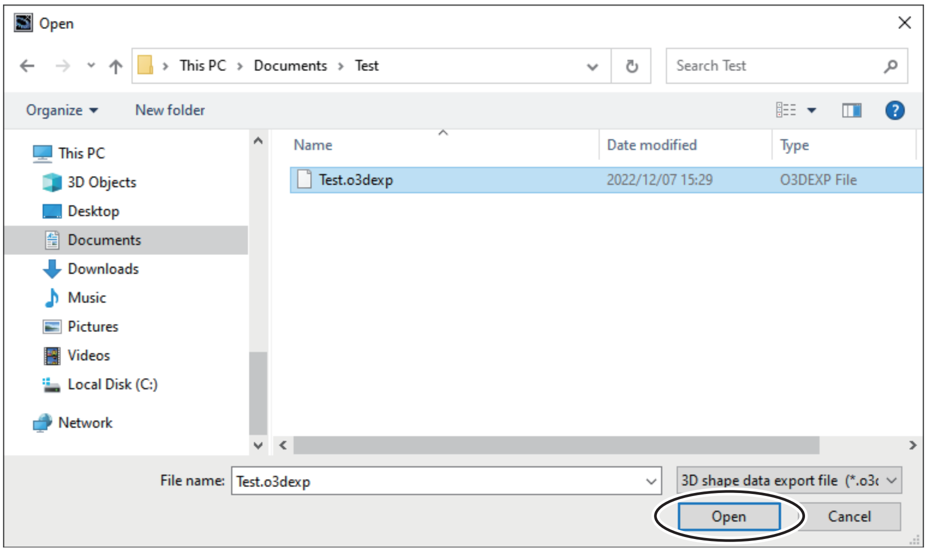
Importing 3D Shape Data

1 Right-click **3D Visualization** under **Configurations and Setup** in the Multiview Explorer and click **Import 3D Shape Data** from the menu.



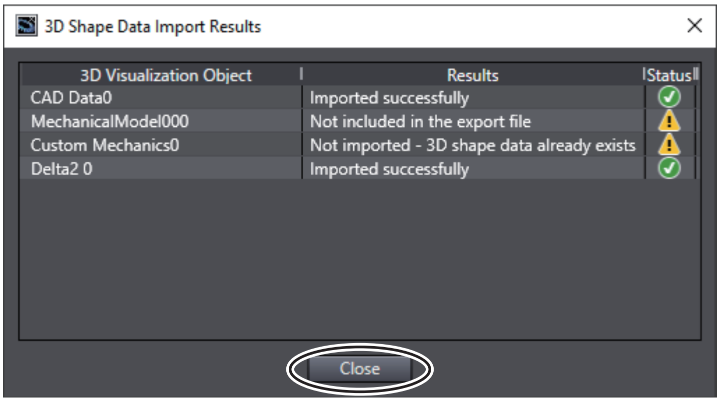
The select file dialog box is displayed.

2 Select the .o3dexp file to import, and then click the **Open** button.



The data is imported, and the **3D Shape Data Import Results** dialog box is displayed.

3 Check the import results, and then click the **Close** button.



Precautions for Correct Use

- The import is executed only for the 3D Visualization object whose 3D shape data was not transferred to the target.
- If a 3D Visualization object included in the export file does not exist in the import-destination project, the target 3D Visualization object will not be imported, so it will not be displayed on **3D Shape Data Import Results** either.

5-5 Robot Vision Manager

Adds the settings for using Robot Vision Manager.

5-5-1 Robot Vision Manager Setting Items

Robot Vision Manager has setting items shown below.

Item	Description	Reference
Belt Latch Calibration	Belt Latch Calibration configures robot operations with respect to a conveyor belt latch. The sensor detects a part reaching to a specific point on the conveyor belt, and generates a latch signal. This latch signal synchronizes positions of the robot and part.	*5-5-2 Belt Latch Calibration* on page 5-15
Belt Calibration	Belt Calibration configures a robot position with respect to a conveyor belt. This calibration will establish a relationship between the belt, its encoder, and the robot. It is necessary when the robot will handle parts that are moving on a conveyor belt.	*5-5-3 Belt Calibration* on page 5-21
Camera Calibration	Camera Calibration configures a robot position with respect to a camera. This calibration is necessary if you will be using a vision system with a robot.	*5-5-4 Camera Calibration* on page 5-27
Gripper Offset Table	The Gripper Offset Table defines where on the part a robot can pick up a part, giving the relationship between the pick point, the part model, and the robot flange center.	*5-5-5 Gripper Offset Table* on page 5-34
Vision Sequence	A Vision Sequence let you see the order and dependency of vision tools executed. A means for retrieving results from vision tools will be given to a V+ program. The Vision Sequence shows the list of tools that will be executed as part of the sequence, the order in which they will be executed, and the Index showing the execution order of each tool. Executing the Overlap Tool and the Communication Tool can control timings of sending results processed by a vision tool to the controller.	*5-5-6 Vision Sequence* on page 5-37
Overlap Tool	The Overlap Tool ensures that parts moving on a conveyor belt are processed only once if located in multiple images.	*5-5-7 Overlap Tool* on page 5-39
Communication Tool	The Communication Tool is called in a Vision Sequence. It sends conveyor-associated instances from the Vision to the Robot Integrated CPU Unit.	*5-5-8 Communication Tool* on page 5-41

5-5-2 Belt Latch Calibration

Belt Latch Calibration configures robot operations with respect to a conveyor belt latch.
Set a belt calibration used for this setting in advance. Refer to *5-5-3 Belt Calibration* on page 5-21 for how to configure the belt calibration.
Follow this order.

- Add a calibration setting
- Execute the calibration
- Test the calibration

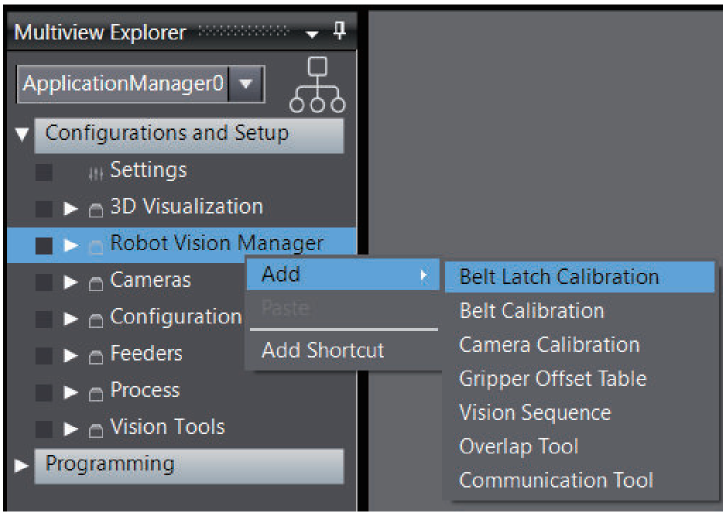
Precautions for Correct Use

Belt Latch Calibration is available only in the Emulation mode.

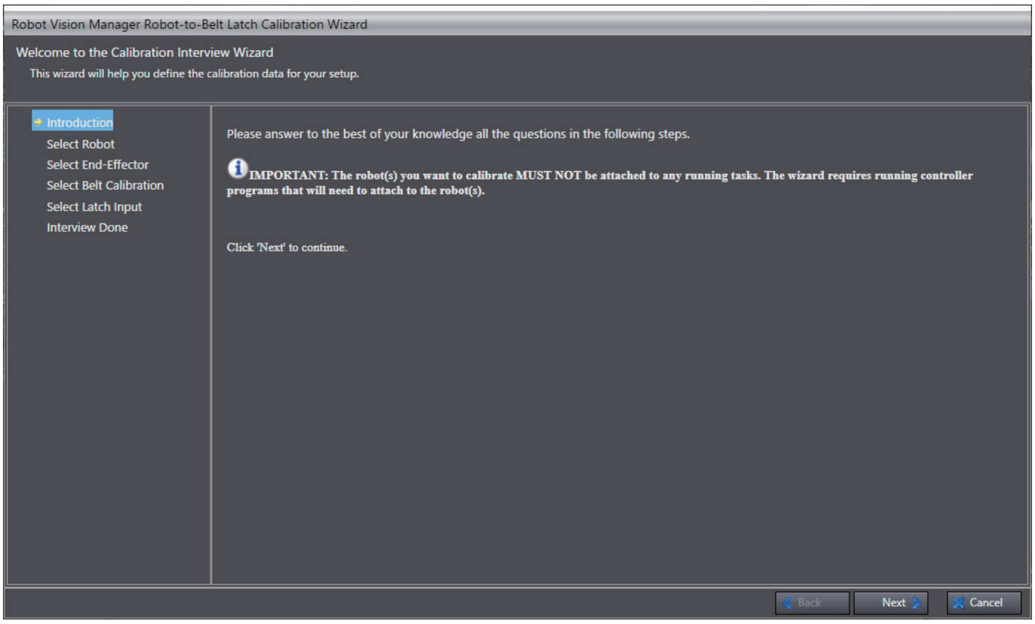
Adding Belt Latch Calibration Settings

Set the data for a calibration. Establish an online connection to the Simulator or actual machine in advance.

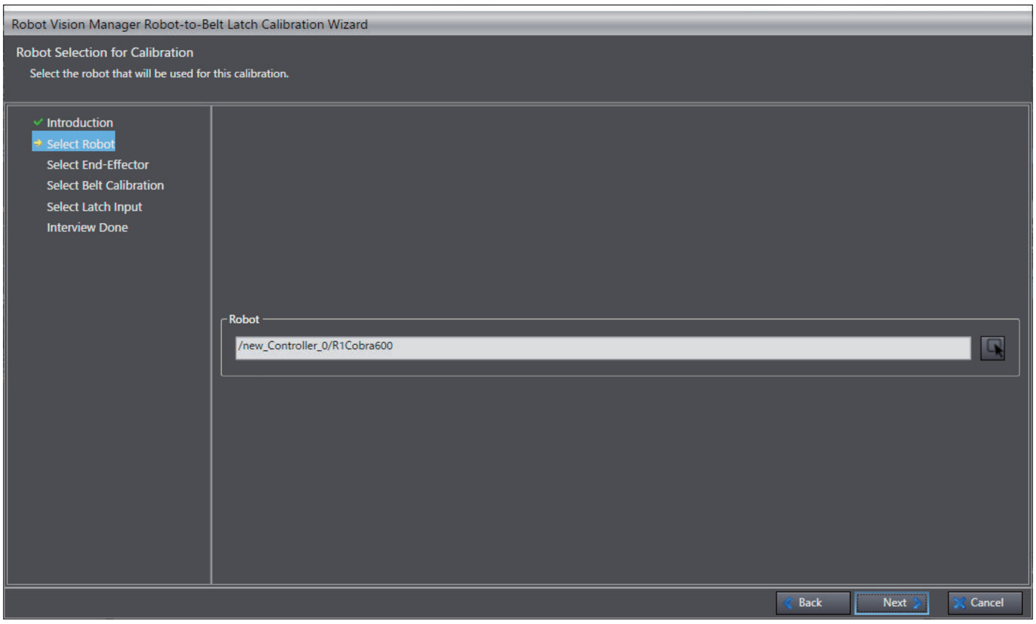
1 Right-click **Robot Vision Manager** under **Configurations and Setup** in the Multiview Explorer. Select **Add**, and then click **Belt Latch Calibration**.



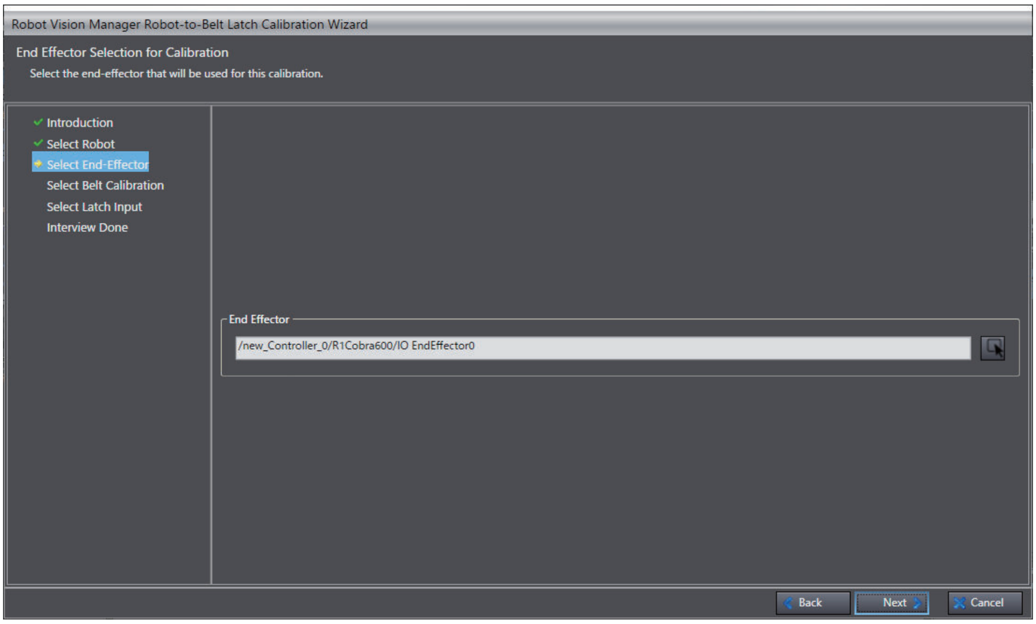
Robot Vision Manager-Robot-to-Belt Latch Calibration Wizard will open.



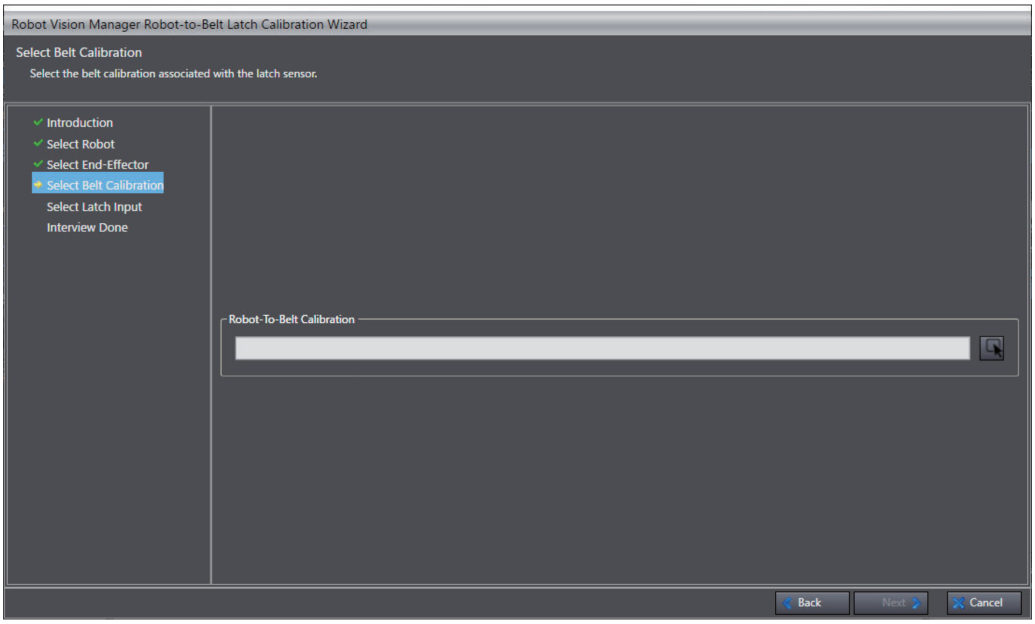
2 Click the **Next** button.
The **Robot Selection for Calibration** window is displayed.



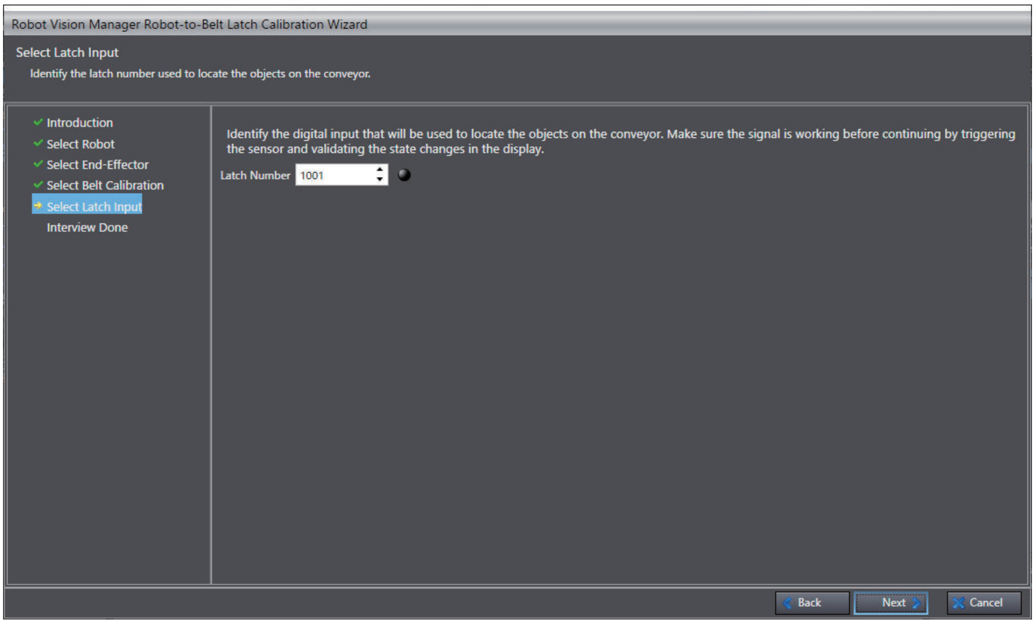
3 Select a robot used for a calibration with the robot selection button. Then click the **Next** button. The **End Effector Selection for Calibration** window is displayed.



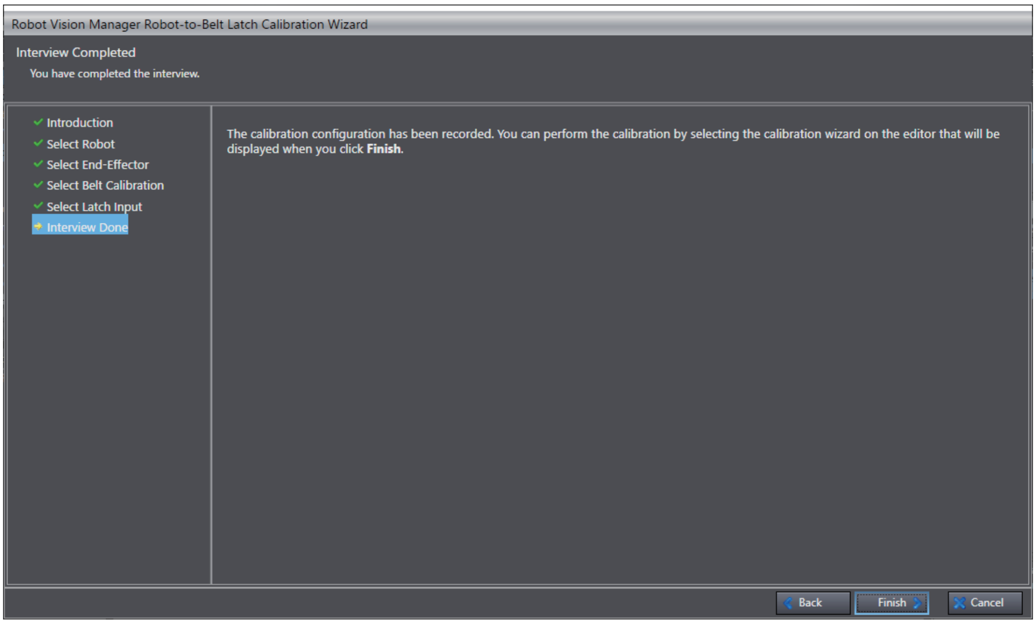
4 Select an end effector used for the calibration with the end effector selection button. Then click the **Next** button.
The **Select Belt Calibration** window is displayed.



5 Select a Belt Calibration you want to connect to the latch sensor with the belt calibration selection button. Then click the **Next** button.
The **Select Latch Input** dialog is displayed.



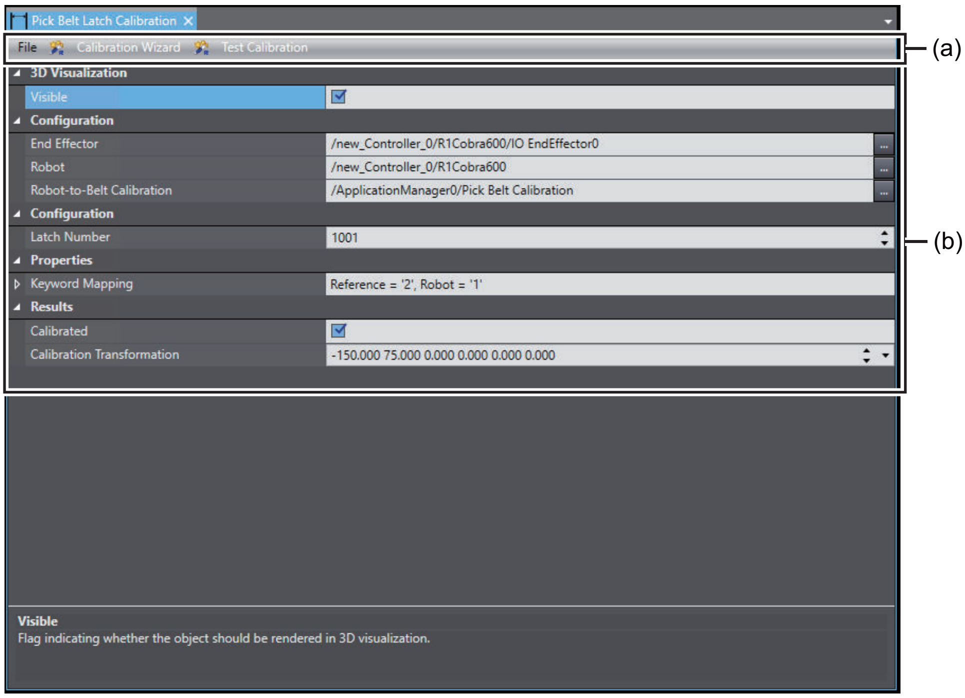
6 Enter a latch signal number of the encoder used for latch input in the **Latch Number** field. Then click the **Next** button.
The **Interview Completed** window is displayed.



Now you have completed setting the calibration data.
Clicking the **Finish** button will add a new **Belt Latch Calibration** to **Robot Vision Manager**.

Belt Latch Calibration Setting Items

You can edit setting values, execute a calibration, and perform a calibration test in a **Belt Latch Calibration**.
Right-click the **Belt Latch Calibration** then click **Edit** to display the edit window.



Items on the editor window are as follows.

	Item			Description
(a)	File			The calibration data can be saved after a calibration. The data can be reused. **Save To**: The data is saved to the specified directory **Load From**: The saved data is loaded from the specified directory
	Calibration Wizard			Execute a calibration with conditions you have set. Refer to *Execute the Calibration* on page 5-20 for details of the calibration wizard.
	Test Calibration			Perform a calibration test with conditions you have set. Refer to *Test the Calibration* on page 5-20 for details of the calibration test.
(b)	3D Visualization	Visible		Check this box to show an object on the 3D Visualizer.
	Configuration	End Effector		Select an end effector used for a calibration.
		Robot		Select a robot used for a calibration.
		Robot-to-Belt Calibration		Specify the Robot-to-Belt Calibration data used in a calibration process.
		Latch Number		Select a latch number used for a calibration.
	Properties	Keyword Mapping		Keyword Mapping is a reference method for a V+ Robot Vision Manager keyword.
			Robot Index	The robot index is passed through the "resultIndex" parameter in the V+ Robot Vision Manager keyword.
			Reference Index	The index is passed through the "toolID" parameter in the V+ Robot Vision Manager keyword.
	Results	Calibrated		The box will be checked if the robot-to-latch sensor calibration is successfully completed.
		Calibration Transformation		Associate the origin of the robot-to-belt calibration with the origin of the latch sensor.

Execute the Calibration

Execute a calibration. Connect to the Simulator in advance.

1 Click the **Calibration Wizard** button in the edit window.
The **Robot-to-Belt Latch Calibration Sequence** window is displayed.

2 Specify an end effector used for an application sample robot, then click the **Next** button.
The **Teach the Sensor Offset** window is displayed.

3 Teach a sensor offset.
A distance between the sensor and an origin of the belt is the sensor offset.
Click and drag the edit handles in the 3D display to adjust the belt calibration. To rotate the calibration, edit the transform manually with the editing buttons at the bottom of the 3D display.
After completed the teaching, click the **Finish** button. Now you have finished the calibration.

Test the Calibration

Test the calibration. Connect to the Simulator in advance.

1. Click the **Test Calibration** button in the edit window.
   The **Latch Calibration Test** window is displayed.

2. Specify an edge of the end effector used for an adjustment, then click the **Next** button.
   The **Position Part** dialog is displayed.

3. Check the displays then click the **Next** button.
   The **Advance the Belt** window is displayed.

4. Move the belt so that the robot reaches to the target.
   Click the button at the right of the **ON/OFF** to run the belt.

5. After a confirmation, click the **Next** button.
   The **Positioning Robots** dialog is displayed.

6. Click the **Move** button to confirm the robot moves. Then click the **Next** button.
   The **Continue testing the calibration** dialog is displayed.

7. Select either of the following options, then click the **Next** button.
   Selecting **I wish to continue testing** returns the screen to the **Position Part** dialog. Selecting **Do not continue testing** closes the window.
   Now you have finished the calibration.

5-5-3 Belt Calibration

Calibrate a belt connected to a robot to acquire a belt position with respect to the robot.
Belt Calibration is required for a robot to handle parts moving on a belt.
Refer to *3-3-2 Robot-to-Belt Calibration* on page 3-9 for technical contents of this calibration.
Follow this order.

- Add a calibration setting
- Execute the calibration
- Test the calibration

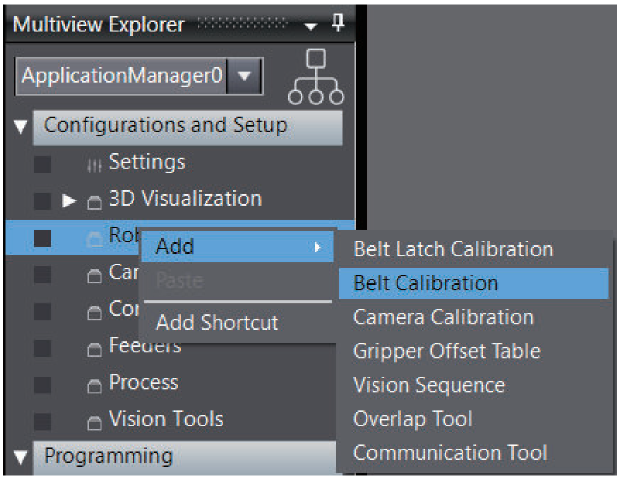
Precautions for Correct Use

Belt Calibration is available only in the Emulation mode.

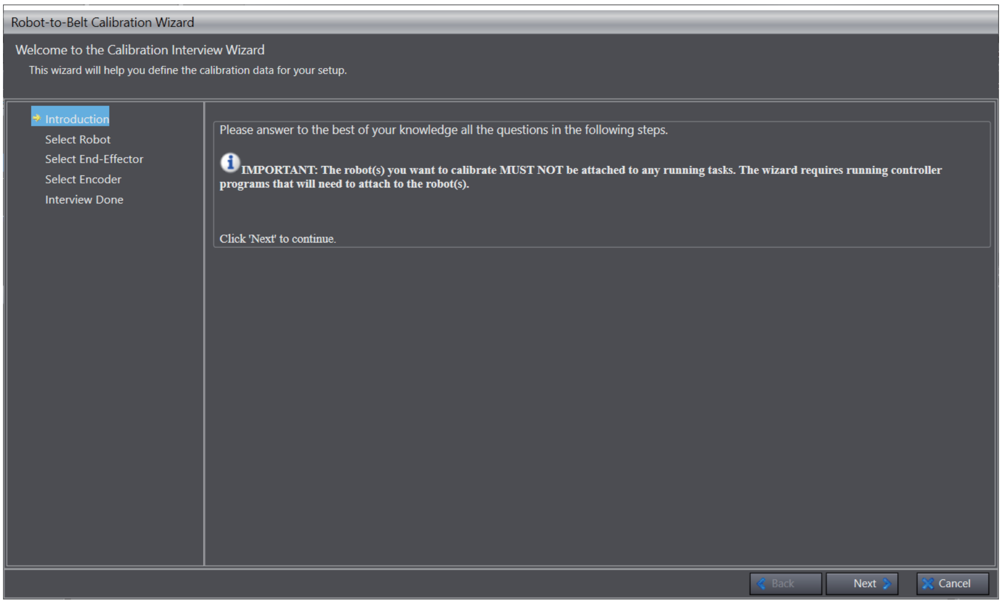
Adding Belt Calibration Settings

Set the data for a calibration. Establish an online connection to the Simulator or actual machine in advance.

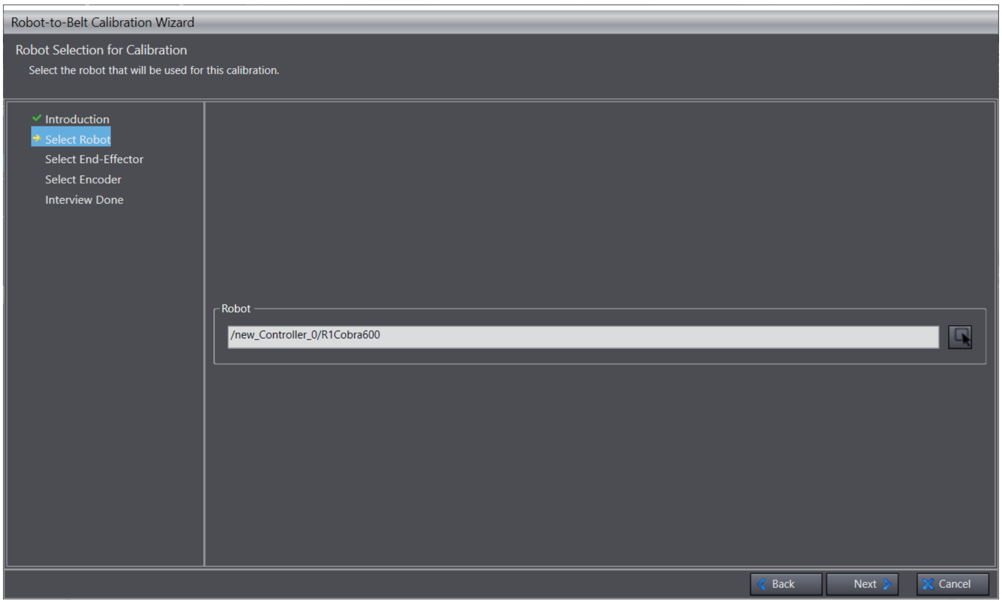
1. Right-click **Robot Vision Manager** under **Configurations and Setup** in the Multiview Explorer. Select **Add**, and then click **Belt Calibration**.



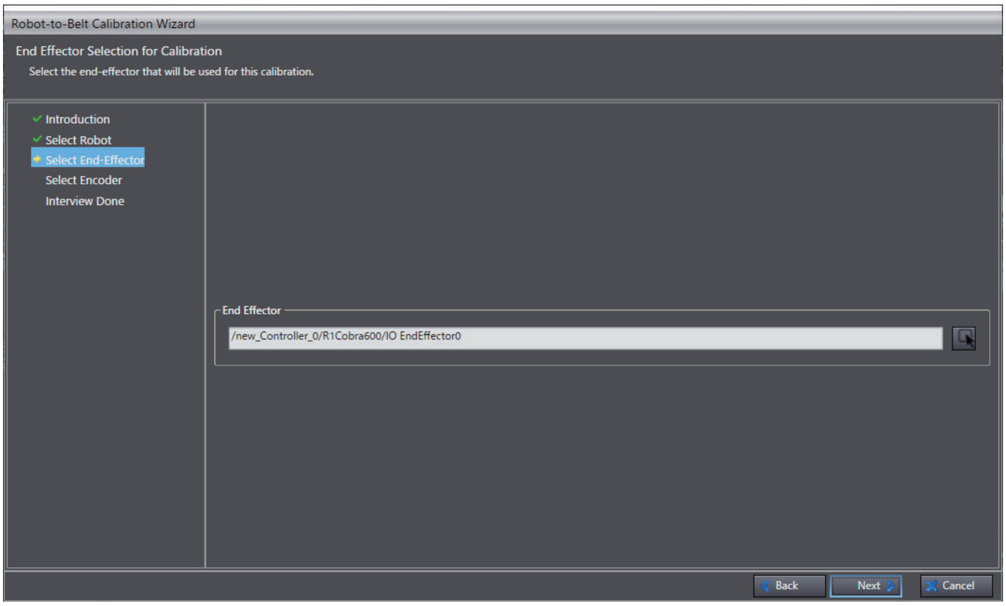
The **Robot-to-Belt Calibration Wizard** is displayed.



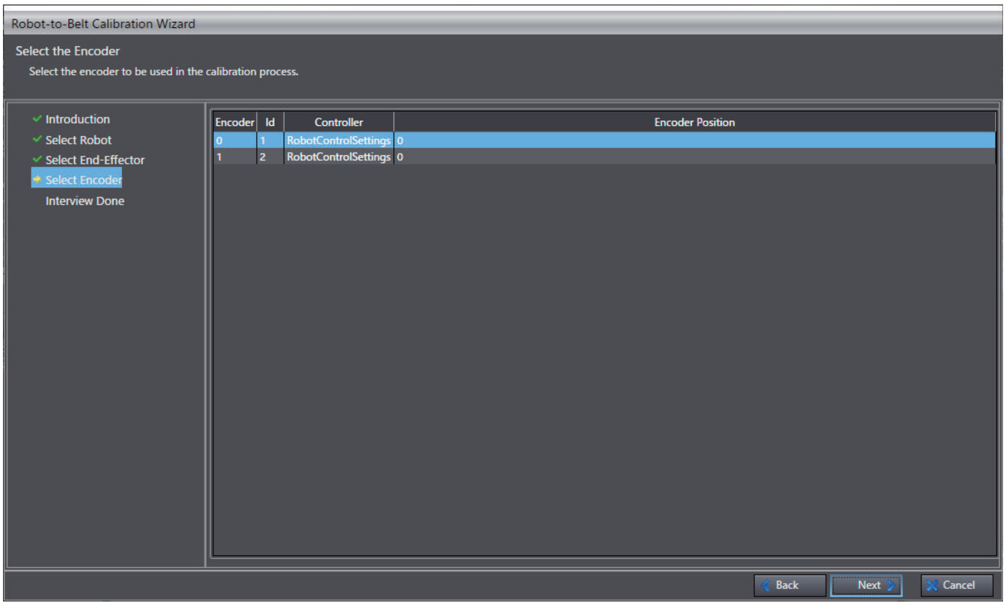
2 Click the **Next** button.
The **Robot Selection for Calibration** window is displayed.



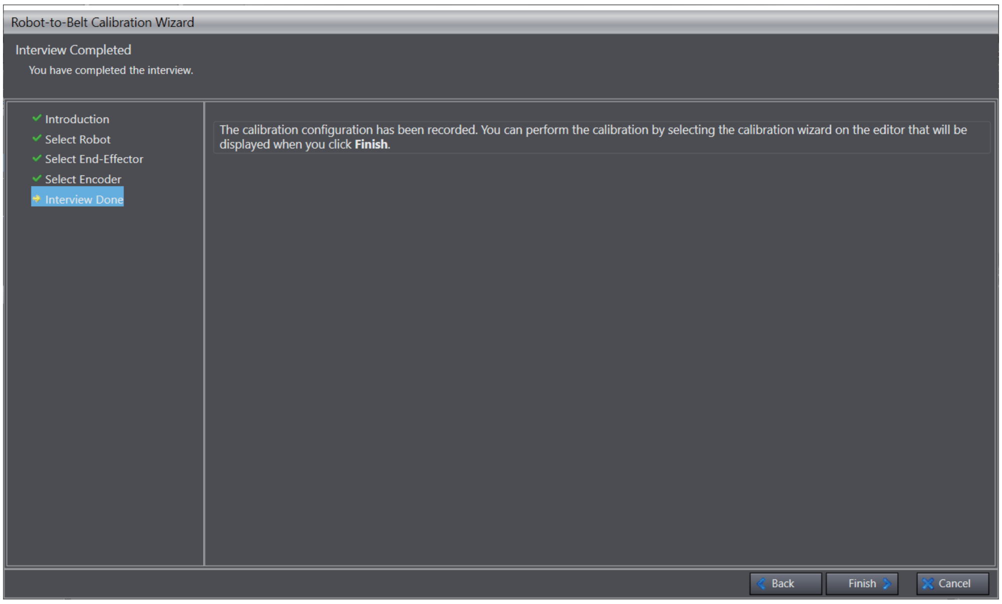
3 Select a robot used for a calibration with the robot selection button. Then click the **Next** button.
The **End Effector Selection for Calibration** window is displayed.



4 Select an end effector used for the calibration with the end effector selection button. Then click the **Next** button.
The **Select the Encoder** dialog is displayed.



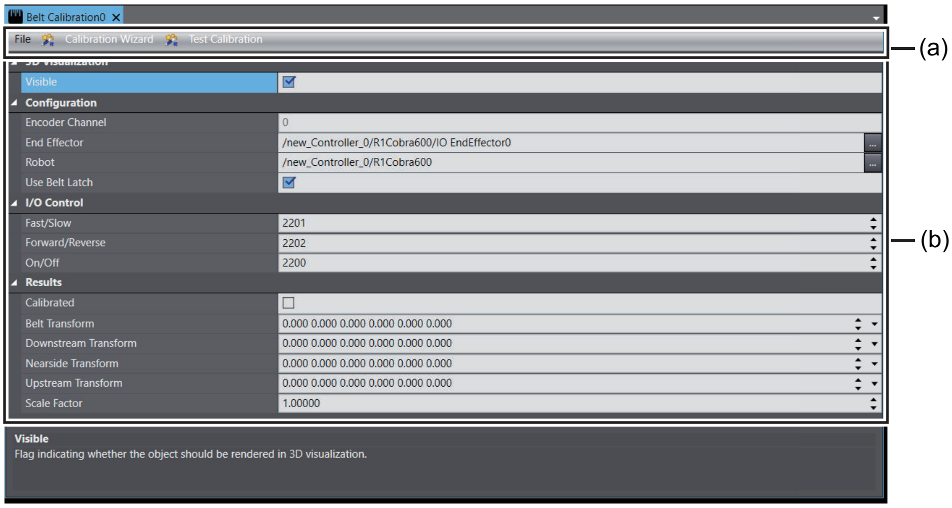
5 Specify an encoder used in a calibration process, then click the **Next** button.
The **Interview Completed** window is displayed.



Now you have completed setting the calibration data.
Clicking the **Finish** button will add a new **Belt Calibration** to **Robot Vision Manager**.

Belt Calibration Setting Items

You can edit setting values, execute a calibration, and perform a calibration test in a **Belt Calibration**.
Right-click the **Belt Calibration** then click **Edit** to display the edit window.



Items on the editor window are as follows.

Different items are shown depends on the conditions, such as automatic or manual control, with or without a belt. The following shows an example.

	Item	Description
(a)	File	The calibration data can be saved after a calibration. The data can be reused. **Save To**: The data is saved to the specified directory **Load From**: The saved data is loaded from the specified directory
	Calibration Wizard	Execute a calibration with conditions you have set. The Wizard differs depends on the conditions, such as automatic or manual control, with or without a belt. Refer to *Executing the Calibration* on page 5-26 for details of Calibration Wizard.
	Test Calibration	Perform a calibration test with conditions you have set. The Wizard differs depends on the conditions. Refer to *Testing the Calibration* on page 5-27 for details of the calibration test.

	Item		Description
(b)	3D Visualization	Visible	Check this box to show an object on the 3D Visualizer.
	I/O Control	ON/OFF	Control signal for ON/OFF. Asserting a signal moves the belt.
		Fast/Slow	Control signal for fasting/slowing the belt. Asserting a signal moves the belt fast.
		Forward/Reverse	Control signal for forwarding/reversing the belt. Asserting a signal reverses the direction of belt travel.
	Configuration	Encoder Channel	A channel where the belt encoder wired to the robot controller.
		End-Effector	Specify an end effector used for a calibration.
		Use Belt Latch	Check this box to read out the belt latch when acquiring images by a camera. Uncheck it if you substitute a current belt position for the latch.
		Robot	Select a robot used for a calibration.
	Results	Belt Transform	Robot-to-Belt Calibration transform.
		Downstream Transform	Downstream robot position.
		Nearside Transform	Position of a robot nearby.
		Upstream Transform	Upstream robot position.
		Scale Factor	A scale factor of an encoder.
		Calibrated	This box will be checked when a Robot-to-Belt calibration has been successfully completed.

Executing the Calibration

Execute a calibration. Connect to the Simulator in advance.

1 Click the **Calibration Wizard** button in the edit window.
The **Robot-to-Belt Calibration Sequence** dialog is displayed.

2 Specify an end effector used for an application sample robot, then click the **Next** button.
The **Test Encoder Operation** dialog is displayed.

3 Check the operation of the encoder by moving the belt.
Click the button at the right of the **ON/OFF** to run the belt. After a confirmation, click the **Next** button.
The **Teach the belt window** window is displayed.

4 Teach the belt.
Click and drag the edit handles in the 3D display to adjust the belt calibration. To rotate the belt calibration, edit the transform manually with the editing buttons at the bottom of the 3D display.
After completed the teaching, click the **Next** button.
The **Test Belt Calibration** window is displayed.

5 Place the robot over a target object on the belt. Then click the **Start Tracking** button.
The belt moves, and the robot tracks the part on the belt. After completed the test, click the **Finish** button.

Now you have finished the calibration.

Testing the Calibration

Test the calibration. Connect to the Simulator in advance.

1. Click the **Test Calibration** button in the edit window.
   The **Test Belt Calibration** window is displayed.

2. Place the robot over a target object on the belt. Then click the **Start Tracking** button.
   The belt moves, and the robot tracks the part on the belt. After completed the test, click the **Finish** button.
   Now you have finished the calibration test.

5-5-4 Camera Calibration

Create a Camera Calibration object. Calibrate a camera connected to a robot to acquire a camera view with respect to the robot.
Follow this order.

- Add a calibration setting
- Execute the calibration
- Test the calibration

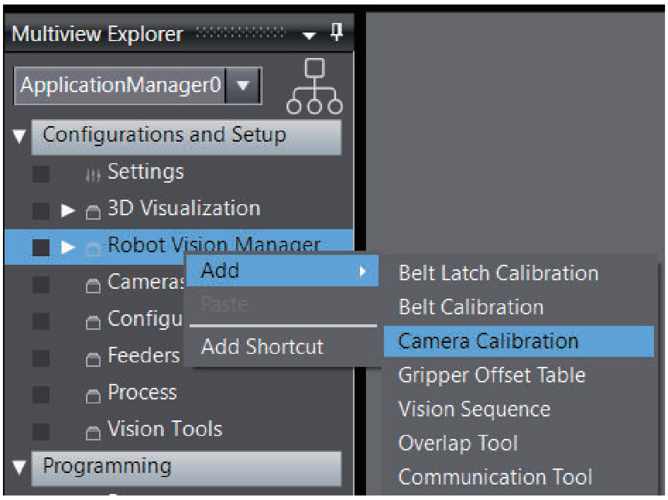
Precautions for Correct Use

It is necessary to calibrate the camera in advance. Refer to *5-6-1 Virtual Camera* on page 5-47 for details.
To use a belt, you also must calibrate the belt in advance. Refer to *5-5-3 Belt Calibration* on page 5-21 for details.

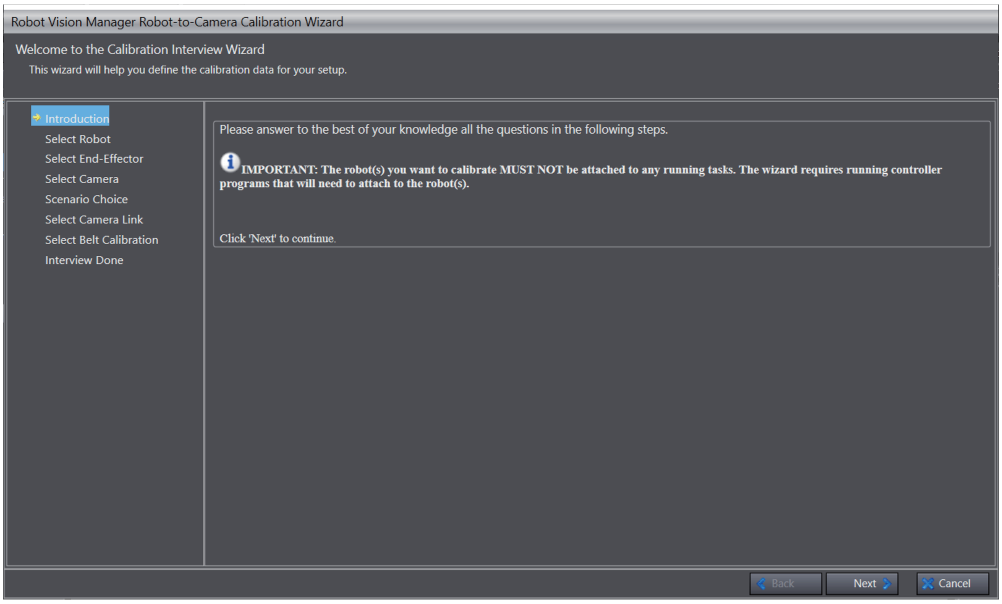
Adding Camera Calibration Settings

Set the data for a calibration. Establish an online connection to the Simulator or actual machine in advance.

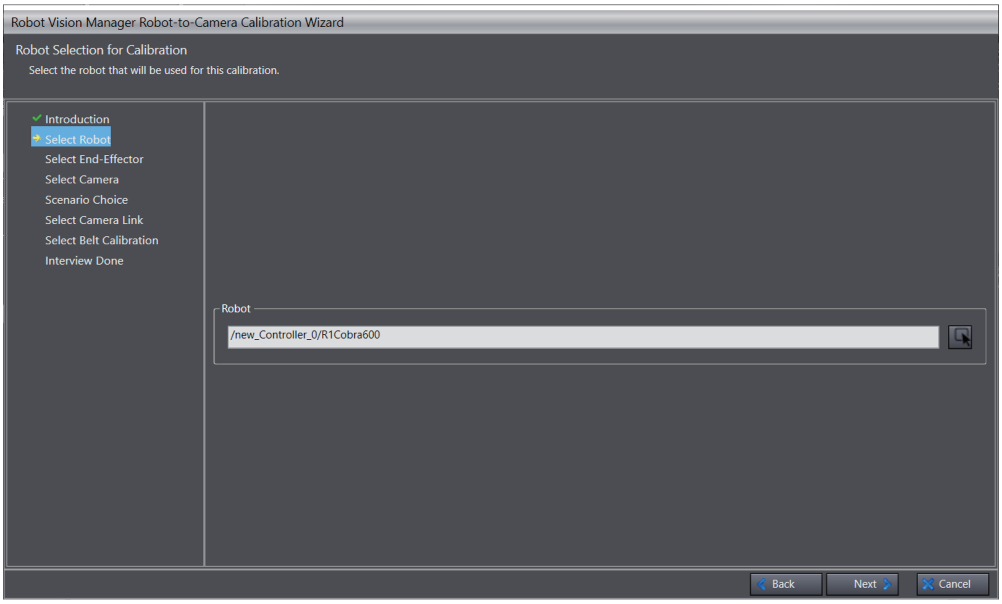
1. Right-click **Robot Vision Manager** under **Configurations and Setup** in the Multiview Explorer. Select **Add**, and then click **Camera Calibration**.



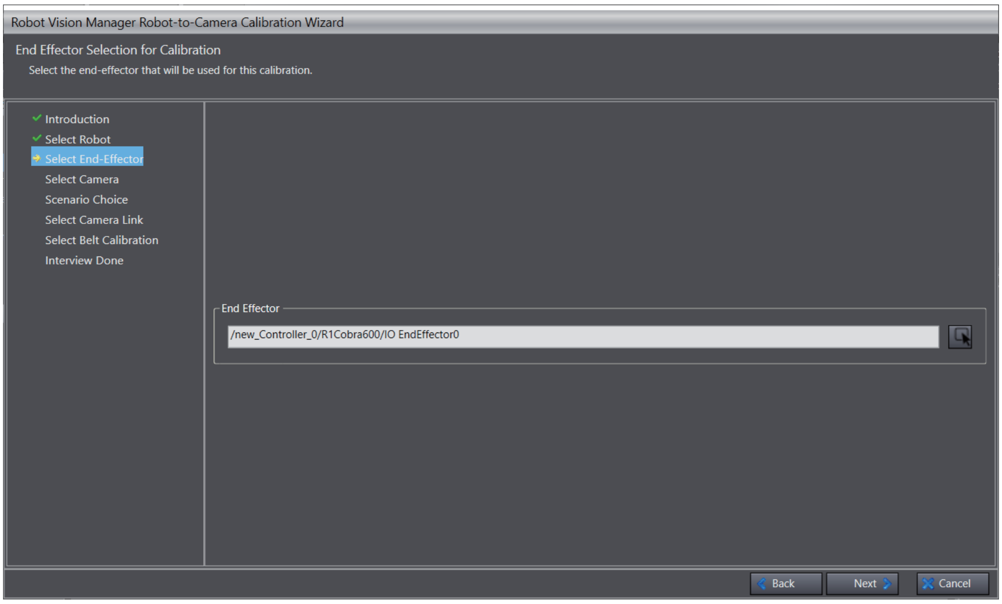
Robot Vision Manager Robot-to-Camera Calibration Wizard will open.



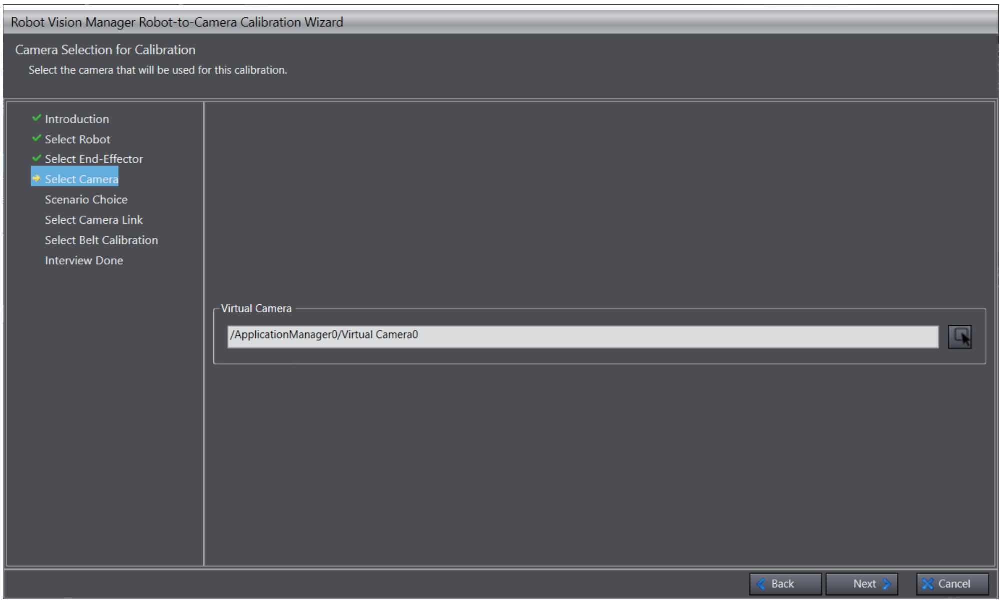
2 Click the **Next** button.
The **Select Robot** dialog is displayed.



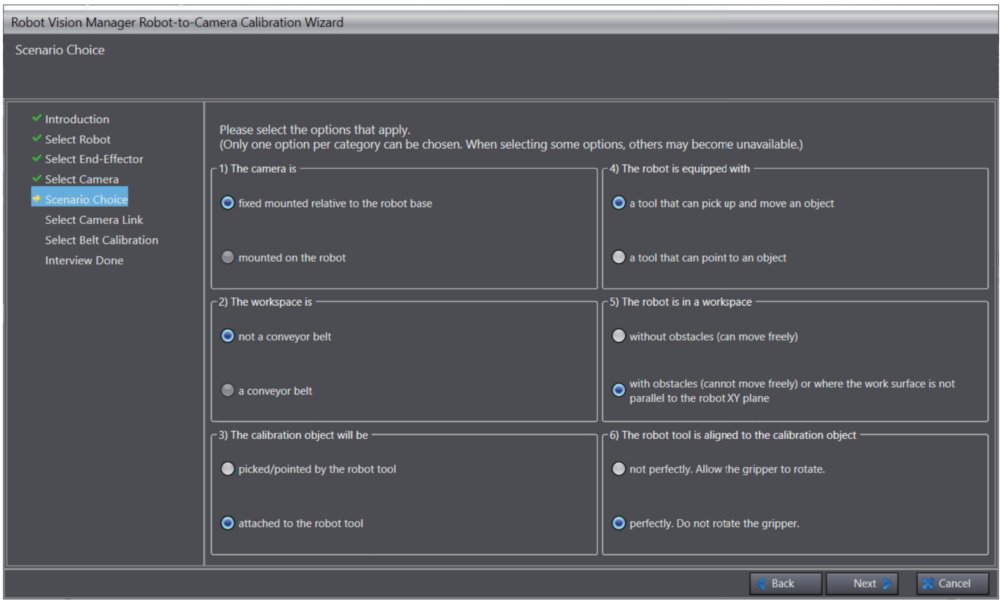
3 Select a robot used for a calibration with the robot selection button. Then click the **Next** button.
The **Select End-Effector** dialog is displayed.



4 Select an end effector used for the calibration with the end effector selection button. Then click the **Next** button.
The **Select Camera** dialog is displayed.



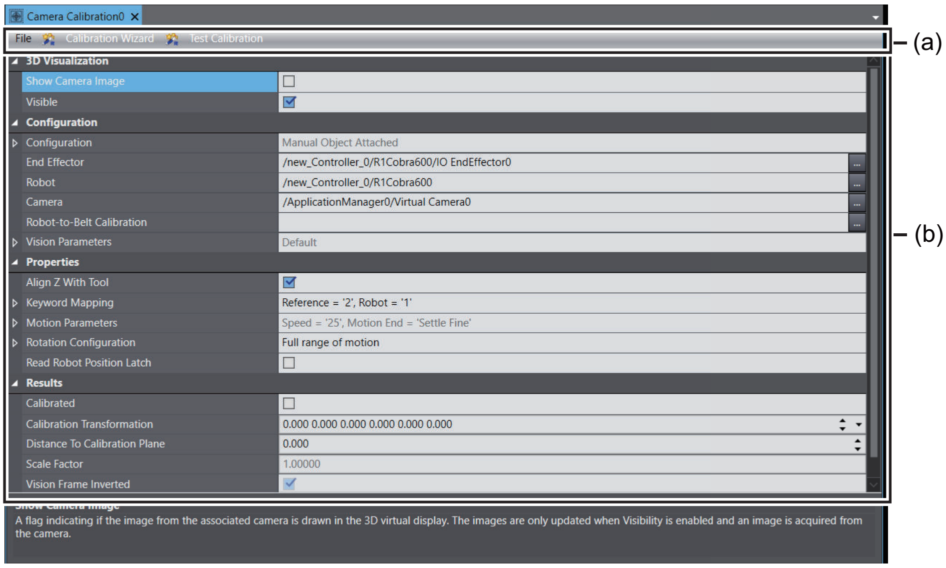
5 Specify a camera used for a calibration with the camera selection button. Then click the **Next** button.
The **Scenario Choice** dialog is displayed.



6 Select options for 1 to 6, then click the **Next** button.
The **Interview Done** window is displayed.
Clicking the **Finish** button will add a new **Camera Calibration** to **Robot Vision Manager**.

Camera Calibration Setting Items

You can edit setting values, execute a calibration, and perform a calibration test in a **Camera Calibration**.
Right-click the **Camera Calibration** then click **Edit** to display the edit window.



Items on the editor window are as follows.

Different items are shown depends on the conditions, such as automatic or manual control, with or without a belt. The following shows an example.

	Item		Description
(a)	File		The calibration data can be saved after a calibration. The data can be reused. **Save To**: The data is saved to the specified directory **Load From**: The saved data is loaded from the specified directory
	Calibration Wizard		Execute a calibration with conditions you have set. The Wizard differs depends on the conditions, such as automatic or manual control, with or without a belt. Refer to *Executing the Calibration* on page 5-33 for details of Calibration Wizard.
	Test Calibration		Perform a calibration test with conditions you have set. The Wizard differs depends on the conditions. Refer to *Testing the Calibration* on page 5-33 for details of the calibration test.
(b)	3D Visualization	Show Camera Image	Check this box to display an image from the connected camera in the 3D Visualizer.
		Visible	Check this box to show an object in the 3D Visualizer.

Item			Description
Configuration	End-Effector		Specify an end effector used for a calibration.
	Camera		Select a camera used for a calibration.
	Configuration	Arm Mounted	Check this box to connect an arm-mounted camera. Uncheck it to connect a fixed mounted camera.
		Object Attached	Check this box to attach an object to a robot tooltip.
		Gripper	Check this box if a gripper can be rotated during a calibration.
		Joint Number	Specify a joint number of the robot where the camera mounted. The joint number is available when an arm-mounted camera is connected.
		No Tool Offset	Check this box when a tool is placed on the center precisely . It is effective when the tool can pick and touch an object at the origin of a model accurately and the tool ‘s diameter is regarded as 0.
		Belt	Check this box for a calibration with a belt.
		Pointer	Check this box when you use a pointer in a calibration process.
		Auto Move	Check this box when the robot is allowed to move automatically and freely in the workspace. Uncheck it to move the robot manually.
	Robot		Select a robot used for a calibration.
	Robot-to-Belt Calibration		Specify the Robot-to-Belt Calibration data.
	Vision Parameters		Specify a Vision Tool used in a calibration process.
Properties	Align Z with tool		Check this box to align the Z-axis to the specified tool axis.
	Keyword Mapping		Keyword Mapping is a reference method for a V+ Robot Vision Manager keyword.
		Robot Index	The robot index is passed through the "resultIndex" parameter in the V+ Robot Vision Manager keyword.
		Reference Index	The index is passed through the "toolID" parameter in the V + Robot Vision Manager keyword.
	Motion Parameters		Parameters used for a calibration motion.
	Read Robot Position Latch		Check this box to use a latch robot position when acquiring an image by the camera. Uncheck it if you substitute a current belt Robot for the latch robot position. This parameter is used only when an arm-mounted or table-mounted camera is connected.
	Rotation Configuration		The range of travel for the robot theta rotation used in the Automatic Calibration Wizard. If the starting angle is a negative value and the ending angle is a positive value, the gripper will rotate in the -180 to +180 degree direction. If the starting angle is a positive value and the ending angle is a negative value, the gripper will rotate in the 0 degree direction.

	Item		Description
	Results	Calibrated	This box will be checked when a Robot-to-Vision calibration has been successfully completed.
		Calibration Transformation	Connect the robot's origin to that of the camera.
		Distance To Calibration Plane	Distance between the camera to the calibration plane.
		Scale Factor	A scale factor from the Vision coordinates to the robot coordinates.
		Vision Frame Inverted	This box will be checked when the Z-axis of Vision Frame is in the same direction as the Z-axis of the robot flange.

Executing the Calibration

Execute a calibration. Connect to the Simulator in advance.

1 Click the **Calibration Wizard** button in the edit window.
The **Robot-to-Fixed or Arm-Camera Manual Calibration Sequence** dialog is displayed.

2 Specify an end effector used for an application sample robot, then click the **Next** button.
The **Teach the Camera Location** dialog is displayed.

3 Teach a camera position.
Click and drag the edit handles in the 3D display to adjust the belt calibration. To rotate the calibration, edit the transform manually with the editing buttons at the bottom of the 3D display.
After completed the teaching, click the **Finish** button. Now you have finished the calibration.

Testing the Calibration

Test the calibration. Connect to the Simulator in advance.

1 Click the **Test Calibration** button in the edit window.
The **Test Calibration** dialog is displayed.

2 Specify a tip of the end effector used for an adjustment, then click the **Next** button.
The **Move Picture** pane is displayed.

3 Move the robot to the position of an image, then click the **Here** button.

4 After moving the robot, click the **Next** button.
The **Teach Model** dialog is displayed.

5 Detects a position of a model.

1) Place a target part at the center of the registration area, then click the **Run** button.
2) Move and adjust the green frame to specify the size and location of the area.
3) Click the **Center** button to place the center point.

6 After completed the teaching, click the **Next** button.
The **Locate Object** dialog will open.

7 Place the target area in the area where the robot can access, then click the **Next** button.
The **Positioning Robots** dialog is displayed.

8 Align the robot with the part, then click the **Next** button.
The **Continue testing the calibration** dialog is displayed.
Selecting **I wish to continue testing** returns the screen to the **Move to Picture** dialog. Selecting **Do not continue testing** closes the window.
Now you have finished the calibration.

5-5-5 Gripper Offset Table

The Gripper Offset Table defines where on the part a robot can pick up a part, giving the relationship between the pick point, the part model, and the robot flange center, and set the values as a offset table.
A gripper can have two types of offsets:

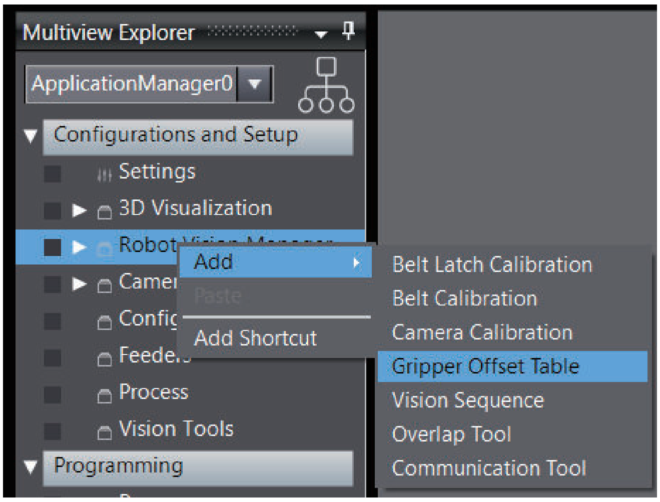
- A tip offset from the center of the tool flange to the gripper tip(s).
  This offset is established when the robot end effector is defined. This offset will be applied to that gripper whenever it is used, including calculating values for the Gripper Offset Table. Refer to *4-4-1 Setting Up IO EndEffectors* on page 4-10 for more information.
- A offset from the actual pick point to the part origin
  It indicates where the robot must pick the part in relation to the origin of the part. It is is defined in the Gripper Offset Table and assigned to a specific robot.

Follow this order.

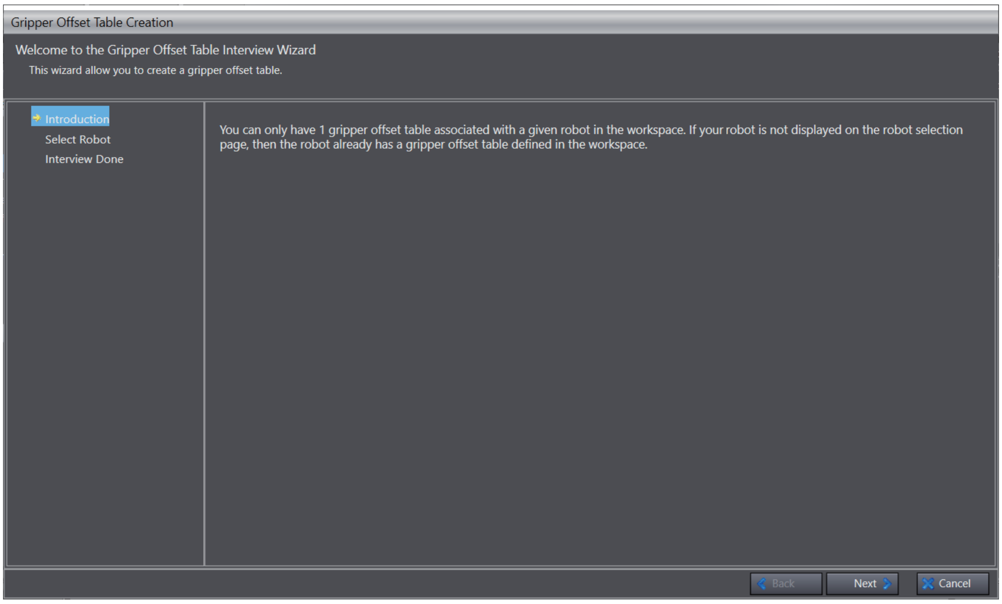
- Adding a Gripper Offset Table
- Adding a Gripper Offset Index
- Teaching

Adding a Gripper Offset Table

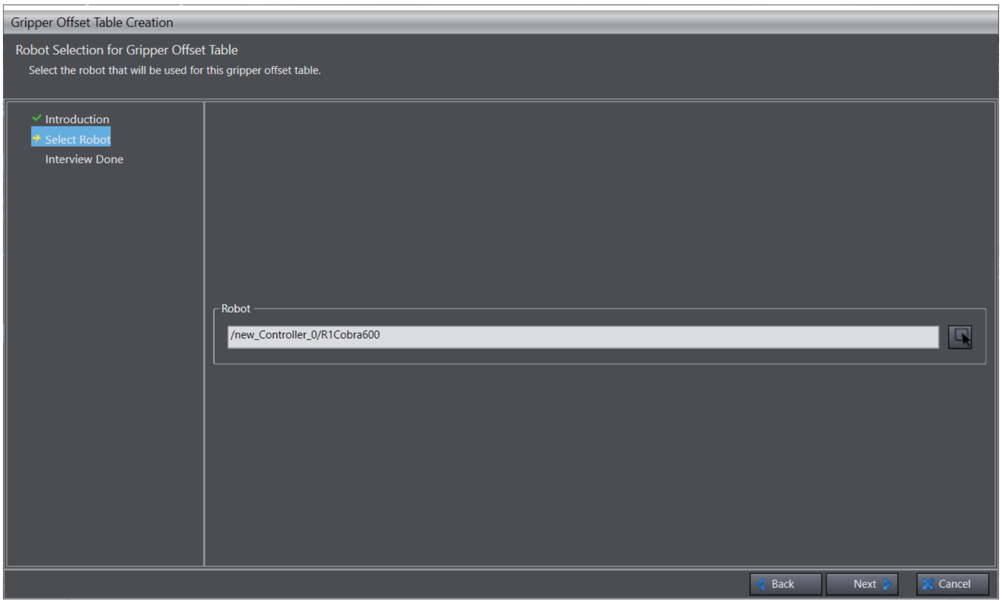
1 Right-click **Robot Vision Manager** under **Configurations and Setup** in the Multiview Explorer. Select **Add**, and then click **Gripper Offset Table**.



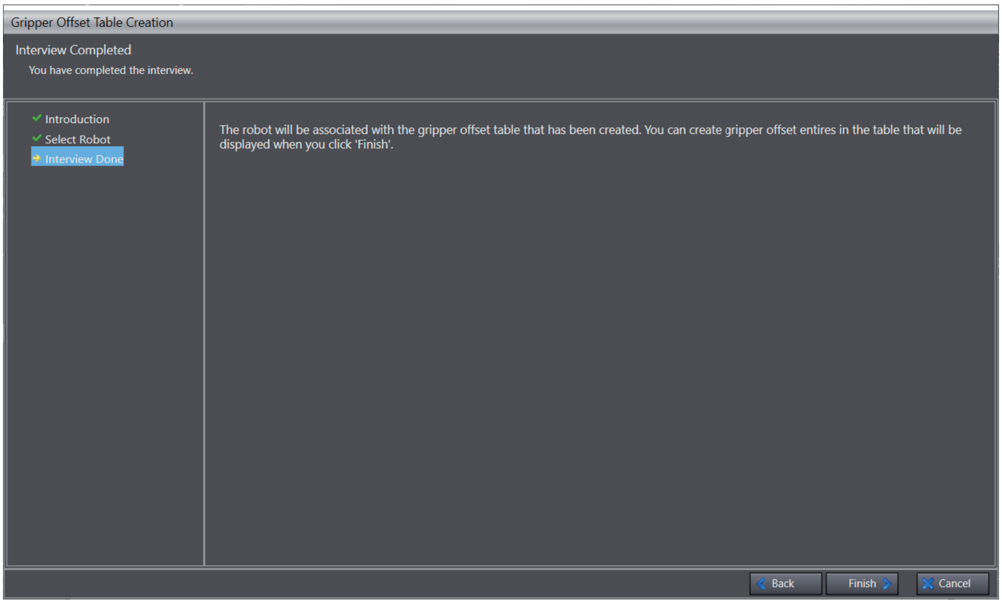
The **Gripper Offset Table Creation** wizard will open.



2 Click the **Next** button.
The **Select Robot** dialog is displayed.



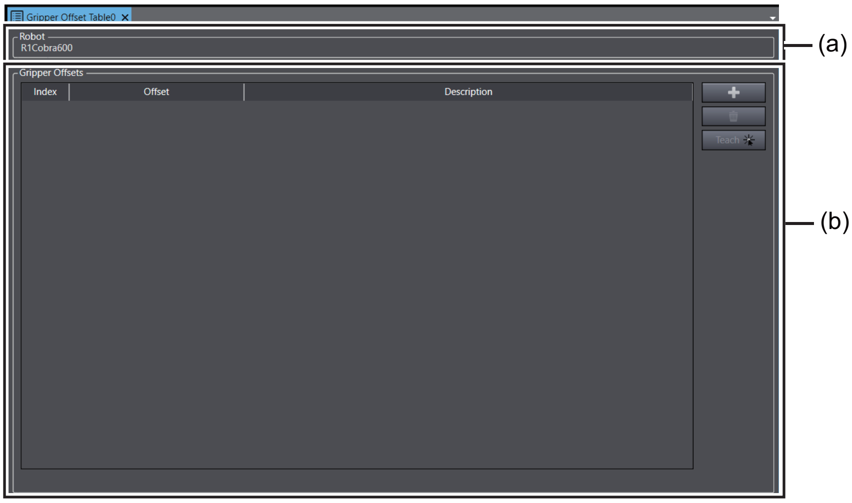
3 Specify a robot used for the Gripper Offset Table, then click the **Next** button.
The **Interview Done** window is displayed.



The robot is connected to the Gripper Offset Table.
Clicking the **Finish** button will add a new **Gripper Offset Table** to **Robot Vision Manager**.

Gripper Offset Table Setting Items

Setting values are registered to **Gripper Offset Table**. You can edit those values.
Right-click the **Gripper Offset Table** you want to edit. Then click **Edit** to display the edit window.



Items on the editor window are as follows.

	Item		Description
(a)	Robot		A robot assigned to the Gripper Offset Table is shown.
(b)	Gripper Offsets		To add a new offset, click the + button. To delete an existing offset, click the Delete button. Click the **Teach** button to set a Gripper Offset. Refer to *Teaching* on page 5-37 for details of Teaching.
		Index	Enter a unique value that reference a Gripper Offset value.
		Offset	Enter a Gripper Offset value that represented in (X, Y, Z, yaw, pitch, roll).
		Description	Enter a description for a Gripper Offset Index entry.

Teaching

Set a Gripper Offset.

Pressing the **Teach** button opens the Gripper Offset Teach Wizard. The wizard guides you through several steps to teach an Offset value. The following other objects that may exist in the project are taken into account.

- 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 con- figured 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 executed. A means for re-trieving results from vision tools will be given to a V+ program. A Vision Sequence shows the list of tools that will be executed as part of the sequence, the order in which they will be executed, and the Index showing the execution order of each tool.

Precautions for Correct Use

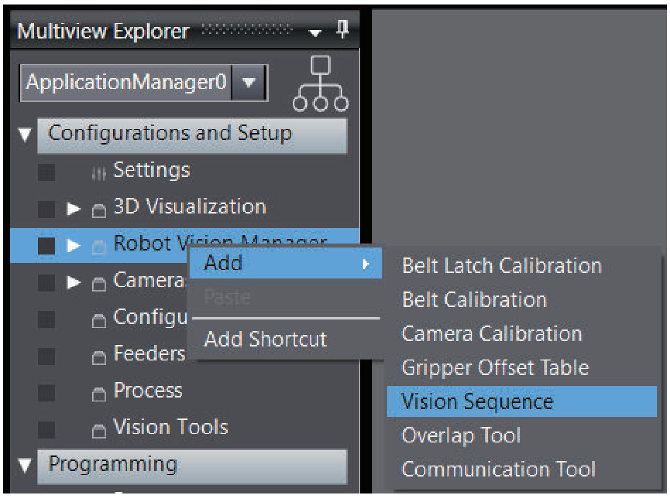
It is necessary to register vision tools you use in advance.

Additional Information

A **sequence number** and an **index** are mandatory for retrieving results from vision tools with a V+ program. Make sure to enter the sequence number and index.

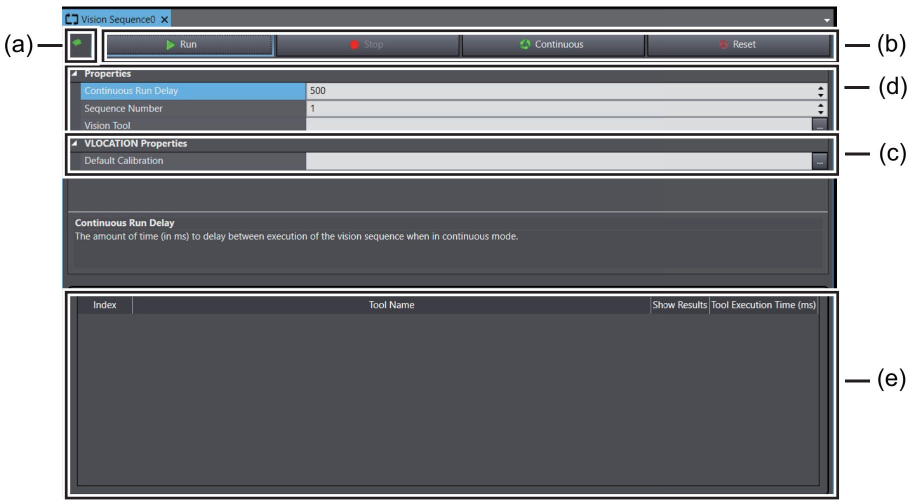
Adding a Vision Sequence

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



A **Vision Sequence** will be added to the **Robot Vision Manager**.

2 Right-click the **Vision Sequence**, then click **Edit**.
The editor window will be displayed.



Items on the editor window are as follows.

	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 **Run** button is clicked. Hover over the red flag for information about the incomplete configuration.
(b)	Sequence Control Buttons	Run	Runs the sequence once.
		Stop	Stops the running sequence.
		Continuous	Runs the sequence repeatedly.
		Reset	Resets the communication queue and the Overlap tool tracking of previously-located instances.
(c)	VLOCATION Properties	Default Calibration	Default camera calibration used to apply to a VLOCATION transformation function from V+ programs.
(d)	Properties	Sequence Number	The sequence number of the specified vision tool.
		Vision Tool	Top-level vision tool this sequence references. The top-level vision tool runs last.
		Continuous Run Delay	Amount of time in milliseconds to delay between execution of the Vision Sequence in continuous mode. The unit is ms.
(e)	Sequence		The order of vision tools executed.

Vision Sequence Settings

The Vision Sequence editor shows the list of tools that will be executed as part of the sequence, the order in which they will be executed, and the associated index number of each. The tools are executed in ascending Index value.

The main property of a Vision Sequence is the Vision Tool parameter that defines what tool marks the end of the sequence. Once you have selected the last tool in the sequence, the Sequence area will be populated by a tool list in the execution order, from the initial image source down to the last one.

5-5-7 Overlap Tool

When a camera captures a part moving on a conveyor for detection, the part may be found in multiple images. The Overlap Tool ensures that parts moving on a conveyor belt are identified only once to prevent picked or processed repeatedly.

A part found by the Locator Vision Tool (or other input tools) may be present in multiple images acquired by the camera and this tool ensures that the robot is not instructed to pick up or process the same part more than once. The input required by the Over- lap 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.

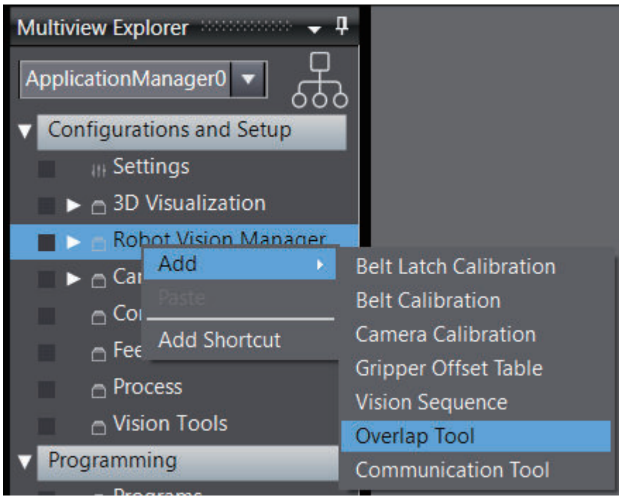
The Overlap Tool should be placed near the beginning of a sequence, just after the input tool and before any inspection tools in the sequence. This ensures that the same instance is not processed multiple times by the inspection tools. Refer to *5-5-6 Vision Sequence* on page 5-37 for details.

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. It makes it possible to get two images of the same part. In this case, the Overlap Tool can be used to filter out duplicates.

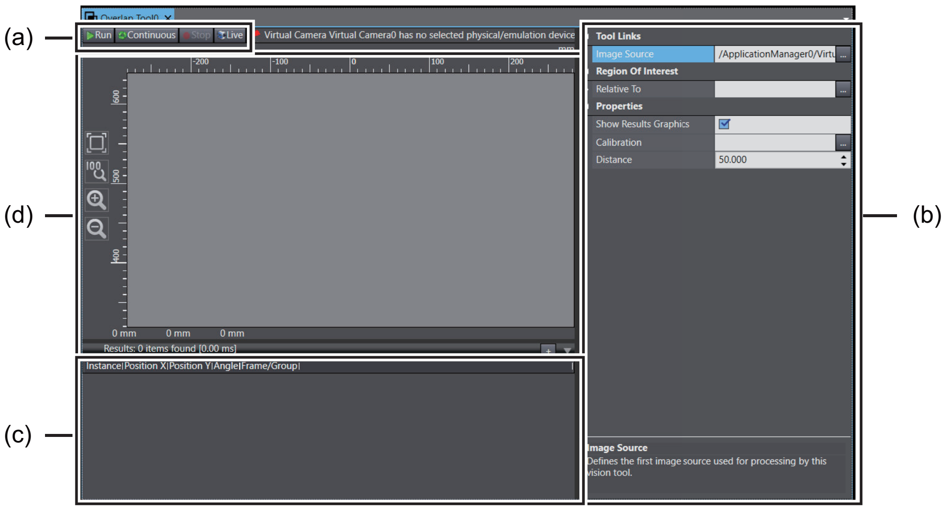
Adding an Overlap Tool

1 Right-click **Robot Vision Manager** under **Configurations and Setup** in the Multiview Explorer. Select **Add**, and then click **Overlap Tool**.



A **Overlap Tool** will be added to the **Robot Vision Manager**.

2 Right-click **Overlap Tool**, then click **Edit**.
The editor window will be displayed.



Items on the editor window are as follows.

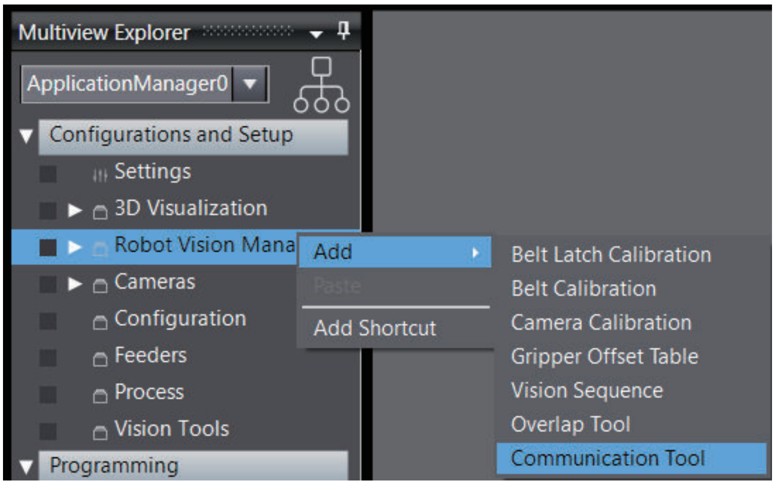
	Item				Description
(a)	Run				Run the tool once.
	Continuous				Run the tool continuously.
	Stop				Stop the tool or **Live** operation.
	Live				Show live images without running the tool.
(b)	Configuration	Tool Links	Image Source		Specify an image source for the tool.
		Properties	Calibration		Specify the robot-to-belt camera calibration for calculations.
			Distance		The minimum distance required between parts. The unit is 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		Check this box to show images on the Vision Window.
		Region of Interest			Specify a vision tool for inputs.
				Relative To	Specifies the data source.
				Link Name	The linked property name.
(c)	Instance				Inspection results are displayed.
(d)	Vision Window				Images specified in **Image Source** are shown.

3 Make the following settings.
The configured Overlap Tool should be placed between the input tool which begins the sequence and the inspection tools in the sequence.
Refer to *5-5-6 Vision Sequence* on page 5-37 for how to configure a Vision Sequence.

5-5-8 Communication Tool

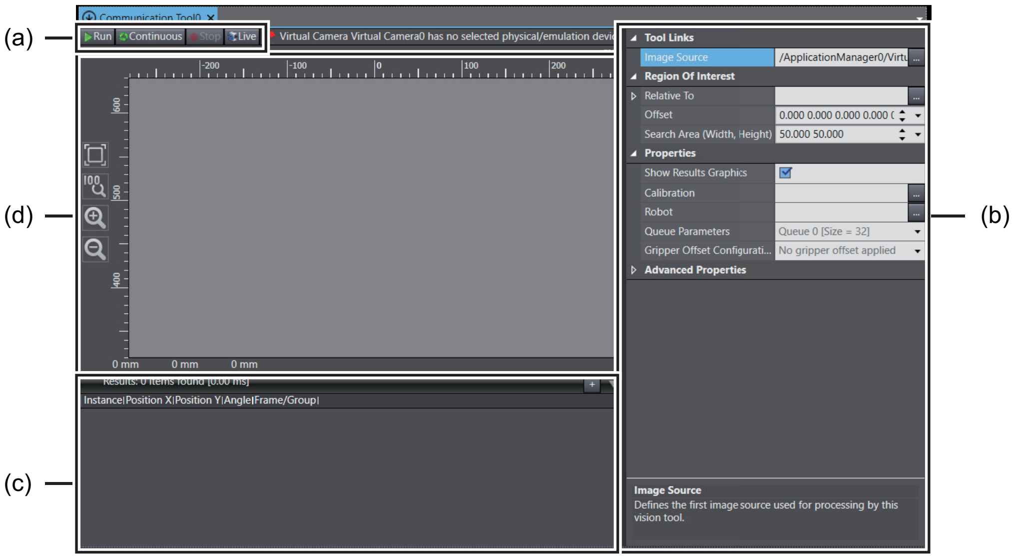
The Communication Tool is called in a Vision Sequence. It sends conveyor-associated instances from the Vision to the Robot Integrated CPU Unit. The Communication Tool typically receives instances from an Overlap Tool, which prevents different images of the same instance from being interpreted as different instances.

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



A **Communication Tool** will be added to the **Robot Vision Manager**.

2 Right-click **Communication Tool**, then click **Edit**.
The editor window will be displayed.



Items on the editor window are as follows.

	Item	Description
(a)	Run	Run the tool once.
	Continuous	Run the tool continuously.
	Stop	Stop the tool or **Live** operation.
	Live	Show live images without running the tool.

	Item			Description
(b)	Configuration	Tool Links	Image Source	Specify an image source for the tool.
		Properties	Calibration	Specify the robot-to-camera calibration for calculations.
			Queue Parameters	Configure the destination queue to send inspection results. Set the following parameters. • **Queue Index**: Identifies the queue to which instances will be sent. • **Queue Size**: Specifies the number of instances that can be written to the queue. Choose a value from 1 to 100. • **Queue Update**: Specifies how often the Communication Tool will write new instance data to the queue by selecting either of **After Every Instance** and **After Last Instance**. The recommended setting is **After Every Instance**. • **Use Soft Signal**: Check this box to enable the **Soft Signal**. • **Soft Signal**: Sets the value of the Soft Signal. The signal can be used by a V+ program to synchronize the controller and the PC. This signal notifies the controller that all instances detected by the input tool have been sent to the controller.
			Gripper Offset Configuration	Configure an offset in the Gripper Offset Table. This offset is applied to results. Without this parameter, Gripper Offset will not be applied. Make the following settings. • **Selection Mode**: Make one of the following selections: **Disabled**, **Use Default**, and **Use Model Identifier**. • **Default Offset Index**: The value of this parameter will be used when **Use Default** is selected in **Selection Mode**. • **Transform**: Specify a transformed value returned from the tool. • **Model offset Index**: The value of this parameter will be used when **Use Model Identifier** is selected in **Selection Mode**.
			Robot	Specify a robot to which the instances will be sent.
			Show Results Graphics	Check this box to show images on the Vision Window.
		Advanced Properties	Tool Relative Coordinates	Check this box to return the location relative to the robot tip position when the image was taken.
		Region of Interest	Offset	Specify the region of interest center and rotation with values of X, Y, and angle.
			Search Area (Width, Height)	The region of interest is defined by the width and height.
			Relative To	Specify an input tool for this tool.
			Tool Links	The linked property name.

	Item	Description
(c)	Results	Instances are shown.
(d)	Vision Window	Images specified in **Image Source** are shown.

3 Make the following settings.
The configured Communication Tool should be placed after the Overlap Tool, receiving instances from the tool or after the inspection tools in the sequence.
Refer to *5-5-6 Vision Sequence* on page 5-37 for details on how to configure the Vision Sequence settings.

Communication Tool Parameter Details

The following provides useful setting examples of the 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 Robot Vision Manager 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.

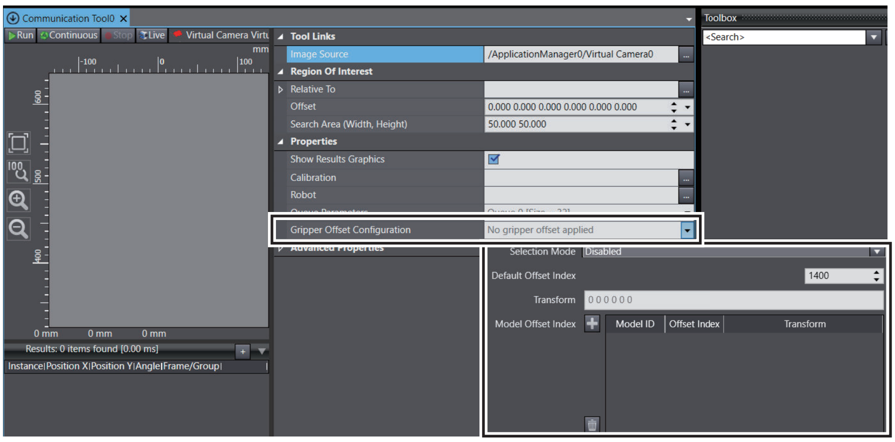
Both Queue Update settings have the same function when only one instance is found.

● 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* on page 5-34 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

Default Offset Index 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 per- formed with the calibration object attached to the robot tool (e.g. an upward looking camera).

5-6 Camera

Add cameras to your robot system. In a robot system, the vision tools processes images shot by cameras, and the processed images are used to identify parts.
Emulation cameras and Visualizer Capture Devices are added and used instead of actual cameras in offline simulation.
There are the following types of cameras.

Camera	Description	Reference
Virtual Camera	A Virtual Camera provides an interface between the vision tools and camera images. It generates input images to the vision tools from part images shot with actual cameras, or images from emulation cameras or Visualizer Capture Devices.	*5-6-1 Virtual Camera* on page 5-47
Emulation Camera	An Emulation Camera is a stored collection of image data, which has been shot in advance. Read out of the Virtual Camera, It can be treated as if it were coming from a physical camera.	*5-6-2 Emulation Cameras* on page 5-53
Visualizer Capture Device	A Visualizer Capture Device is image data that captured the 3D shape data in the 3D Visualizer. Read out of the Virtual Camera, it can be treated as if it were coming from a physical camera.	*5-6-3 Visualizer Capture Device* on page 5-55
Basler Camera	A camera made by Basler AG. It is used to acquire images. You can use images shot by the Basler camera that assigned to the Virtual Camera.	*5-6-4 Basler Cameras* on page 5-64
Sentech Camera	A camera made by OMRON Sentech. It is used to acquire images. You can use images shot by the Sentech camera that assigned to the Virtual Camera.	*5-6-5 Sentech Cameras* on page 5-69
Custom Device	The Custom Device processes image data from external cameras or the vision system through C# scripts. It is assigned to the Virtual Camera to use.	*5-6-6 Custom Device* on page 5-70

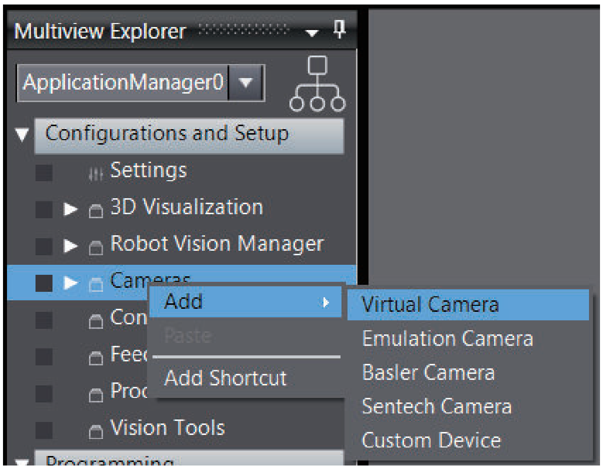
5-6-1 Virtual Camera

A Virtual Camera provides an interface between the vision tools and camera images. It generates input images to the vision tools from part images shot with actual cameras, or images from emulation cameras or Visualizer Capture Devices.

Precautions for Correct Use

To use the Virtual Camera, it must be added and assigned to a newly added physical camera, or a newly created emulation device or Visualizer Capture Device. Refer to *In the Offline State* on page 5-65, *5-6-5 Sentech Cameras* on page 5-69, *5-6-2 Emulation Cameras* on page 5-53, and *5-6-3 Visualizer Capture Device* on page 5-55 for details.

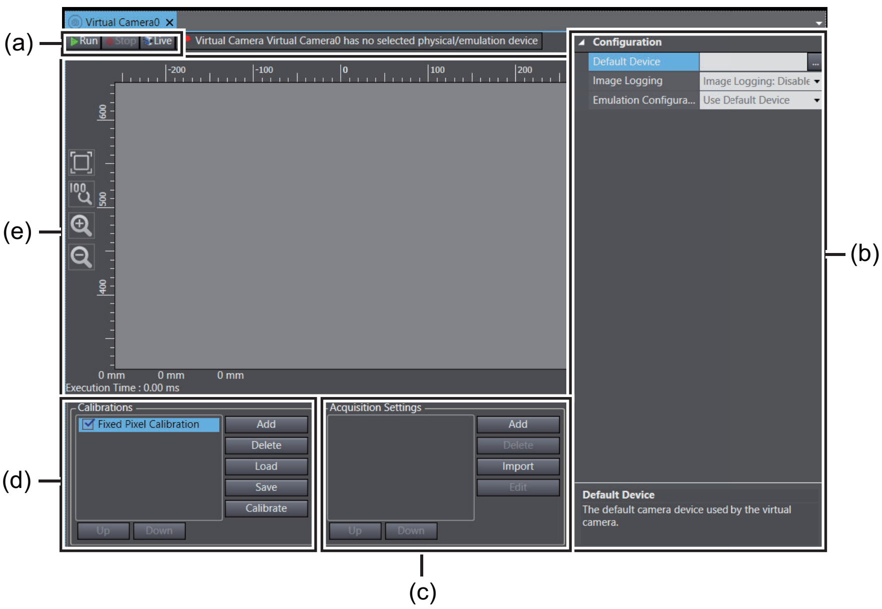
1 Right-click **Cameras** in **Configurations and Setup** under the Multiview Explorer. Select **Add**, and then click **Virtual Camera**.



Virtual Camera will be added under **Cameras**.

2 Right-click **Virtual Camera**, then click **Edit**.

The Edit Pane is displayed.



Items on the Edit Pane are as follows.

	Item	Description
(a)	Run	Acquires an image once from the camera device specified.
	Stop	Stops continuous Live image acquisition.
	Live	Starts continuous image acquisition from the camera device specified.

	Item		Description
(b)	Configuration	Default Device	Select a default camera device used by the Virtual Camera.
		Image Logging	Save the image data for offline settings or adjustments. Make the following settings. • **Enabled**: Check the box to enable the image logging. • **Directory Name**: Select a directory where images will be stored. • **Image Count**: Enter the number of images to store. Up to 1000 images can be stored.
		Emulation Configuration	You can setup the camera running in the Emulation mode. Select an **Emulation Configuration Behavior** from the following to setup. • **Use Default Device**: This setting will use the device specified in the **Default Device** field. • **Random Instances**: Specifies the **minimum value** and **maximum value**. This setting allows acquisition of random quantity of randomly oriented vision results. • **Use Alternative Device**: Specifies other available **camera devices**. • **Images Replay**: This setting allows acquisition of images by specifying **Directory Name**. .hig files only.
(c)	Acquisition Settings		This area is used to adjust image acquisition settings for the camera. Refer to *Acquisition Settings* on page 5-51 for more information.
(d)	Calibrations		This area is used to calibrate the Virtual Camera. Refer to *Virtual Camera Calibration* on page 5-49 for more information.
(e)	Vision Window		Images specified in **Default Device** are shown.

Virtual Camera Calibration

Virtual camera calibration is required before you use vision tools: adjustment for perspective or 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 a Sysmac Studio installation. Find these in the default installation directory with the following file names:

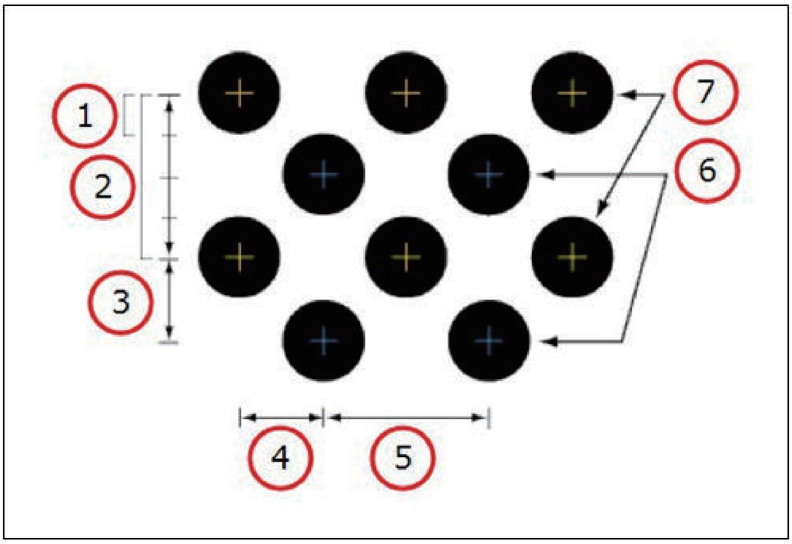
- DotPitchOthers_CalibrationTarget.pdf
- DotPitch10_CalibrationTarget.pdf

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 or Visualizer Capture Device, 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 con- figured 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.

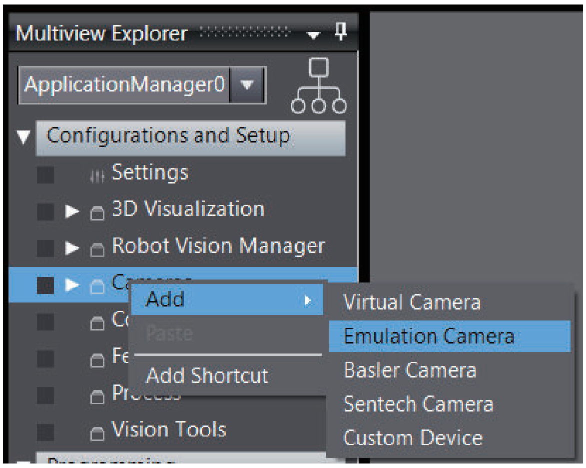
● Other Acquisition Settings Functions

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 Application Manager.
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.

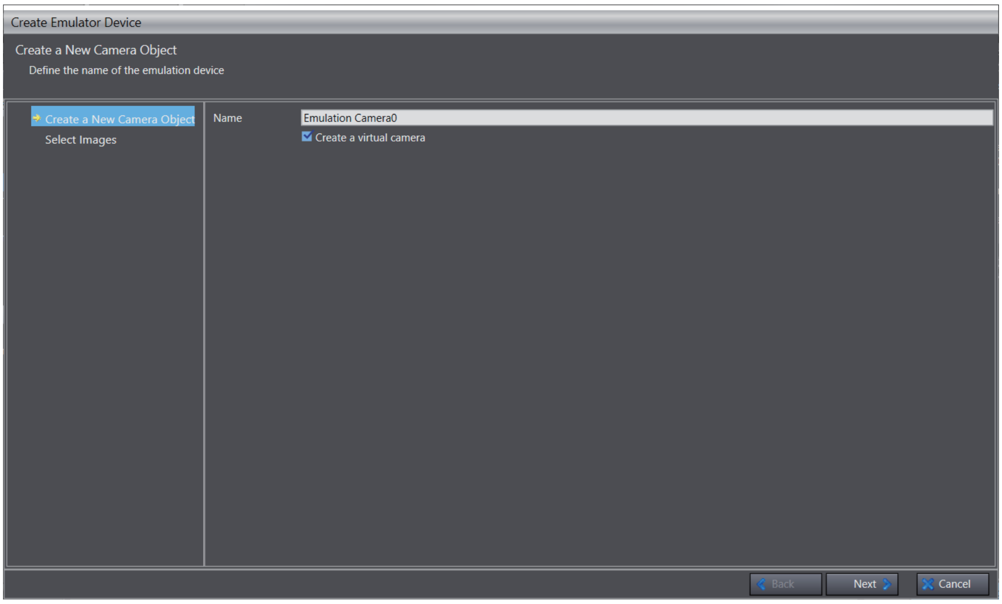
5-6-2 Emulation Cameras

An Emulation Camera is a stored collection of image data, which has been shot in advance. Assigned to the Virtual Camera, the Emulation Camera can be processed as if it were coming from a physical camera.
You can use the Emulation Camera for offline tasks, e.g., a robot system simulation before making an actual equipment, or image display from an application in a remote equipment.

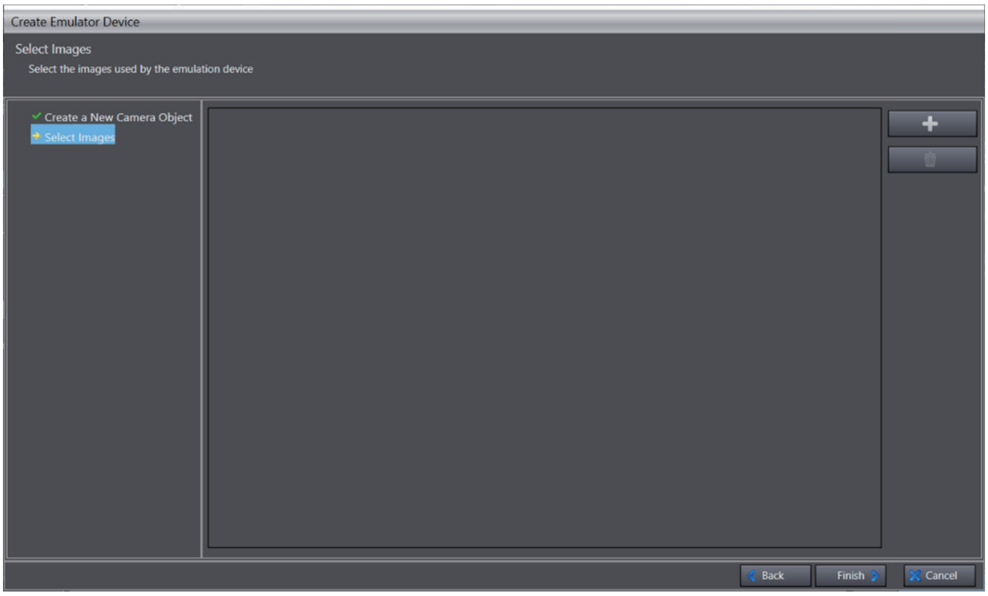
1 Right-click **Cameras** in **Configurations and Setup** under the **Multiview Explorer**. Select **Add**, and then click **Emulation Camera**.



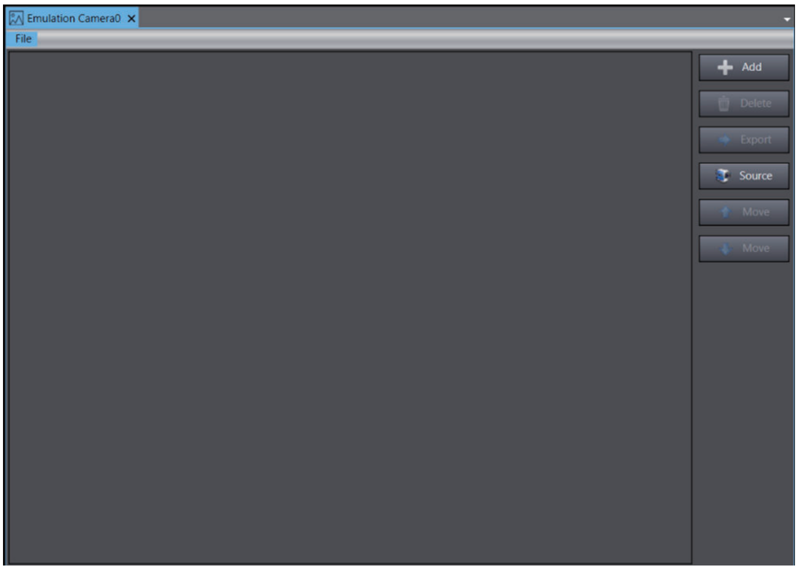
The **Create Emulator Device** window will open.



2 Provide a name for the Emulation Camera, then click the **Next** button.
Check the **Create a virtual camera** box to exchange images between physical cameras or Emulation Cameras through the Virtual Camera. Refer to *5-6-1 Virtual Camera* on page 5-47 for details.
The **Select Images** window is displayed.



3 Select an image for the Emulation Camera, then click the **Finish** button.
Emulation Camera will be added under **Cameras**.
The selected image is registered to **Emulation Camera**. You can add images to the Emulation Camera. Also can export and edit the Emulation Cameras.
Right-click the **Emulation Camera** (number) you want to edit. Then click **Edit** to display the edit window.



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

5-6-3 Visualizer Capture Device

Use the Visualizer Capture Device to capture objects in the 3D Visualizer. Selecting the Visualizer Capture Device for Virtual Camera allows you to process image data as if a physical camera captured it. It is used to perform offline simulations before making physical equipment. The Visualizer Capture Device always captures images in parallel projection.
You need to perform the following procedure in advance.

1. Add a Virtual Camera.
   Refer to *5-6-1 Virtual Camera* on page 5-47 for details.

2. Add a robot.
   To associate a robot with the Visualizer Capture Device, you need to add it. Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for details.

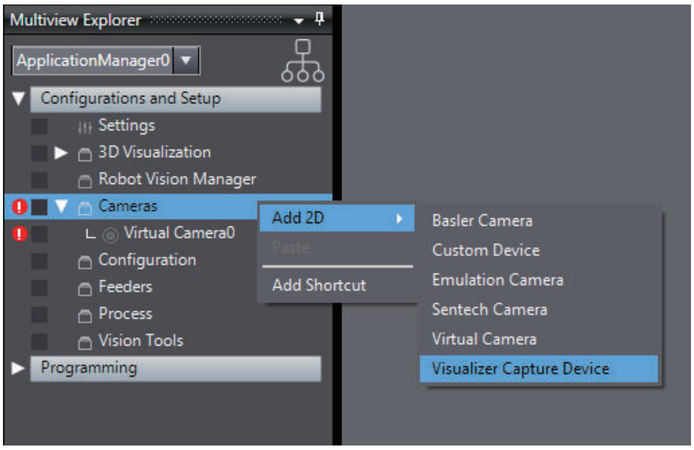
3. Add an IO EndEffector.
   To associate an IO EndEffector with the Visualizer Capture Device, you need to add it. Refer to *4-4-1 Setting Up IO EndEffectors* on page 4-10 for details.

4 Add Camera Calibration.
Refer to *5-5-4 Camera Calibration* on page 5-27 for details.

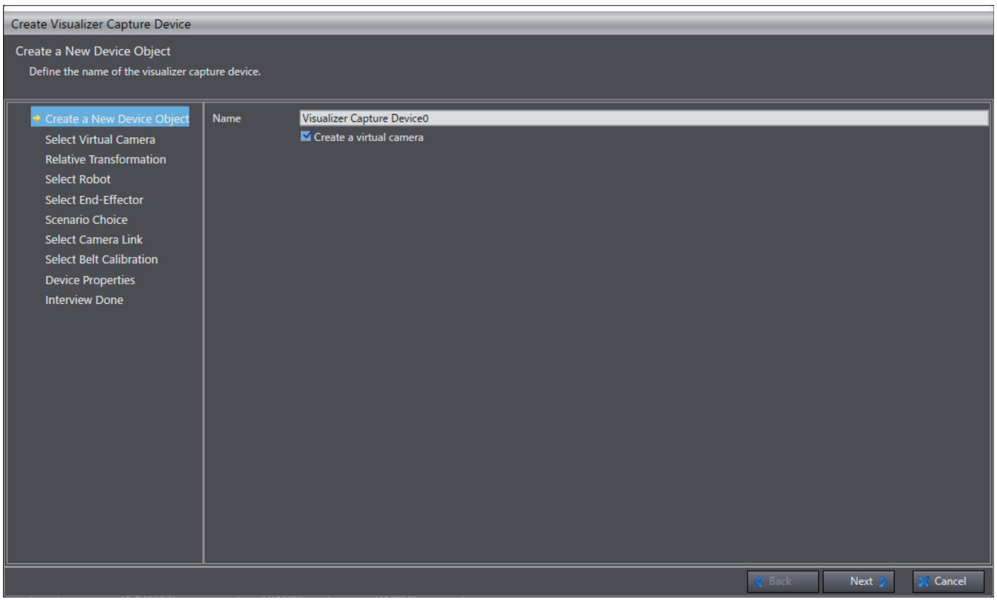
Adding a Visualizer Capture Device

The following describes how to add a Visualizer Capture Device.

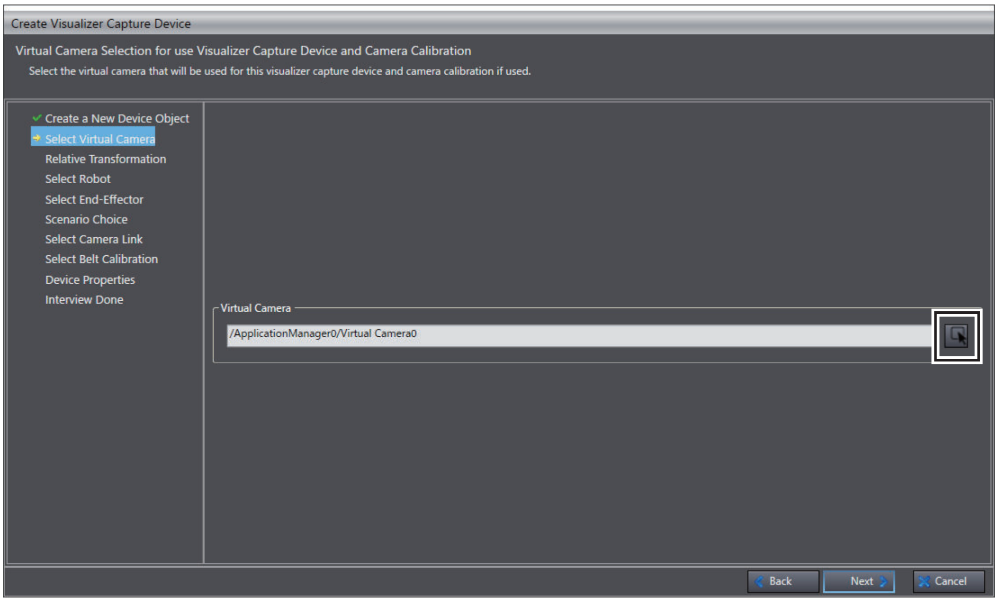
1 Right-click **Cameras** under **Configurations and Setup** in the **Multiview Explorer** and select **Add 2D** – **Visualizer Capture Device**.



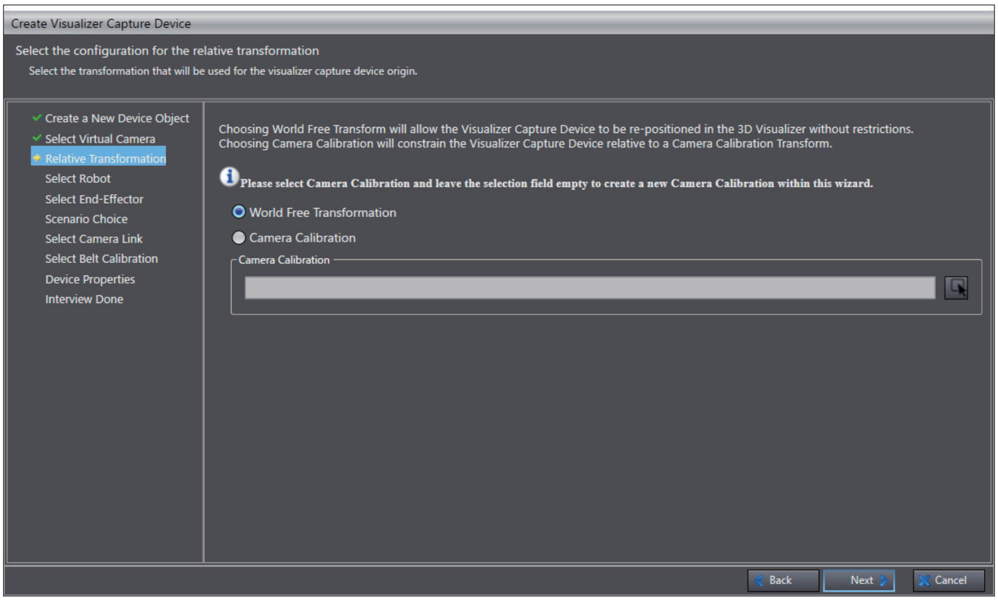
The **Create Visualizer Capture Device** wizard starts and the **Create a New Device Object** page is displayed.



2 Here, clear the **Create a virtual camera** check box and then click the **Next** button.
The **Virtual Camera Selection for use Visualizer Capture Device and Camera Calibration** page is displayed.

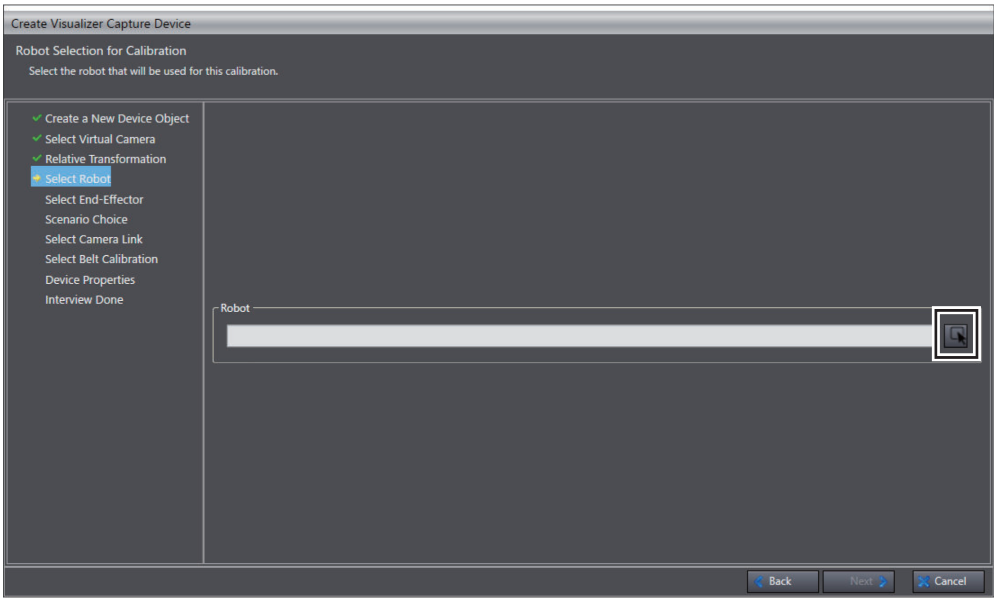


3 Click the **Virtual Camera Selection** button, select a pre-registered Virtual Camera, and then click the **Next** button.
The **Select the configuration for the relative transformation** page is displayed.

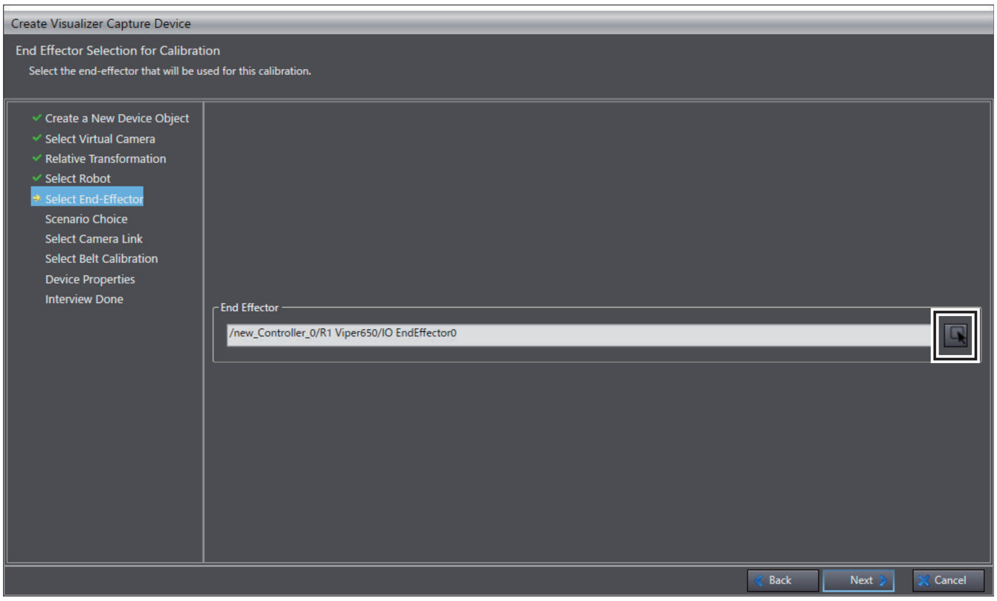


Selecting the **World Free Transformation** option directly brings you to the **Device Properties** page. Selecting the **Camera Calibration** option allows you to use a pre-set camera calibration. If you select **Camera Calibration** with the **Camera Calibration** field left empty, a new calibra-tion camera will be created.

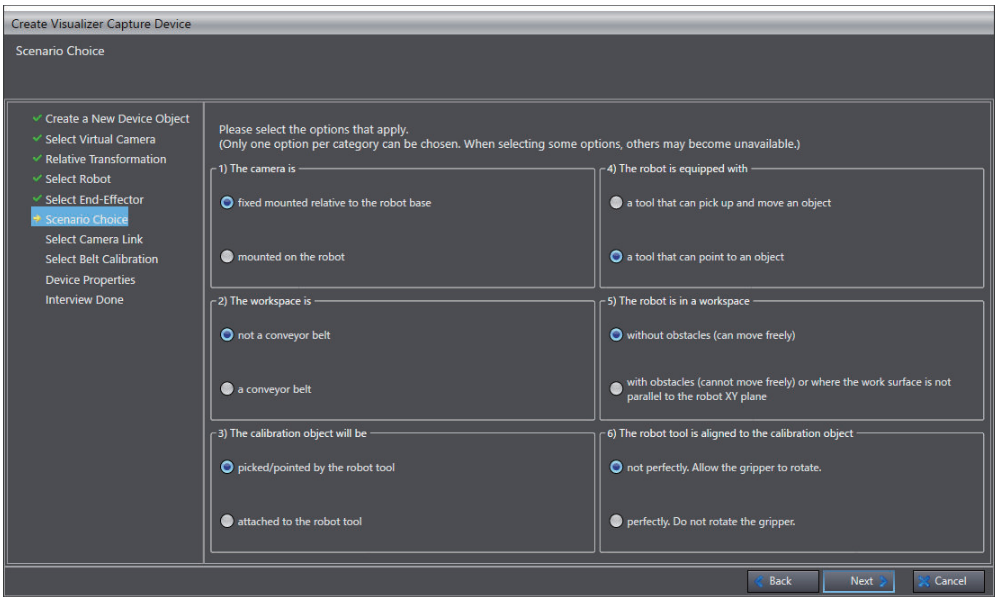
4 Here, select **Camera Calibration** and then click the **Next** button.
The **Robot Selection for Calibration** page is displayed.



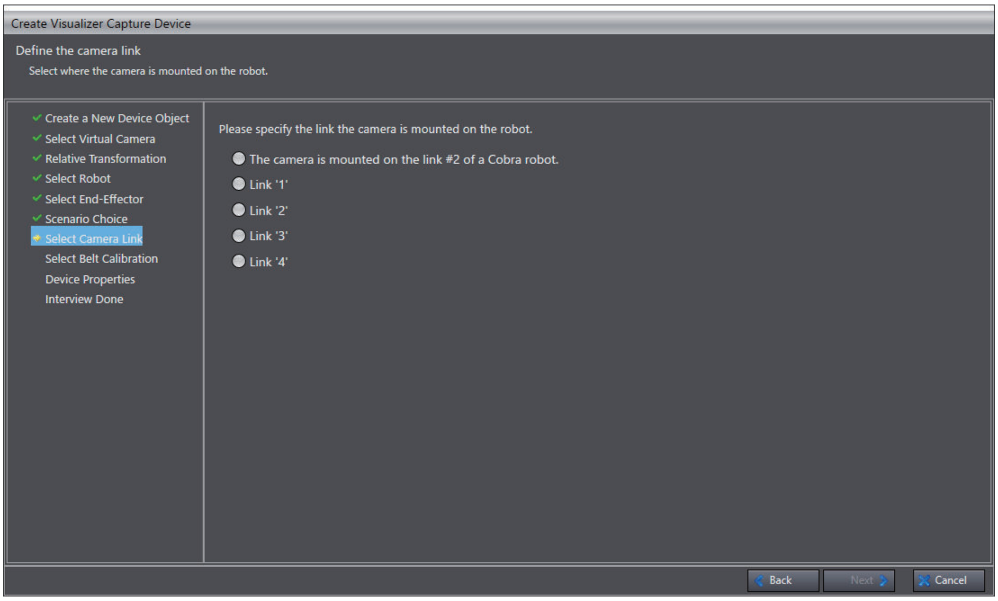
5 Click the **Robot Selection** button, select a robot to use for calibration, and then click the **Next** button.
The **End Effector Selection for Calibration** page is displayed.



6 Click the **End Effector Selection** button, select an end effector to use for calibration, and then click the **Next** button.
The **Scenario Choice** page is displayed.

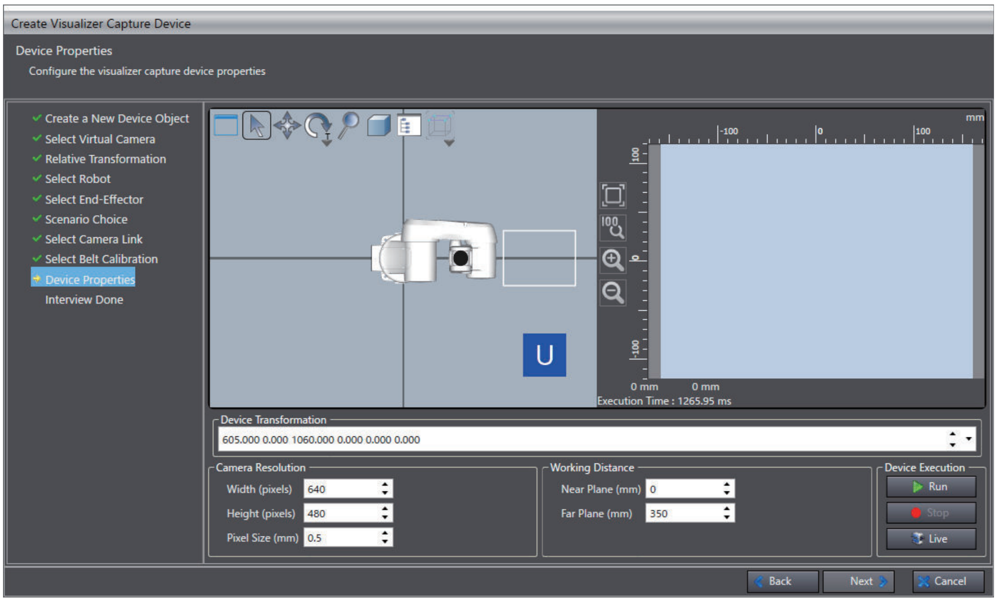


7 Select options for categories 1 to 6 in order and then click the **Next** button. Here, for **1) The camera is**, select the **mounted on the robot** option.
The **Define the camera link** page is displayed.

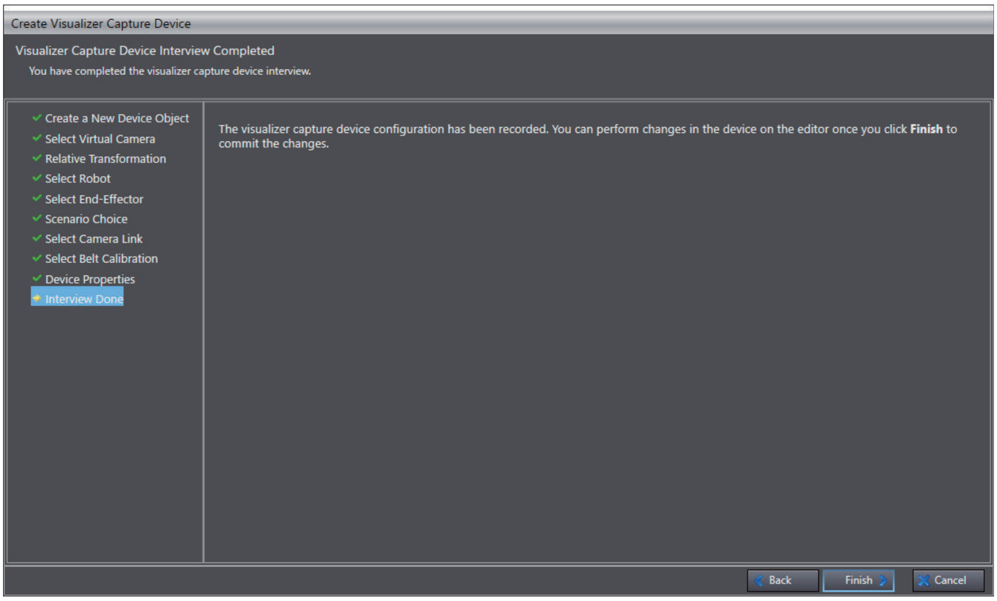


This screen is displayed only when you selected the **mounted on the robot** option for **1) camera is** on the **Scenario Choice** page.

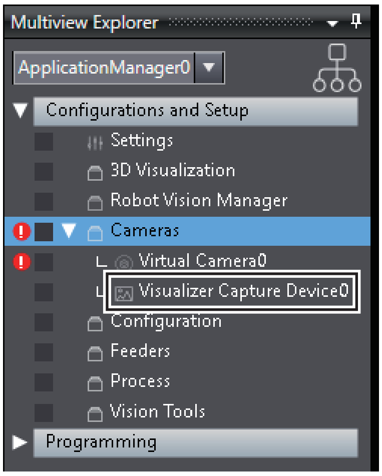
8 Specify the link to the robot on which the camera is mounted and then click the **Next** button.
The **Device Properties** page is displayed.



9 Configure the properties and then click the **Next** button.
The **Visualizer Capture Device Interview Completed** page is displayed.

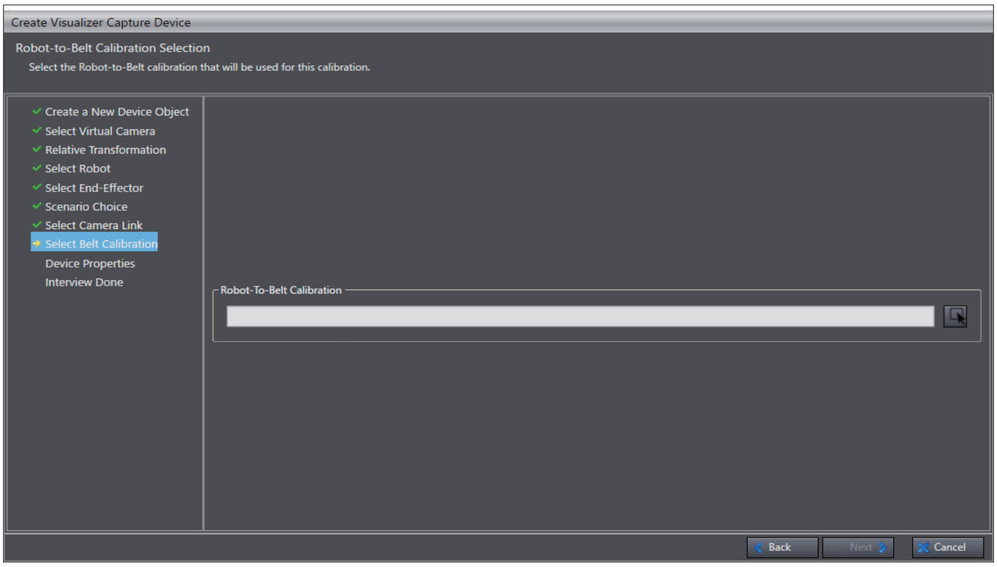


Click the **Finish** button. **Visualizer Capture Device0** is now added under **Configurations and Setup** – **Cameras** in the **Multiview Explorer**.



Additional Information

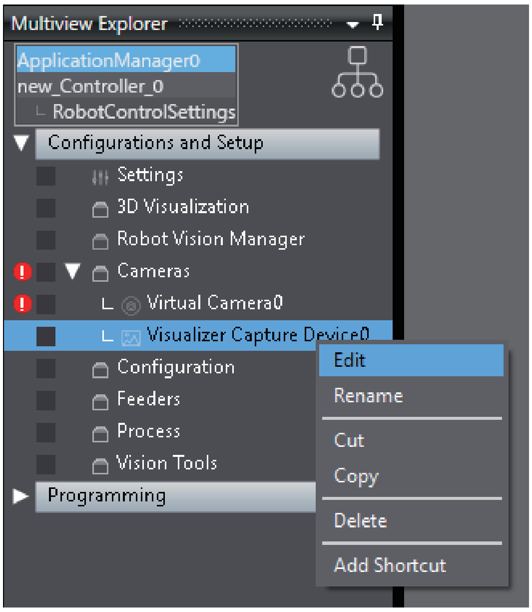
If you selected the **a conveyor belt** option for **2) The workspace is** on the **Scenario Choice** page of the **Create Visualizer Capture Device** wizard, the **Robot-to-Belt Calibration Selection** page appears. Specify the belt calibration to use for calibration on this page.



Editing the Visualizer Capture Device

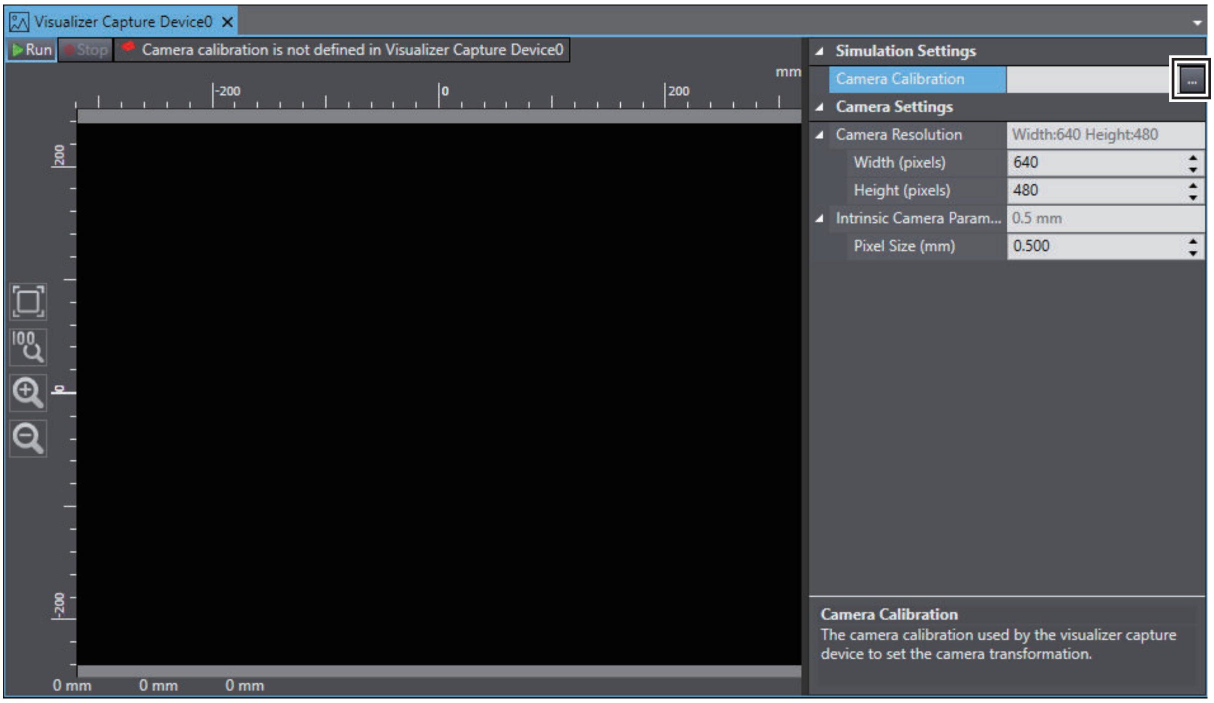
The following describes how to edit the Visualizer Capture Device.

1 Select **Configurations and Setup** in the **Multiview Explore**. Then right-click **Visualizer Capture Device0** under **Cameras**. Select **Edit**.



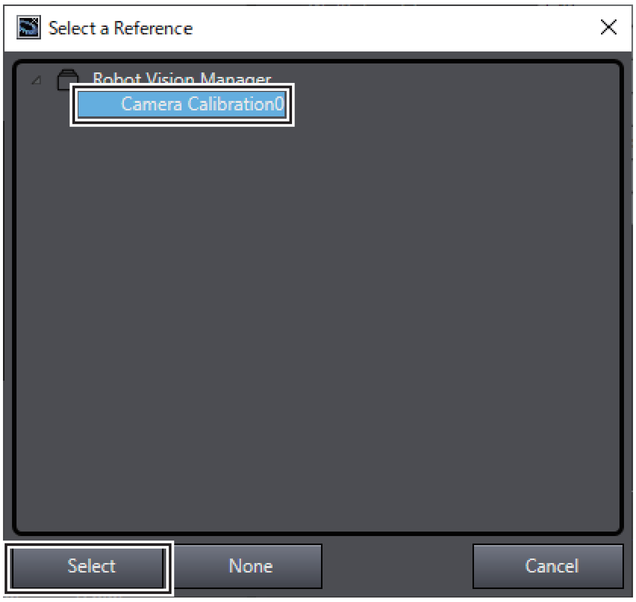
The **Visualizer Capture Device Edit** window is displayed.

2 Click the **...** button at the right of **Camera Calibration** under **Simulation Settings** on the upper right corner of the window.



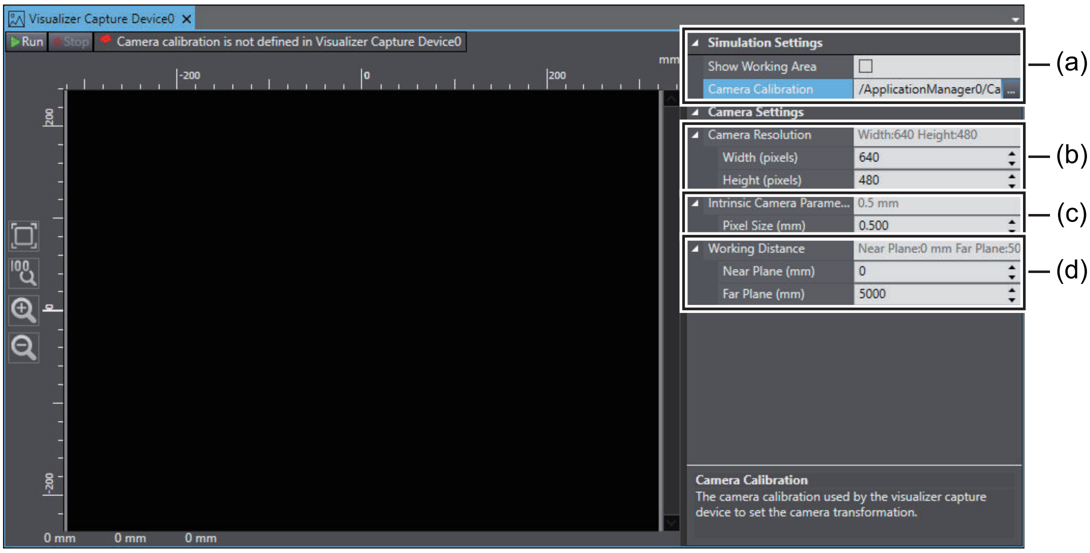
The **Select a Reference** dialog is displayed.

3 Select **Camera Calibration0** and click the **Select** button.



The **Select a Reference** dialog is closed.

4 Change the values of **Width (pixels)** and **Height (pixels)** under **Camera Resolution** and the value of **Pixel Size** under **Intrinsic Camera Parameters** in the upper right corner of the window.



	Item		Description
(a)	Simulation Settings	Show Working Area	Check this box to display the working area in the 3D Visualizer.
		Camera Calibration	Specify the camera calibration to use for the Visualizer Capture Device.
(b)	Camera Resolution	Width (pixels)	Set the image's width in pixels in the range of 1 to 4096.
		Height (pixels)	Set the image's height in pixels in the range of 1 to 4096.
(c)	Intrinsic Camera Parameters	Pixel Size (mm)	Set the pixel size in the range of 0.001 to 15.000.

	Item		Description
(d)	Working Distance	Near Plane (mm)	Set the distance to the near plane that cuts the work area perpendicular to the direction of sight in the range of 0 to 5000.
		Far Plane (mm)	Set the distance to the far plane that cuts the work area perpendicular to the direction of sight in the range of 0 to 5000.

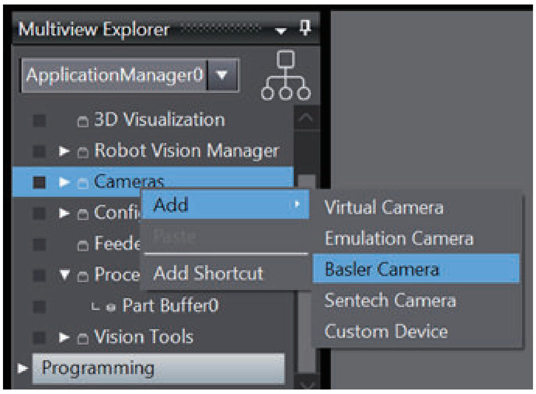
5-6-4 Basler Cameras

Cameras made by Basler AG. The are used to acquire images. You can use images shot by a Basler camera that assigned to the Virtual Camera.

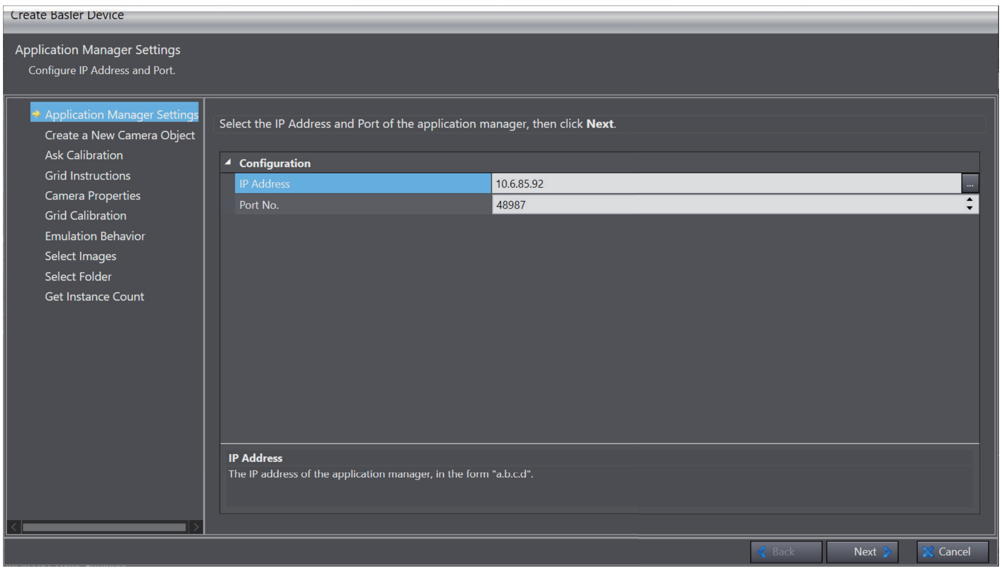
Additional Information

Robot Vision Manager supports Basler Pylon cameras.

1 Right-click **Cameras** in **Configurations and Setup** under the **Multiview Explorer**. Select **Add**, and then click **Basler Camera**.



The **Application Manager Settings** dialog is shown.

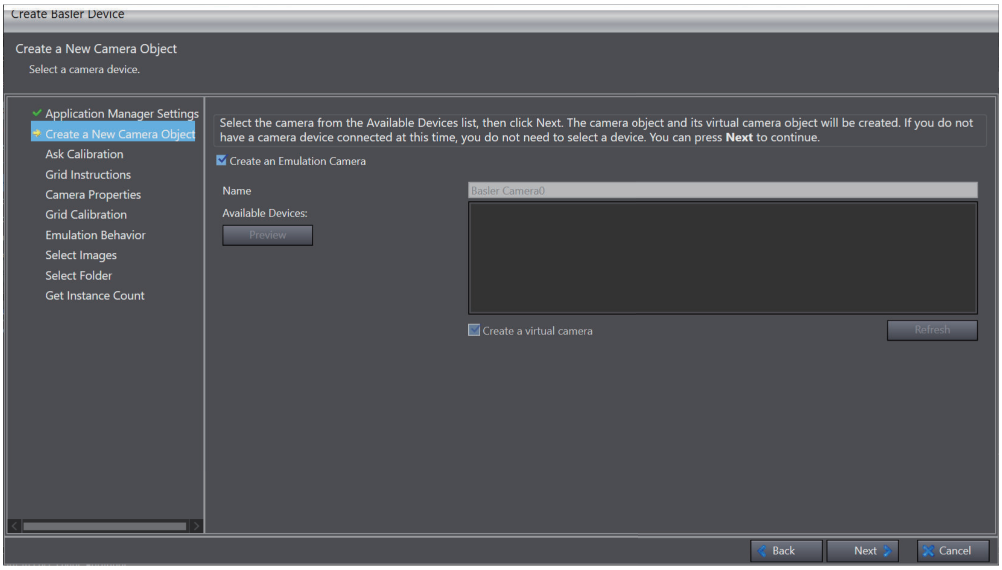


The following window screen shots differs from offline and online to an actual equipment.

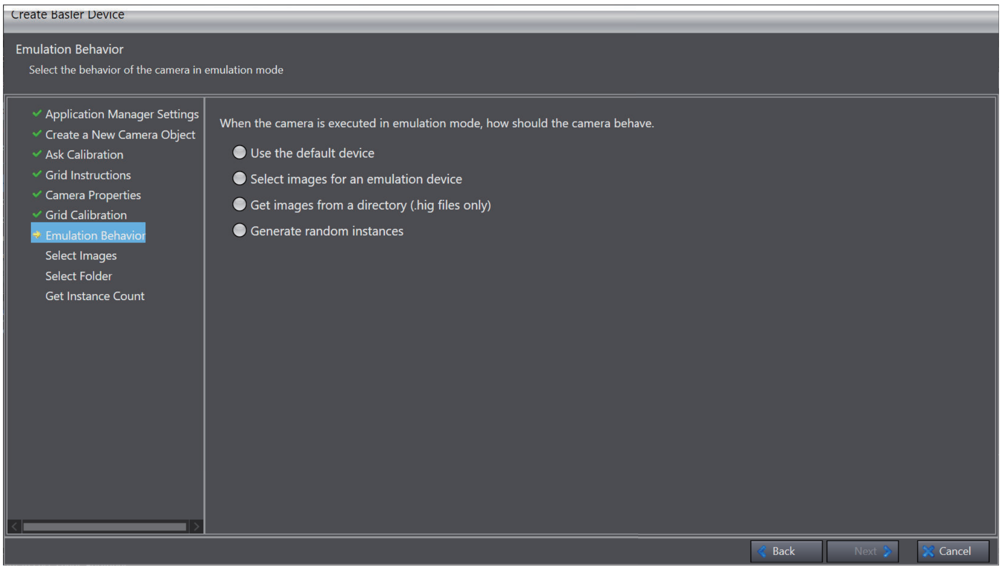
Refer to *In the Offline State* on page 5-65 and *In the Online Connection State* on page 5-66 respectively.

● In the Offline State

1 Click the **Next** button.
The **Create a New Camera Object** window will open.



2 Check the **Create an Emulation Camera** box, then click the **Next** button.
Emulation Behavior window is displayed.



3 Check a camera behavior in the Emulation mode, then click the **Next** button.
Select camera behaviors in the Emulation mode from the following.

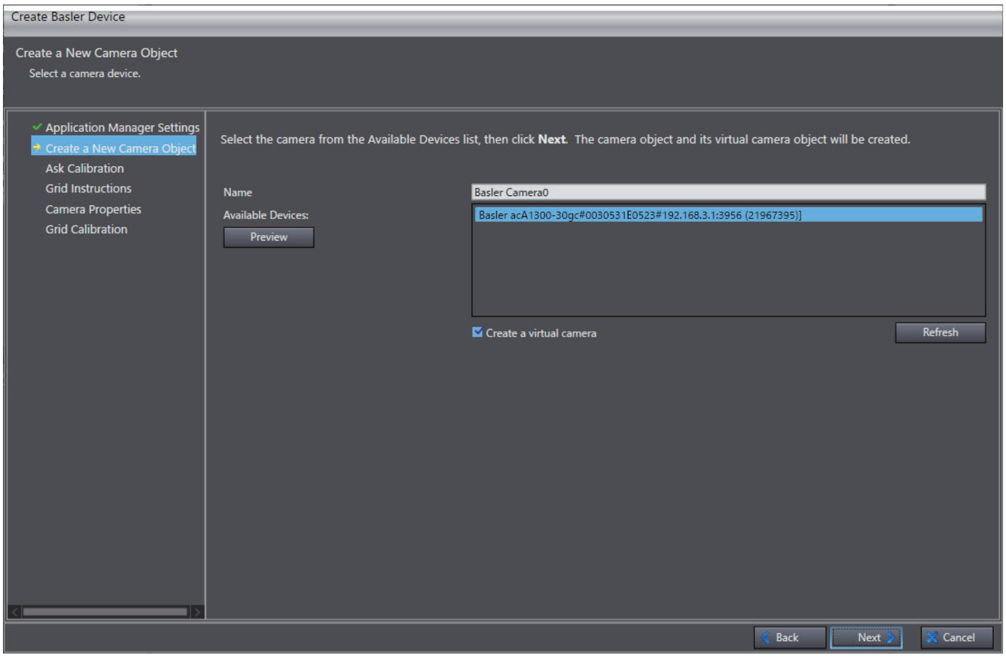
Behavior	Description
Use the default device	No need to specify a camera. An image is processed in the Emulation mode.

Behavior	Description
Select images for an emulation device	The **Select Images** dialog is displayed. Select an image to use in the Emulation mode.
Get images from a directory (.hig files only)	The **Select Folder** dialog is displayed. Specify the folder where images used in the Emulation mode are.
Generate random instances	The **Get Instance Count** dialog is displayed. Specify the minimum and maximum instances. A random number of images will be retrieved in the specified range of values.

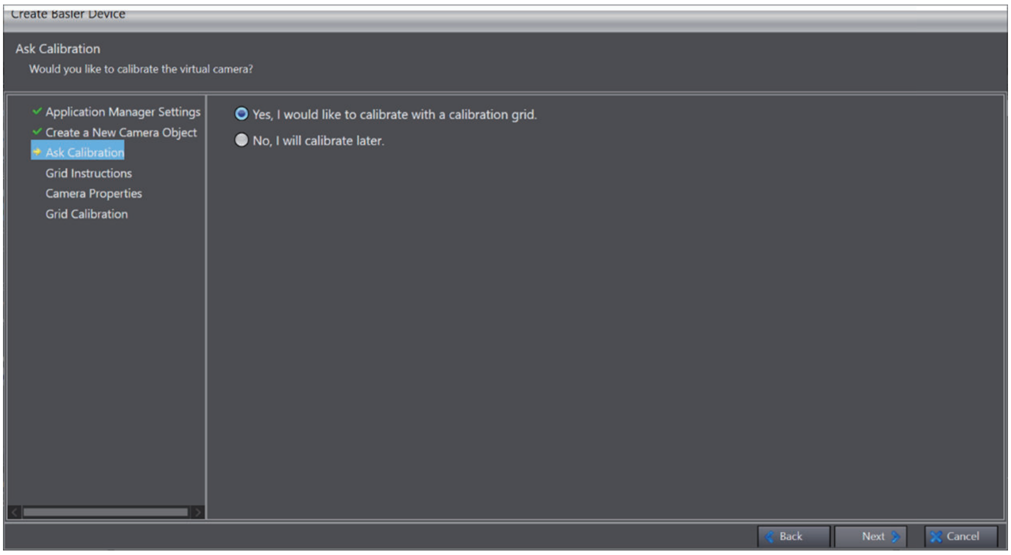
4 Configure required settings following a guidance given by the selected item. Then click the **Finish** button.
Basler Camera, **Emulation Camera**, and **Virtual Camera** will be added to **Cameras**.

● In the Online Connection State

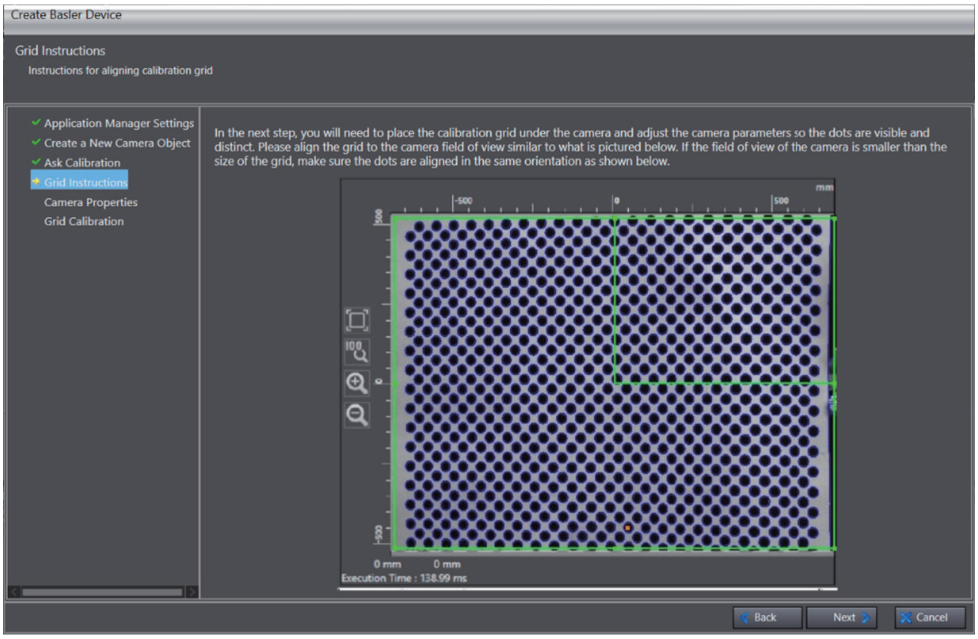
1 Display the **Select a controller address** dialog with the selection button. Then select the IPC Application Controller to which the camera connected. Click the **Next** button.
The **Create a New Camera Object** window will open.



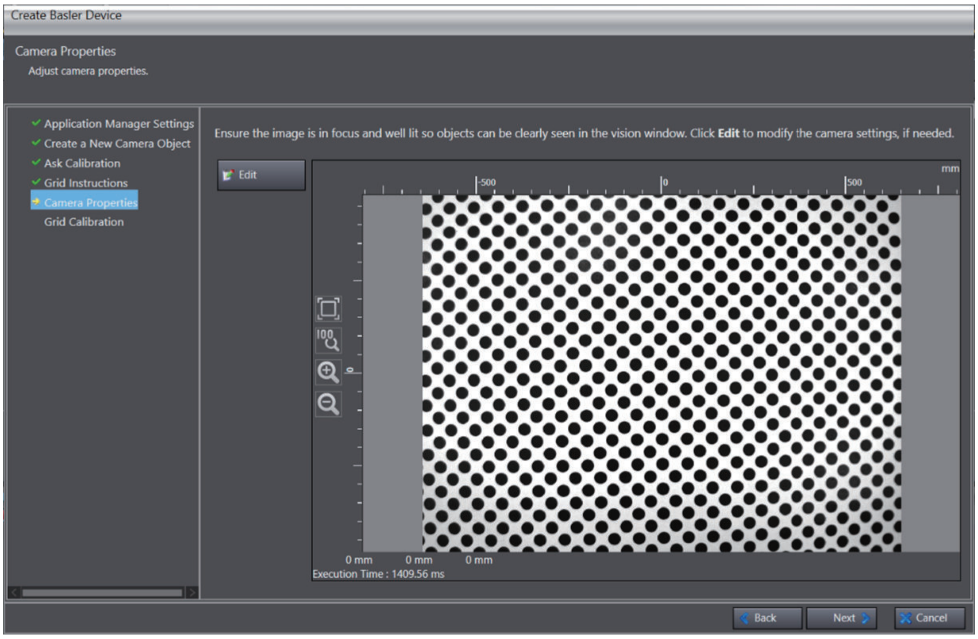
2 Select the camera to connect, then click the **Next** button.
Check the **Create a virtual camera** box to use a Virtual Camera. Refer to *5-6-1 Virtual Camera* on page 5-47 for details.
The **Ask Calibration** dialog is displayed.



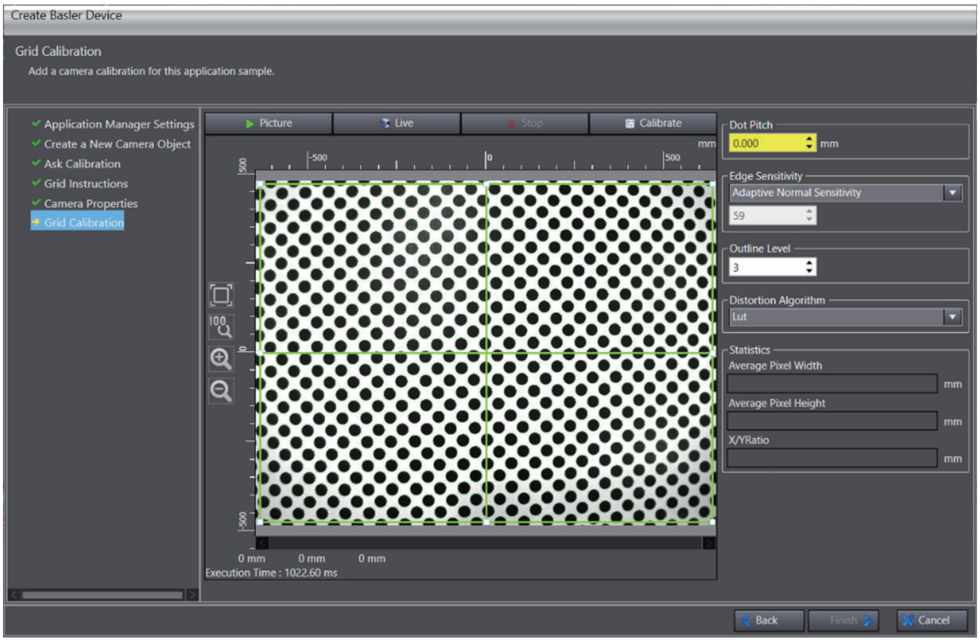
3 Select “**Yes, I would like to calibrate with a calibration grid**”, then click the **Next** button.
If you select “**No, I will calibrate later**”, go to Step 6.
The **Grid Instructions** window opens.



4 Read the directions and click the **Next** button.
The **Camera Properties** window is displayed.



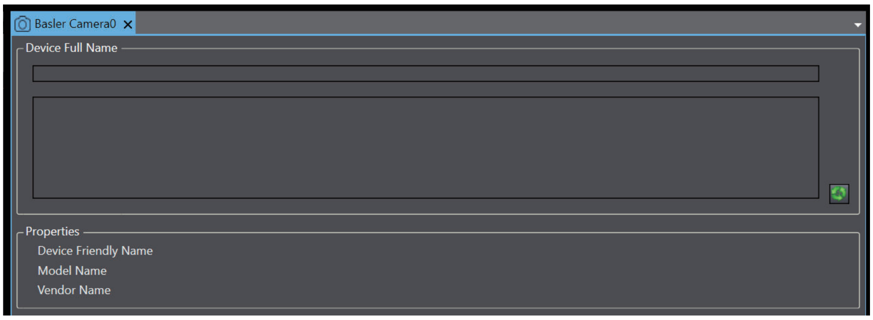
5 If necessary, click the **Edit** button to modify the camera settings, then click the **Next** button. The **Grid Calibration** dialog is displayed.



6 Shoot the calibration sheet and do a calibration. Then click the **Finish** button.
The **Basler Camera** will be added to **Cameras**. If you have checked the **Create a virtual camera** box, a **Virtual Camera** will be added.

Basler Camera Configuration

The properties of a selected camera is registered to a **Basler Camera**.
Right-click the **Basler Camera** then select **Edit** to see properties of the connected camera.

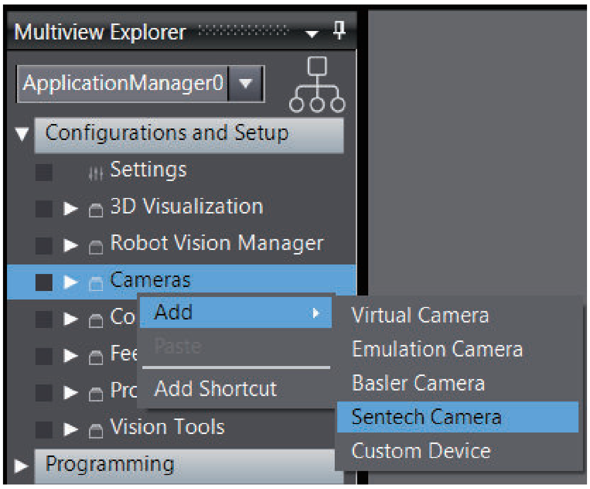


Displays vary depends on the type of the selected Basler camera. Refer to Basler camera manuals for more information.

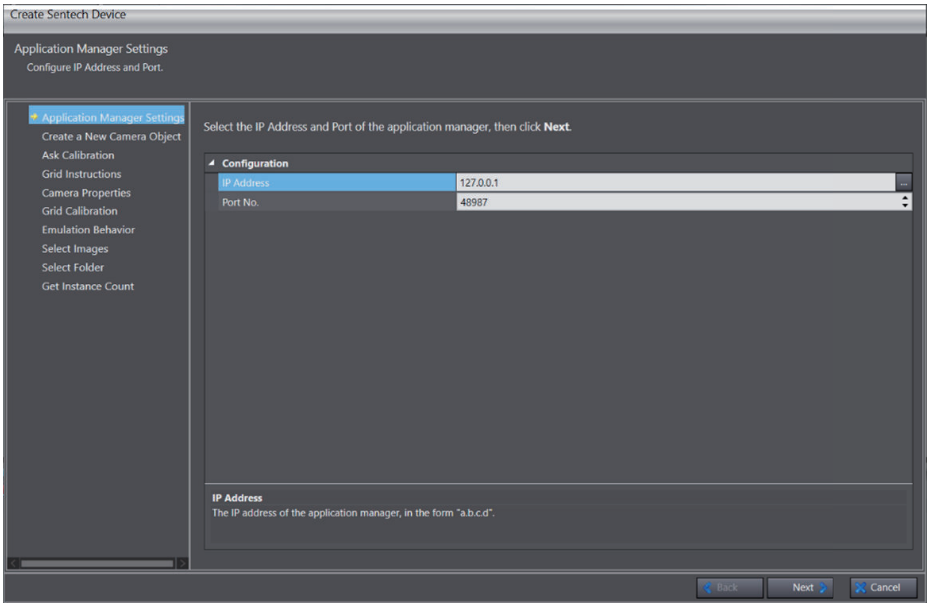
5-6-5 Sentech Cameras

Cameras made by OMRON Sentech. The are used to acquire images. You can use images shot by a Basler camera that assigned to the Virtual Camera.

1 Select **Configurations and Setup** in the **Multiview Explorer**, then right-click **Cameras**. Select **Add**, and then click **Sentech Camera**.



The **Create Sentech Device** window opens.



2 Display the **Select a controller address** dialog with the selection button. Then select the IPC Application Controller to which the camera connected. Click the **Next** button.
Operations for the subsequent screens are the same as those for adding a *Basler Camera*. Refer to *5-6-4 Basler Cameras* on page 5-64.
Clicking the **Finish** button adds a **Sentech Camera** to **Cameras**. An **Emulation Camera** and a **Virtual Camera** will be added according to the settings.

Sentech Camera Configuration

The properties of a selected camera is registered to a **Sentech Camera**.
Right-click the **Sentech Camera** then select **Edit** to see properties of the connected camera.
Displays vary depends on the type of the selected Sentech camera. Refer to Sentech camera manuals for more information.

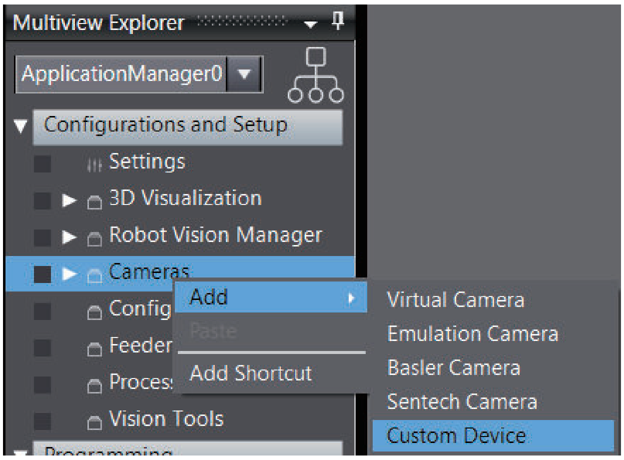
5-6-6 Custom Device

The Custom Device processes image data from external cameras or the vision systems through C# scripts. It is assigned to the Virtual Camera to use.

Additional Information

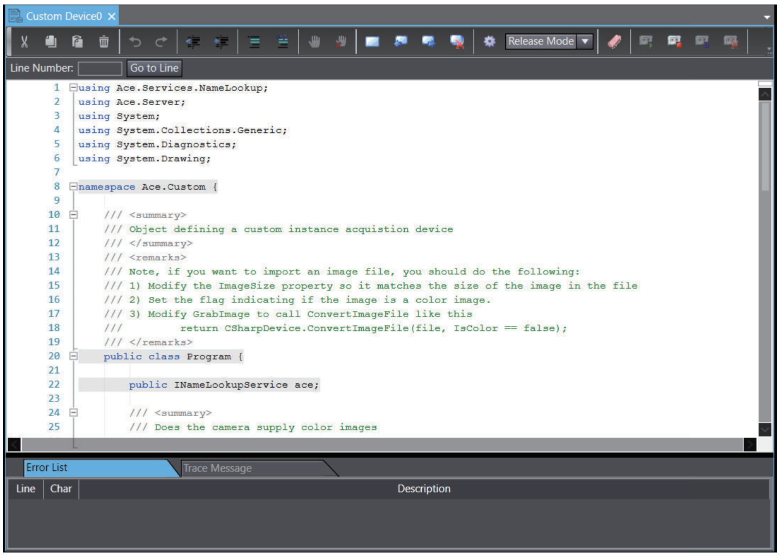
Refer to manuals of the external cameras or vision systems for procedures to detect images with them.

1 Select **Configurations and Setup** in the **Multiview Explore**. Then right-click **Cameras**. Select **Add**, and then click **Custom Device**.



A new **Custom Device** will be added to **Cameras**.

2 Right-click the **Custom Device()**, then click **Edit**.
The editor window will be displayed.



Refer to *5-11-1 C# Program Editor* on page 5-192 for how to use the C# Program editor.

5-7 Configuration

You can configure the Application Manager's settings, such as production recipe creation, Recipe Manager used for management, and Data Mapping which relates inputs and outputs in the workspace. The setting items are as follows.

Item	Description	Reference
Controller Connection Startup	Configure an automatic connection to the Robot Integrated CPU Unit or program execution when the Application Manager on the IPC Application Controller is started.	*5-7-1 Controller Connection Startup* on page 5-72
Data Mapper	Data Mapper provides a method to associate different data items within the Sysmac Studio project.	*5-7-2 Data Mapper* on page 5-74
Note	Note object provides a means for creating documentation. Use this object to create a note about an application.	*5-7-3 Note* on page 5-76
OPC Container	Create an 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 OPC Container* on page 5-77
Program System Startup	Configure settings regarding startup of the Robot Integrated Systems. The Program System Startup object is used to specify a C# program to run when the Robot Integrated System project opens.	*5-7-5 Program System Startup* on page 5-80
Recipe Manager	Recipe Manager is used to define every source that is to be used for creating a recipe.	*5-7-6 Recipe Manager* on page 5-81
Recipe Manager Script	A Recipe Manager Script is a C# program invoked when certain events occur in the lifetime of a Recipe.	*5-7-7 Recipe Manager Script* on page 5-90

5-7-1 Controller Connection Startup

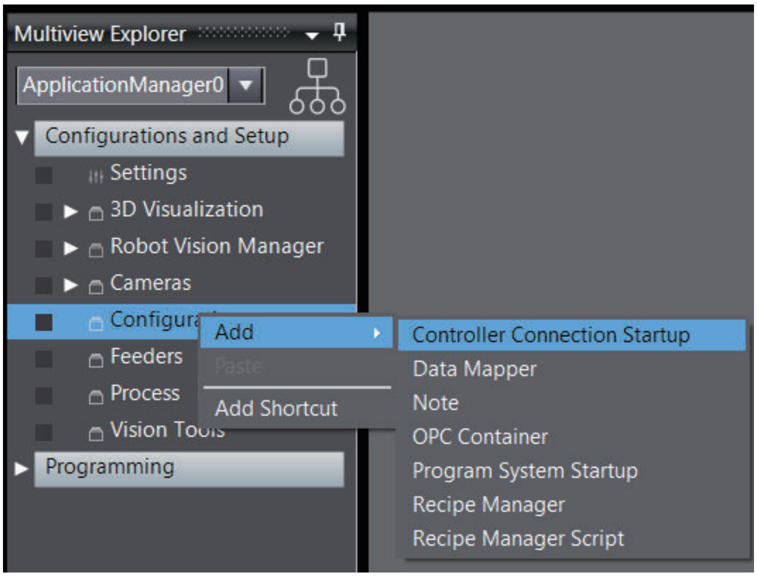
Controller Connection Startup enables an automatic connection to at least one Robot Integrated CPU Unit when the Application Manager on the IPC Application Controller is started. It can also be configured to start a V+ program on a specified task number upon startup.

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

⚠ WARNING

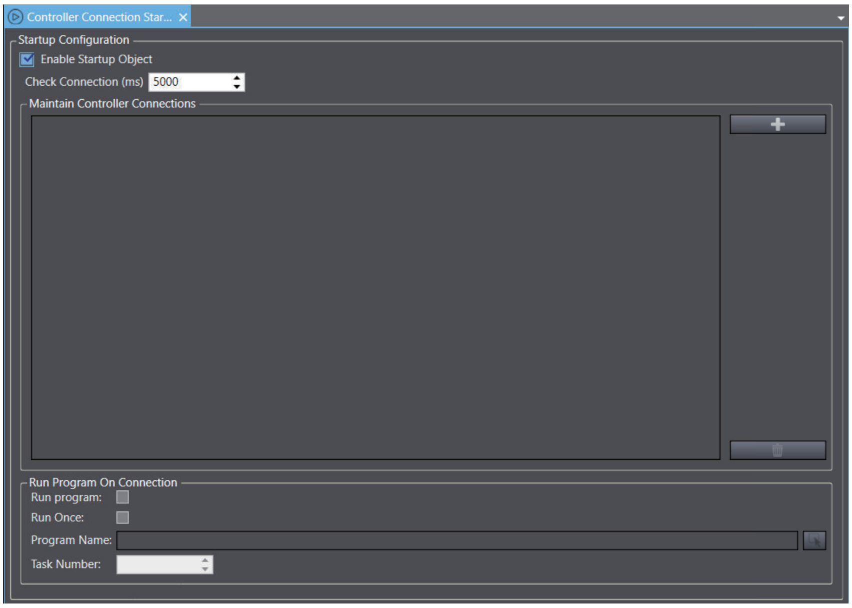
If an execution program and task number are specified and the automatic execution of V+ program is enabled, it may possibly happen that the robot operates after the CPU Unit and robot are turned on. Make sure that the movement of the robot does not cause a danger.

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



A new **Controller Connection Startup** will be added to **Configuration**.

2 Right-click the **Controller Connection Startup**, then click **Edit**.
The editor window will be displayed.



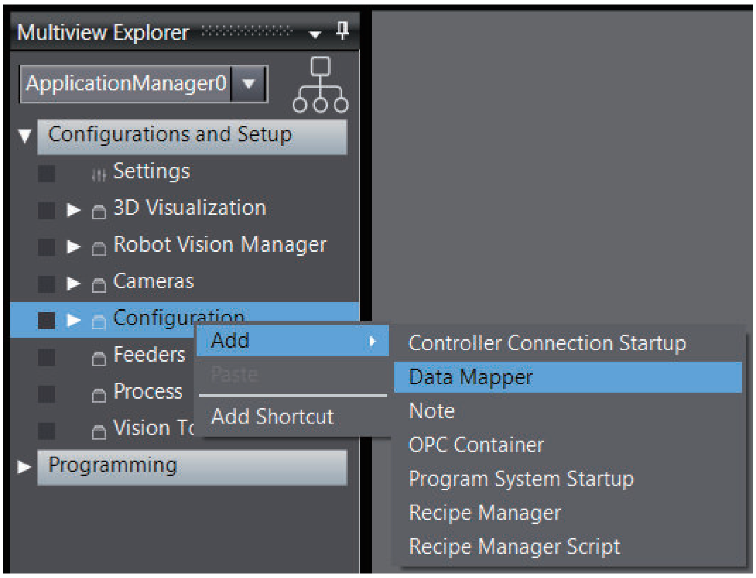
Items on the editor window are as follows.

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 the IPC Application Controller.
	Add/Remove Buttons	Add or remove controllers to/from the **Maintain Controller Connections** list. You can only add a Robot Integrated CPU Unit that already exists in the project.
	Check Connection (ms)	Specifies the interval in milliseconds for checking the connections to the controllers specified on the **Maintain Controller Connections** list.
	Maintain Controller Connections 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 on the IPC Application Controller is first connected to the controller. It will not re-run when disconnected and reconnected. If the program needs to run again, you will need to exit the Application Manager and restart it.
	Program Name	Specifies the program to run when the selected controller is connected.
	Task Number	Specifies the V+ task number for the V+ program. The specified V+ task number must be idle. Otherwise, the program will not run on connection.

5-7-2 Data Mapper

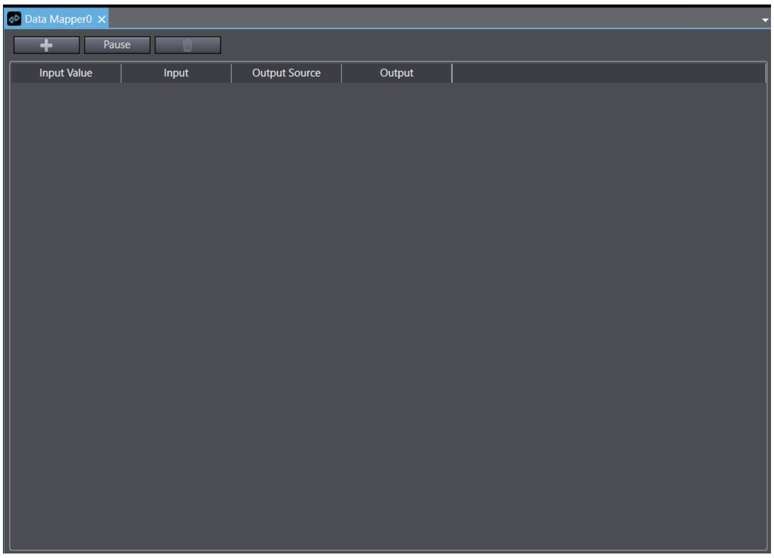
The Data Mapper provides a method to associate different data items within a Sysmac Studio project. For example, you can trigger a Process Manager object to run when a digital input signal turns on.
Any data items that are associated in the Data Mapper will be continuously checked while the Sysmac Studio project is opened.
Create a Data Mapper object. Use the Data Mapper to map the data among objects in the workspace.

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



A new **Data Mapper** will be added to **Configuration**.

2 Right-click the **Data Mapper** and select **Edit**.
The editor window will be displayed.



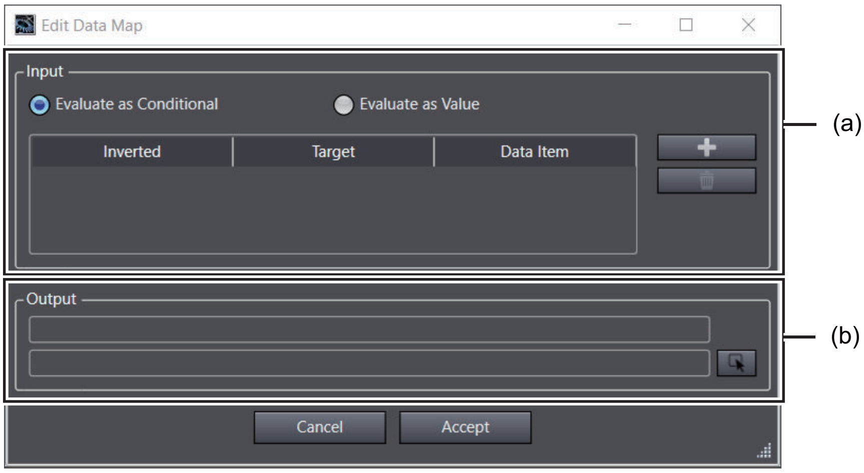
Items on the editor window are as follows.

Item	Description
Add Button	Adds a Data Mapper. Refer to *Data Mapping* on page 5-75 for details.
Pause Button	Pauses the Data Mapper processing.
Delete Button	Delete the selected Data Mapper from the list.
List	List of Data Mappers you have created.

Data Mapping

When the **Add** button is clicked in the Data Mapper editor, the **Edit Data Map** dialog 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.
The Data Mapping input and output items that are available depend on the objects present in the project.

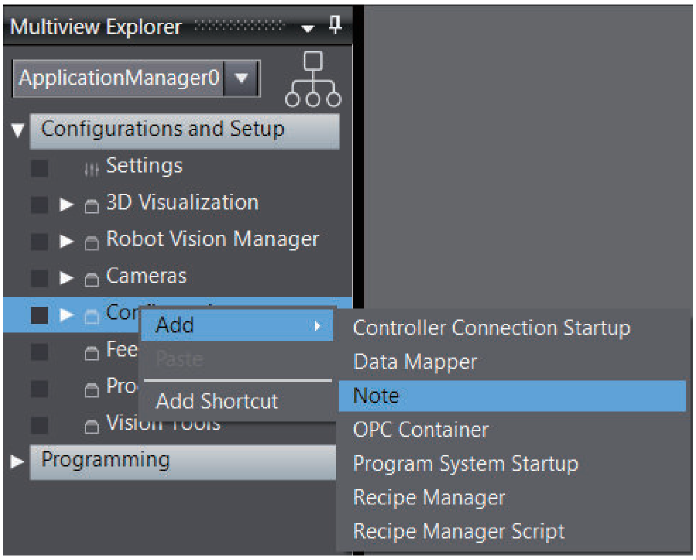


	Item		Description
(a)	Input	Evaluate as Conditional	When **Evaluate as Conditional** is selected, the Data Mapper will interpret all input conditions as a boolean item. If the value of the input item is 0, the condition is considered OFF. If the value is non-zero, the condition is considered ON. If all items in the input list are ON, then the output condition is asserted. If any item in the input list is OFF, the output condition is not asserted. Additionally, when **Evaluate as Conditional** is selected, you can invert the expected value of an input item.. In that case, if the value is 0, the condition is considered to be ON.
		Evaluate as Value	When **Evaluate as Value** is selected, the value of all input conditions are added together and written to the output value.
		Add/Remove Buttons	Clicking the **Add** button will display the **Data Item Selection** dialog. Select input items from the items of **RobotControlSettings**, the robot control function module of the Robot Integrated CPU Unit.
		Data Item List	Input item list registered to the data map.
(b)	Output		Clicking the output item selection button will display the **Data Item Selection** dialog. Select output items from the items of **RobotControlSettings**, the robot control function module of the Robot Integrated CPU Unit.

5-7-3 Note

Note object provides a means for creating documentation. Use this object to create a note about an application.

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



A new **Note** will be added to **Configuration**.

2 Right-click the **Note**, then click **Edit**.
The editor window will be displayed. Enter a note.

5-7-4 OPC Container

Create an 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.

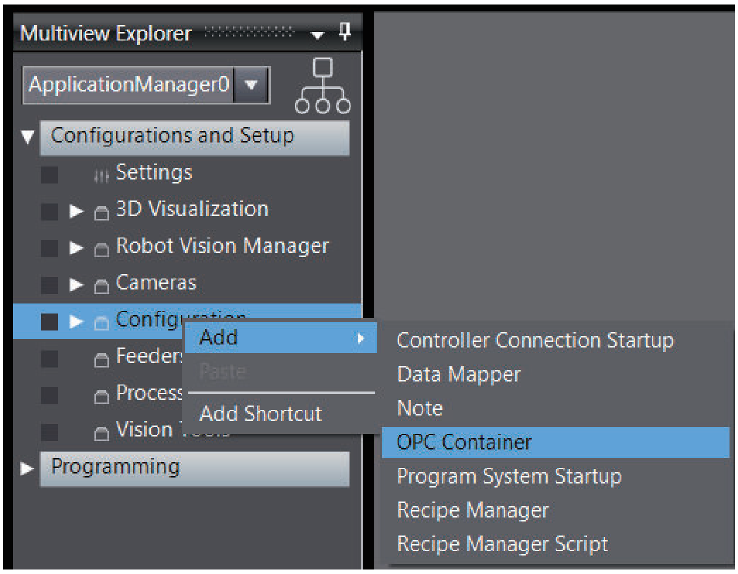
- Communicate values of V+ Global Variables and Process Manager statistics.
- Start or stop a process within a Robot Integrated System application.

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.

Visit the *OPC Foundation* website below for more information.

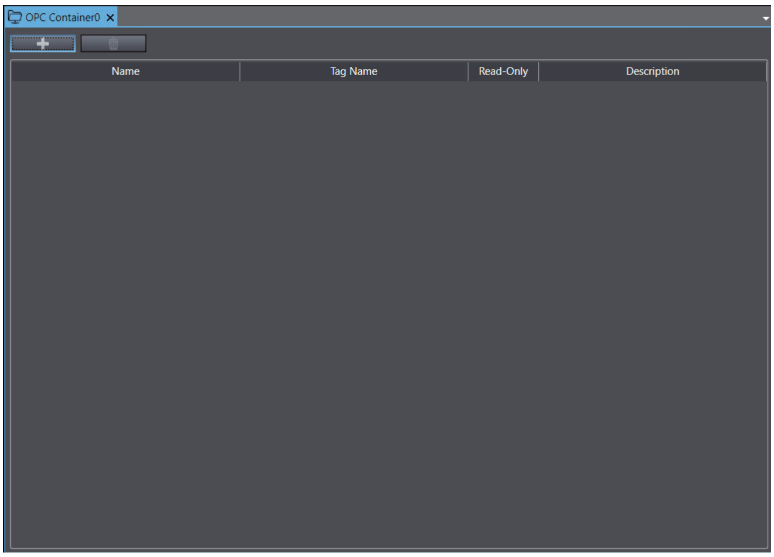
https://www.opcfoundation.org

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



A new **OPC Container** will be added to **Configuration**.

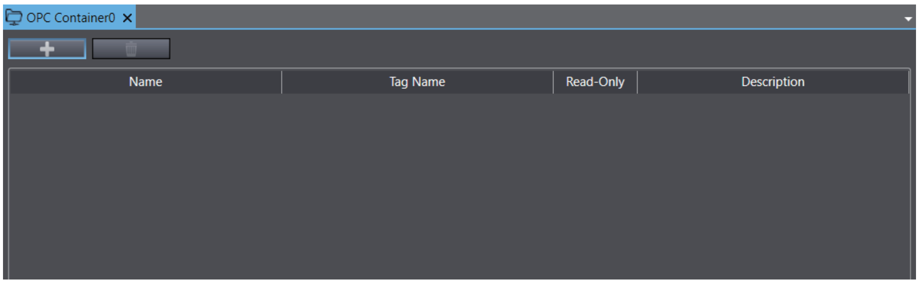
2 Right-click the **OPC Container**, then click **Edit**.
The editor window will be displayed.



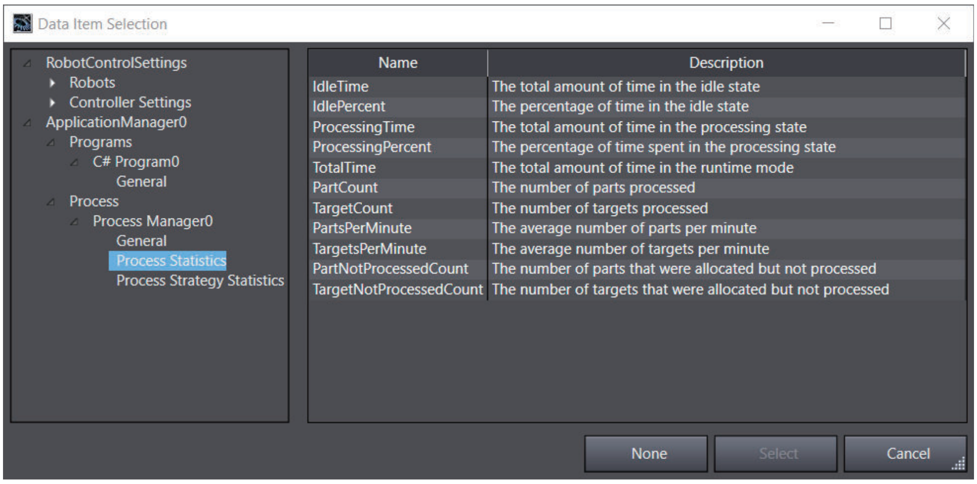
OPC Container Configuration Procedure

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 has one 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.
It will open the **OPC Container** editor in the Edit Pane.



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



3 Select an item from the list and then click the **Select** button.
The item will be added to the OPC Container publish list in the Edit Pane.

4 Check the **Read-Only** option.
If the option is 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.
Further OPC client configurations will be done on the IPC Application Controller.

Refer to *Automation Control Environment (ACE) Version 4 User's Manual (I633)* for more information.

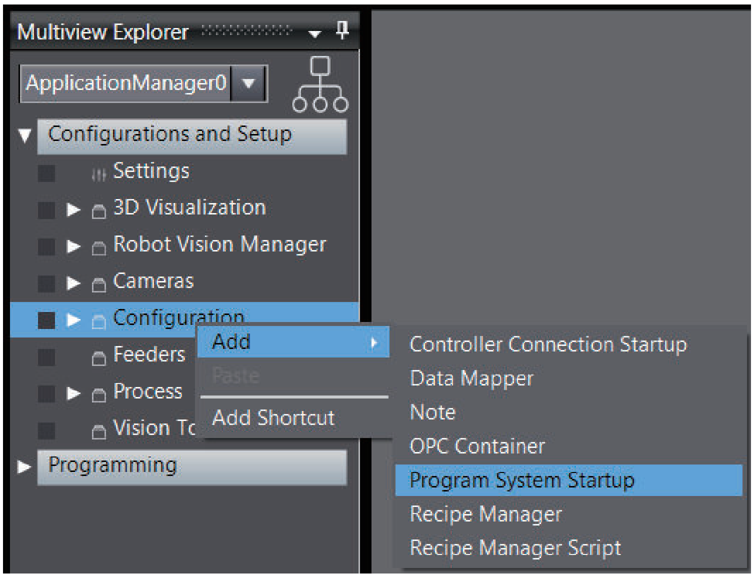
5-7-5 Program System Startup

Configure settings regarding startup of the Robot Integrated Systems. The Program System Startup object is used to specify a C# program to run when the Robot Integrated System project opens.

Additional Information

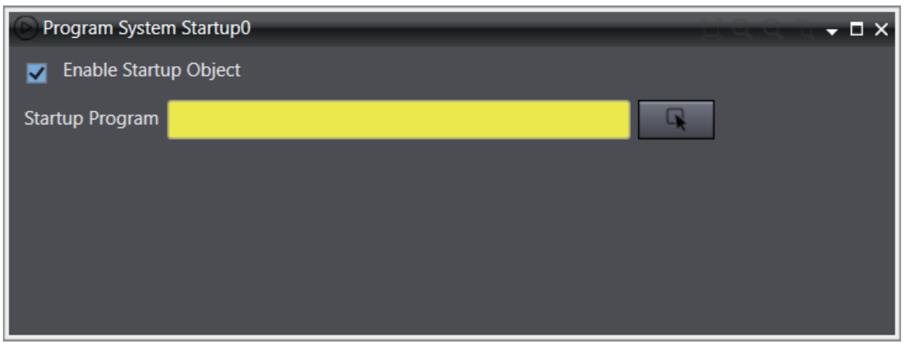
Only one program System Startup object can be created in the workspace.

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



A new **Program System Startup** will be added to **Configuration**.

2 Right-click the **Program System Startup**, then click **Edit**.
The editor window will be displayed.



Items on the editor window are as follows.

Item	Description
Enable Startup Object	Check this box to run the C# program specified with the selector button at a system startup.

Item	Description
Selector Button	Use the Selector button to make a startup C# program selection. You can specify only one object.

5-7-6 Recipe Manager

Recipe Manager is used to define every source that is to be used for creating a recipe. You must add sources to the Recipe Manager before creating a recipe.
Save the information set to the controller and vision tools to the object. The saved information will be 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. It 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 Sources

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

Item	Description
V+ Variable	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. • Calculation Tools • Image Process Tools • Custom Tools
Virtual Camera	Specify Virtual Camera object data to be included in a recipe.
AnyFeeder	Specify AnyFeeder object data to be included in a recipe.

Item	Description
CAD Data	Specify CAD data to be included in a recipe.

● Creating and Modifying a Recipe

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. It minimizes the amount of parameters in Source objects that need to be individually edited in each recipe. Although recommended, it is not mandatory.
Task Status Control provides the Recipe Editor. The editor 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.

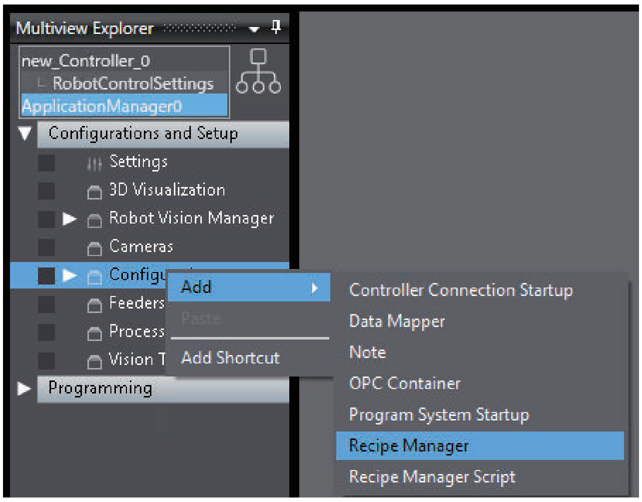
Refer to *Recipe Editor* on page 5-85 for details about creating, modifying, and deleting recipes.

● Selecting a Recipe

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.

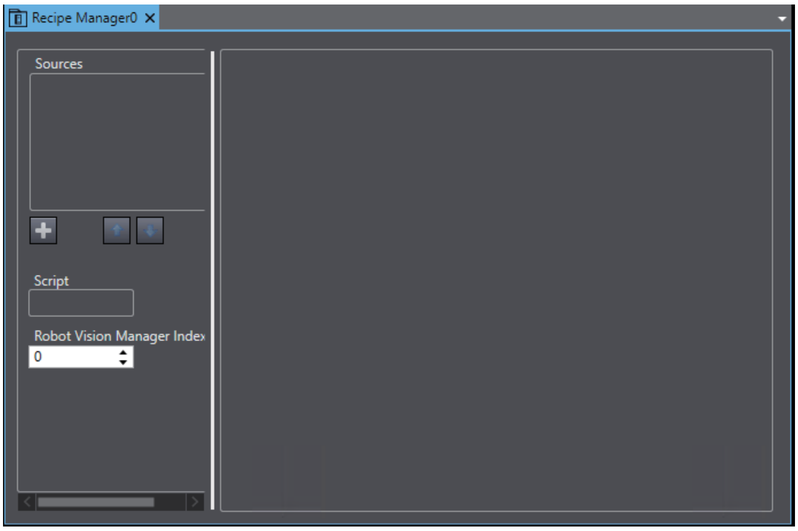
Adding a Recipe Manager

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



A new **Recipe Manager** will be added to **Configuration**.

2 Right-click the **Recipe Manager** and select **Edit**.
The editor window will be displayed.
Click the **Configuration** tab to display the next screen.



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 can change the orders of the items in the list and in the Recipe Editor tab. Frequently used items should be placed at 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.

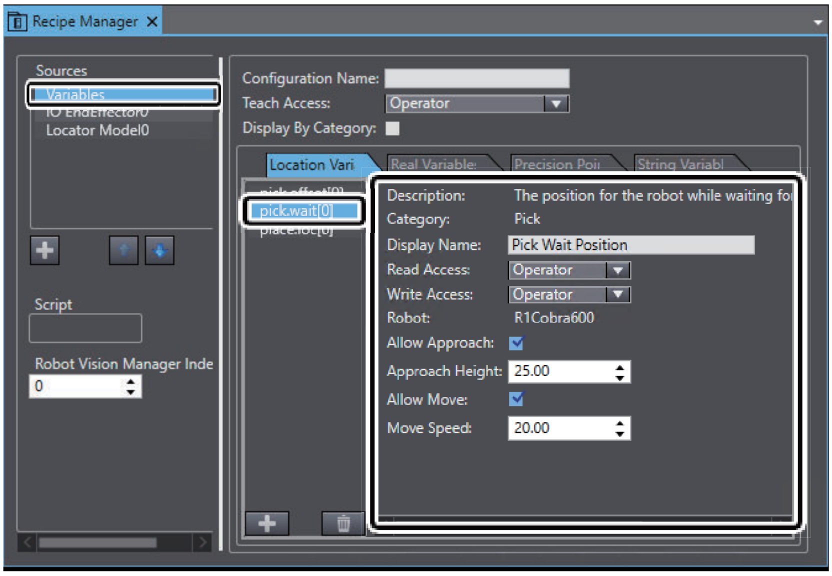
- Configuration Name: Provide a unique name for the Source. This is the name that is displayed in the Recipe Editor. It is the name that is displayed in the Recipe Editor.
- Teach Access: You can restrict write access to Recipe Manager sources by setting user access levels. Refer to *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for details.

● V+ Variable Sources

To set a V+ Variable Source in the Recipe Manager, you must add individual variables in the configuration area. Make a V+ Variable list with the Add and Remove buttons, then use the list in each Recipe.

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 containing a vision tool is selected, the Recipe will be linked with a corresponding vision tool object in the Sysmac Studio project. 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* on page 5-85 for more information.

● CAD Data Sources

CAD data can be changed by switching recipes. Switching them together with other recipe data also makes it easy to manage projects.

The CAD data sources can be used by Application Manager version 4.0 or higher.

● Recipe Script Selection

Select a Recipe Script object created with the Recipe Manager Script editor. Refer to *5-7-7 Recipe Manager Script* on page 5-90 for more information.

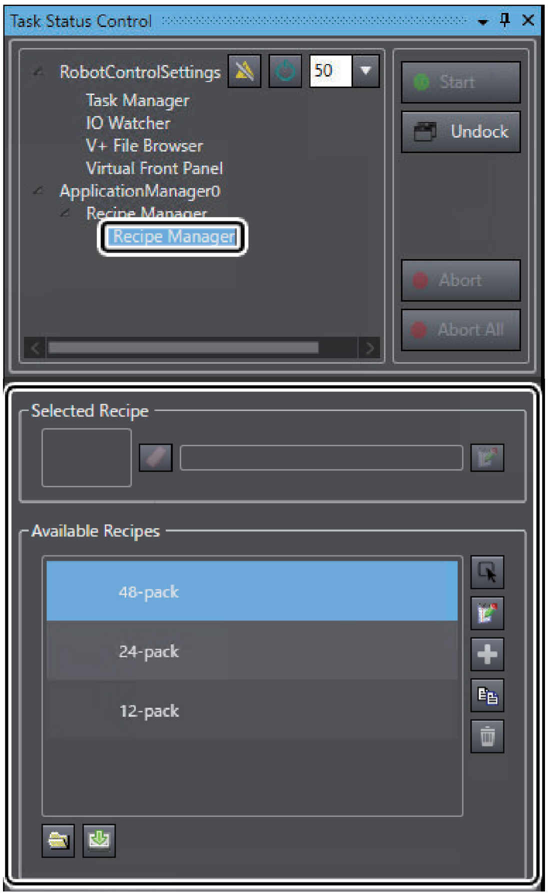
● Robot Vision Manager.Index Setting

The Robot Vision Manager index setting defines the index used as the *sequence_id*. Refer to Recipe Manager properties in the *ACE Reference Guide* for details.

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 access from the Task Status Control area. Refer to *7-5 Task Status Control* on page 7-30 for more information. The Recipe Editor is described below.



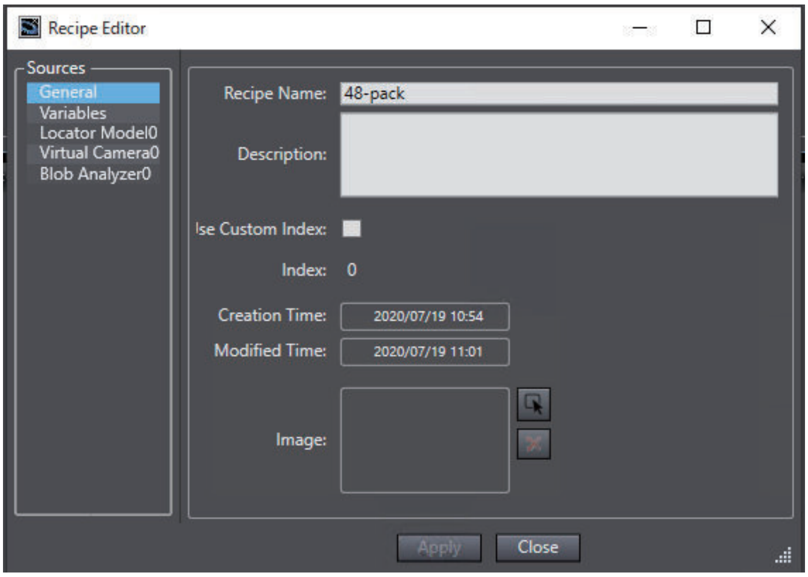
Group	Button	Description
Selected Recipe		Edit the selected Recipe.
		Clear the selected Recipe.
Available Recipes		Makes the highlighted Recipe in the Available Recipe list the active Recipe.
		Edit the selected Recipe.
		Add a new Recipe.
		Create a copy of the selected Recipe.
		Delete the selected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

● 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.
The **Recipe Editor** dialog opens.



The Recipe Editor settings are listed in the following table.

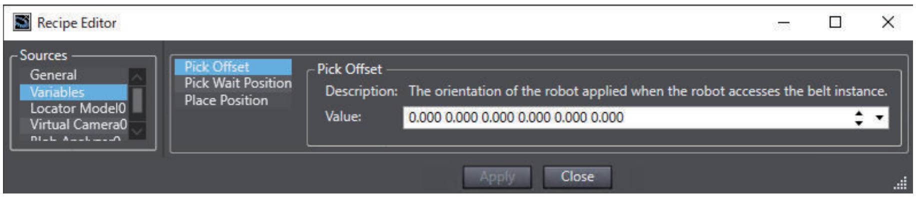
Item	Description
Recipe Name	Provide a unique name for the Recipe.
Description	Provide a description associated with the Recipe.
Index	If the **Use Custom Index** option is selected, you can set a unique index number. This is the index of the Recipe used when accessing the Recipe through the Robot Vision Manager or with a C# program. Refer to the *ACE Reference Guide* for details.
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 the 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.

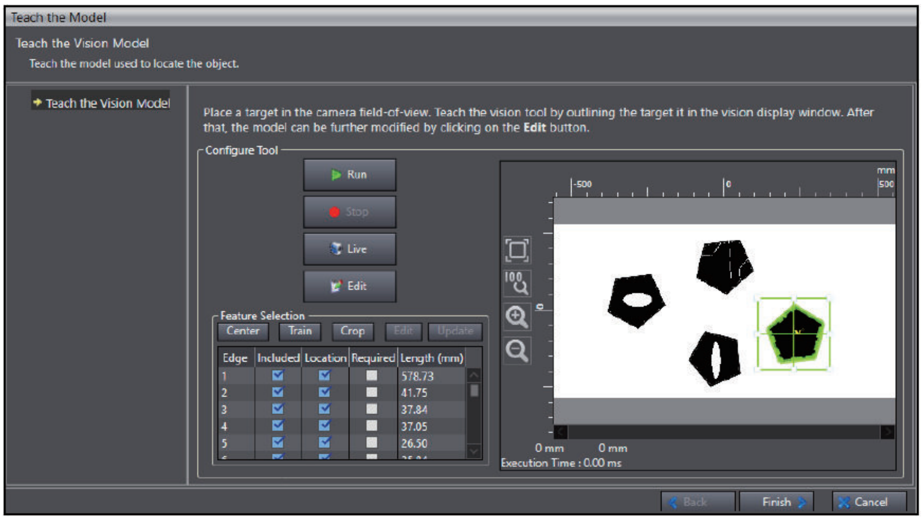
● V+ Variable Sources

Each selected variable is displayed in a list. The display will be changed based on the settings in the Recipe configuration as each variable is selected.



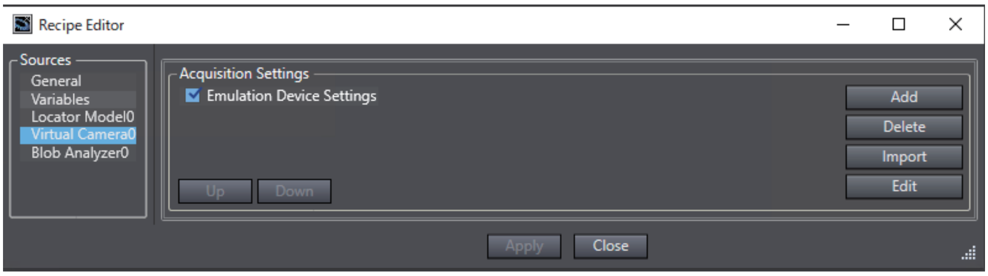
● Locator Model Sources

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



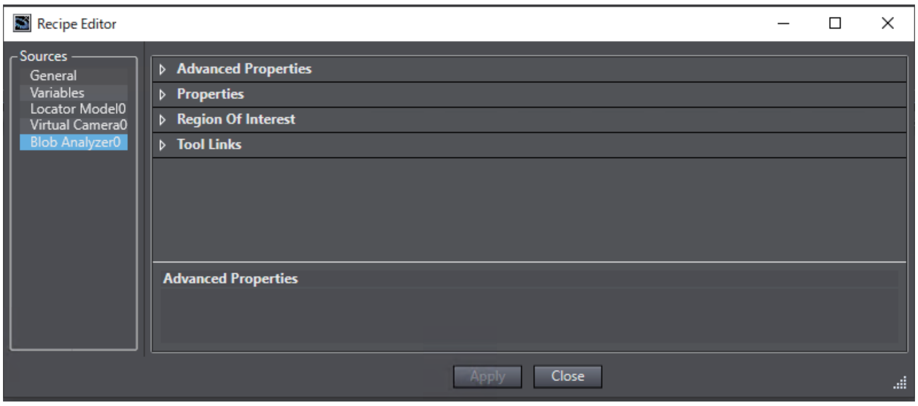
● Virtual Camera Sources

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



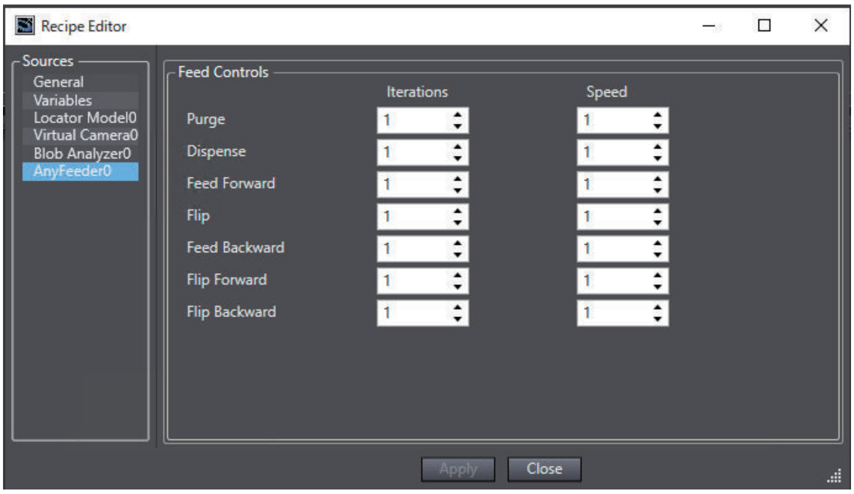
● Other Vision Tool Sources

Adjust the parameters of the vision tool.



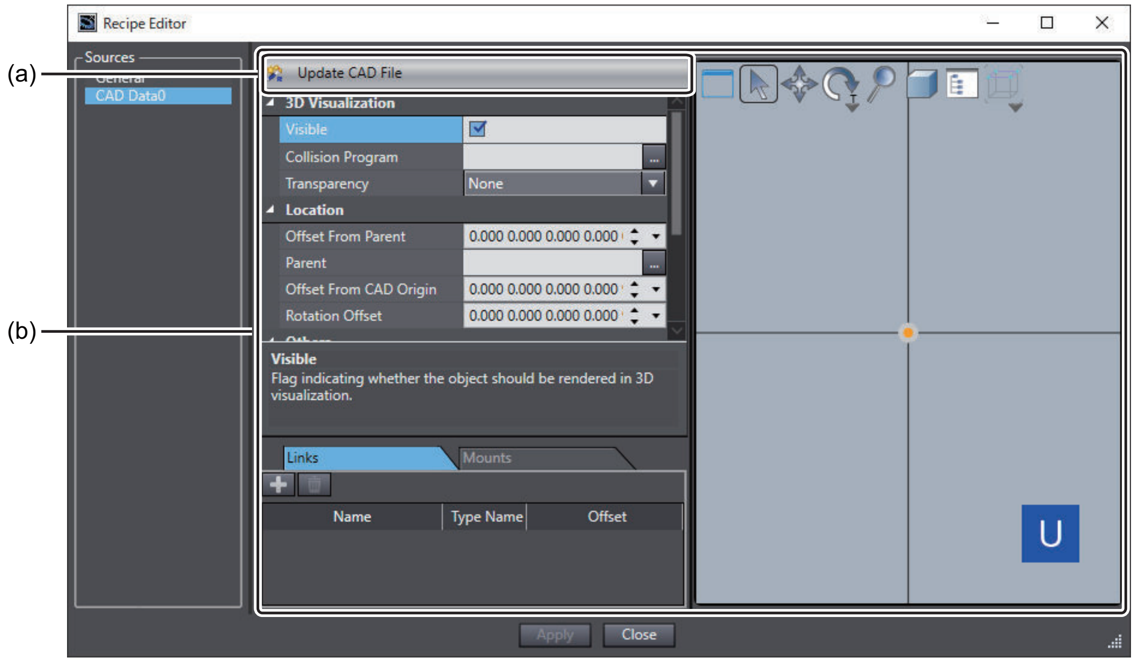
● AnyFeeder Sources

Adjust the Feed Controls of the AnyFeeder.



● CAD Data Sources

Each selected CAD data is displayed.



The setting items of CAD data sources are as follows.

	Item	Description
(a)	Update CAD File	Click this button to display a wizard for updating the CAD file. Refer to the *Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618)* for details on the wizard.
(b)	CAD DATA set-up pane	Display the CAD DATA setup pane. Refer to the *Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618)* for details on the setup pane.

Precautions for Correct Use

At the time of project synchronization, if it is transferred to the Controller with the synchronization check item **Transfer 3D shape data to the target when performing Transfer To Target** unchecked, the 3D shape data of the CAD data registered in the recipe will be empty at the transfer destination. Therefore, a simulation and a collision judgment based on the 3D shape data cannot be performed. Refer to *5-4-3 Exporting/Importing 3D Shape Data* on page 5-11 for the operation when you want to perform a simulation and a collision judgment in the project that includes no 3D shape data.

Version Information

- Switching recipes including CAD data sources can be used by Application Manager version 4.0 or higher.
- The file in .recipe format including CAD data can be read by Application Manager version 4.0 or higher.
- If the Recipe Manager including CAD data sources created by Application Manager version 4.0 or higher is copied and pasted to Application Manager version 3.0 or lower, the CAD data sources will be deleted.

5-7-7 Recipe Manager Script

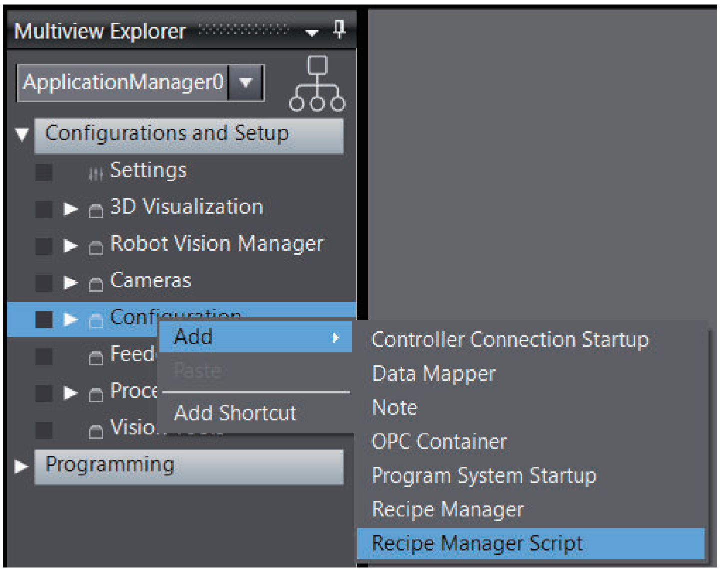
A Recipe Manager Script is a C# program invoked when certain events occur in the lifetime of a Recipe. Select a Recipe Manager Script in the **Script** under the Recipe Manager to use in the Recipe Manager.

The following script methods are available in a Recipe Manager Script.

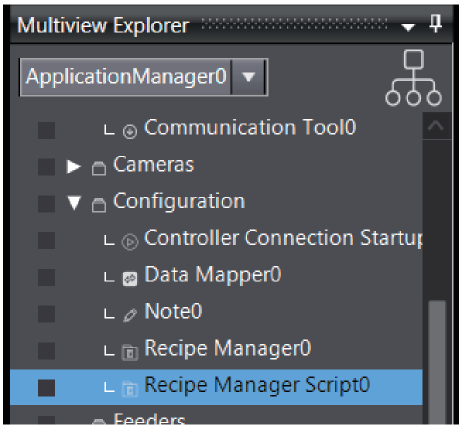
Script Method	Description
string CanEdit(Recipe recipe)	Called to check if a Recipe can be edited. If this method returns an empty string, the Recipe can be edited. If it returns a non-empty string, editing will be prevented and the string will be displayed.
void BeforeEdit(Recipe recipe)	If a Recipe can be edited, this method is called before the editor is displayed.
void AfterEdit(Recipe recipe)	This method is called after the Recipe Editor is closed.
string CanSelect(Recipe recipe)	Called to check if a Recipe can be selected. If this method returns an empty string, the Recipe can be selected. If it returns a non-empty string, selection will be prevented and the string will be displayed.
void BeforeSelection(Recipe recipe)	If a Recipe can be selected, this method is called before the Recipe Editor is selected.
void AfterSelection(Recipe recipe)	This method is called after the Recipe Editor is selected.

Adding a Recipe Manager Script

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

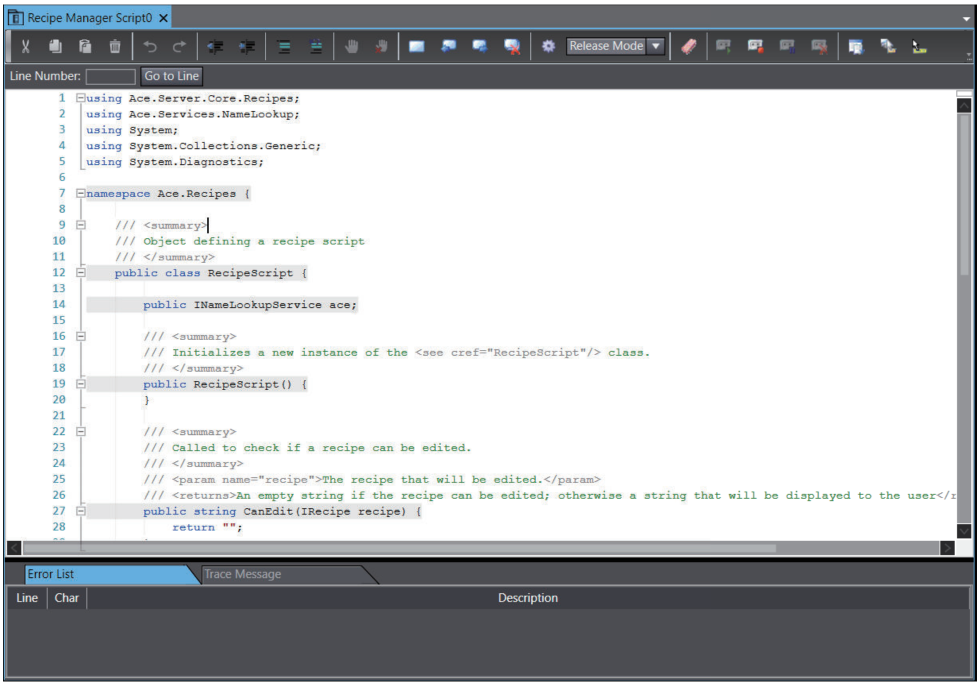


A new **Recipe Manager Script** will be added to **Configuration**.



2 Right-click the **Recipe Manager Script** and select **Edit**.

The C# program editor is displayed.



Refer to *5-11-1 C# Program Editor* on page 5-192 for how to use the C# Program editor.

5-8 Feeder

Feeders are objects that represent devices used to dispense and present parts to a robot. The following types of Feeder are available.

Feeder Type	Description
AnyFeeder	An integrated parts feeding system optimized to work together with the vision sensors, motion devices, and robots.
IO Feeder	Typically used feeder device. Controlled by in-out signals from the connected Robot Integrated CPU Unit.

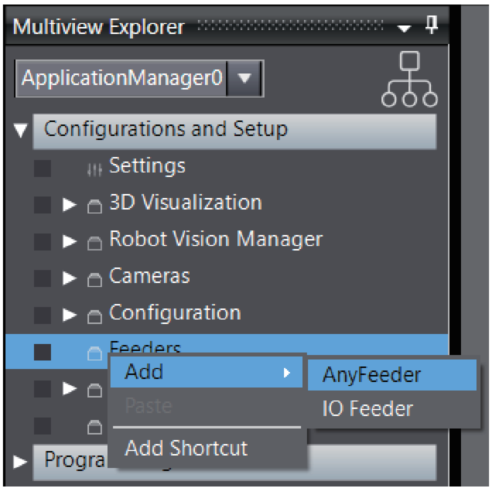
5-8-1 AnyFeeder

AnyFeeder is an integrated parts feeding system optimized to work together with the vision sensors, motion devices, and robots. Settings (e.g. feed control, execution order) and tests regarding to AnyFeeder are available.

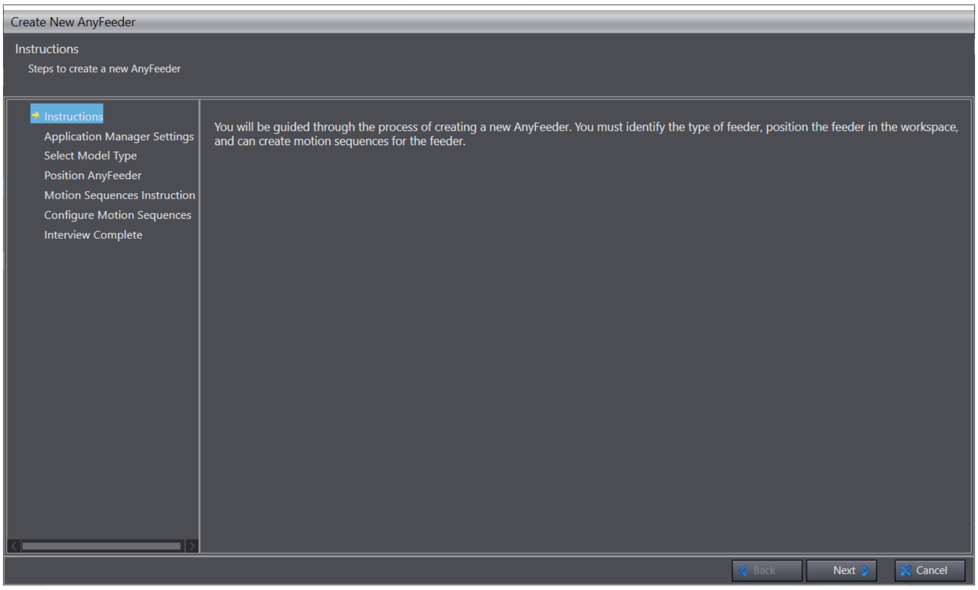
Additional Information

Refer to the operation manual of AnyFeeder for an installation.

1 Right-click **Feeder** in **Configurations and Setup** under the **Multiview Explorer**. Select **Add**, and then click **AnyFeeder**.

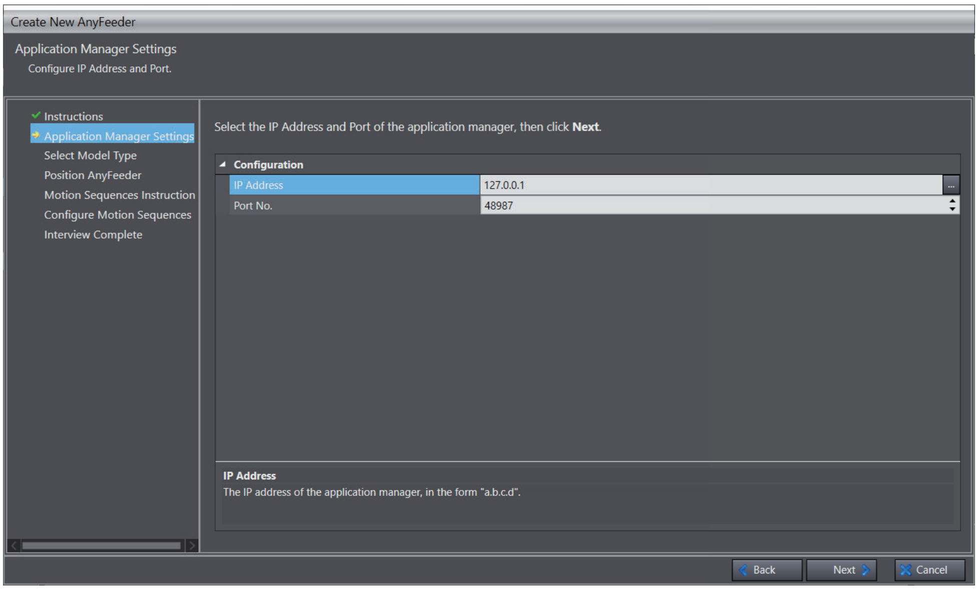


The **Create New AnyFeeder** wizard will open.



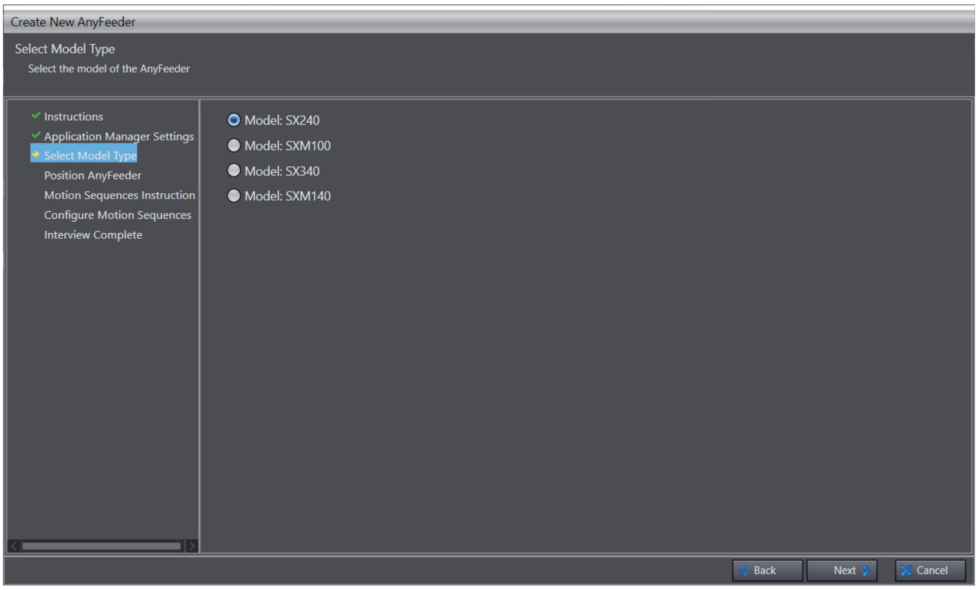
2 Click the **Next** button.

The **Application Manager Settings** dialog is shown.

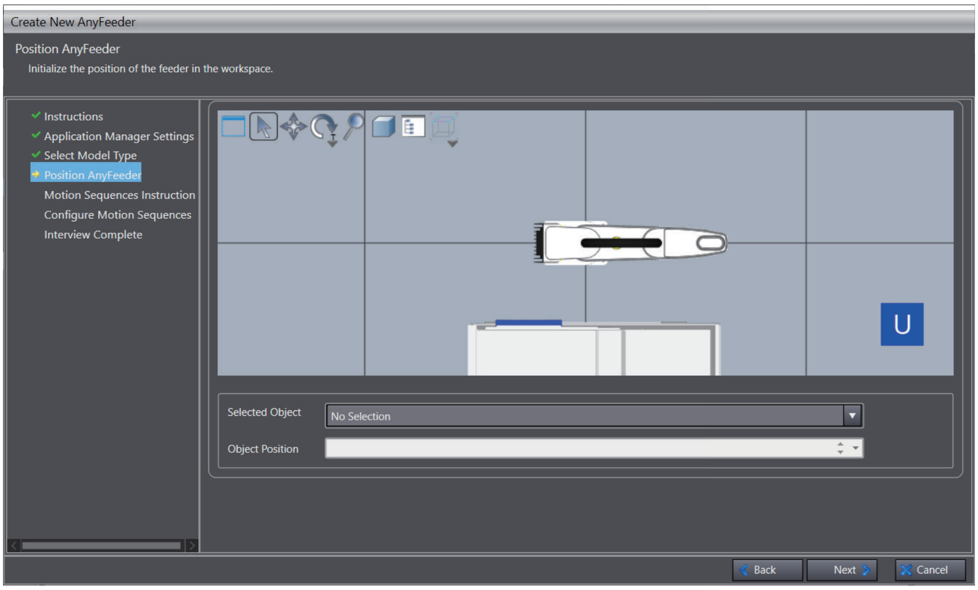


3 Provide the IP address and port of the Application Manager, then click the **Next** button.

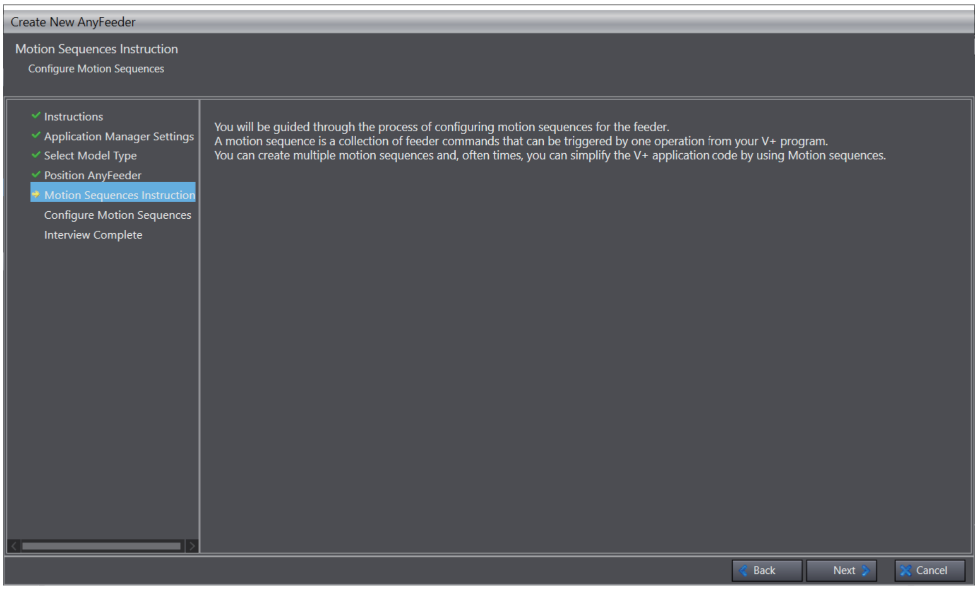
The **Select Model Type** window is displayed.



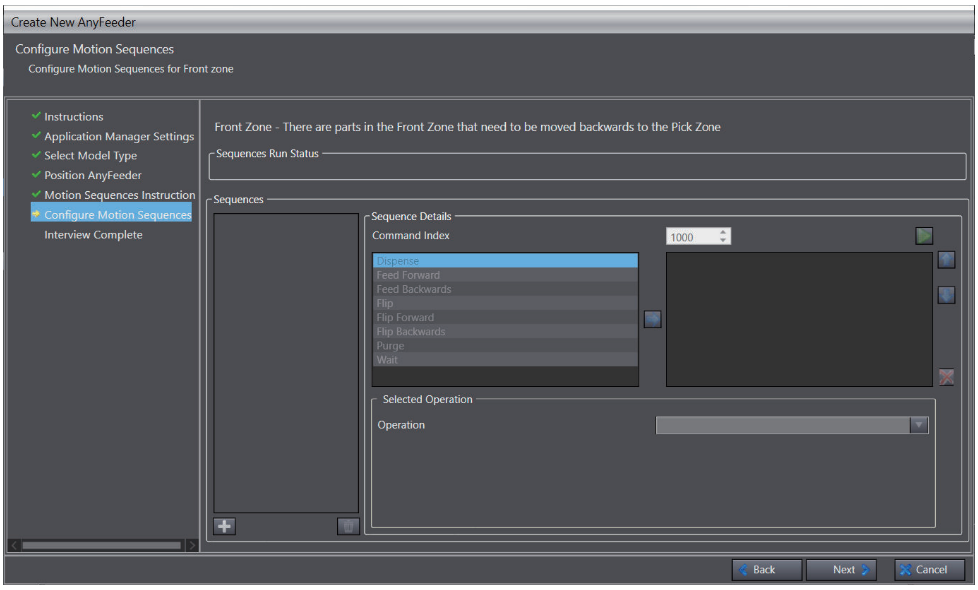
4 Select the AnyFeeder model to use, then click the **Next** button.
The **Position AnyFeeder** window opens.



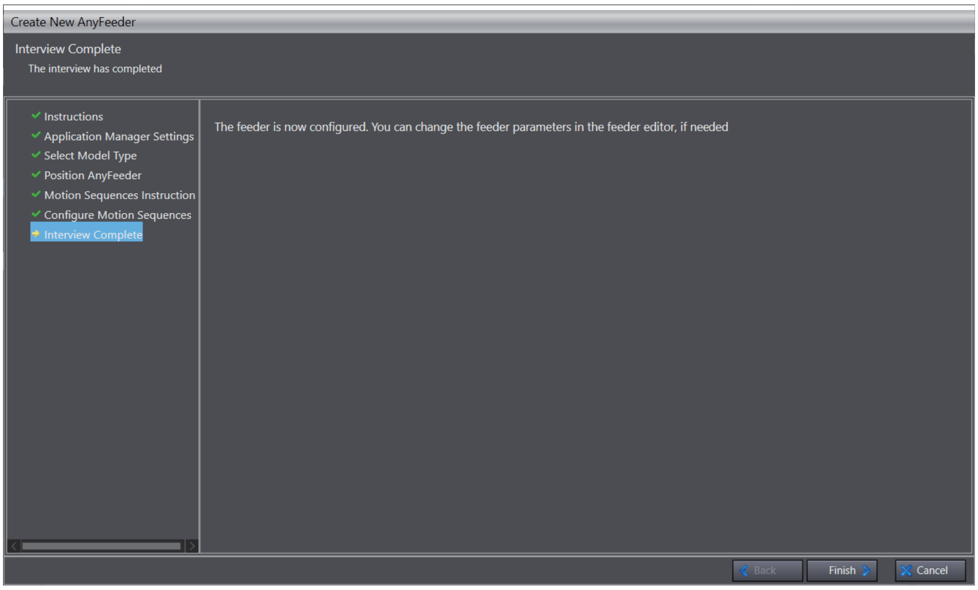
5 Specify the position of the AnyFeeder in the workspace, then click the **Next** button.
The **Motion Sequences Instruction** window opens.



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



7 Configure a Motion Sequence for the Front zone. Then click the **Next** button.
Specify operations as the sequence.
Executing the selected sequence runs operations in the configured order.
Motion Sequences can be triggered through the AnyFeeder interface, C# scripts, or V+ Robot Vision Manager protocols.
The **Interview Complete** window is displayed.

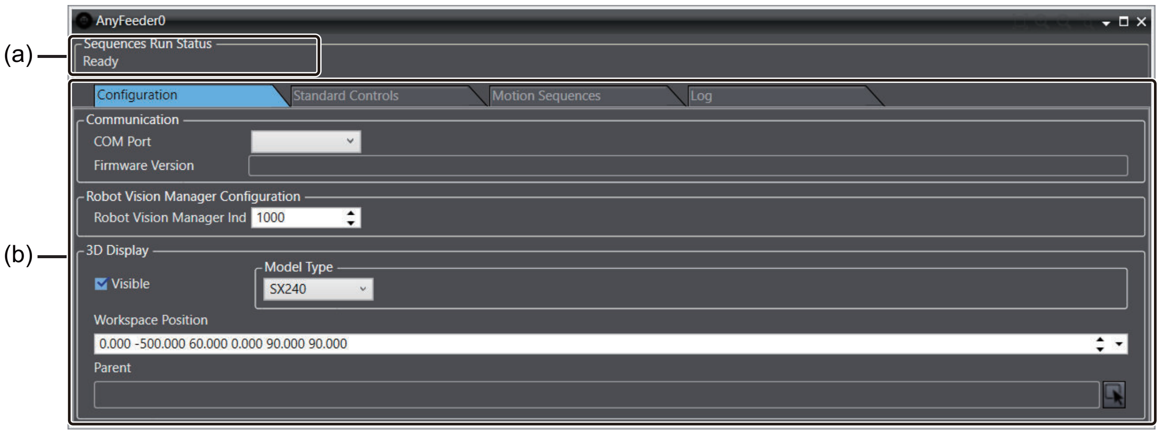


Now you have finished setting AnyFeeder.
Click the **Finish** button after completing all steps and then the **AnyFeeder** will be added to **Feeder**.

AnyFeeder Setting Items

Values are registered to **AnyFeeder** setting items. You can edit those values.

Right-click the **AnyFeeder** you want to edit. Then click **Edit** to display the edit window.



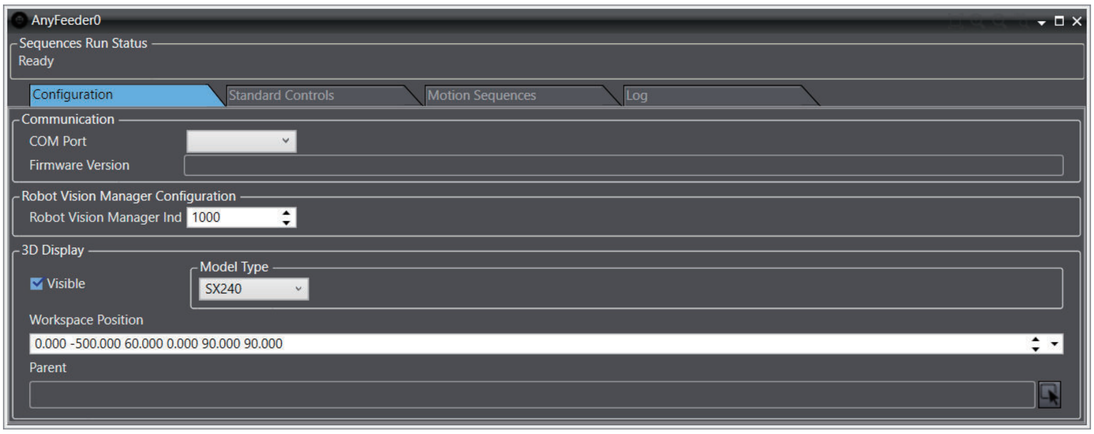
	Item	Description	Reference
(a)	Sequences Run Status	Displays the information about operations and errors regarding to AnyFeeder objects. A current state is shown in the top field, and the description is displayed below.	-
(b)	Configuration	General setting items for the AnyFeeder.	*Configuration* on page 5-98
	Standard Controls	Setting items used for manually controlling the AnyFeeder device.	*Standard Controls* on page 5-98

	Item	Description	Reference
	Motion Sequences	Shows a list of high level motion sequences associated with the AnyFeeder device. 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* on page 5-99
	Log	Shows a history of the communications between the AnyFeeder device and the PC.	*Log* on page 5-101

● Configuration

General setting items for the AnyFeeder.

1 Click the **Configuration** tab to display the next sreen.



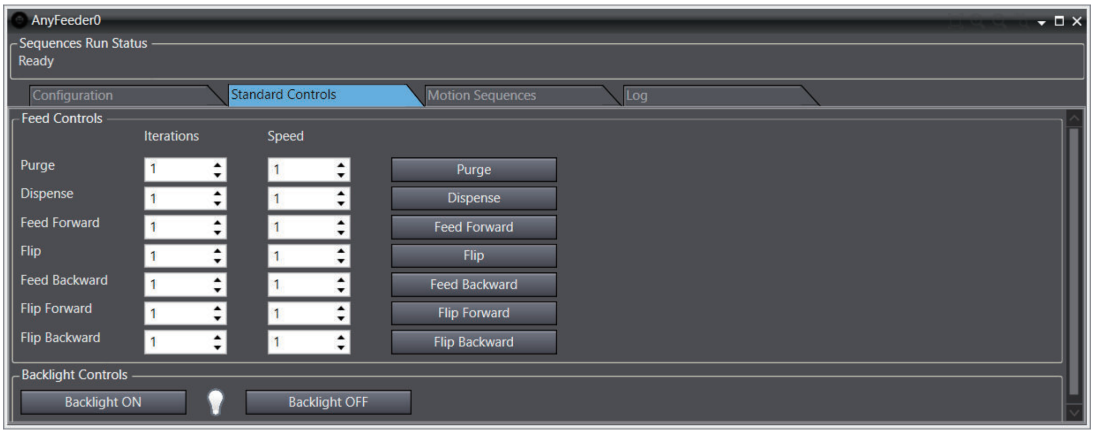
Item		Description
Communication	COM Port*1	Specifies the COM (serial) port that is used to send commands to the AnyFeeder device.
	Firmware Version*1	Displays the firmware version of the connected AnyFeeder device.
Robot Vision Manager Configuration	Robot Vision Manager Index	The index to identify a Robot Vision Manager. It corresponds to the ID identifier of the AnyFeeder. A value not used for a Robot Vision Manager sequence should be specified for the AnyFeeder. Specify a unique number between 0 to 9999.
3D Display	Visible	Check this box to display the AnyFeeder object on the 3D Visualizer.
	Model Type	Select the AnyFeeder model type from the drop-down selection menu.
	Workspace Position	Enter the location of the AnyFeeder device in the workspace.
	Parent	Specify a 3D shape data which is the Parent of the AnyFeeder in the workspace.

*1. Not available in the Emulation mode.

● Standard Controls

Setting items used for manually controlling the AnyFeeder device.

1 Click the **Standard Controls** tab to display the next screen.



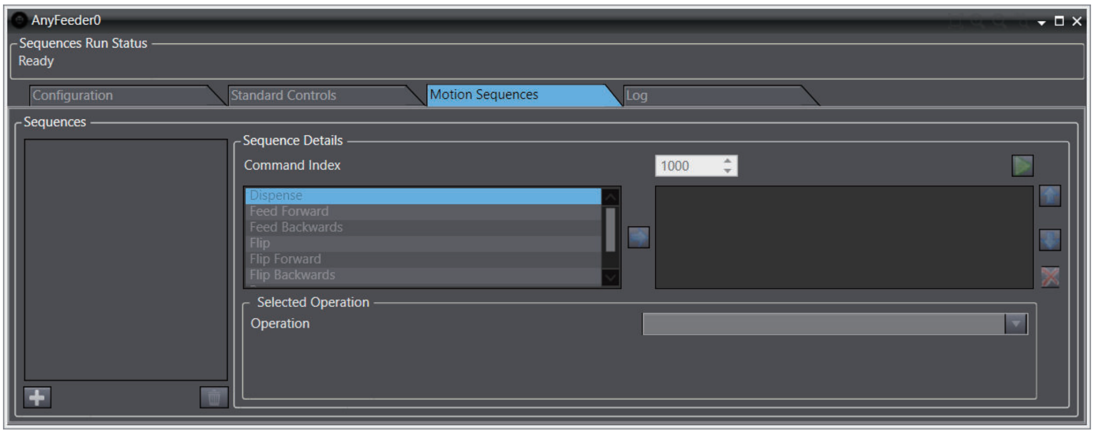
Item			Description
Feed Controls	Feeder Functions	Purge	Feed parts through the purge gate backwards.
		Dispense	Dispense parts from the bulk container onto the conveyor belt.
		Feed Forward	Feed parts forward on the conveyor belt.
		Flip	Flip parts without moving for- wards or backwards.
		Feed Backward	Feed parts backward on the conveyor belt.
		Flip Forward	Flip parts and move them forward.
		Flip Backward	Flip parts and move the backward.
	Iterations		Specify the number of times this action is performed. There are some actions that are designed to occur at the end of all motion and will only occur once, regardless of iteration number. For example, the Dispense command causes the lip in the top bin of the AnyFeeder to retract at the beginning and then come back up at the end. This will only occur once when iteration is set to a number greater than 1.
	Speed		Set the speed of the Feeder Function. Set- tings can range from 0-10 where 0 is the minimum speed and 10 is the maximum speed.
	Execution Buttons		Clicking a button executes the function as many as the setting, at the set speed.
Backlight Controls	Backlight ON/OFF		Turn the backlight ON or OFF.

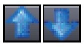
● Motion Sequences

The Motion Sequences tab shows a list of high level motion sequences associated with the Any-Feeder device. 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.

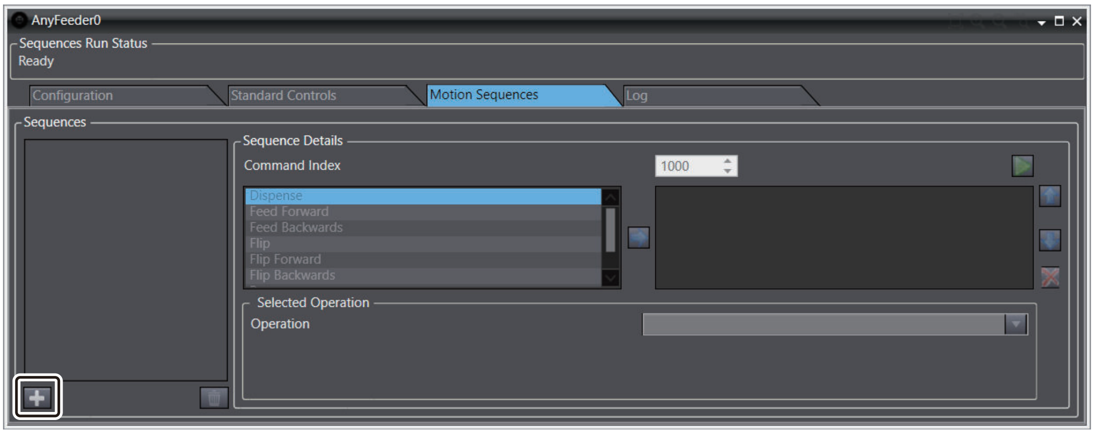
Motion sequences can be triggered through the AnyFeeder user interface, a C# program, or with a V+ program. Motion sequences are stored as Command Index numbers between 1000 to 10000. A motion sequence is referenced with this number in C# and V+ programs.

1 Click the **Motion Sequences** tab to display the next screen.



Item		Description
Sequences		Indexes of commands are shown. Add a new sequence with the **Add** button.
Sequence Details		Set sequences for commands. Select the first Feeder Function from the list and then click the **Select** button to add it to the action list.
	**Play** Button	A sequence can be tested with the button.
	**Up**/**Down** Buttons	Sequence actions can be rearranged with the buttons.
	**Delete** Button	Sequences and sequence steps can be removed using the button.
	**Select** Button	Adds a selected command to the action list.

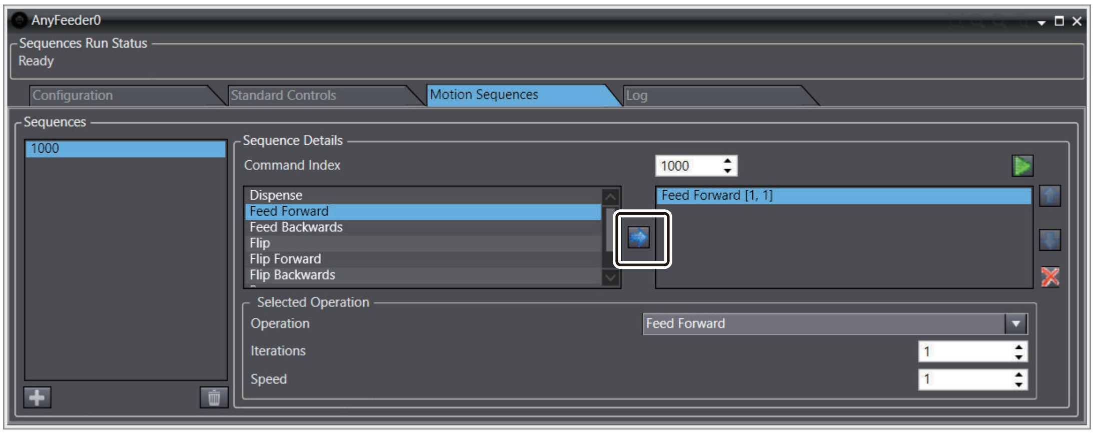
2 Add a new sequence with the **Add** button in **Sequences**.



A new sequence will be placed in the sequence list.
Adjust the Command Index value if necessary.

3 Select the first Feeder Function from the list and then click the **Select** button to add it to the action list.

The function is added to the action list. The Flip function has been added in the example below.



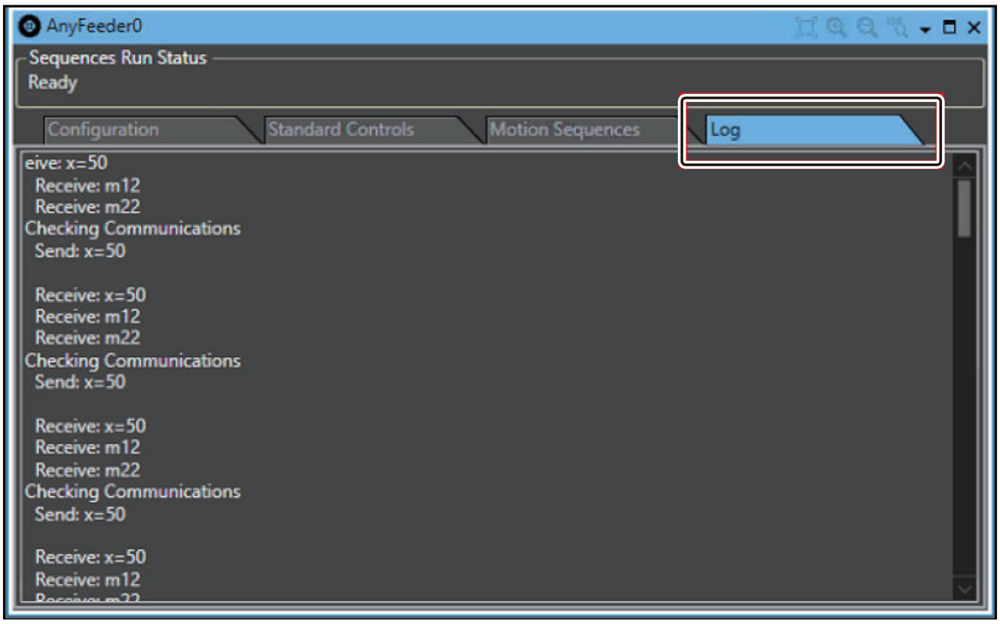
4 Select the action from the Selected Operation list and then make any necessary adjustments to Iterations and Speed.

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.
It will move the connected AnyFeeder and execute the sequence. If the sequence executes correctly, the procedure is complete.

● Log

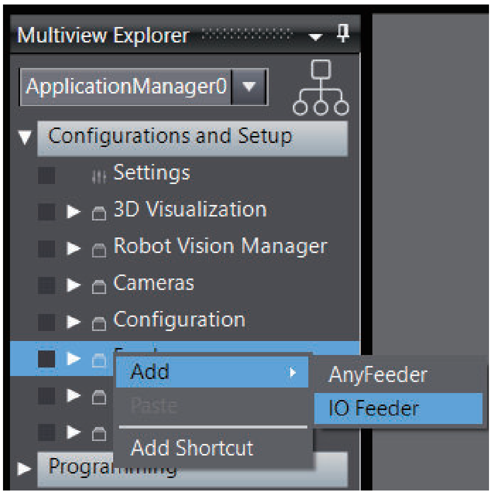
The log page shows a history of the communications between the AnyFeeder device and the PC.



5-8-2 IO Feeder

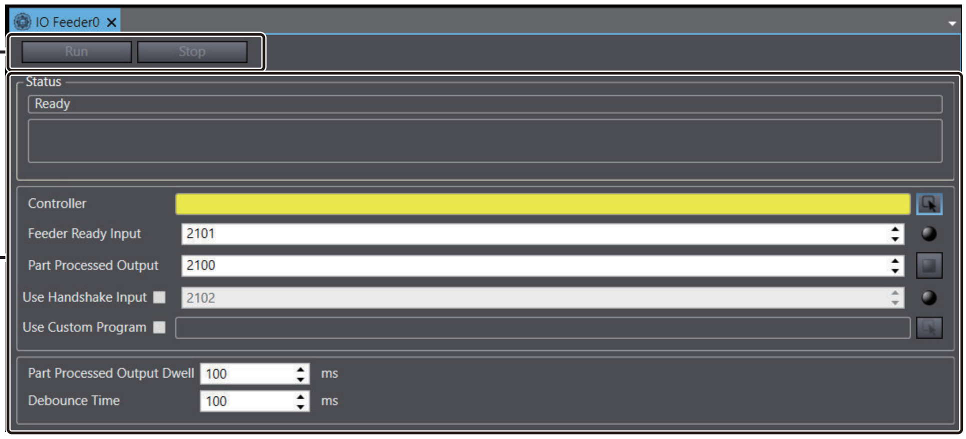
IO Feeder is a generic feeder device controlled by in-out signals from a connected Robot Integrated CPU Unit. Configure settings and perform a test in this section.

1 Right-click **Feeder** in **Configurations and Setup** under the **Multiview Explorer**. Select **Add**, and then click **IO Feeder**.



IO Feeder will be added to **Feeder**.

2 Right-click the **IO Feeder** and select **Edit**.
The editor window will be displayed.



Items on the editor window are as follows.

	Item	Description
(a)	**Run** Button	Used to perform one test cycle of the feeder. The operation stops when the cycle has completed or if the **Stop** button is clicked before the end of the cycle.
	Stop Button	Stops (interrupts) the test cycle.
(b)	Status	Provides operation and error information about the IO Feeder object. A current state is shown in the top field, and the description is displayed below.
	Controller	Specifies the controller that will process the feeder signals.
	Feeder Ready Input*1	Specifies the input signal number that indicates a part detection result when the feeder is ready.
	Part Processed Output*1	Specifies the output signal number that indicates a part detection result has been processed by the robot.
	Use Handshake Input*1	If enabled, the feeder will assert a signal indicating it has acknowledged the part processed signal.

	Item	Description
	Use Custom Program*1	If enabled, the selected feeder interface code runs as a V+ program on the specified robot controller. This program can be overwritten if some custom logic needs to be applied.
	Part Processed Output Dwell	Specifies the dwell time (time to wait) in milliseconds after the part processed output signal is turned ON before turning it OFF. The unit is ms.
	Debounce Time	Specifies the amount of time that a signal must be detected in the ON state before it is considered logically ON. The unit is ms.

*1. Disabled in the Emulation mode. A test can be performed independently from signal status or settings.

5-9 Process

In the Process setting group, you can configure settings associated with processes, which necessary to pick up a part and place it at the target position following the specified execution order.
The setting items are as follows.

Item	Description	Reference
Part Buffer	Defines an object that is an overflow area. The overflow area is a place where parts can be temporarily stored when part targets are unavailable to accept more parts.	*5-9-1 Part Buffer* on page 5-104
Part Target	Defines an object that is a possible destination for a part.	*5-9-2 Part Target* on page 5-107
Part	Defines a part shape and conditions that the robot follows to pickup and place the part.	*5-9-3 Part* on page 5-113
Belt	Defines a physical conveyor belt used in the system. The Belt object maintains a list of encoders that are associated with the physical conveyor.	*5-9-4 Belt* on page 5-115
Process Manager	Process Manager is 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 behaviors to meet the requirements of your application, if necessary. The Process Manager runtime handler is the supervisory control for the entire packaging system, managing allocation and queuing for multiple controllers, robots, conveyors, parts, and targets.	*5-9-5 Process Manager* on page 5-127
Allocation Script	The Allocation Script object is used to create and edit custom part-allocation programs for use with the Process Manager.	*5-9-6 Allocation Script* on page 5-174
Pallet	Defines the layout of a pallet, which can be used to pick parts from or place parts to.	*5-9-7 Pallet* on page 5-175
Visual Refinement Station	Defines an object that is used to refine the part to gripper orientation for improved placement accuracy.	*5-9-8 Vision Refinement Station* on page 5-179

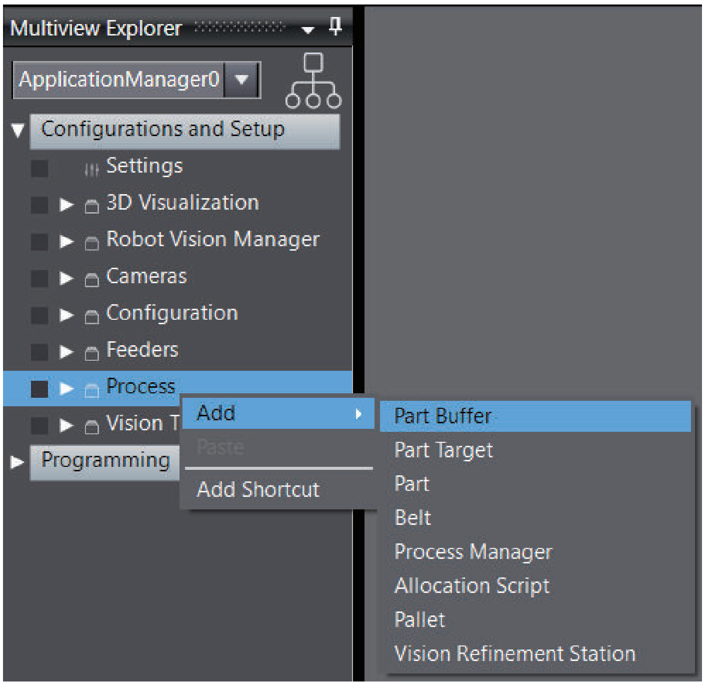
5-9-1 Part Buffer

Create a Part Buffer object. It is necessary to store parts temporarily when more parts cannot be placed on the output conveyor belt or feeder.
The Part Buffer object defines a physical object that is an overflow area The overflow area is a place where parts can be temporarily stored when the output conveyor belt or feeder are unavailable to accept more 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

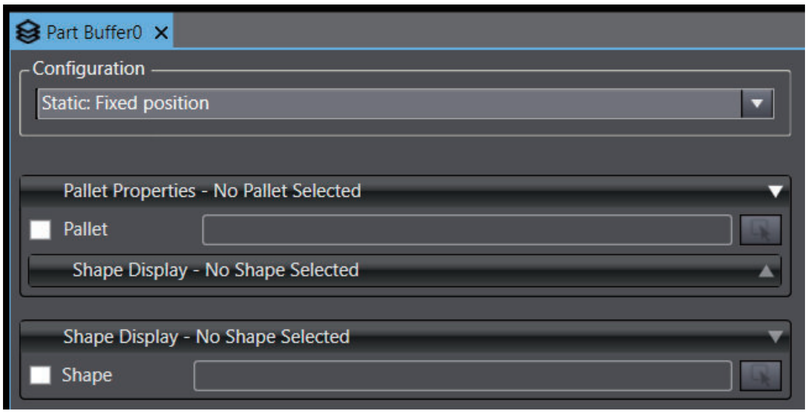
Pallets are available to carry parts. For pallets, you can also control the access method in the following ways: LIFO (Last In, First Out) and FIFO (First In, First Out). Register a pallet before use. Refer to *5-9-7 Pallet* on page 5-175 for details.

1 Right-click **Process** in **Configurations and Setup** under the **Multiview Explorer**. Select **Add**, and then click **Part Buffer**.



A new **Part Buffer** object will be added to the **Process**.

2 Right-click **Part Buffer** and select **Edit**.
The editor window will be displayed.



Items on the editor window are as follows.

Item			Description
Configuration			Select **Static: Fixed position**. Parts are placed at a static location.
Pallet Proper-ties	Pallet		Check this box to use a pallet to carry parts, and specify a pal-let.
	Shape Dis-play	Ena-ble	Check this box to specify a shape to represent the pallet in the 3D Visualizer.
		Shap e	This property defines the shape of a pallet.

Item		Description
Shape Display	Shape	Check this box to specify a shape to represent the Part in the 3D Visualizer.

Adding a Part Buffer to a Process

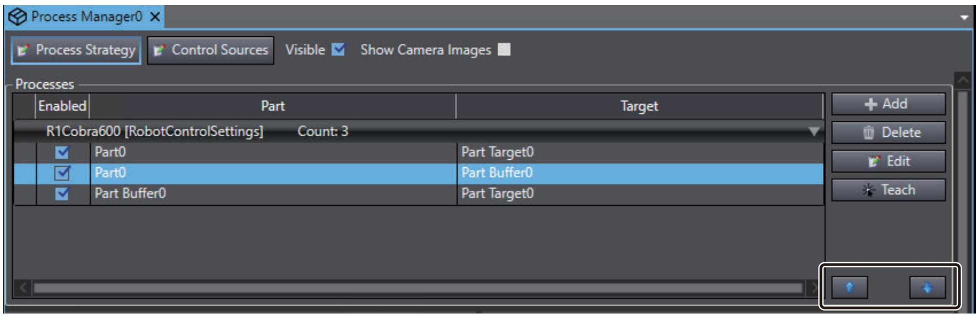
Use the following procedure to add a Part Buffer to a Process after creating and configuring the Part Buffer object.
The default Process Strategy can choose the appropriate process based on part/target availability.

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.



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

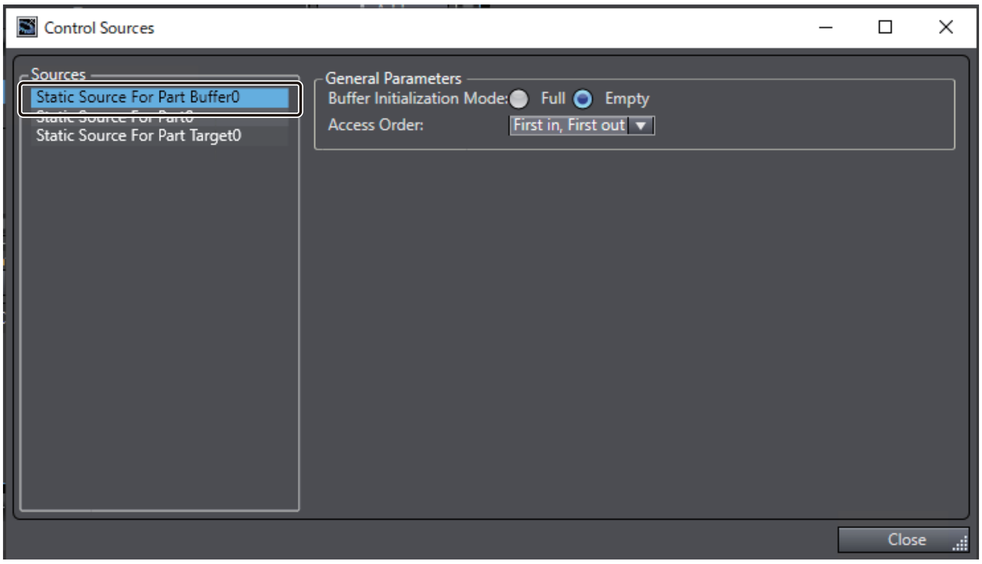
When a pallet is used for the part buffer, you need to specify how the parts will be accessed as the buffer is being emptied. You can select from the following options.

- 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* on page 5-136 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).



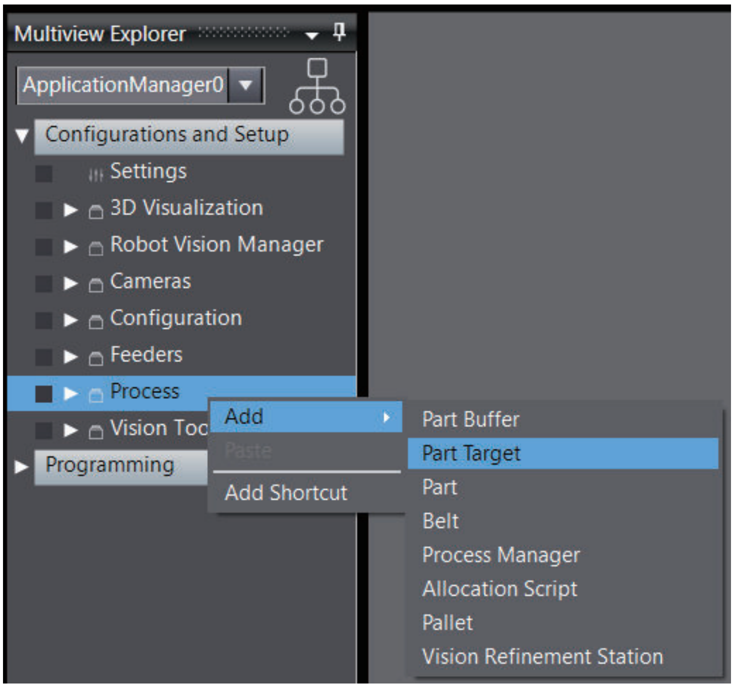
2 If necessary, select the desired Buffer Initialization Mode to indicate the state of the part buffer when it is initialized.
The default state is **Empty**, which means the buffer is empty when initialized.

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

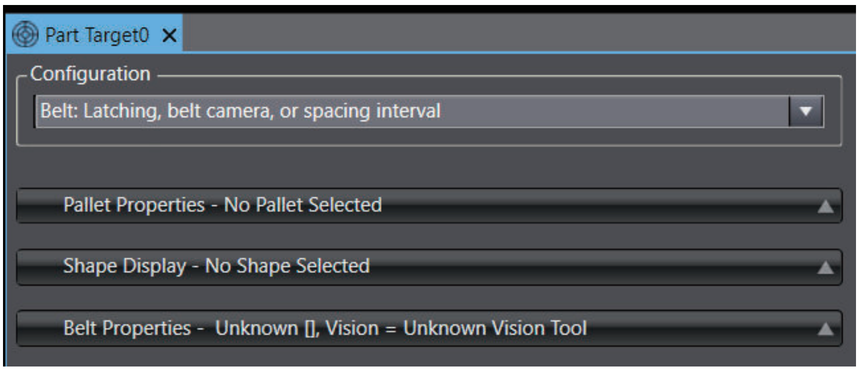
Define the target position where a part will be placed.

1 Right-click **Process** under **Configurations and Setup** in the **Multiview Explorer** and select **Add - Part Target**.



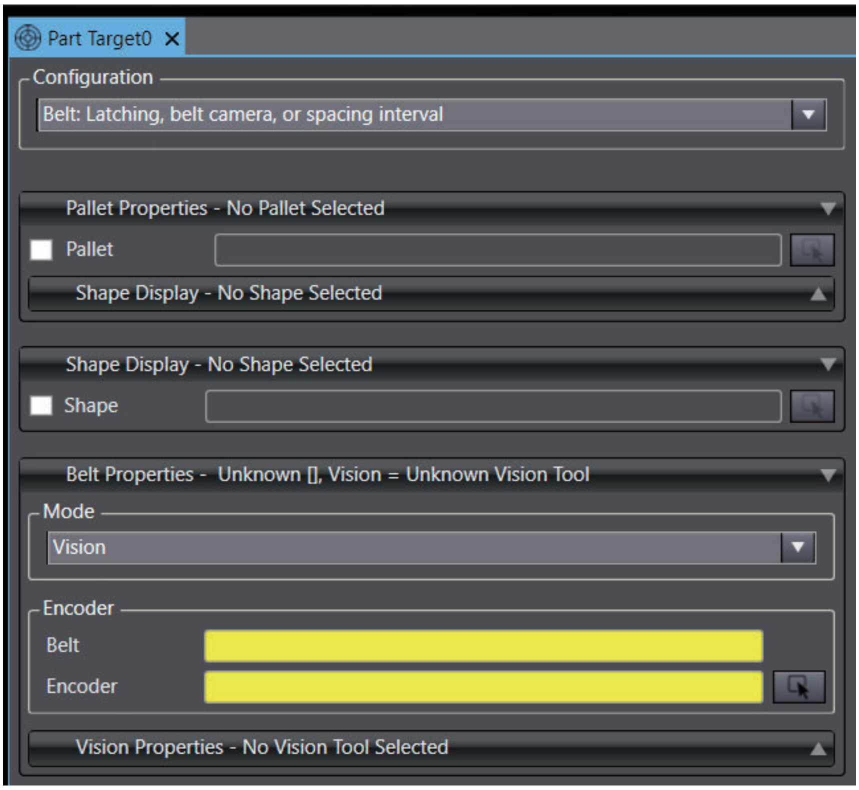
Part Target is added under **Process**.

2 Right-click **Part Target** to select **Edit**.
The edit window will open.



Setting Part Targets

The setting items are as follows.



Item			Description
Configuration			Select the method to pick and place parts.
Pallet Properties (optionally) *1*2*3	Pallet		Check this box to use a pallet to carry parts, and specify a pallet.
	Shape Display	Shape	Check this box to specify a shape to represent the pallet in the 3D Visualizer.
Shape Display *1*2*3	Shape		Check this box to specify a shape to represent the Part in the 3D Visualizer.
Belt Properties *1	Mode		Select the operation mode of the belt.
	Encoder *4*5*6	Belt	Specify the Belt that is used to handle parts.
		Encoder	Specify the belt encoder that is used to control the belt.
	Vision Properties *4	Vision Tool	Specify the vision tool that is used to detect a part on the conveyor belt.
		Use Named Instance (optionally)	Checking this box displays detection results with name . Generally a model name.
	Latch Period in Emulation Mode *7		Specify the Latch Period.
	Spacing *6		Specify the spacing between parts on the conveyor belt. (Unit: mm)
Vision Properties *3	Vision Tool		Specify the vision tool that is used to detect a part on the conveyor belt.
	Use Named Instance		Checking this box displays detection results with name . Generally a model name.

*1. Displayed when **Belt** is selected in the **Configuration** drop-down list box.

*2. Displayed when **Static** is selected in the **Configuration** drop-down list box.

*3. Displayed when **Vision Properties** is selected in the **Configuration** drop-down list box.

*4. Displayed when **Vision Properties** is selected for **Mode** in the **Belt Properties** area.
*5. Displayed when **Latches** is selected for **Mode** in the **Belt Properties** area.
*6. Displayed when **Spacing** is selected for **Mode** in the **Belt Properties** area.
*7. Displayed when **Latches** is selected for **Mode** in the **Belt Properties** area in emulation mode.

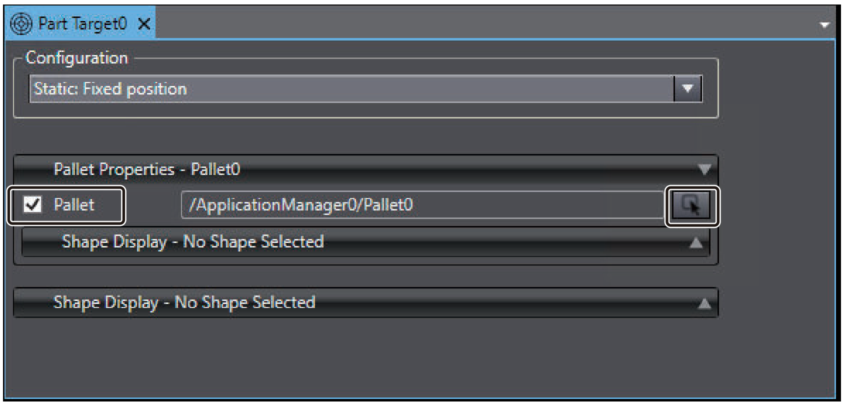
Configuration

The **Configuration** drop-down list box is used to specify how the target is input to the system. 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 *Belt: Latching, Belt Camera, or Spacing Interval Configuration* on page 5-111 for more information.
Static: Fixed position	Targets / instances are acquired from a static location such as a part feeder or a pallet. Refer to *Static: Fixed Position Configuration* on page 5-112 for more information.
Vision: Fixed camera not relative to a belt	Locations are acquired through camera that is not located over a belt. Refer to *Vision: Fixed Camera Not Relative to Belt* on page 5-112 for more information.

Pallet Properties

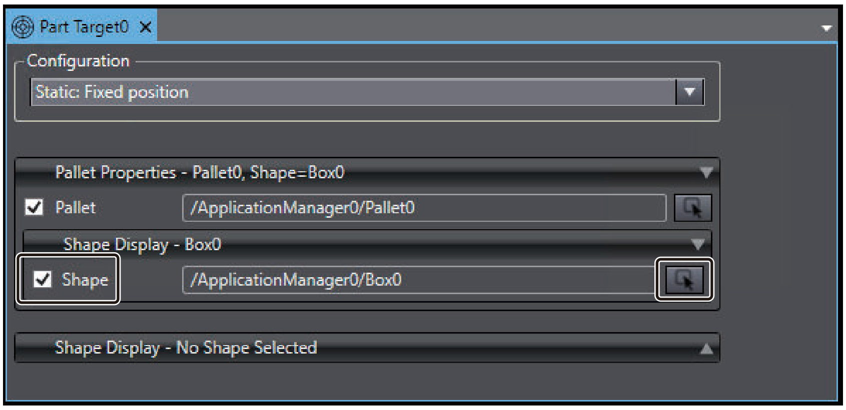
Use the Pallet Properties area if you need to specify a pallet that is used to hold the part(s). Make reference to a Pallet object by checking **Pallet** and then use the **Select** button to specify that object as shown below.



A Pallet object must already exist in the project. Refer to *5-9-7 Pallet* on page 5-175for more information.

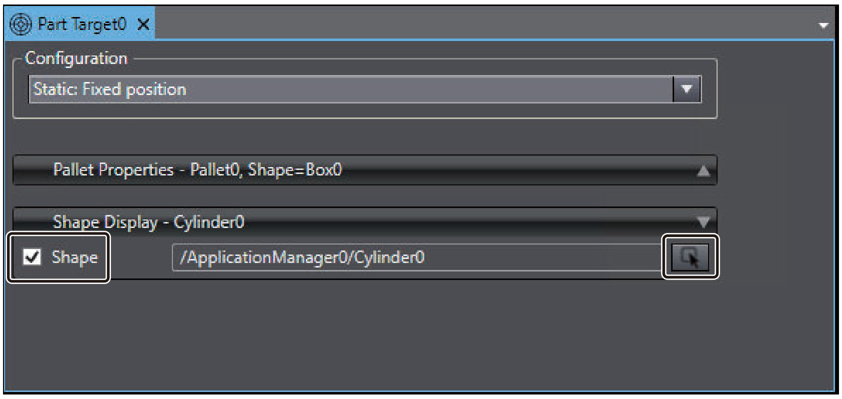
● Shape Display (Pallet Properties)

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.

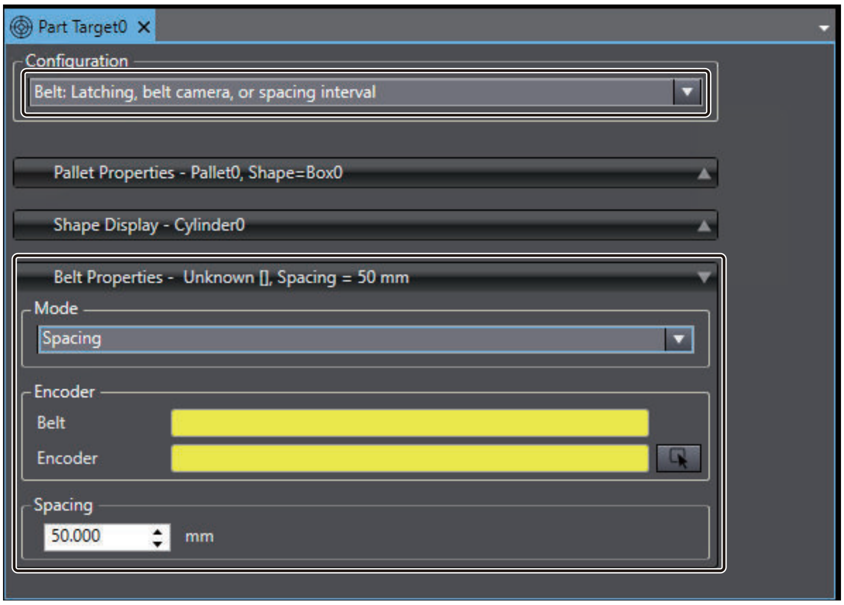
Vision: A vision tool is used to detect the part on the belt.

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 Selection button to reference an exiting 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.

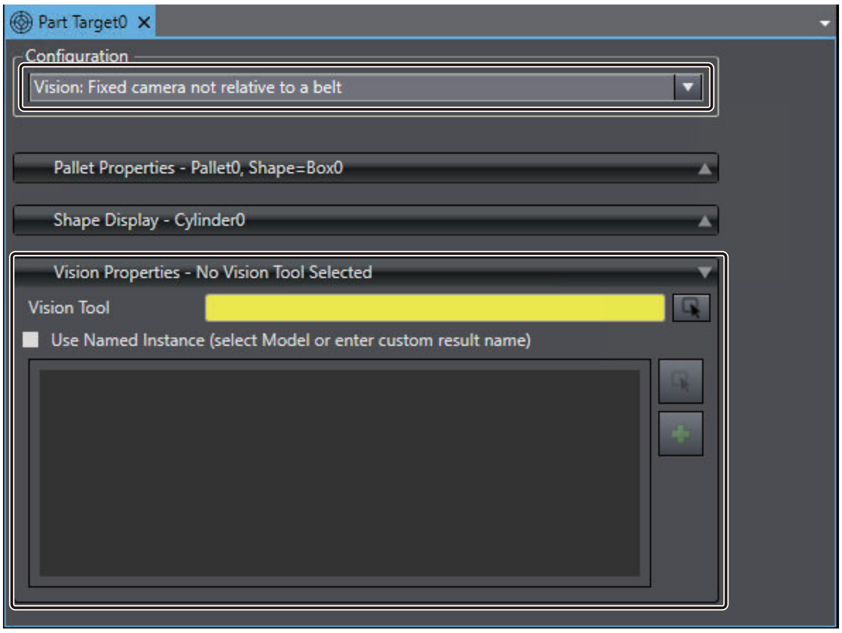
Vision: Fixed Camera Not Relative to Belt

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.

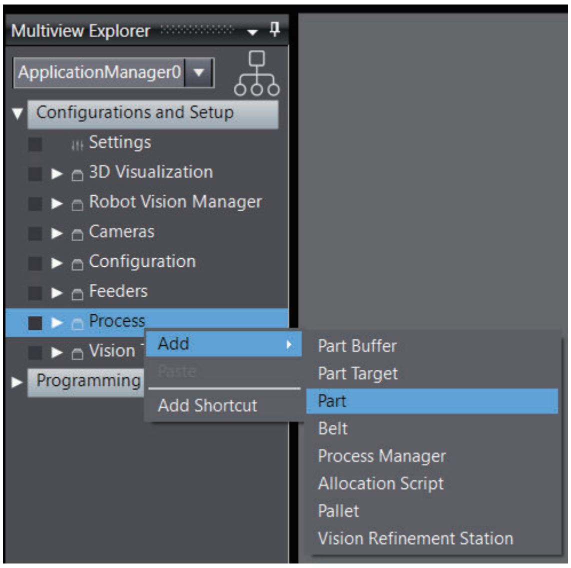
You can optionally specify a named instance and then select a Model or enter a custom result name.



5-9-3 Part

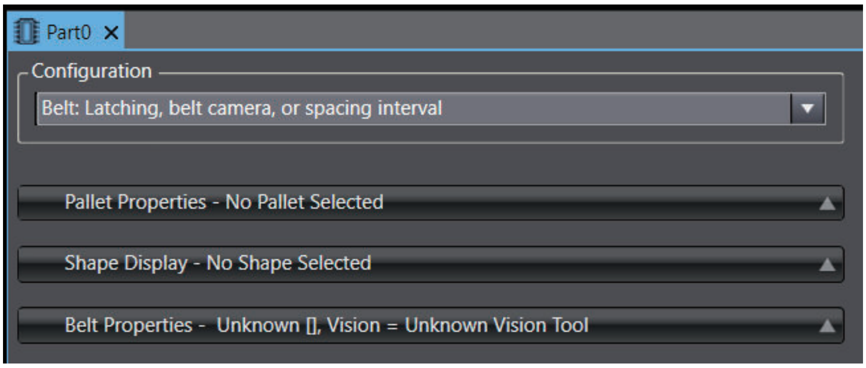
Define a part shape and conditions that the robot follows to pickup and place the part.

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



Part is added under **Process**.

2 Right-click **Part** to select **Edit**.
The edit window will open.



Items on the editor window are as follows.

Item			Description
Configuration			Select the method to pick and place parts. The methods that you can select are as follows. **Belt**:A part is picked from or placed onto a conveyor belt. It may use latching, a camera, or a spacing interval to determine the position. **Static**: Pick a part from a fixed position or place it in a fixed position. Because it is a fixed location, no camera or belt is used. **Vision**: A fixed-location camera is used to determine where to pick or place parts. There is no belt used. Properties you can set depend on your selected method.
Pallet Properties (optionally) *1*2*3	Pallet		Check this box to use a pallet to carry parts, and specify a pallet.
	Shape Display	Shape	Check this box to specify a shape to represent the pallet in the 3D Visualizer.
Shape Display *1*2*3	Shape		Check this box to specify a shape to represent the Part in the 3D Visualizer.
Belt Properties *1	Mode		Specify a belt that is used to handle parts and enter the description of the belt mode. Properties you can set depend on your selected mode.
	Encoder *4*5*6	Belt	Specify the Belt that is used to handle parts.
		Encoders	Specify the belt encoder that is used to control the belt.
	Vision Properties *4	Vision Tool	Specify the vision tool that is used to detect a part on the conveyor belt.
		Use Named Instance (optionally)	Checking this box displays detection results with name . Generally a model name.
	Latch Period in Emulation Mode *7		Specify the Latch Period.
	Spacing *6		Specify the spacing between parts on the conveyor belt. (Unit: mm)
Vision Properties *3	Vision Tool		Specify the vision tool that is used to detect a part on the conveyor belt.
	Use Named Instance		Checking this box displays detection results with name . Generally a model name.

*1. Displayed when **Belt** is selected in the **Configuration** drop-down list box.

*2. Displayed when **Static** is selected in the **Configuration** drop-down list box.

*3. Displayed when **Vision Properties** is selected in the **Configuration** drop-down list box.
*4. Displayed when **Vision Properties** is selected for **Mode** in the **Belt Properties** area.
*5. Displayed when **Latches** is selected for **Mode** in the **Belt Properties** area.
*6. Displayed when **Spacing** is selected for **Mode** in the **Belt Properties** area.
*7. Displayed when **Latches** is selected for **Mode** in the **Belt Properties** area in emulation mode.

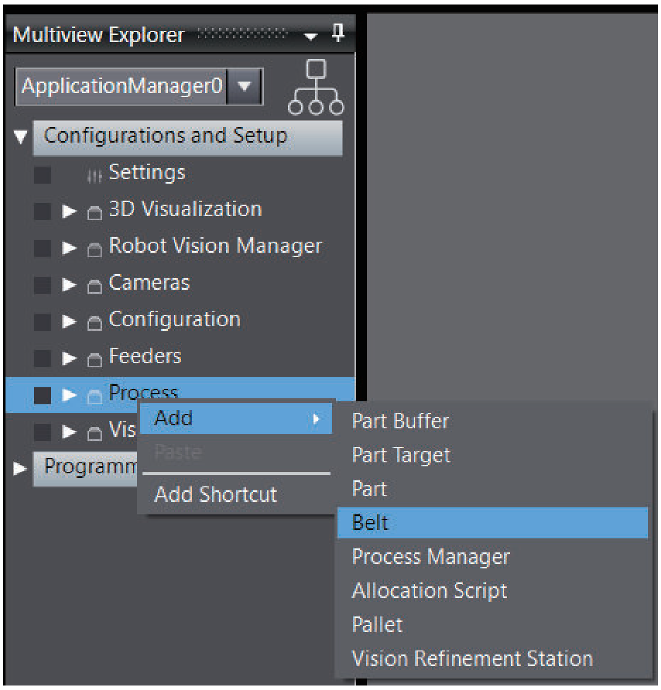
5-9-4 Belt

A Belt 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. After setting up a configuration related to the belt control, place the belt in the 3D Visualization to associate a latch and a controller with the belt.

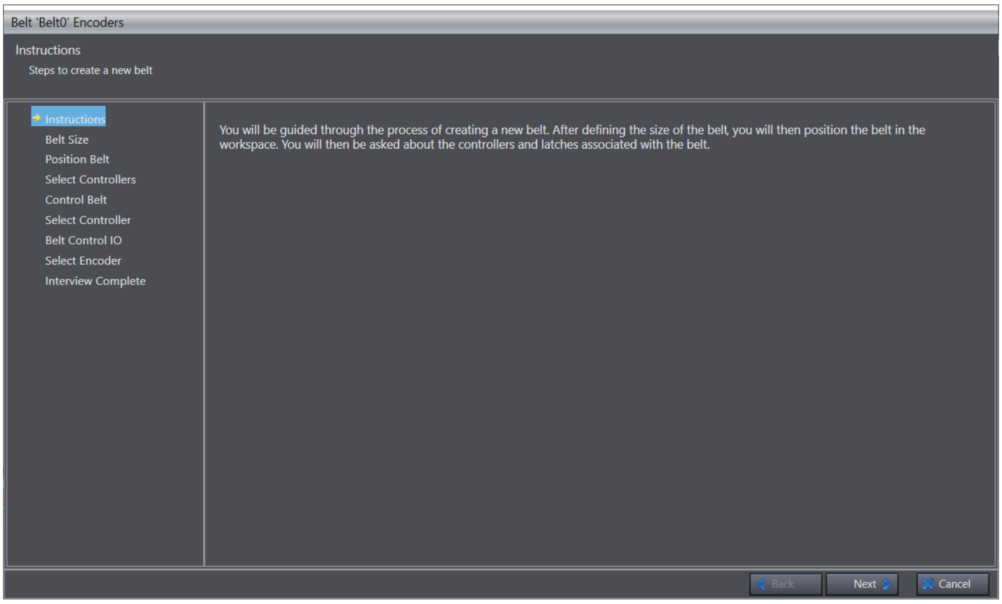
Adding a Belt

Use the following steps to add a Belt.

1 Right-click **Process** under **Configurations and Setup** in the **Multiview Explorer** and select **Add** - **Belt**.

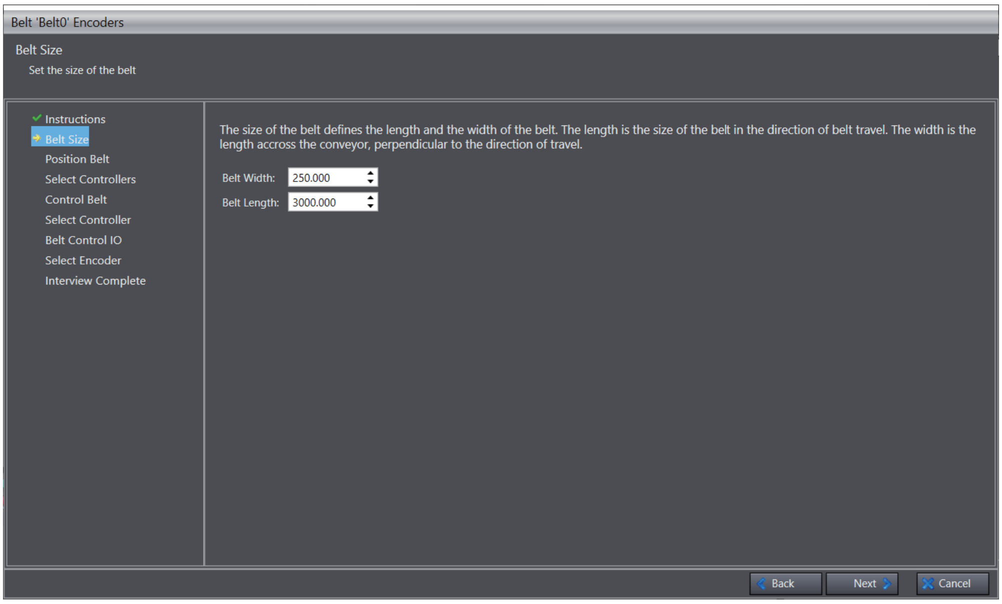


The confirmation message is displayed if you are offline. Follow the message and click the **OK** or **Cancel** button.
Click the **OK** button to open the **Belt** pane.



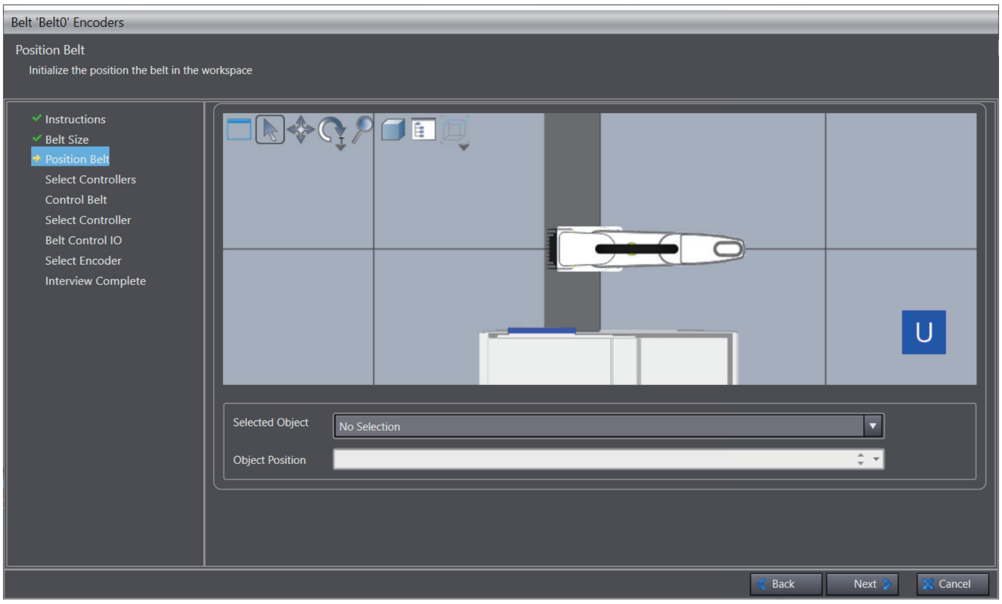
2 Click the **Next** button.

Then the **Belt Size** pane will open.

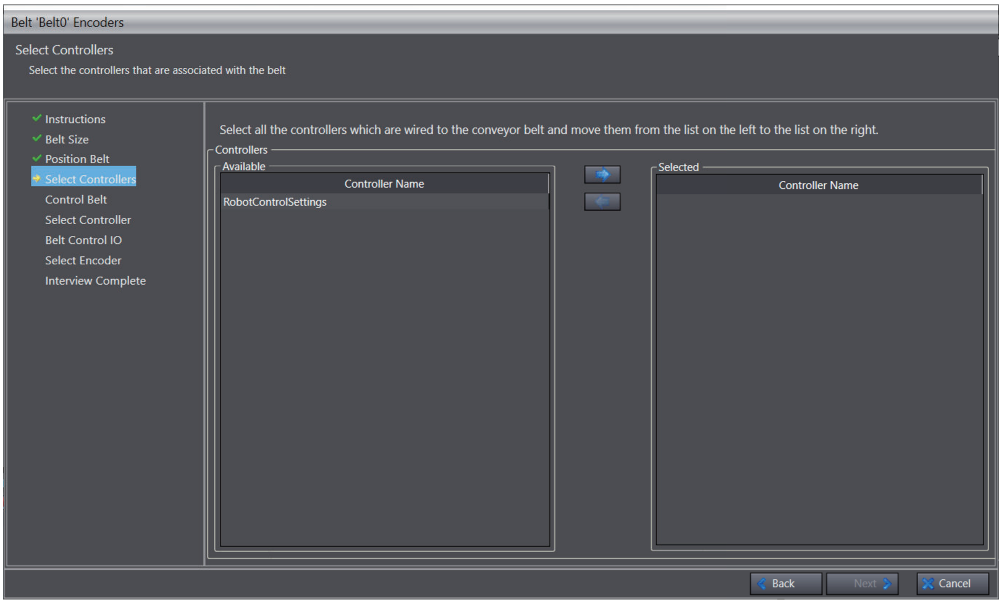


3 Define the width and length of the belt and click the **Next** button.

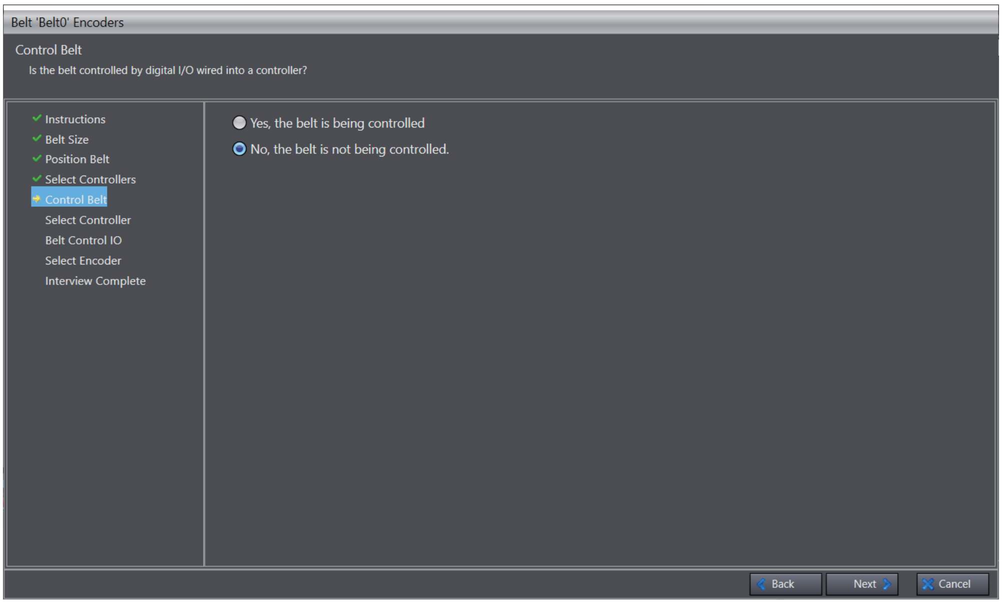
Then the **Position Belt** pane will open.



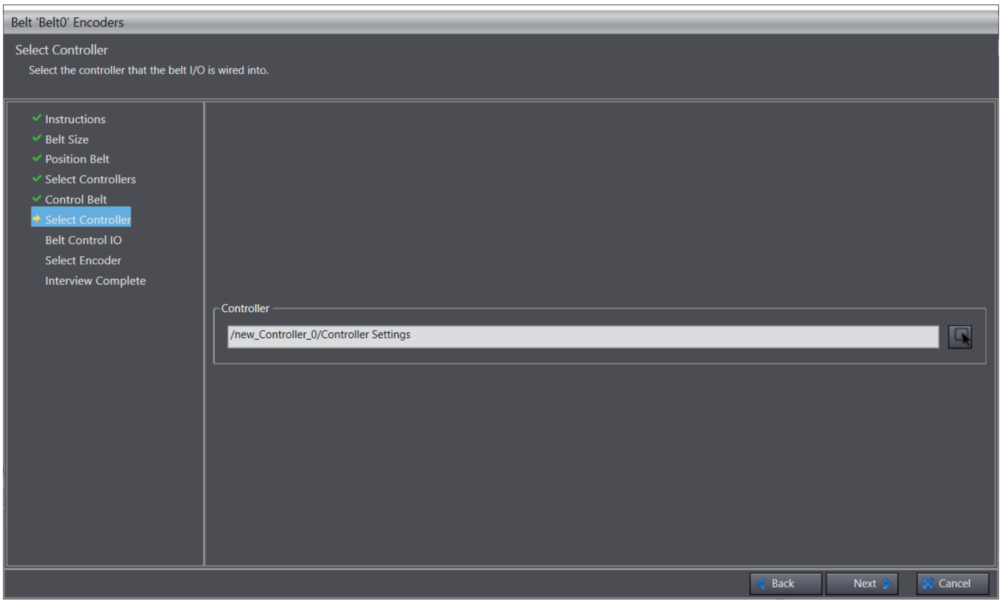
4 Define the belt position in the workspace and click the **Next** button.
Then the **Select Controllers** pane will open.



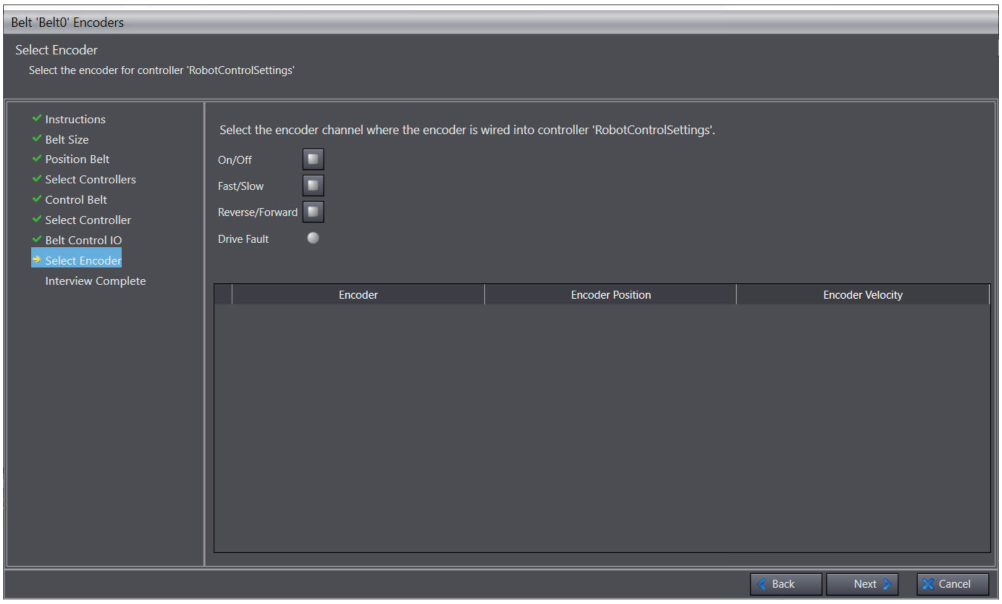
5 Select controllers to associate with the belt and click the **Next** button.
Then the **Control Belt** pane will open.



6 Check the applicable item to see if the belt is controlled by the digital I / O connected to the controller, and click the **Next** button.
If you check **Yes, the belt is being controlled**, the **Select Controllers** pane will open. Proceed to step 7.

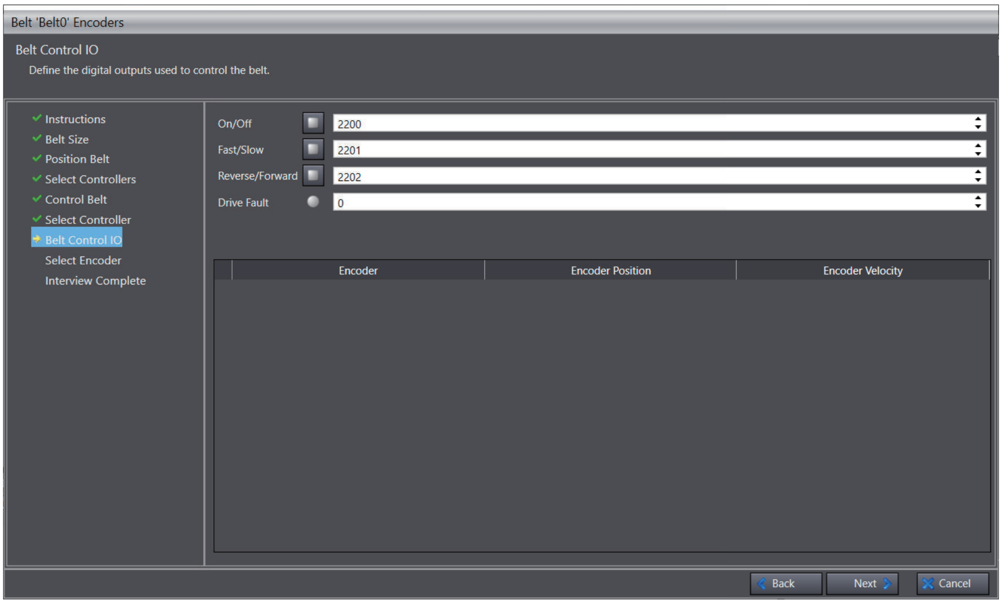


No, the belt is not being controlled, If you check this, the **Select Encoder** pane will open.



Proceed to step 9.

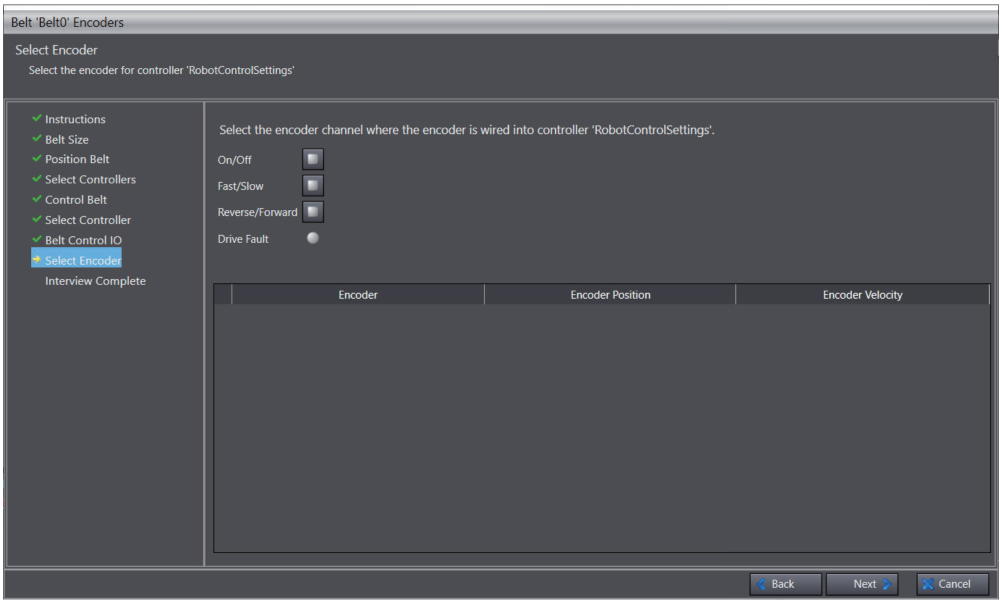
7 Define the connected controller and click the **Next** button.
The **Belt Control IO** pane will open.



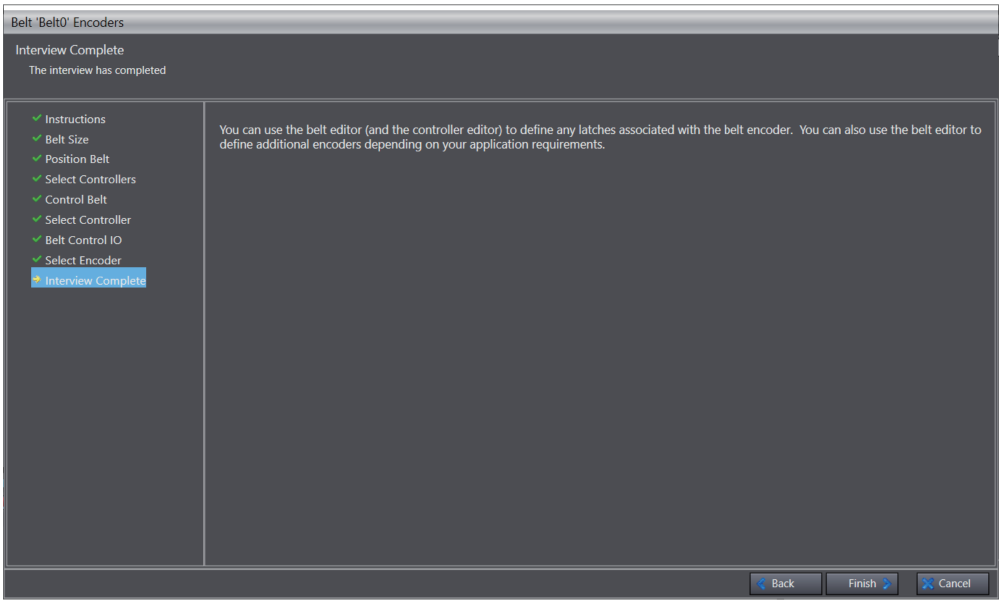
The setting items and descriptions are as follows.

Item	Description
On/Off	Specifies the number of digital I/O signals that are used to start and stop the belt. The belt moves when the signal is turned on.
Fast/Slow	If fast/slow speeds are supported in the belt, enter the number of the digital I/O signals that are used to switch fast/slow speeds of the belt. When the signal is turned on, the belt moves at fast speed.
Reverse/ Forward	If reverse/forward movements are supported in the belt, enter the number of the digital I/O signals that are used to switch reverse/forward movements of the belt. When the signal is turned on, the belt moves in the reverse direction.
Drive Fault	The display turns on to signal the error if the belt control fails.

8 Set the digital output used to control the belt and click the **Next** button.
Then the **Select Encoder** pane will open.



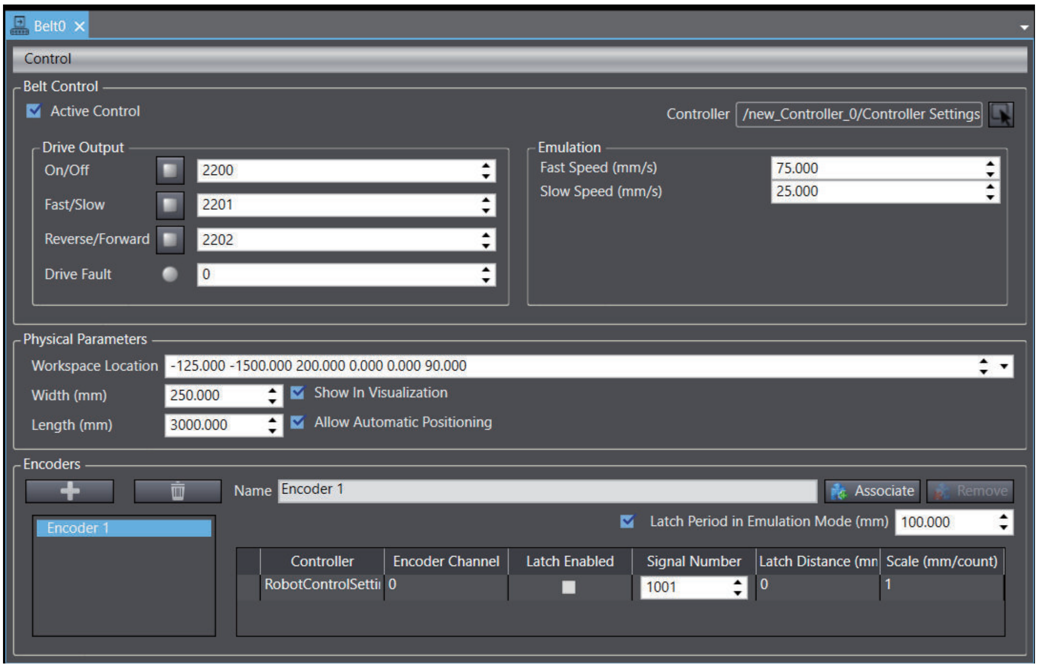
9 Select the encoder channel to connect to the controller and click the **Next** button.
The **Interview Complete** pane is displayed.



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

Belt Configuration Items

Various configuration values are registered in **Belt**. You can edit those values. Right-click the **Belt** you want to edit and select **Edit** to display the edit window.



Items on the editor window are as follows.

Item			Description
Control	View Belt Encoders		Views the operation of the belt encoder to check the values when the belt moves.
	Test Latch Signal		Displays a wizard to perform the test of the encoder latch signal.
Belt Control	Active Control		If you want to control the robot with the defined controller, check this box.
	Controller		Defines the robot controller that is used to control the belt.
	Drive Output	On/Off	Specifies the number of digital I/O signals that are used to start and stop the belt. The belt moves when the signal is turned on.
		Fast/Slow	If fast/slow speeds are supported in the belt, enter the number of the digital I/O signals that are used to switch fast/slow speeds of the belt. When the signal is turned on, the belt moves at fast speed.
		Reverse/ Forward	If reverse/forward movements are supported in the belt, enter the number of the digital I/O signals that are used to switch reverse/forward movements of the belt. When the signal is turned on, the belt moves in the reverse direction.
		Drive Fault	The display turns on to signal the error if the belt control fails.
	Emulation*1	Fast Speed (mm/s)	Defines the distance traveled per second when fast speed is enabled.
		Slow Speed (mm/s)	Specifies the distance traveled per second when slow speed is enabled.
Physical Parameters	Workspace Location		Define the position of the belt in the workspace. The position is represented by the world coordinate system.
	Width (mm)		Defines the width of the belt.
	Length (mm)		Defines the length of the belt.
	3D Visualization		Check this box to display the belt in the 3D Visualizer.
	Allow Automatic Positioning*1		The belt object in the workspace is automatically placed.

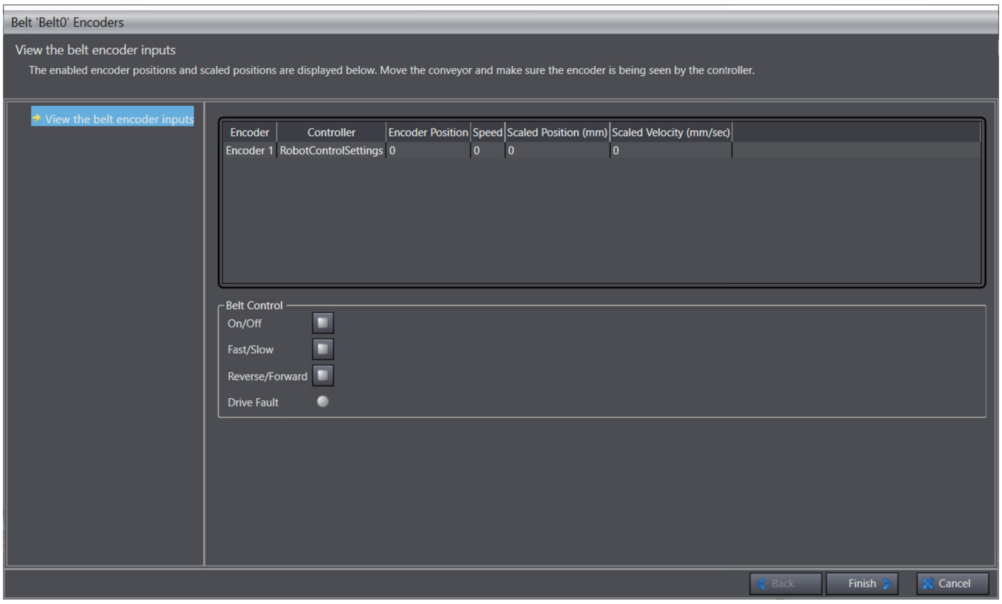
Item		Description
Encoders	Add Button	Adds new encoders.
	Delete Button	Deletes encoders.
	Name	Edits the encoder names.
	Associate/Remove	Defines robot controllers associated with the selected encoder. In ordrer to define the robot controllers, select the encoder that you want to associate and click the **Associate** button. If you want to remove the association, select the rows for the controllers to be removed from the table and click the **Delete** button.
	Latch Period in Emulation Mode (mm)*1	Specifies the Latch Period.

*1. Displayed when Emulation Mode is enabled.

View Belt Encoders

View the operation of the belt encoder to check the values when the belt moves.

1 Click **Control** in the **Belt** display to select **View Belt Encoders**.
The **Belt Encoders** wizard will open.



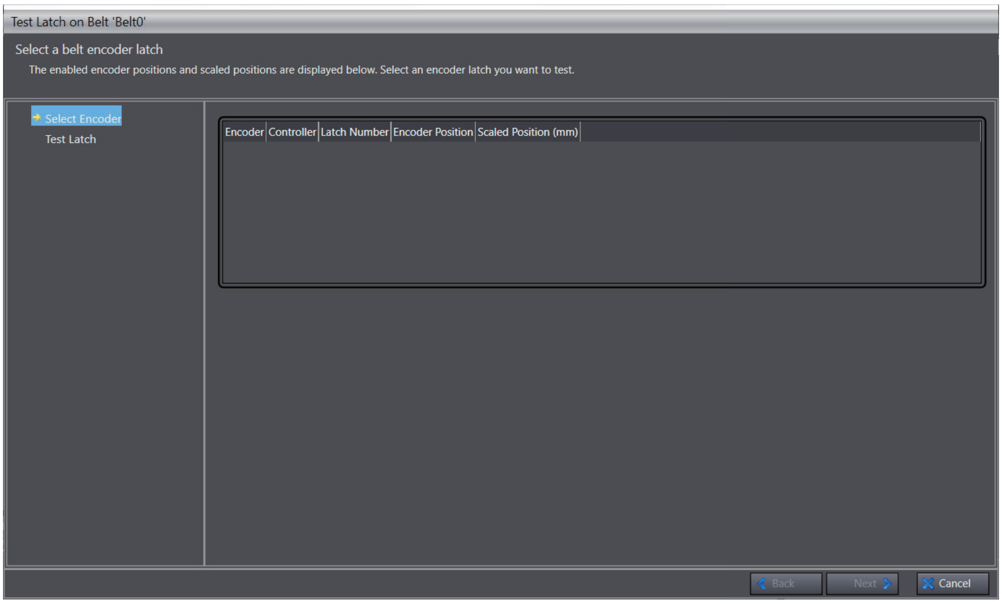
2 Select the belt encoder from the list and then activate the belt and observe the values.
These values should change when the belt moves. You can use the belt controls in the wizard if Active Control is enabled or you are in Emulation mode.
After completion of observing the values, click the **Finish** button to finish the wizard.

Test Latch Signal

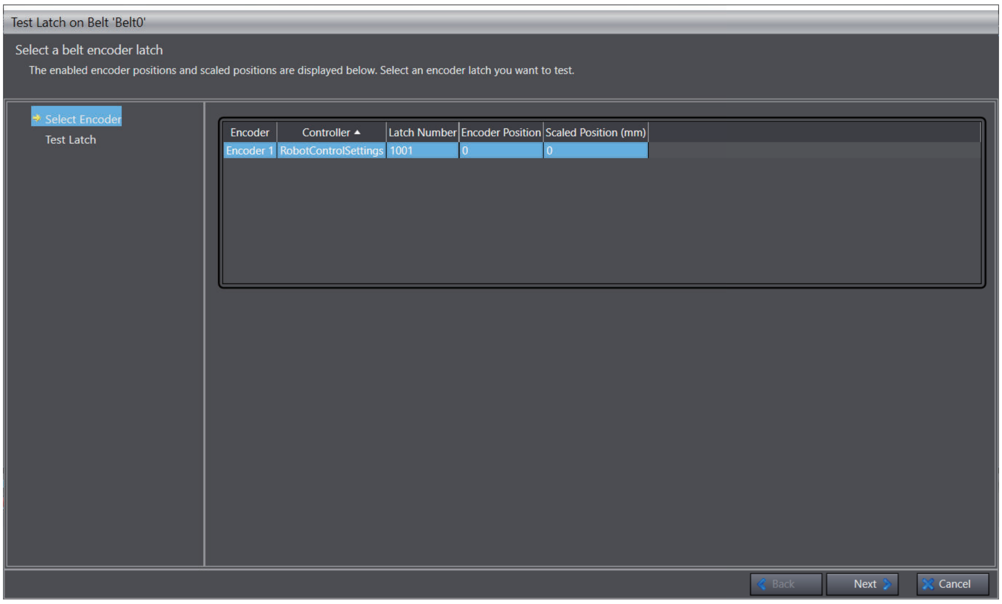
Perform an encoder latch signal test.
A Belt Encoder Latch must be configured and the latch must be enabled in the Encoders section to make the latch signal appear in the Test Latch Signal wizard. Refer to *Sysmac Studio Robot*

Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595) for Configure Belt Encoder Latches.

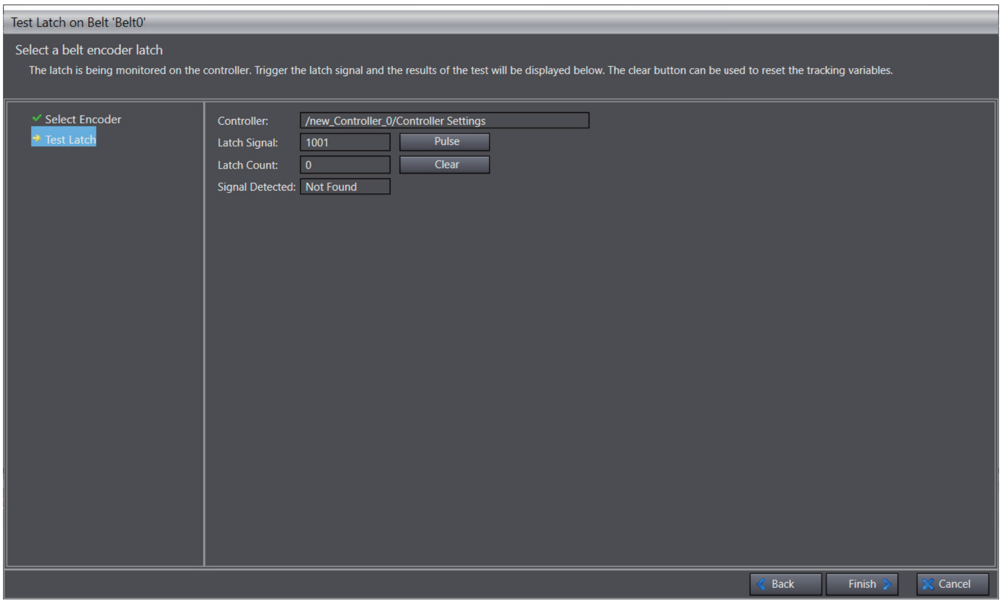
1 Click **Control** in the **Belt** display to select **Test Latch Signal**.
The **Test Latch on Belt** wizard will open.



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



3 Use the **Pulse** button to ensure the Latch Count increments correctly.



4 After the test is completed, click the **Finish** button to finish the wizard.

Active Control

Active Control can control the belt during the calibration and teaching process. It can also optionally be set to control the belt during run time based on part or part target instance location.

If the conveyor belt can be controlled by Robot Integrated CPU unit 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. 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 5-134 for more information.

Physical Parameters

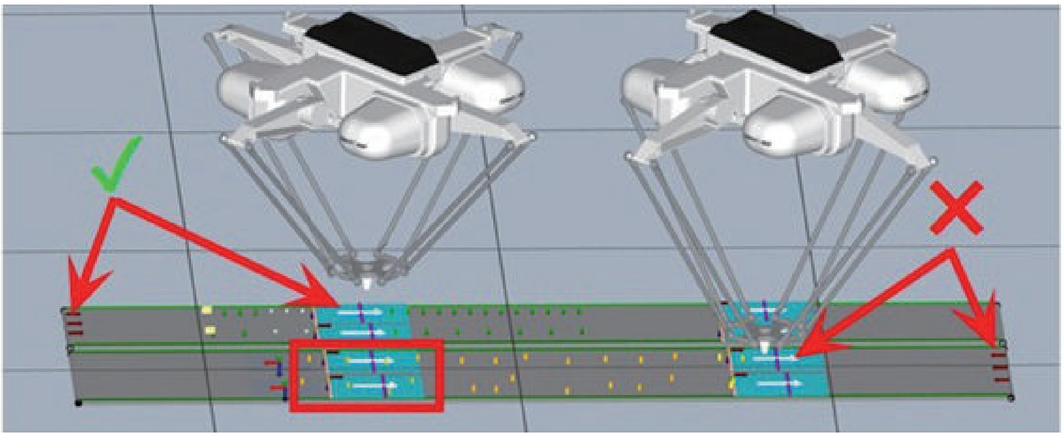
The Physical Parameters are generally set when creating the Belt object, but they can be modified as needed when the Process Manager is not executing. The Workspace Location, Width, and Length of the Belt object should be configured to closely approximate the position of the physical belt relative to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

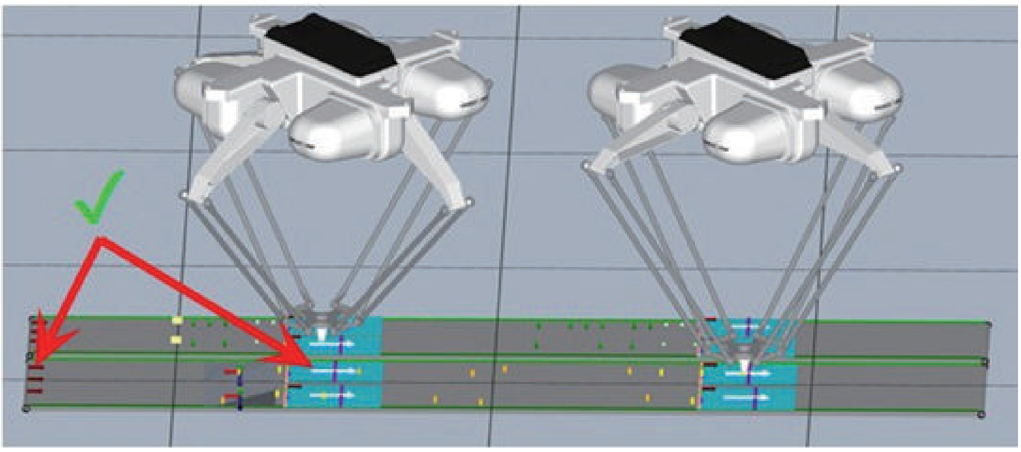
Belt Direction of Travel

In the 3D Visualizer, three red arrows are displayed 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 should allocate parts to the downstream robot first. Therefore, downstream robots process the parts while all upstream robots do not pick pickable parts.



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. The following figure shows, once corrected, that both robots can process parts as expected. Any parts not processed by the upstream robot can be processed by the downstream robot.



Encoders and Encoder Associations

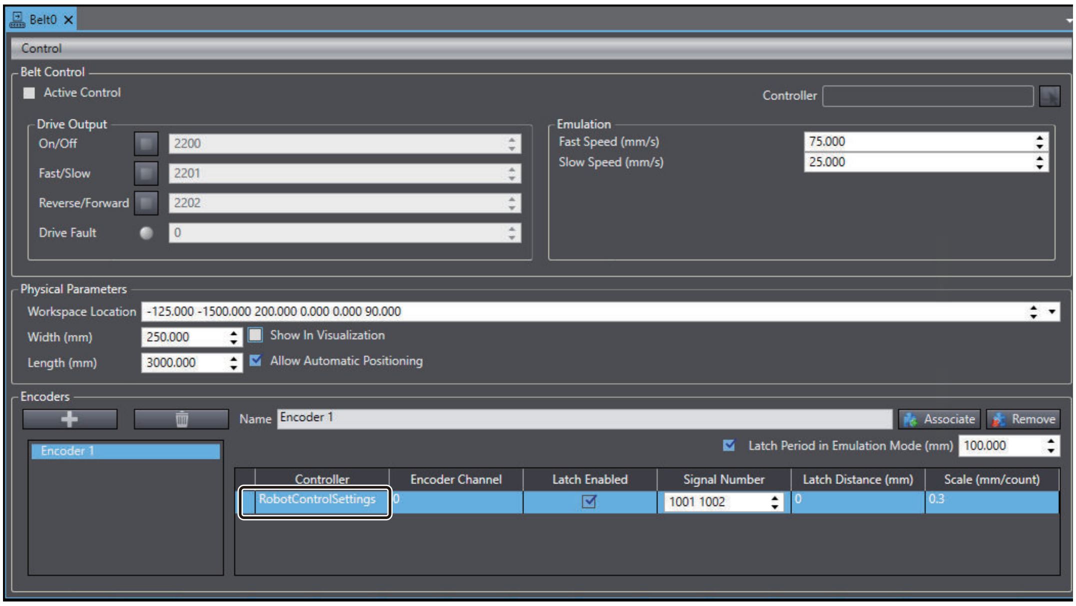
The Belt provides an interface for defining Encoders and Encoder Associations. A single Belt object may be used as a belt source for multiple part and part target types. And it also may be associated with belt encoder inputs for multiple controllers.

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.

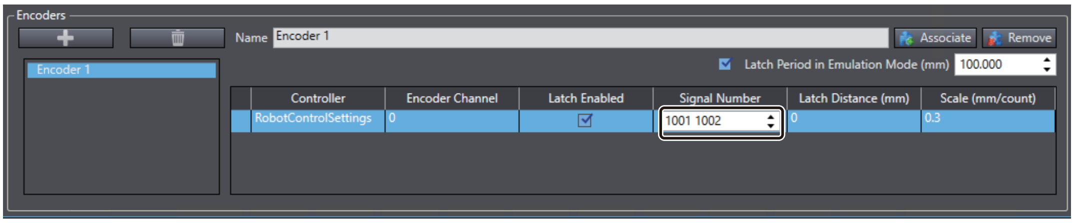
Multiple Cameras Relative to a Single Robot-to-belt Calibration

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

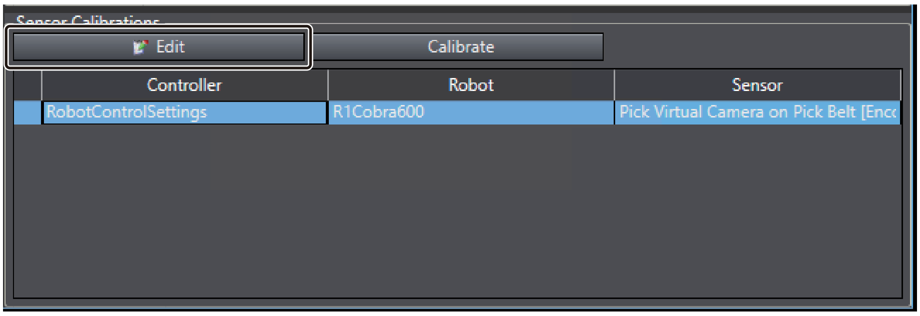
1 Open the Belt and verify that you have associated a controller with the belt encoder.



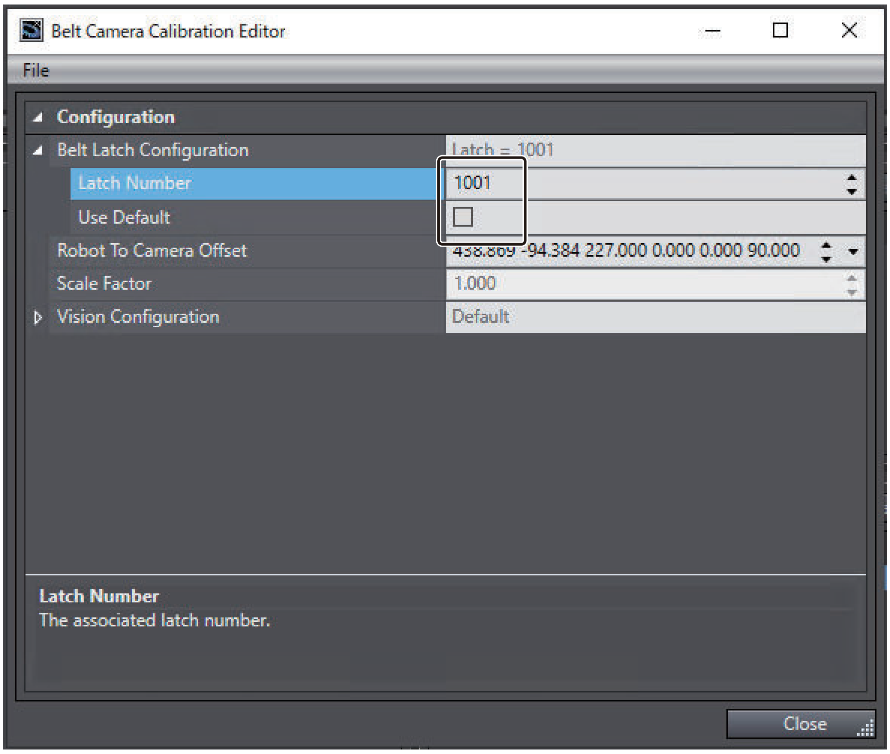
2 Specify the latch signals used by each camera in the Signal Number field.
Multiple signals must be separated by a space, as shown in the following figure.



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



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



5 Repeat steps 3 and 4 for the other belt camera calibrations and corresponding latch signals. Once all belt camera calibrations and latch signals are accounted for, the procedure is complete.

5-9-5 Process Manager

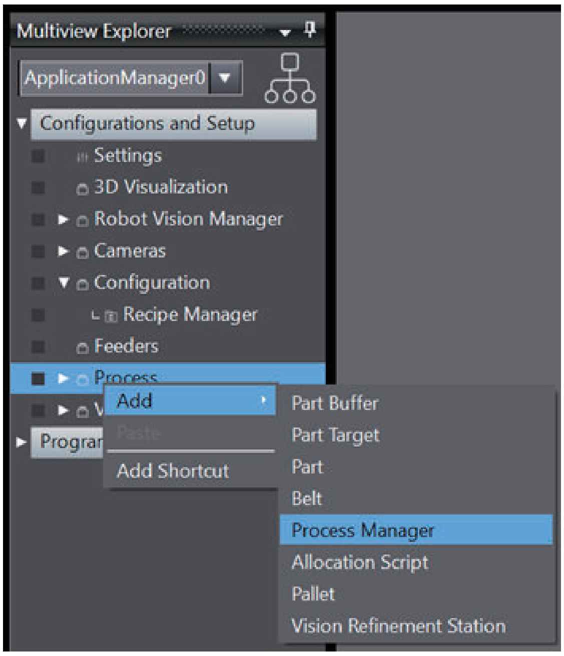
The Process Manager manages operation processes of the Robot Integrated system. It manages part processes such as picking and placing a part in the target position based on the specified execution order.

Additional Information

The Process Manager allows you to create complex applications without writing any programming code.
If you need advanced adjustment, you can modify your application by accessing the V + programming controls.

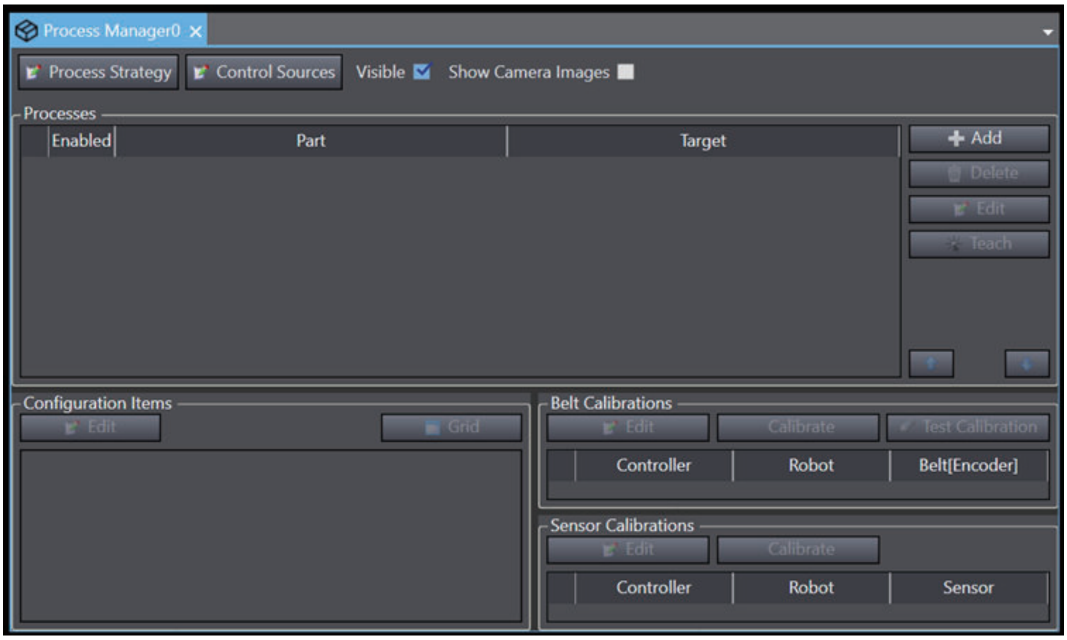
Adding a Process Manager

1 Right-click **Process** under **Configurations and Setup** in the **Multiview Explorer** and select **Add - Process Manager**.



A new **Process Manager** will be added under **Process**.

2 Right-click the **Process Manager** and select **Edit**.
The edit window will open.



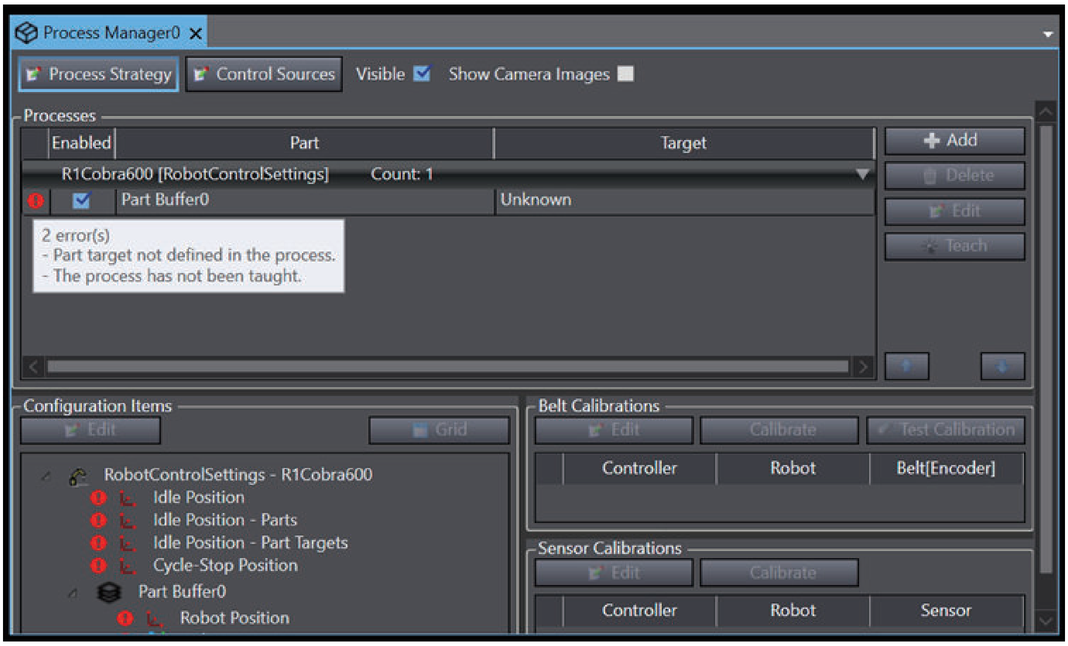
Process Manager Configuration Items

Items on the editor window are as follows.

Item	Description	Reference
Process Strategy	Edits a Process Strategy on the edit screen. The Process Manager allocates the Parts and Part Targets to specific robots based on the configured Process Strategy. The output process is passed to the robot controller queue and operates based on the Process Strategy.	*Process Strategy* on page 5-130
Control Sources	Edits the property of a selected item.	*Control Sources* on page 5-136
Visible	Displays the detection result of calibrations or Parts/ Part Targets in the 3D Visualizer. Displays all calibrations if checked.	-
Show Camera Images	Displays the camera image in the camera calibration / field of view in the 3D Visualizer if checked.	-
Processes	Define a robot, pick location, and place location that will be controlled by the Process Manager.	*Process List* on page 5-139
Configuration Items	Show the relationships between the workcell items in a tree structure.	*Configuration Items* on page 5-148
Belt Calibrations	Calibrate a robot to a conveyor belt.	*Belt Calibrations* on page 5-157
Sensor Calibrations	Define the robot-to-sensor calibrations for the selected workcell process.	*Sensor Calibrations* on page 5-161

● Process Manager Configuration Errors

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



Process Strategy

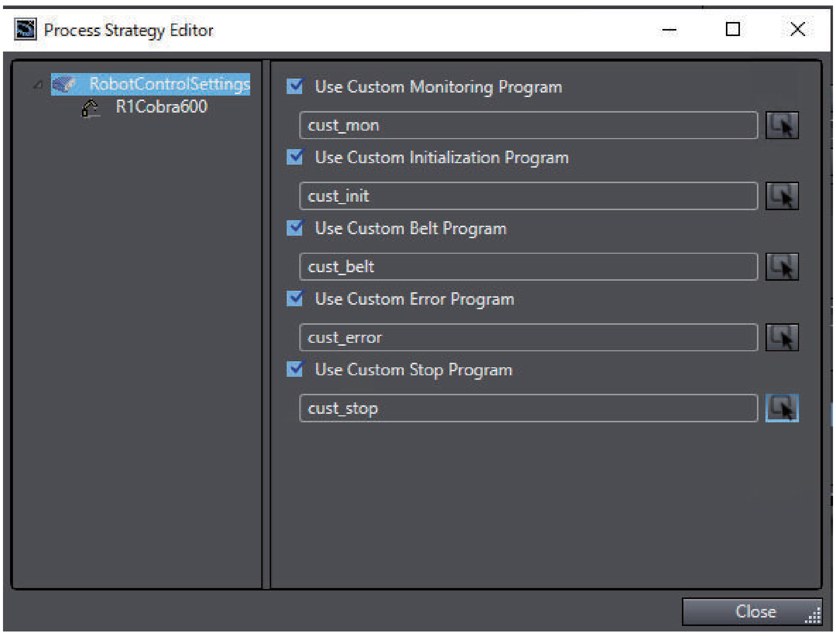
The Process Manager invokes a Process Strategy to determine how to allocate the Parts and Part Targets identified by the Process Manager. It uses the list of Part Processes to allocate the Parts and Part Targets to specific robots. The output of this process is passed to the controller queue by the Process Manager. Each Process Strategy operates under certain assumptions based on the process being monitored. Those assumptions determine which algorithms are used to perform the allocation.
The Process Strategy Editor provides access to the following parameters editors.

- Controller Parameters: Refer to *Process Strategy Controller Parameters* on page 5-130 for more information.
- Robot Parameters: Refer to *Process Strategy Robot Parameters* on page 5-131 for more information.
- Belt Control Parameters: Refer to *Process Strategy Belt Control Parameters* on page 5-134 for more information.

To access the Process Strategy editor, click the **Process Strategy** button. he appropriate editor is shown based on the object selected in the left pane of the Process Strategy Editor.

● Process Strategy Controller Parameters

The **Controller Parameters** are displayed when the controller is selected in the Process Strategy Editor. The **Controller Parameters** group is used to specify custom V+ programs for the selected controller.

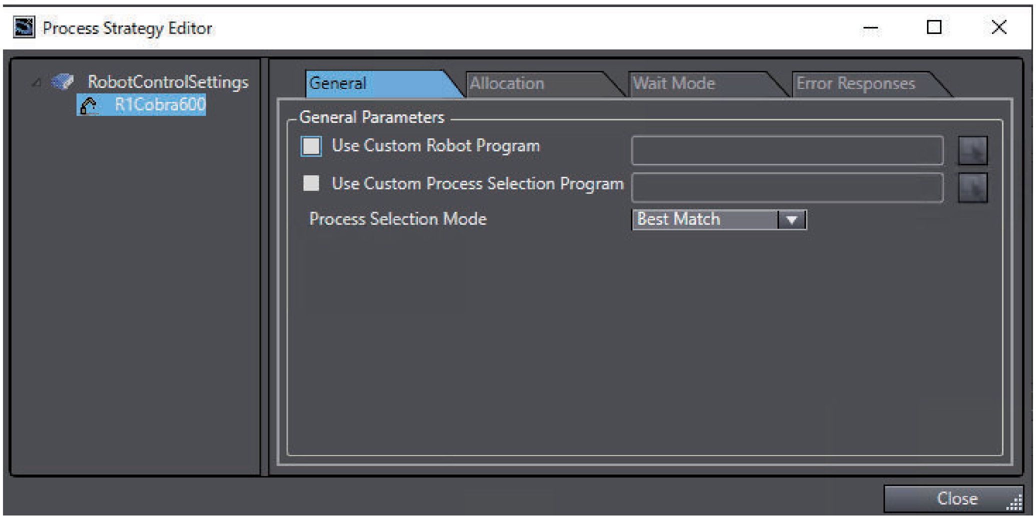


Item	Description
Use Custom Monitoring Program	The default process monitoring has the following functions. • Checks for updates to process strategies. • Handles belt monitoring. • Monitors parts and part targets. You can copy the default V+ monitoring program for editing, or select an existing program. Most applications do not require a Custom Monitoring Program.

Item	Description
Use Custom Initialization Program	The default initialization program that executes before the control programs (robot, belt, process strategy, etc.) are started. This can be used to initialize system switches, parameters, and variables or execute V+ programs that need to be started in parallel to the Process Manager. You can copy the default V+ monitoring program for editing, or select an existing program.
Use Custom Belt Program	The default belt program monitors the speed / ON / OFF status of all belts. You can copy the default V+ belt program for editing, or select an existing program. Most applications do not require a Custom Belt monitoring program and can be sufficiently controlled using Process Strategy Belt Control Parameters as described in this section.
Use Custom Error Program	The default error program handles the processing and reporting of errors during the execution of a process. 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. All Process Manager V+ program error handling will lead to this program. Use this program to automate error handling of errors that are reported to the PC by default. This program will check if any user-defined error responses exist in the Process Strategy - Robot parameters.
Use Custom Stop Program	The custom stop program can be used to perform certain operations after the application has stopped. You can copy the default V+ stop program for editing, or select an existing program.

● Process Strategy Robot Parameters

The **Robot Parameters** are displayed when the robot is selected in the Process Strategy Editor. There are four tabs of robot parameters: the **General** tab, the **Allocation** tab, the **Wait Mode** tab, and the **Error Responses** tab that are described below.



Group	Item	Description
General	Use Custom Robot Program	Allows you to specify a custom V+ main robot-control program. For example, this would allow you to customize exactly when a given robot executes each process. You can copy the default V+ program for editing, or select an existing program.

Group	Item	Description
	Use Custom Process Selection Program	Allows you to specify a custom V+ process-selection program. For example, this would allow you to customize the selection and ordering of the processes. 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 downstream in the belt window of the associated robot. If no process is belt-relative, it will select the first available process in the list. As Ordered (no Timeout): Selects the processes in the order they are listed. For example, if three processes are listed, it will select process 1, then process 2, and then process 3. The currently selected process must be completed before the next one can begin. When the last process has completed, the list repeats. As Ordered with Timeout: Selects the processes based on their order and user-supplied timeout (the timeout is specified in milliseconds). Normally, it selects the next process in the list, but if the process cannot be selected within the given timeout, it will move to the next process for possible selection.
Allocation	Part Filtering Mode/Target Filtering Mode	Specifies how instances are identified for processing by this robot. The filtering process occurs before the allocation process. No Filtering: All instances are sent to the robot. Pick / Skip Instances: Processes a certain number of instances then skips a certain number of instances. Even Distribution: Distributes instances by allocating an equal percentage to all robots. Percentage of Instances: Allocate a percentage of instances evenly across the range of available instances. Skip Rate: The robot should process and skip instances to maintain a balanced flow and robot work load in a multi-robot system. Relative Belt Position: Process instances that are separated by a certain distance. Pick / Skip Pallets: Process a certain number of pallets then skip a certain number of pallets.
	Queue Size	Specifies the queue size for the robot. Each robot has a queue size. It represents the maximum number of Part or Part Target instances that can be sent to the robot. Each part and target is associated with a different queue for a given robot. This parameter defines the size of each queue. This parameter has a default value of 10. To change the parameter, enter the new value into the **Queue Size** field, or adjust the value using the up/down arrows.

Group	Item	Description
	Allocation Distance	Specifies the distance upstream from the belt window that a part instance must be within before the system is allowed to allocate that part instance to the robot. This can be considered the maximum distance upstream that an instance can be allocated to the robot.
	Allocation Limit	Specifies the distance upstream from the Process Limit Line that a part instance must be to allocate that part instance to the robot. This can be considered the minimum distance upstream that an instance must be for allocation. If set to zero, instances will be allocated if they are at or upstream of the process limit line. If greater than 0, it represents the upstream distance (in mm) from the process limit line an instance must be for it to be allocated. This parameter is useful in the case of fast-moving belts where the robot needs additional look ahead to process instances. For example, if a part pallet has four instances and the robot queue size is two, when the robot picks instance one, there is now space in the queue for one more instance. If instance three is upstream of the allocation limit, it will be allocated but if it is downstream of the allocation limit, (>0) it will not be allocated. This prevents allocation of instances that are in the downstream portion of the allocation area.
	Use Custom Allocation Script	The Custom Allocation Script provides two different entry points that are called. One is for allocating nonbelt instances (parts and targets). The other is for belt instances. The program can manipulate the lists to indicate to the system what should be allocated to a given robot. This is called by the Process Manager when it needs to allocate parts. If this option is enabled, click the **Select** button ( ) to select the desired Custom Allocation Script. Refer to the *ACE Reference Guide - Custom Allocation Scripts* for more information.
Wait Mode	Stay at current position	Causes the robot to remain at the current position while waiting for a part or target to process.
	Move to Idle Position	Causes the robot to move to the idle position after the specified **After waiting for** time (in milliseconds) while waiting for a part or target to process. For example, if the **After waiting for** time is 500 ms, when there is a break in the part or part target processing queue, the robot will move to the idle position after 500 ms. This must be enabled for the **Idle Position Parts** and **Idle Position Part Targets** to be used. Refer to *Idle Positions* on page 5-152 for more information.

Group	Item	Description
	Use Custom Wait Program	Allows you to specify a custom V+ wait program. The program would be called when the robot does not have a part or target available. The program could check to see if the robot needs to be moved to the idle location or if it should stay at the current location. This program starts 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	When a cycle stop is issued and this option is enabled, the specified I/O signal will be turned ON when the robot has reached the cycle stop state. When the cycle is resumed (cycle stop state is canceled), it will turn the specified signal OFF and will attempt to enable power, if high power was disabled.
Error Responses	Output Signal - On Error	Defines a digital signal to assert when an error is detected for the selected robot.
	Error Listing (Use the **Add**, **Delete**, **Up / Down** buttons to arrange the list).	Specifies how specific error conditions are to be handled. 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. Single Error Code: User-defined error code. Range of Error Codes: User-defined error code range. Belt Window Access Error: Belt window violations. Robot Power Errors: Problems with power being enabled or enabling power. Gripper Errors: All gripper actuations. All Errors: All other errors. When an error is detected for the robot, it will process the error listing handlers defined and find the first one that can handle the condition. If the error cannot be handled by an item in the list, the error is reported to Task Status Control. Refer to *Process Strategy Controller Parameters* on page 5-130 for more information.

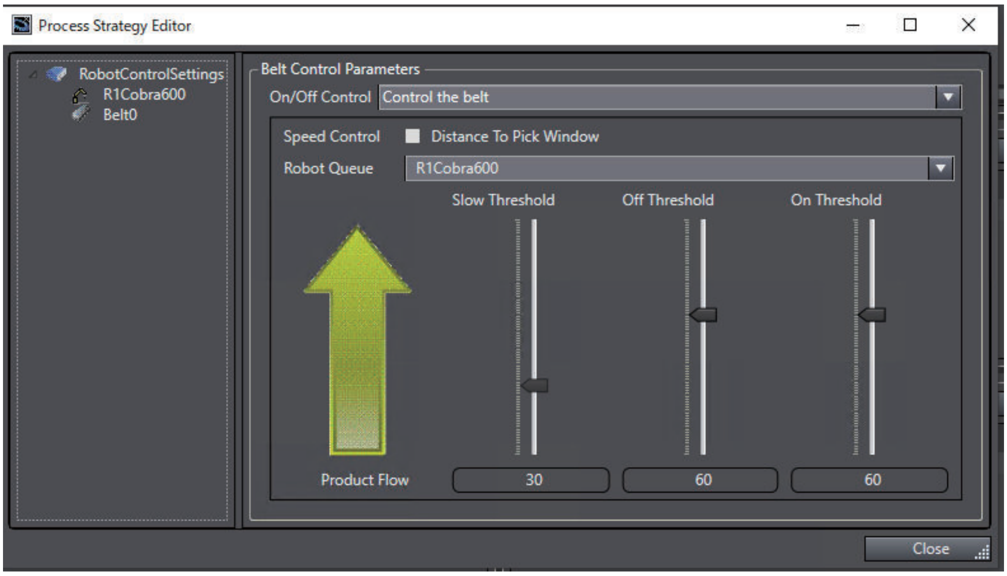
● Process Strategy Belt Control Parameters

The **Belt Control Parameters** are displayed when the belt is selected in the Process Strategy Editor.

Belt Control Parameters are only available when the following items are configured.

- **Active Belt Control** is enabled in the Belt object configuration.
- A **controller** is selected in the Belt object configuration.
- A defined process includes a Part or Part Target that references the Belt Object.

The **Belt Control Parameters** group as shown in the following figure is used to set the belt control parameters for the selected conveyor belt. These parameters can be set to determine when a conveyor belt is turned ON or OFF. An optional speed control parameter is also provided. The decision point for the belt I/O control is based on the selected robot. If objects on the belt in the selected robot queue reach the specified thresholds, the belt will be turned OFF or the belt speed will be adjusted.



Item	Description
On/Off Control	Specifies the ON / OFF control of the belt. There are three selections available: Do not control the belt: Use this option if there are output signals that can control the belt, but you do not want the Process Manager to control the belt during run time. If the belt control is provided by a PLC and output signals are not able to control the belt, disable Active Control in the Belt object. Leave the belt always ON: The belt is turned ON when the process starts and OFF when the Process Manager is stopped. Control the belt: (Default) The belt is controlled based on thresholds described below. The belt is automatically turned OFF when the Process Manager stops the application.
Speed Control	If this is selected, 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.
Robot Queue	Specifies a robot for queue monitoring (the queue size for the robot is set in the Robot Parameters group). The selected robot will typically be the most downstream robot if multiple robots service the same belt. If parts get to the last robot, it needs to slow / stop the conveyor to ensure all Parts or Part Targets are processed.
Slow, Off, and On Thresholds	These thresholds are used to control the belt based on instance position. This is useful for preventing the belt from feeding the robot faster than the robot can pick the parts, or preventing not-processed Part or Part Targets from passing the most downstream robot. Slow Threshold: Specifies the point in the belt window for slowing the conveyor if parts reach that point. For example, 50% means that if a part reaches the midpoint of the belt window, the conveyor will be slowed. Off Threshold: If an instance reaches this threshold, the belt will be turned OFF. This is set as a percentage from upstream belt window limit to downstream process limit and is visualized as a black line in the 3D Visualizer (in the belt window for the selected robot). On Threshold: When a belt is turned OFF by the **Off Threshold**, the belt will remain OFF until all instances are removed between the **Off Threshold** point and the **On Threshold** point. This can be used to minimize the number of times the belt is started and stopped. This is set as a percentage from upstream belt window limit to downstream process limit and is visualized as a green line in the 3D Visualizer (in the belt window for the selected robot).

Item	Description
Product Flow	Shows the product flow (belt window) direction of travel in relation to the **Slow** and **Off Threshold** slide controls. It is a reference for the thresholds. The bottom of the arrow represents the start of the belt window. The top of the arrow represents the Downstream Process Limit.

Control Sources

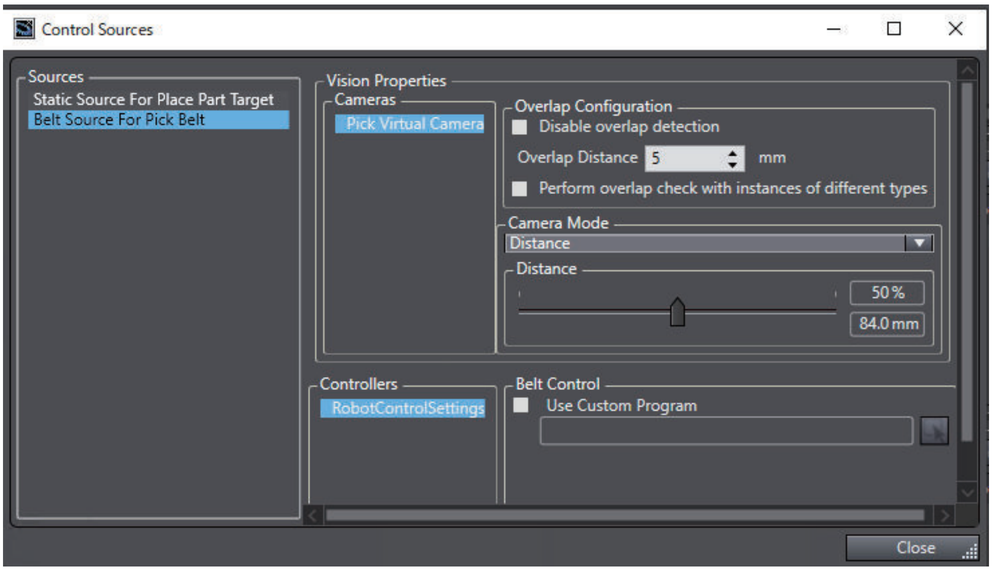
The Control Sources editor provides access to parameters that affect Part and Part Target sources for the defined processes. Sources are responsible for providing instances to a process. These are automatically created based on the Part and Part Target object configuration property. There are three types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



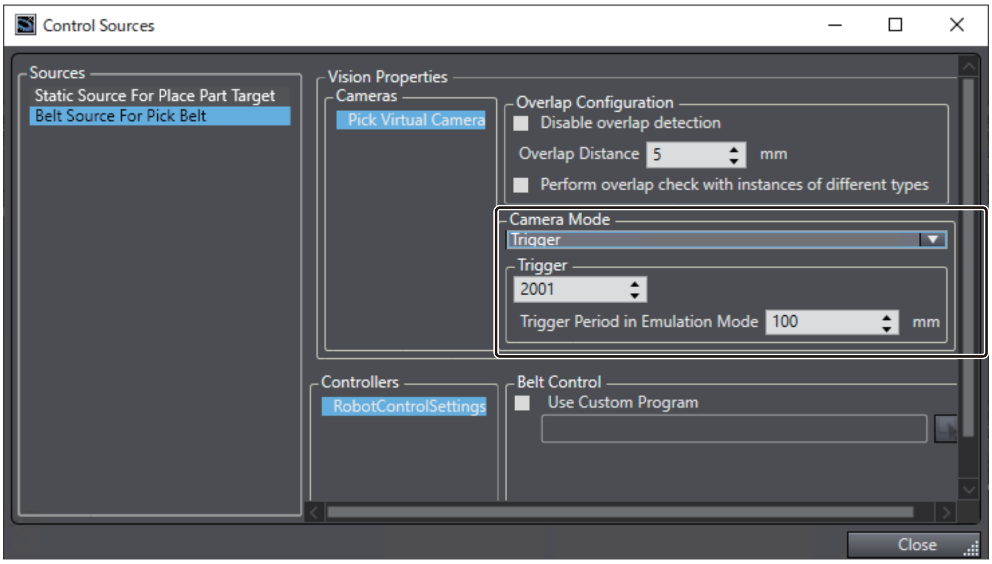
- Cameras
  The Cameras list will display the Virtual Cameras that have been associated with this belt through the Part and Part Target object configurations. You can change the Vision Properties for the selected camera in this area.

- 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 duplicate new instance.
If **Perform overlap check with instances of different types** is selected, the overlap calculation will check for overlap of any parts, rather than just parts of the same type.

- Camera Mode
  There are three camera modes available: Distance, Trigger, and Time that are described below.
  When **Distance** is selected, the Field of View Picture interval control is enabled. This control is used to adjust the picture interval relative to belt travel. The setting is displayed as a percentage of the field of view, and in millimeters (mm) as calculated from the calibration in the selected virtual camera.
  When **Trigger** is selected, the Trigger Signal control is enabled. This specifies the signal number to use for the vision trigger. When the specified trigger signal number is activated, a new vision image will be acquired.
  For example, this can be used in an application where an image only needs to be acquired when an object activates a sensor if it is below the camera. In this case, the trigger signal is wired to the robot controller and should not be confused with applications that require triggering the camera directly. Triggering a camera directly is configured in the Virtual Camera object Acquisition Settings.

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



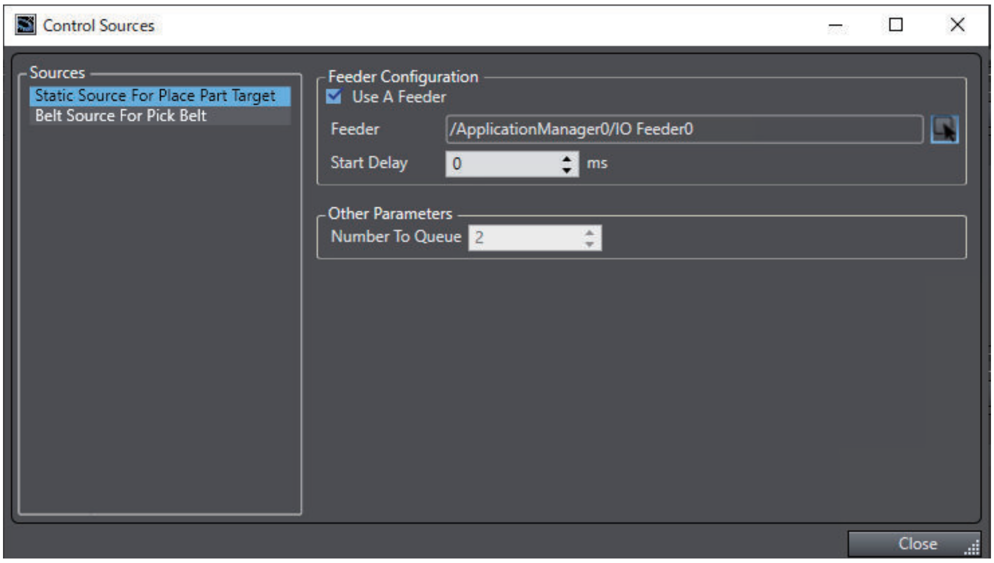
  When **Time** is selected, an image will be requested on the specified time interval.

- 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.

Some 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.

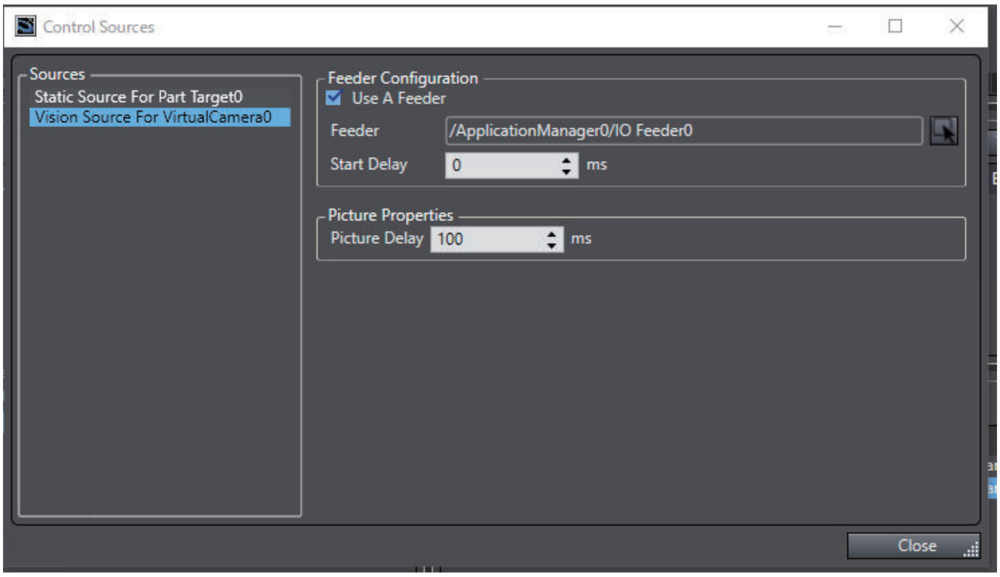


Static Sources are used for Part and Part Targets that are not related to belts or cameras.
If an **IOFeeder** is not enabled, then the PC generates instances for these sources at the Robot Frames defined in configuration items. Each time the controller has emptied the queue of instances, the PC will generate another set of instances and pass them to the controller. The quantity of instances generated is set by the **Number to Queue** property. The default value is two instances to overcome any disturbances in the communication flow between the PC and controller.
Alternatively, select **Use A Feeder** and choose an IOFeeder object that controls when parts are generated. When this is selected, another V+ program is executed for monitoring feeder activity. This can be used for individual parts or pallet configurations. For example, associate an IOFeeder with a target pallet source to use an input signal from a sensor to detect 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.

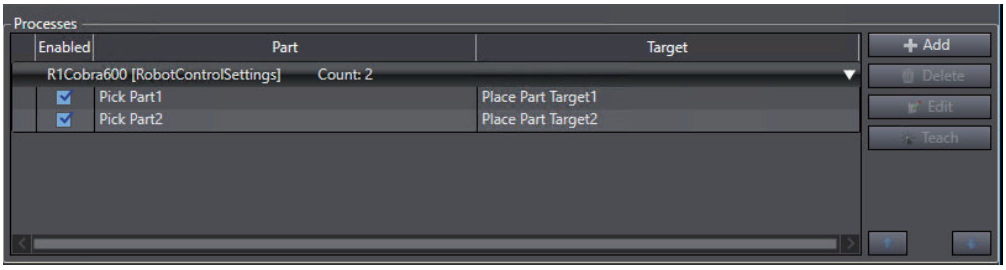


Vision Sources are used for fixed-mounted cameras. They can be associated with IOFeeder objects, similar to Static Sources.

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 Processes List of the Process Manager. It defines the elements (robot, pick location, place location) for a particular process that will be controlled by the Process Manager.

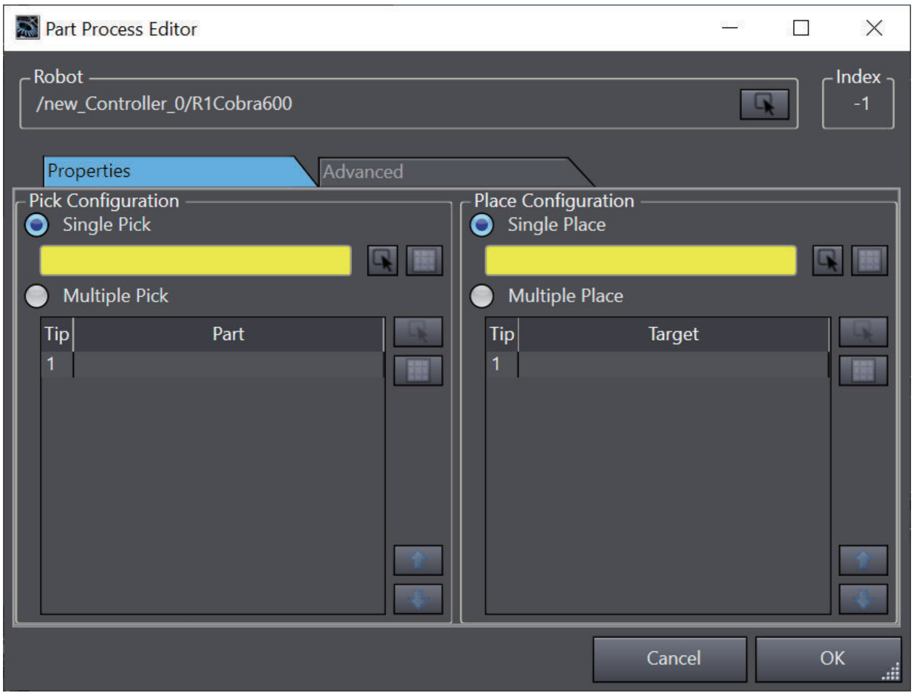


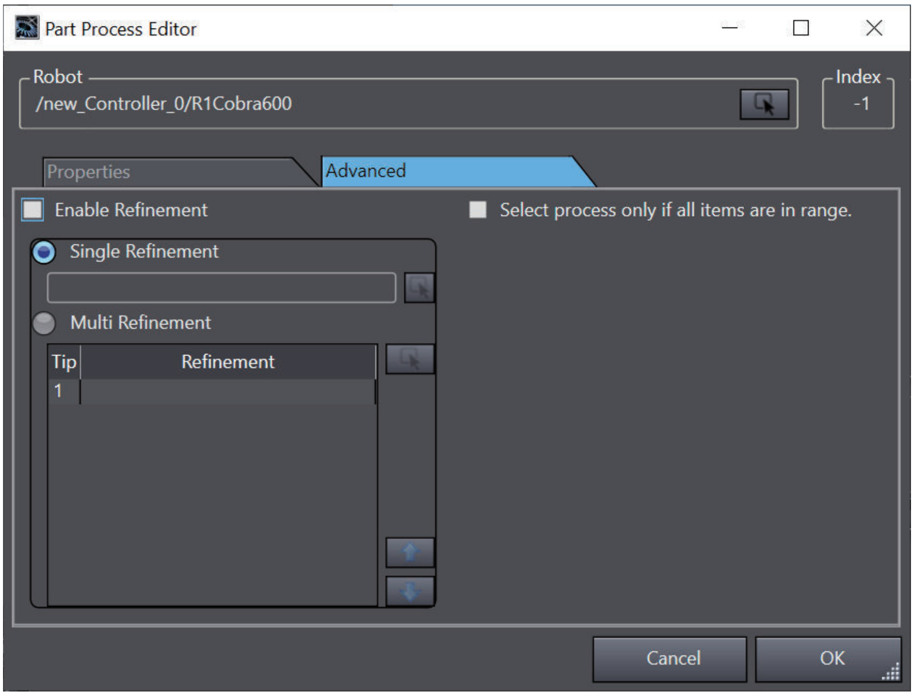
Item	Description
Up/Down Buttons	Sets the priority for the selected process. The process at the top of the list receives the highest priority. Refer to *Changing Process Priority* on page 5-146 for more information.
Enabled	If checked, the process is enabled for use.
Part	The Part(s) specified for the process. You can double-click this item or select the process and then click the **Edit** button to change it.
Target	The Part(s) specified for the process. You can double-click this item or select the process and then click the **Edit** button to change it.
Add Button	Remove the selected process from the Processes list.
Delete Button	Remove the selected process from the Processes list.

Item	Description
Edit Button	Edit the selected process. Refer to *Part Process Editor* on page 5-140 for more information.
Teach Button	Teach the selected process. Refer to *Teaching a Process* on page 5-147 for more information.
Alert Icon	Indicates the process needs to be taught or there is some other problem. Hover the mouse pointer over the icon to view the alert message for a description of the reason(s) for the alert.

Part Process Editor

The Part Process Editor is used to specify the items used in the process. Access the **Part Process Editor** by double-clicking an existing process, or by clicking the **Add** button for a new process.



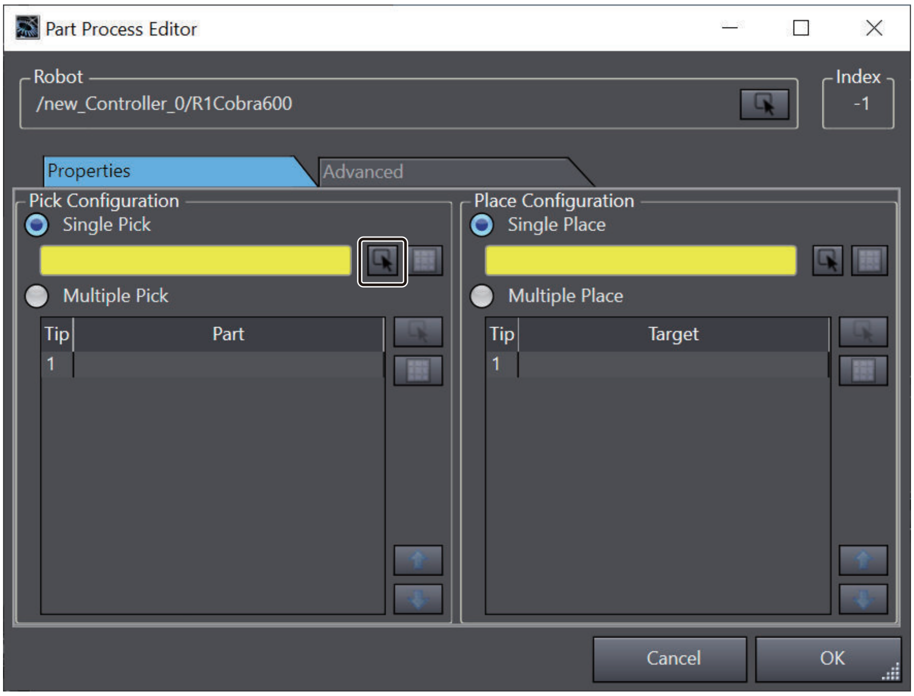


Item		Description
Robot		The Robot reference is used to specify the robot that will be used for the pick-and-place process. Use the **Select** button to open the Select a **Reference** Dialog Box, which allows you to create or specify the robot that will be used in the pick-and-place process.
Index		This item displays the index number of the process. This number can be referenced in V+ programs and C# programs.
Properties Tab	Pick Configuration	The single or multi-pick Part items for the pick and-place process are specified. 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* on page 5-142 and *Multiple Pick* on page 5-142 for details.
	Place Configuration	The single or multi-place Part Target items for the pick and-place process are specified. Refer to *Single Place* on page 5-143 and *Multiple Place* on page 5-143 for details.
Advanced Tab	Enable Refinement	This item is used to perform a position refinement operation over an upward-facing camera to improve part placement accuracy. This is used when placement error needs to be smaller than error introduced during the pick operation. When this option is selected, the part is moved to a predefined vision refinement station for additional processing before being moved to the place (Part Target) location. Refer to *5-9-8 Vision Refinement Station* on page 5-179 for more information. Refer to *Single Refinement* on page 5-144 and *Multi Refinement* on page 5-145 for the selection of refinement targets.
	Select Process Only if All Items are in Range	The check box tells the system to only select this process if all parts and targets are in range of the robot. Refer to *Select Process Only if All Items are in Range* on page 5-145 for usage examples of this item.

● 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* on page 4-10 for more information.

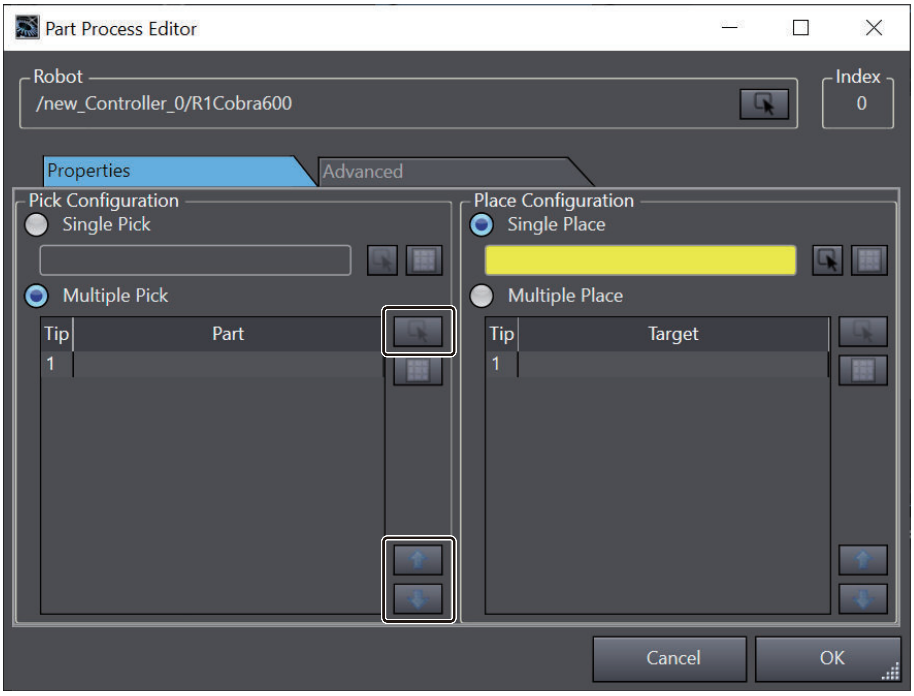


If you want to use a pallet, use the **Pallet Slot Selection** button to select the pallet slots the robot can access for pick or place configurations. This button is only available when a Pallet object is configured for the selected part / target.

● Multiple Pick

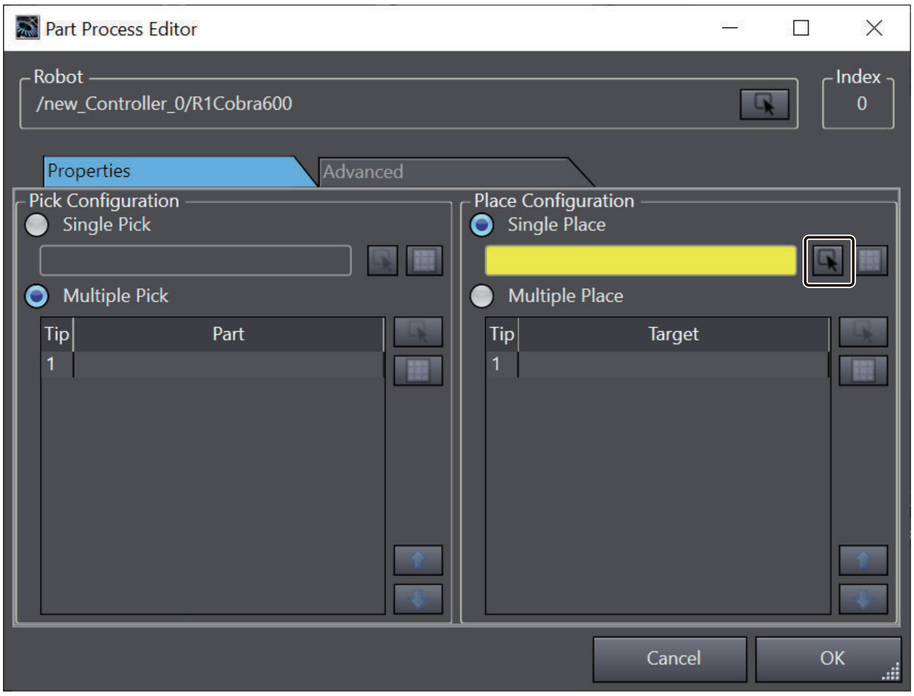
Select this item for a multiple-pick application where multiple parts will be picked before placement at a target. When multi-pick is enabled, the available tip indexes will be provided for the selected robot IO EndEffector. Use the **Select** icon to browse for the part that will be picked by each tip.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

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



● 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.

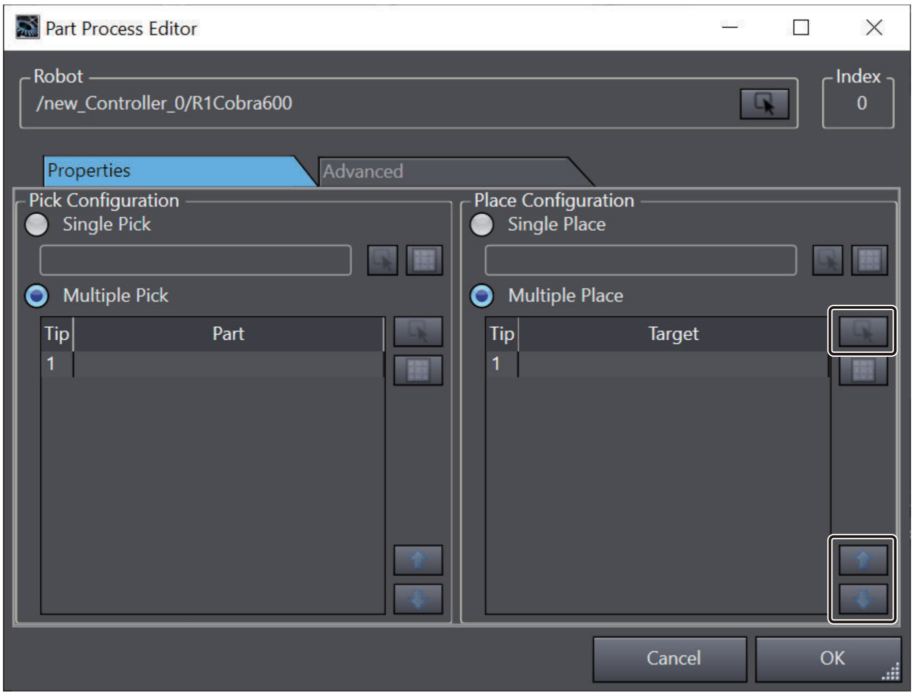


● Multiple Place

Select this item for a multiple-place application where multiple place-motions are performed. Use the **selector** icon to browse for the part that will be picked.

Use the **Up**/**Down** buttons to change the order of the tip processing.

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



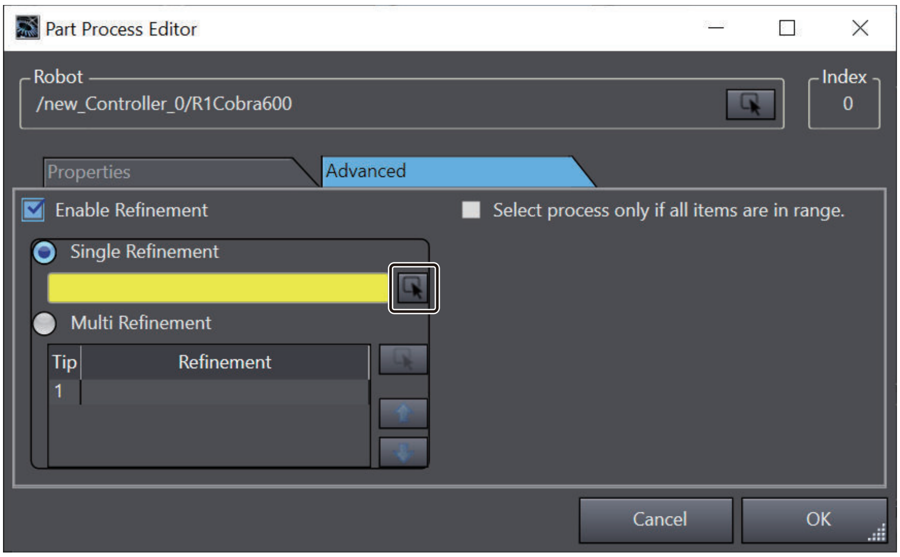
● Single Refinement

Select this item for a single-vision-refinement application under the following conditions.

- Only one part at a time is being moved by the robot.

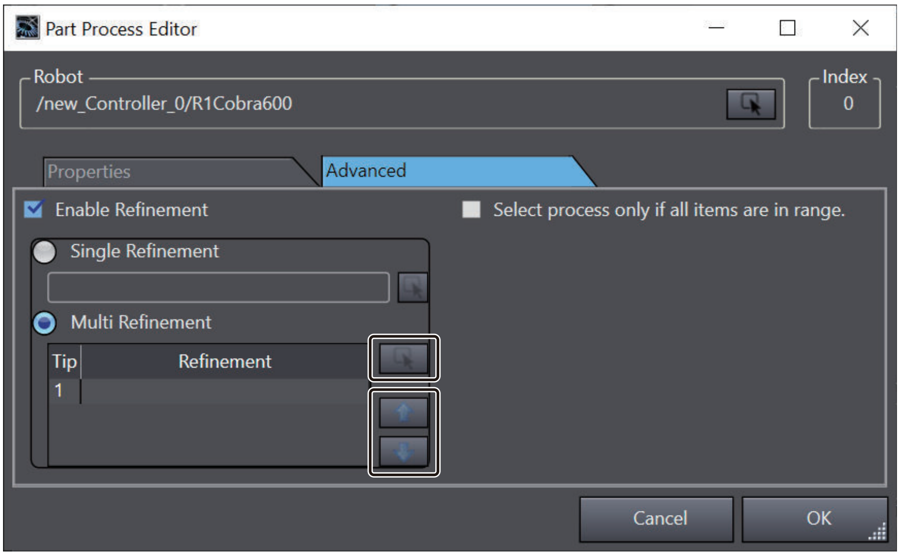
A multi-tip-gripper is moved to the vision refinement station, but only one picture is taken that includes all parts on the gripper. If you want to process an individual refinement for each part, use the **Multi Refinement** option.

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



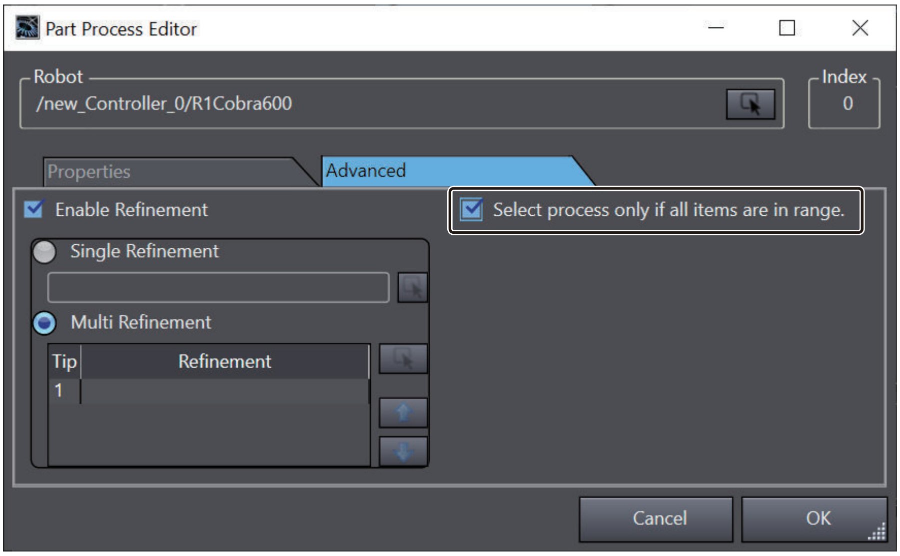
● Multi Refinement

Select this item for a multiple-tip-gripper applications. Each tip will be moved individually to the specified vision refinement station.
Use the **Select** icon to specify the tool tip and corresponding vision refinement.
Use the **UP**/**Down** buttons to change the order of the tip processing.



● Select Process Only if All Items are in Range

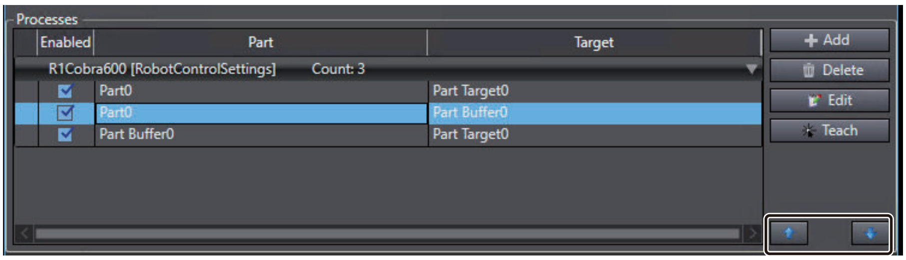
The **Select process only if items are in range** check box tells the system to only select this process if all parts and targets are in range of the robot.
This option is typically disabled, but may be useful in multi-process applications where you want to select this Process only if the required Part(s) 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.
This is typically not necessary, but can be helpful in multi-process configurations. For example, in a situation where Process A has higher priority than Process B, Process A Part is further 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**/**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* on page 5-131 for more information.



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 Part Buffer, place to output Part Target.
3. Pick from input Part, place to output Part Buffer.

In this case, the Process Manager would always execute the first process if there are parts at the input, and move them to the output. If no parts are present at the input, it will then check for process 2 and 3.
However, 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.

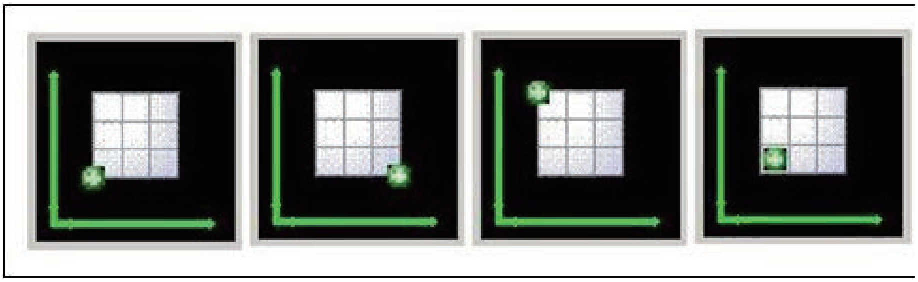
In this case, the Process Manager will always remove parts from the Part Buffer until no parts remain in the Part Buffer, before processing the input (Part).

Teaching a Process

The last step in creating the Process is the teach function. This must be performed after all calibrations are complete. Ideally, all calibrations would be performed as accurately as possible with a calibration pointer and disk. After calibrations are complete, the process end effector can be installed to teach the process. Every step of the process is performed one at a time to capture any necessary motion offsets. Select a **Process** and then click the **Teach** button to open the Teaching Process wizard.

The following process illustrates the steps for a basic pick-and-place application. The steps in your wizard will vary depending on the application and types of Parts and Part Targets you have defined for your application.

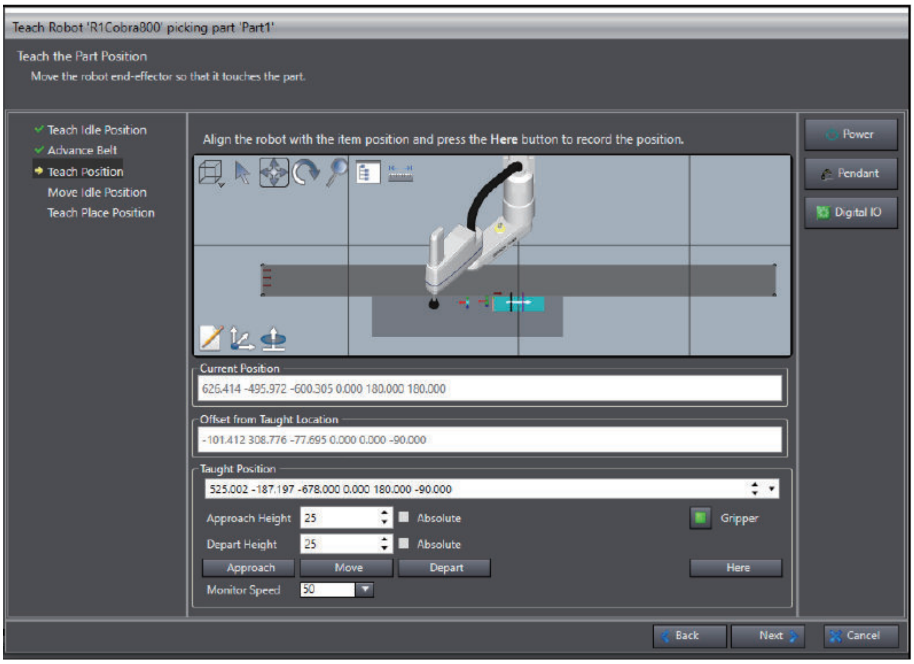
If a Pallet object is used for a Part or Part Target, you will see additional screens in the wizard for teaching the pallet frame (the orientation of the pallet in the workspace) and the first position in the pallet. Each of these steps contain an image of the pallet item being taught, which provides a visual reference. Refer to *5-9-7 Pallet* on page 5-175 for more information.



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

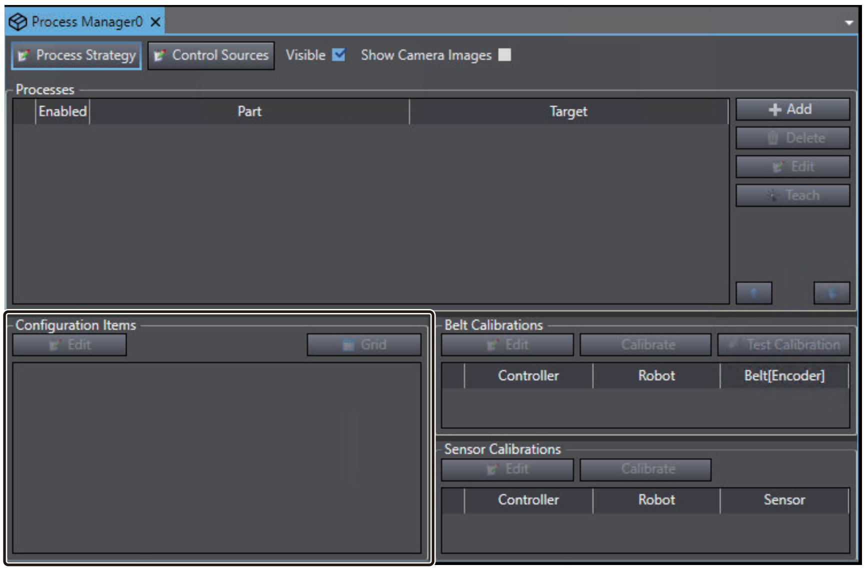
Precautions for Correct Use

When the **Absolute** option is selected, the approach or depart heights will be an absolute Z co-ordinate 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.



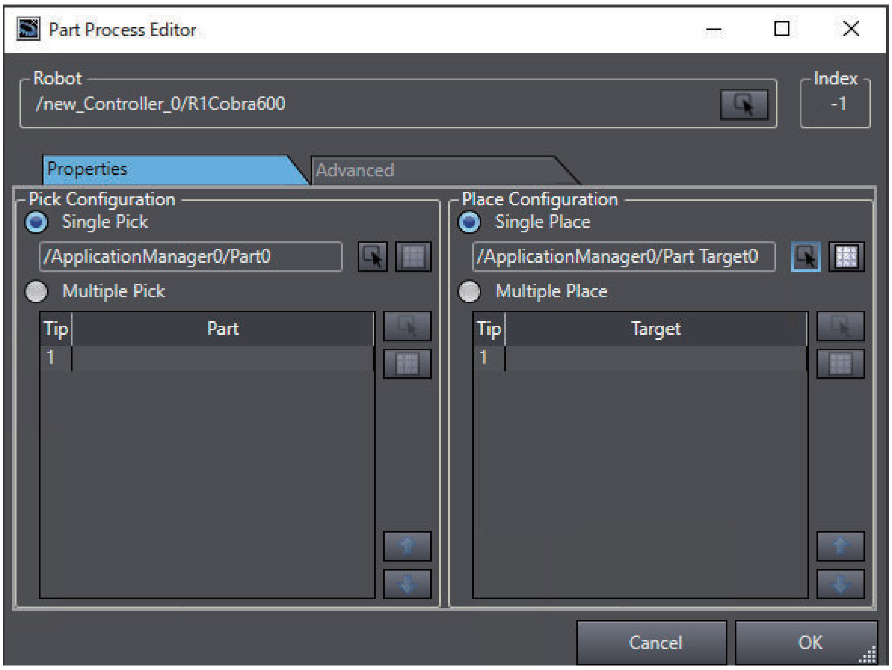
Configuration Items

The **Configuration Items** define the workcell items and the relationships between those items that are associated with a particular workcell configuration. The Configuration Items area also allows quick access to the robot position (location) editors for a particular item, such as the idle, part pick, or part place location.

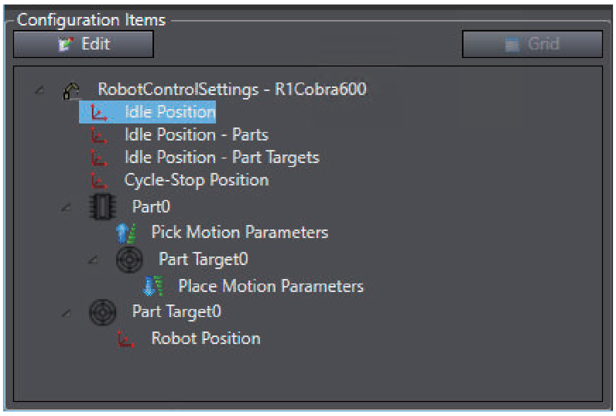


● Creating the Configuration Items

The **Configuration Items** are created automatically as the workcell process is defined through the Part Process editor after you add a new process to the Process list. As items are added/deleted in the Part Process editor, they are added/deleted in the Configuration Items area. For example, a basic pick and place application would look like this in the Part Process editor:



The corresponding Configuration Items area has the following appearance.



● Configuration Item Area Structure and Features

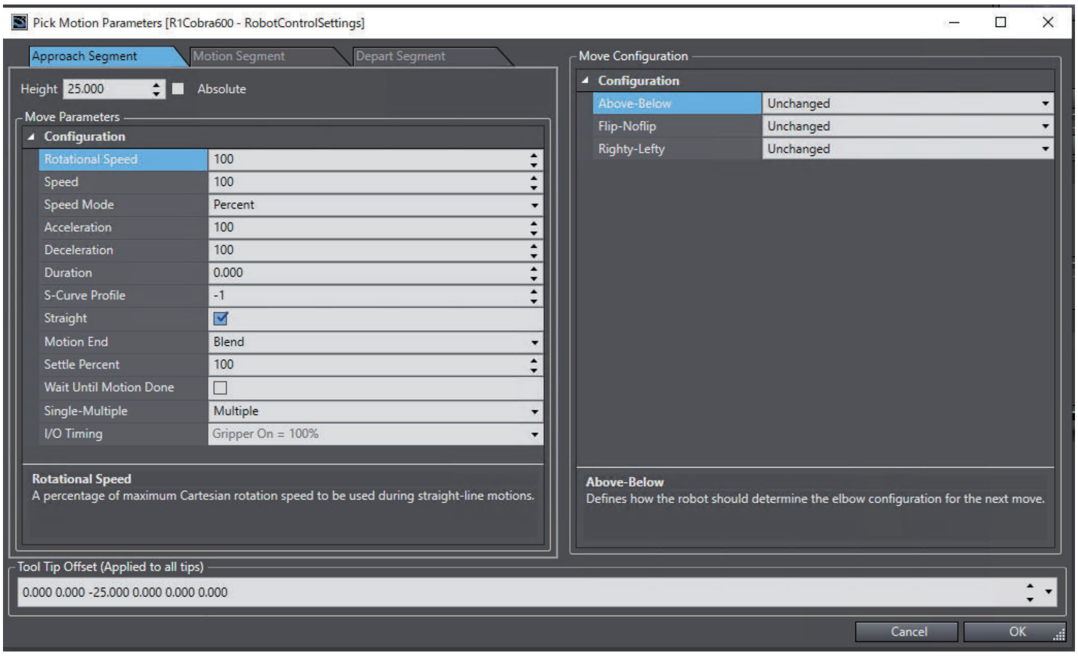
The Configuration Items are arranged in a tree structure to show the relationships between the workcell items. The Configuration Items group contains the following features.

- 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* on page 5-156 for more information.

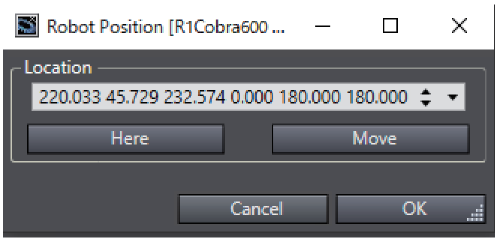
Location Editors

There are two types of location editors. A simple editor allows you to enter location information. And an enhanced position editor contains additional sections such as **Move Parameters**, **Move Configuration**, **Approach/Depart Parameters**, etc.
For example, the Idle Position editor shown in the following figure is an enhanced position editor, which contains additional properties for **Move Parameters** and **Move Configuration**. Refer to *Enhanced Location Editor Parameters* on page 5-150 for more information.

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



The Robot Position editor shown below is a simple position editor. This 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. Use the following examples of various enhanced location editor parameter grids to understand the editor functions.

- Move Configuration Area
  These parameters control the configuration of the robot at the selected location. For example, if your workcell contains a SCARA robot and you want it to be in a **Lefty** configuration, 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. When the **Absolute** option is selected, you must ensure the **approach**/**depart** heights are set correctly. 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. The time value allows you to set the gripper timing (in milliseconds).

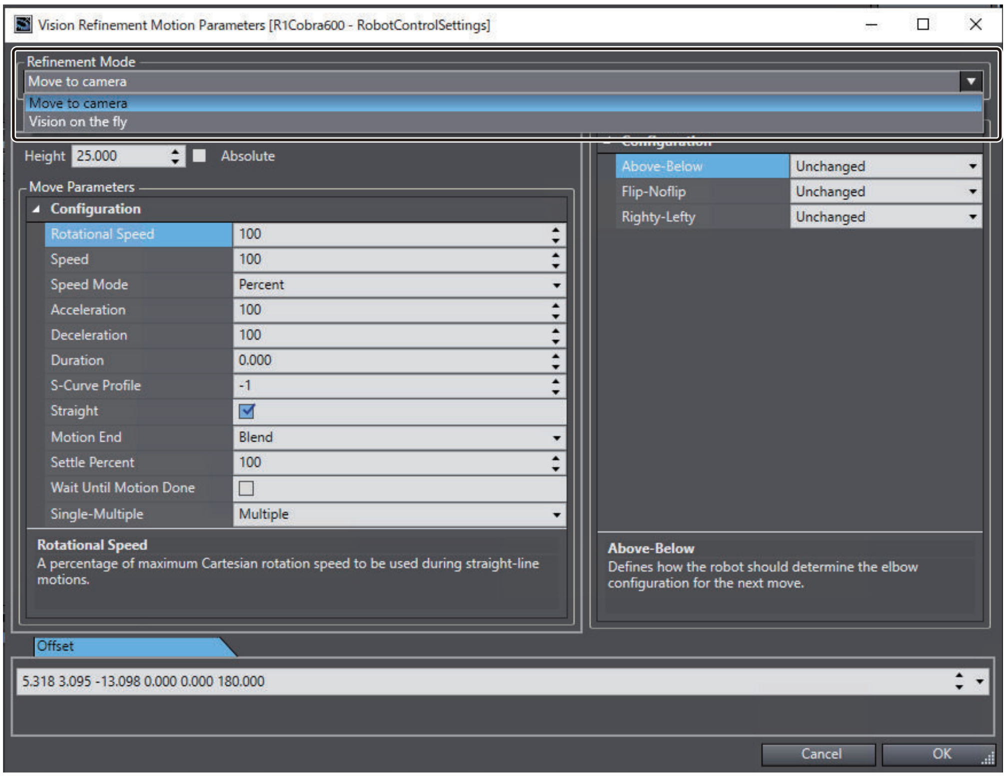
Additional Information

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.

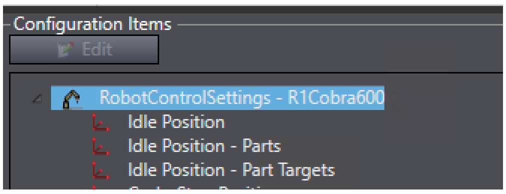
- Vision on the fly
  This is a method of measuring while the robot is operating. This is an in-motion refinement, where the robot passes through the Vision Refinement Station without any pause in the robot motion.
  The **Start Location** and **End Location** tabs allow you to edit the start and end 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 *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.



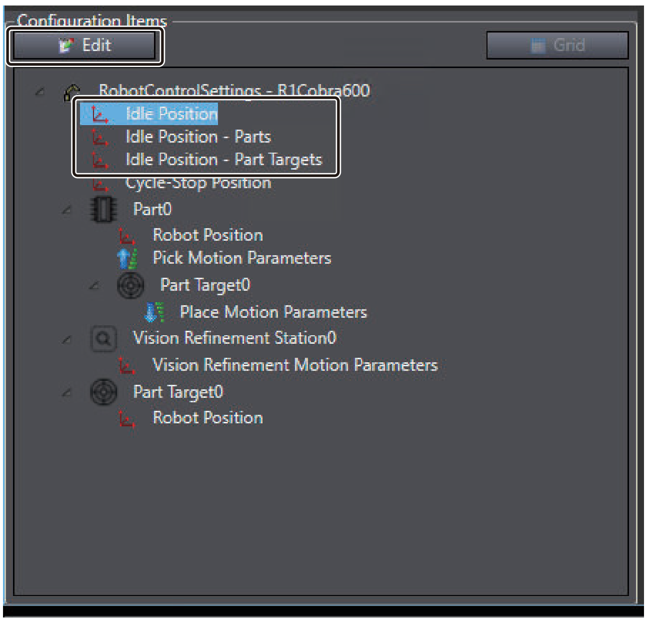
The Wait Mode setting can be found in the Process Strategy area. Refer to *Process Strategy* on page 5-130 for more information.

- Idle Position
  This is the location the robot uses when no Process is selected or when the Process Manager is aborted or stopped.

- Idle Position - Parts
  This is the location the robot uses when it is waiting to pick a part, typically when no parts are available.
  The Idle Position - Parts location is not associated with a specific process. If 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
  This is the location the robot uses when it has picked a part and is waiting for a target to become available.
  The Idle Position - Part Targets location is not associated with a specific process. If 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.



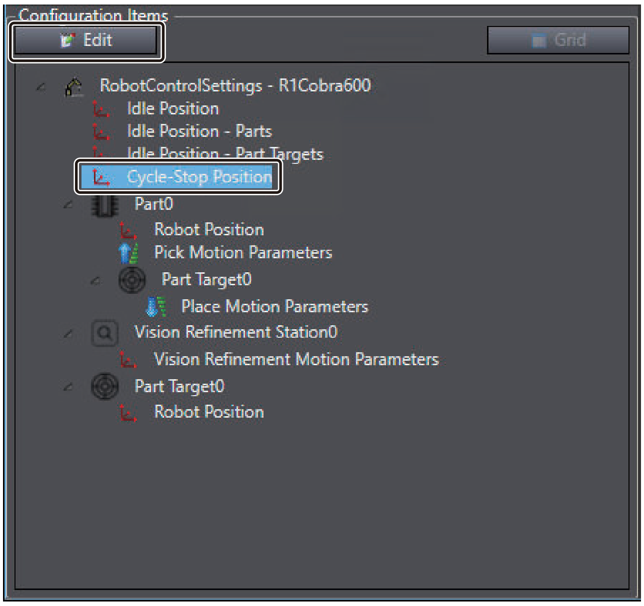
Cycle-Stop Position

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 the **Cycle-Stop** Configuration Item or select the item and click the **Edit** button. The Cycle-Stop Position editor will open.

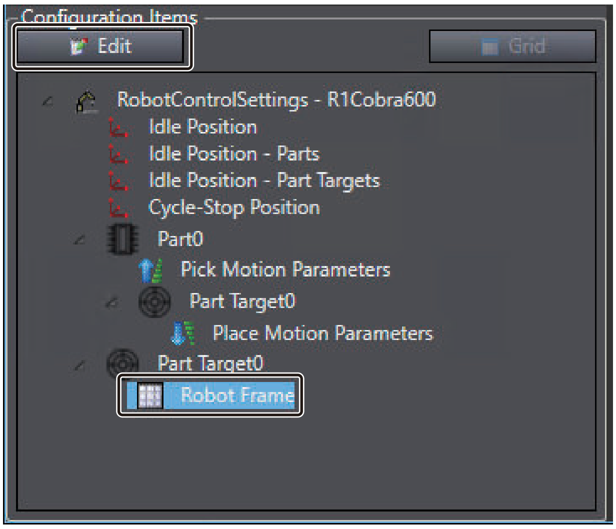


Robot Frames

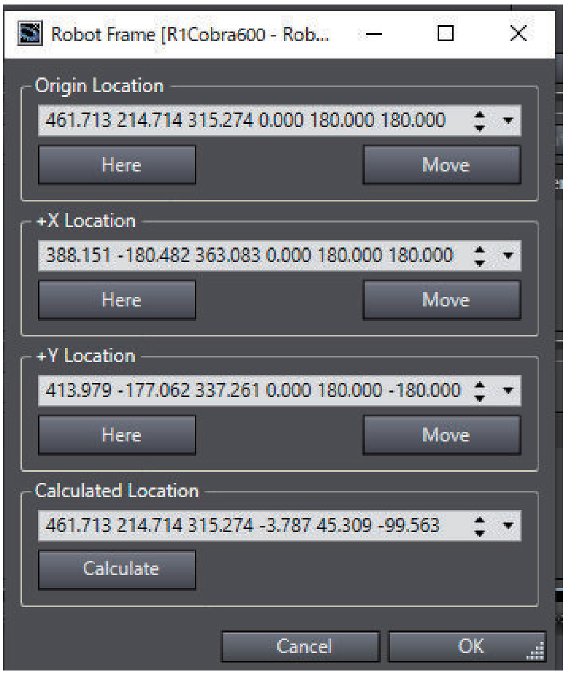
Robot frames (also known as reference frames) are useful because they allow you to teach locations relative to the frame. If the location of the frame changes in the workcell, you simply have to update the frame information to reflect its new location. Then you can use any locations created relative 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 reteach all of those positions if the pallet moves for some reason. Instead, the pallet is taught relative to a frame. If the pallet moves in the workcell, the frame position is re-taught and the part positions relative to that frame remain intact.

The Robot Frame editor is used to teach a reference frame, such as a pallet frame. To access the **Robot Frame** editor, double-click the Robot Frame Configuration Item or select the item and click the **Edit** button.



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

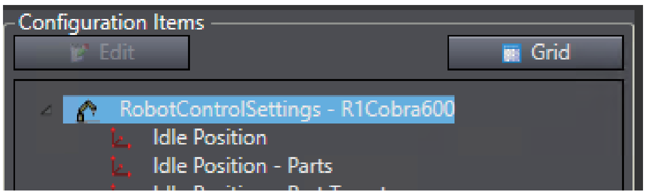
The Motion Sequence Grid Editor provides a grid / table interface that allows you to access and individually edit the common motion parameters used to optimize cycle time. Also, you can change multiple speed / acceleration / deceleration parameters at the same time by clicking on the box and dragging the cursor before you enter a value.

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

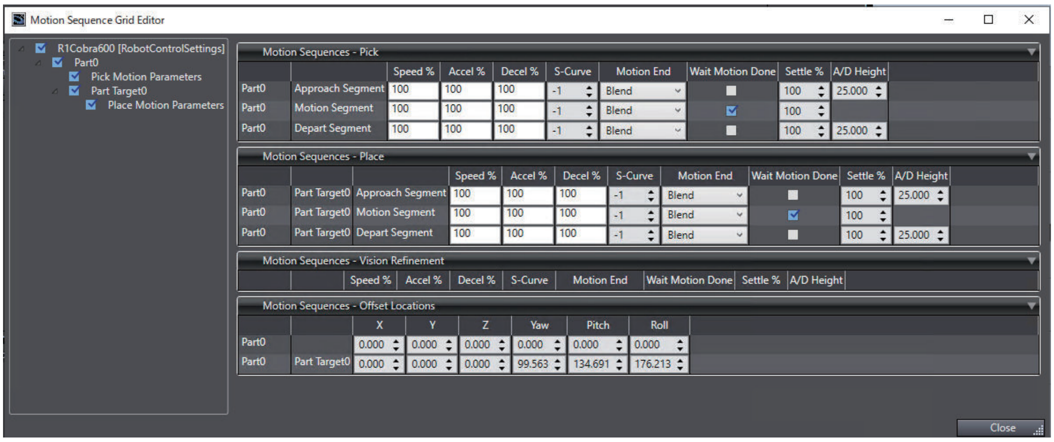
Precautions for Correct Use

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.



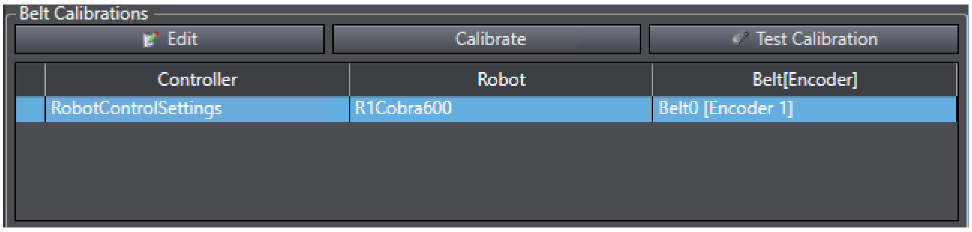
The Motion Sequence Grid Editor opens.



The left pane is used to select the items you wish 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* on page 5-139 for more information.



Item	Description	Reference
Edit Button	Click to edit the selected belt calibration.	*Editing the Belt Calibration Parameters* on page 5-158
Calibrate Button	Click to begin the belt calibration procedure.	*Belt Calibration Wizard* on page 5-158
Test Calibration Button	Click to test the current belt calibration. This button is not available until the belt has been calibrated.	*Testing the Belt Calibration* on page 5-160
Robot	Specifies the robot for the belt calibration. Double-click this item or click the **Edit** button to display the Belt Calibration Editor.	*Editing the Belt Calibration Parameters* on page 5-158
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.	*Editing the Belt Calibration Parameters* on page 5-158

● 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. Access this wizard 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.

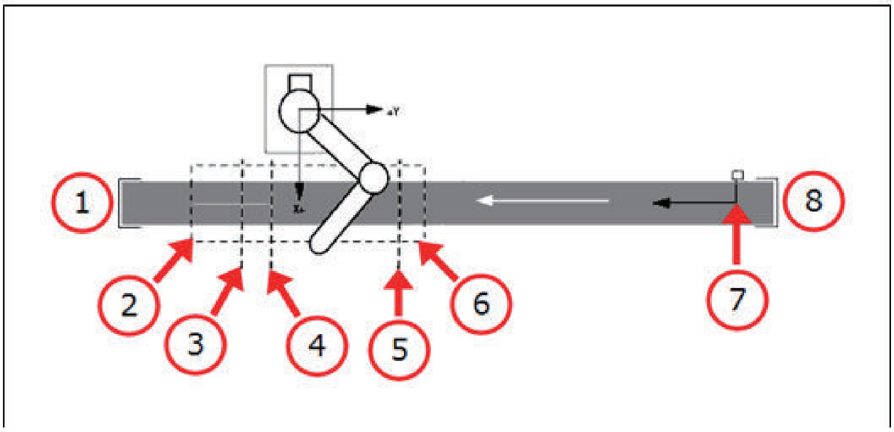
● Belt Calibration Wizard

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 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

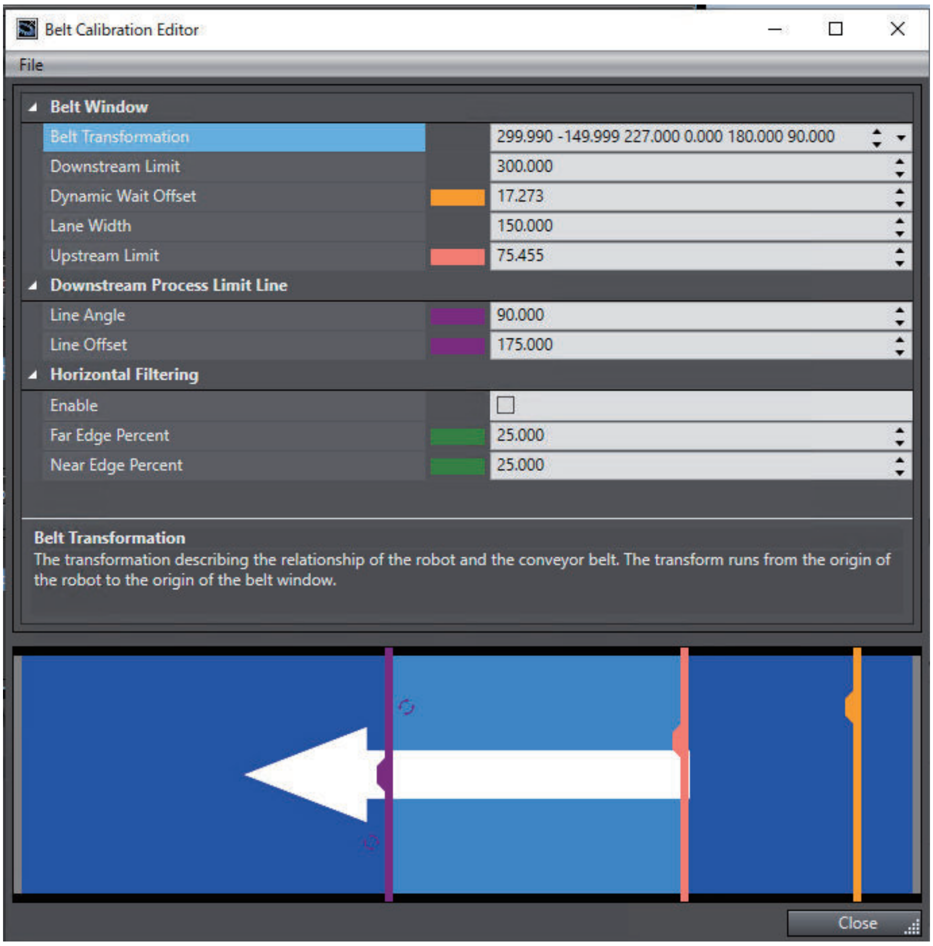
After the belt has been calibrated using the Belt Calibration wizard, you can manually edit the stored belt calibration parameters and allocation limits, such as upstream limit, downstream limit, and downstream process limit. These parameters are edited using the Belt Calibration Editor. The following 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.

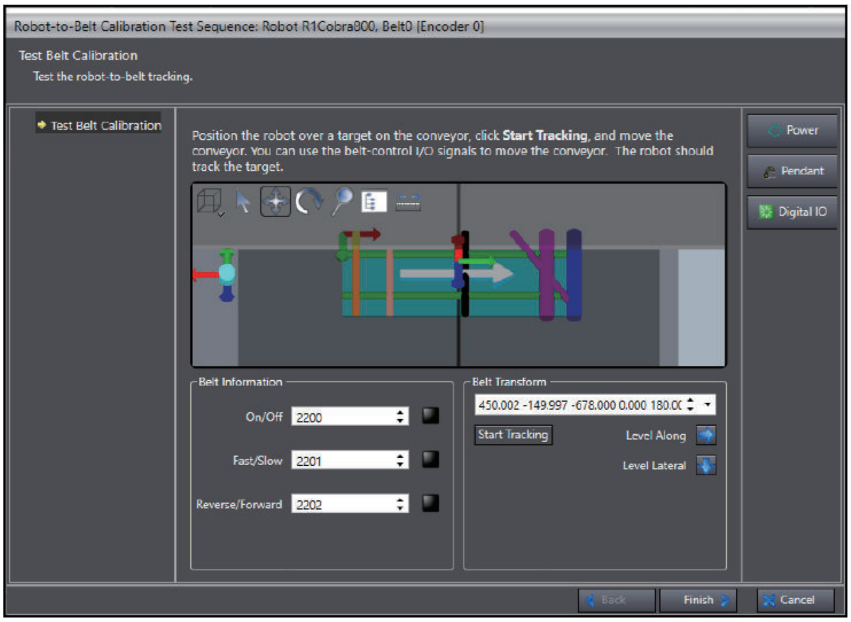


Group	Item	Description
Belt Window	Belt Transformation	The transformation describing the location of the belt window relative to the origin of the robot. This location also defines the upstream tracking limit of the belt window. The X-axis of the belt transformation is often referred to as the belt vector.
	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. This may be upstream or downstream of the Upstream Limit depending on the motion parameters and path of travel to a target.
	Lane Width	The width of the belt window starting from the belt transform, pointing in the positive Y-direction.
	Upstream Limit	Upstream pick limit (mm from belt transformation along the belt vector).
Downstream Process Limit Line	Line Angle	The angle of the downstream process limit line.

Group	Item	Description
	Line Offset	Downstream process limit. If a robot has initiated a move to an instance that is upstream or at this limit, it will proceed with that motion unless a belt window violation occurs at the downstream limit of the belt window. However, if an allocated instance crosses the downstream process limit line before a robot initiates a motion to it, then it will be unallocated from this robot queue. The value must be between the upstream limit and the downstream limit. A good initial value can be calculated by subtracting the distance the belt will travel during a pick or place operation from the Downstream limit (length of belt window), and then subtract an additional 15-25mm. If the process limit is any closer to the downstream limit, 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 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.

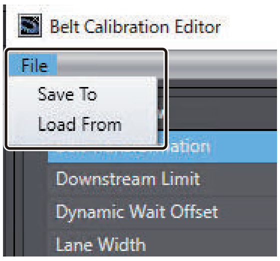
7 Click the **Next** button to proceed. The Robot-to-Belt Calibration wizard closes and the procedure is completed.

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.

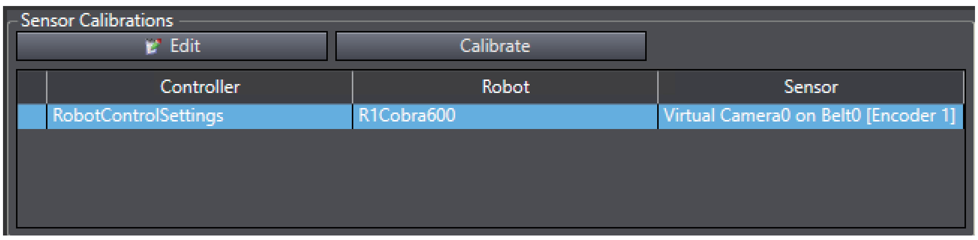
● Saving and Loading a Belt Calibration

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* on page 5-139 for more information. 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 **Edit** button to display the Sensor Calibration Editor.	*Editing the Sensor Calibration Parameters* on page 5-162
Sensor	The sensor specified for the process. Double-click this item or click the **Edit** button to display the Sensor Calibration Editor. A sensor can be any of the following, depending on the type of con- figuration selected for the Part or Part Target. Belt camera Fixed camera (downward-looking camera) Latch sensor Spacing reference Refinement camera (upward-looking camera)	
Edit Button	Click to edit the selected belt calibration.	
Calibrate 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 with 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 Sensor Calibration Parameters

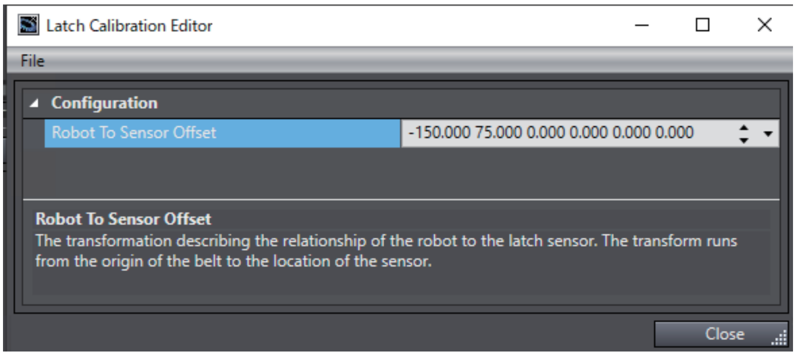
After the sensor is calibrated through the Sensor Calibration wizard, you can manually edit the stored sensor-calibration parameters, such as the robot-to-sensor offset. These parameters are edited using the Sensor Calibration Editor. To access the sensor Calibration Editor, select a sensor and then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

Precautions for Correct Use

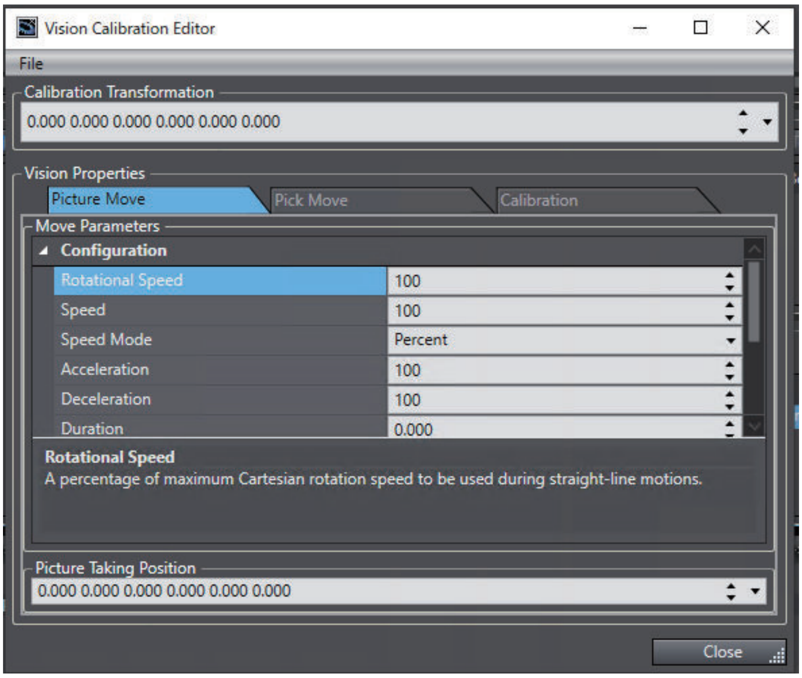
The sensor must be calibrated, or loaded from a previously-saved calibration data file, before the values can be manually edited.

The Sensor Calibration Editor contains the sensor properties configuration parameters. These are used to configure various settings of the selected sensor.

The following figure shows the Latch Calibration Editor, which has one property for controlling the calibration offset.



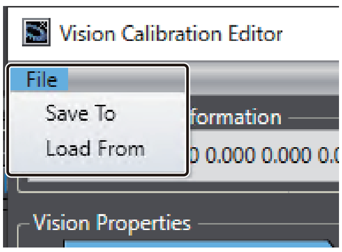
The following figure shows the Vision Calibration Editor, which contains a calibration offset along with additional parameters for controlling the robot motion during the picture-taking and part-pick operations of the automated hardware calibration. These are not used during run time when the robot is performing the process (run time motion parameters will be found in configuration items).



● Saving and Loading a Sensor Calibration

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

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



Calibration Wizard - Automatic Versus Manual Calibrations

The calibrations can be performed using either the automatic calibration (preferred method) or the manual calibration procedure. In the automatic calibration procedure, 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 *3-3-3 Robot-to-Camera Calibration* on page 3-10 for a more detailed explanation of 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 *3-3-3 Robot-to-Camera Calibration* on page 3-10 for a more detailed explanation of 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 *3-3-4 Robot-to-Latch Calibration* on page 3-11 for a more detailed explanation of the calibration process.

Type	Description
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 Calibration	The Spacing Reference Calibration wizard configures the positioning of parts along a belt at defined intervals. This process is performed by setting a stationary point along the belt from which the instances will be generated. This point should be calibrated outside of the belt window to avoid difficulties in allocation. If creating spacing instances for multiple robots, the spacing calibrations must reference the same upstream position.

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 an ACE 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 "Peach Packing." After all Process Manager objects are added, the procedure is complete.

Process Manager Process Components

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 a **Configuration** drop-down list box that is used to specify how the part is input to the system.
A Part Target object defines a possible destination for a part. The possible configurations for a Part Target object are the same as for a Part object. Depending on the selected configuration, additional information can be defined when configuring the Part/Part Target as described below.

Part/Part Target Configuration	Vision Properties	Belt Properties	Pallet
Belt	Optional	Required	Optional
Static	Not used	Not used	Optional
Vision	Required	Not used	Optional

● Part and Part Target Configuration Options

The Part and Part Target configuration defines the part pick or place requirements. The following options are available.

Part and Part Target Configuration	Description	Details
Belt	The part is picked from or placed onto a conveyor belt. It may use latching, a camera, or a spacing interval to determine the position. A pallet is optional.	A belt and encoder must be specified for use with the Part/Part Target. Then, a Belt Mode is defined that describes how the part is related to the belt. For this item, additional information is required based on the options below. Belt Mode: Vision Vision properties are required. Spacing is not used. Belt Mode: Latch The latch information is taken from the Belt object linked to the Belt Properties. Vision properties are not used. Spacing is not used. Belt Mode: Spacing Vision properties are not used. Spacing is required.
Static	The part is picked from or placed to a fixed location. Because it is a fixed location, no camera or belt is used. A pallet is optional.	-
Vision	The part pick or place process requires a fixed-location camera. There is no belt used. A pallet is optional.	A vision tool is specified that is used to locate the part. For example, this could be an inspection tool that filters instances based on some criteria. Additionally, the vision properties can be configured to filter the vision results based on a part name. This will most likely be associated with a named part returned from a locator model.

● 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 Cam- era 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 man- age 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.

● Robot Controller Queue

The Robot 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.
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).

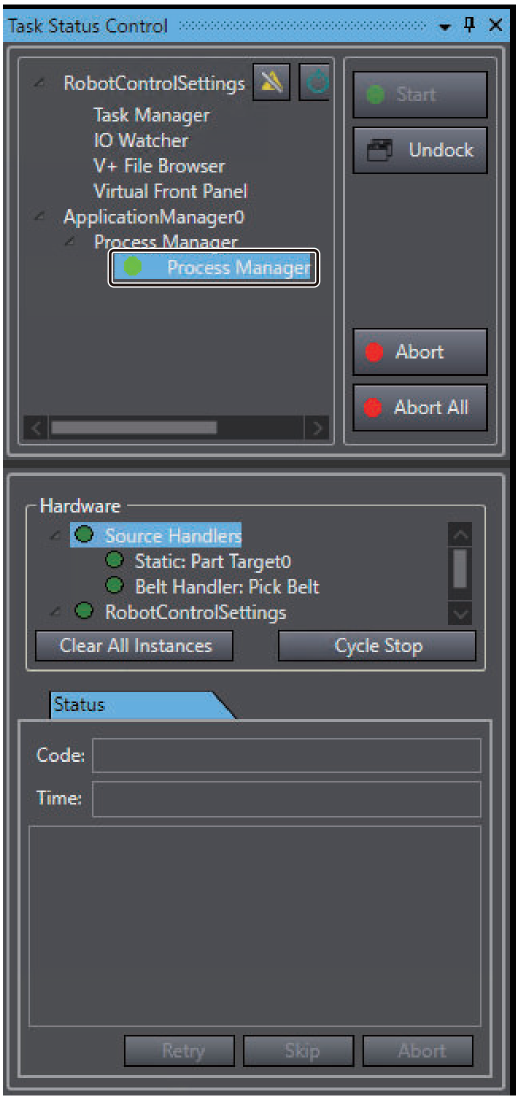
Process Manager Control

The Process Manager you create is executed as a task in the Task Status Control interface.
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* on page 7-30 for details.

● Using the Process Manager Control

The Task Status Control interface is used to monitor and control Process Manager objects in a 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.

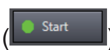
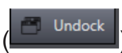
Process Manager control items are added to the Task Status Control area as shown in the following figure. Select a Process Manager control item to view the Hardware and Application information areas. Refer to *7-5-9 Application Manager Control* on page 7-41 for more information.



● Process Manager Tasks

Process Manager tasks are displayed under the Application Manager group in a tree view. Tasks are grouped by type (C# program, Process Manager, etc.). When 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 ()	Executes the selected task.
Undock ()	Undocks the selected task's hardware and application information area. Close the undocked window to restore the view.

Function	Description
Abort ( )	Stops execution of the selected task.
Abort All ( )	Stops execution for all tasks.

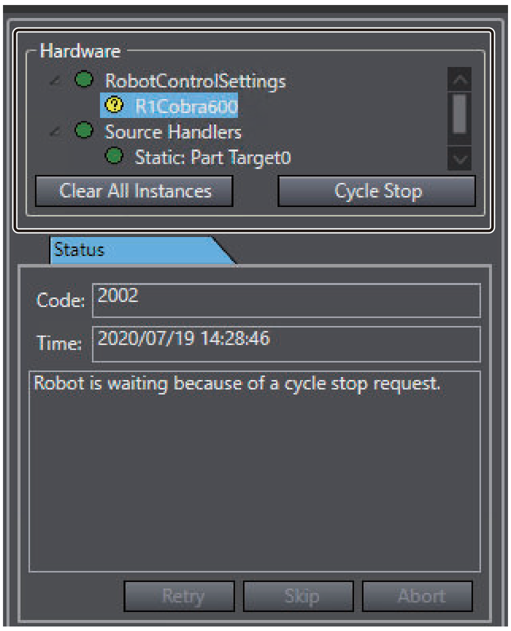
Tasks are marked with icons to indicate operational status.

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

● Hardware Information Area

The Hardware information area displays the hardware items and their status for the selected Process Manager task. Use the information below to understand the functions of the hardware information area.
When a robot is waiting (for example, waiting for Parts or Part Targets to arrive or because of a cycle stop request), a yellow warning condition is displayed on the Process Manager control. Selecting the item in the Hardware Information area will display additional information in the status and instance tabs below.

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

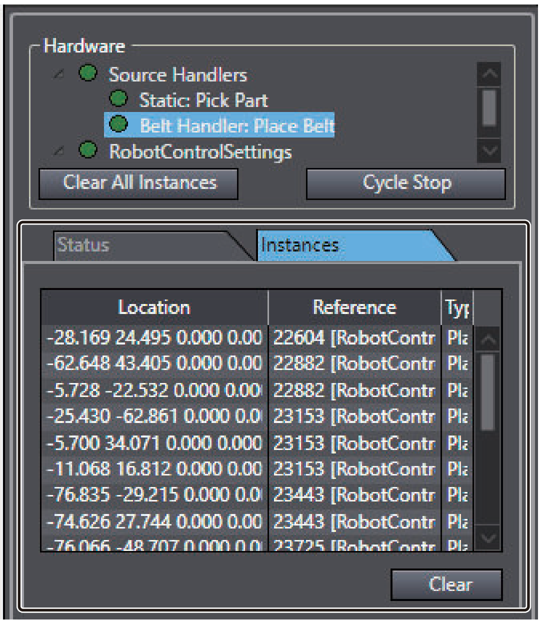


Item	Icon/Button	Description
Idle State Icon		The item is idle.

Item	Icon/Button	Description
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 **Status** tab. Refer to *Status Tab* on page 5-174 for more information.
Error State Icon		An error condition is present for the item. The specific warning message is displayed on the **Status** tab. Refer to *Status Tab* on page 5-174 for more information.
Clear All Instances Button		Clears all Part and Part Target instances from the system.
Cycle Stop Button		Sends a signal to each robot to stop after the current process cycle has completed. Each robot stops after it reaches the end of its current process, and then "*Robot is waiting because of a cycle stop request.*" is displayed in the Status tab. You can resume the robot(s) and process operations by clicking the **Cycle Stop** button. The **Cycle Stop** button provides the capability to implement a variety of situations. For example, you could click the **Cycle Stop** button and leave the system running. When the system is in this state, all tracking is enabled. Therefore, you could either click the **Abort** button which stops everything or you could simply resume the current cycle by clicking the **Cycle Stop** button again.

● Application Information Area

The Application information area displays feedback on the operation of the item selected in the Hardware area. The Application information area has a **Status** tab and an **Instances** tab which are described below.



- **Status Tab**

This tab displays information on the status of the *control components* driving the process. It shows the hardware in the system and the status of the selected item.
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 *ACE Reference Guide: Custom Error Programs* for more information.

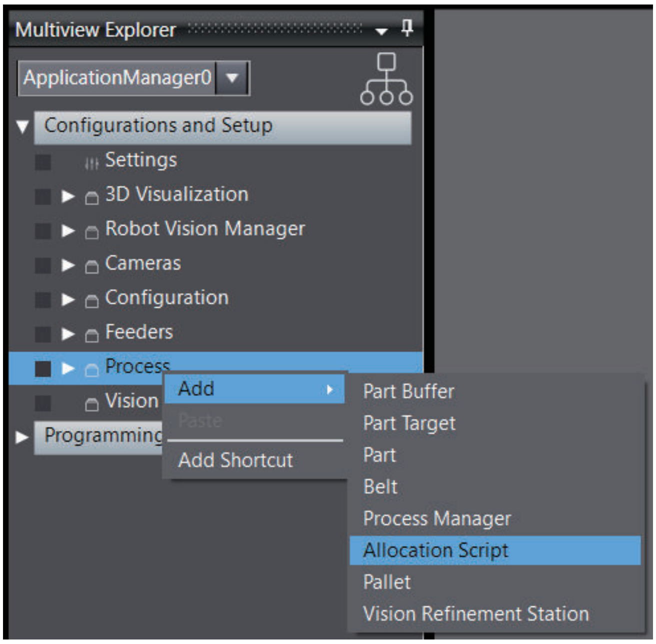
- **Instances Tab**

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

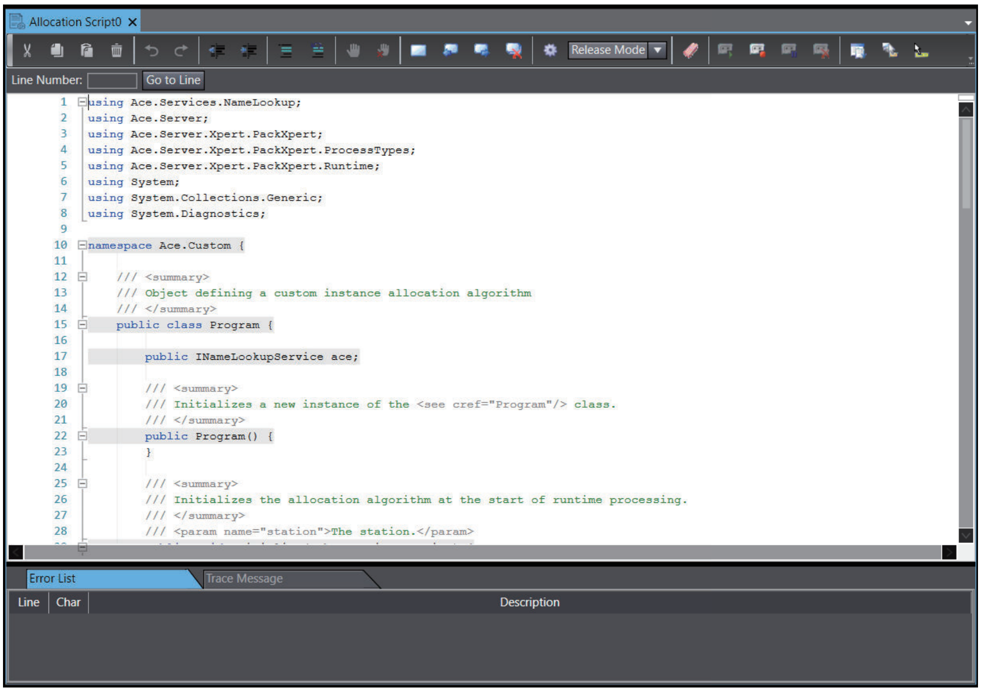
You can create Allocation Script objects that include a simple C# program editor. Use the Allocation Script objects to edit the Process Strategy in the Process Manager.
The Allocation Script object is used to create and edit custom part allocation programs for use with the Process Manager.
The Allocation 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** in **Configurations and Setup** under the **Multiview Explorer**. Select **Add**, and then click **Allocation Script**.



A new **Allocation Script** object will be added to the **Process**.

2 Right-click **Allocation Script** and then select **Edit**.
The Allocation Script editor is displayed.



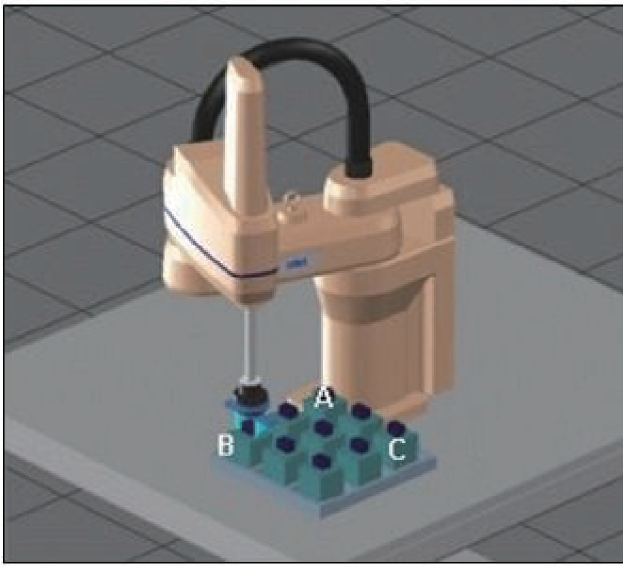
You can edit an Allocation Script with the C# program editor. Refer to *5-11-1 C# Program Editor* on page 5-192 for details the about the C# Program editor.
See *ACE Rererence Guide* for a sample Allocation Script.

5-9-7 Pallet

You can create Pallet objects that can be used to pick parts from or place parts to. Define a pallet layout and position.

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 can 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

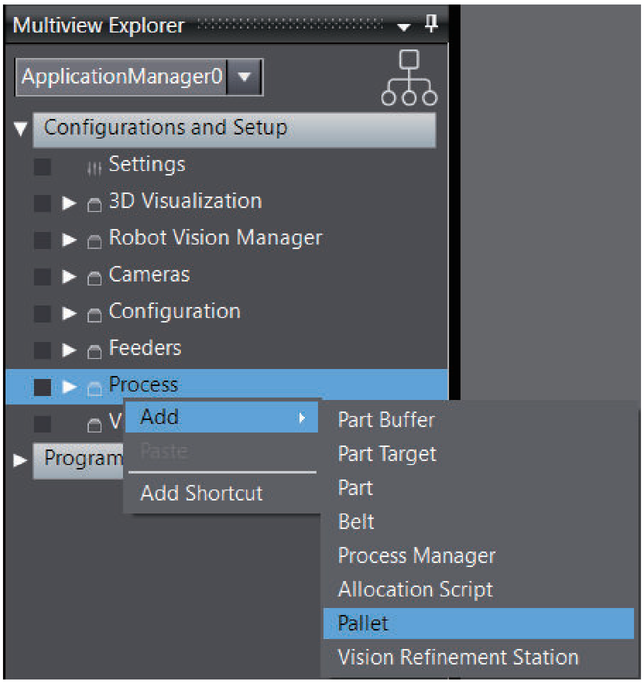
When teaching the pallet using the Sysmac Studio wizard, the system automatically computes the orientation and origin offset of the pallet. Then, the system has all of the information it needs to pick or place parts from or to positions in the pallet.

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* on page 5-127 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

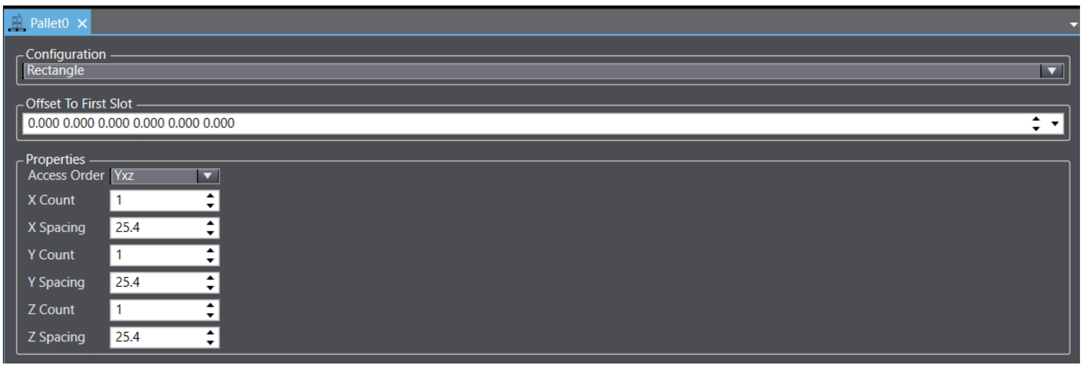
Add a pallet following the procedure described below.

1 Right-click **Process** in **Configurations and Setup** under the **Multiview Explorer**. Select **Add**, and then click **Pallet**.



A new **Pallet** object will be added to the **Process**.

2 Right-click **Pallet** and select **Edit**.
The Edit Pane is displayed.

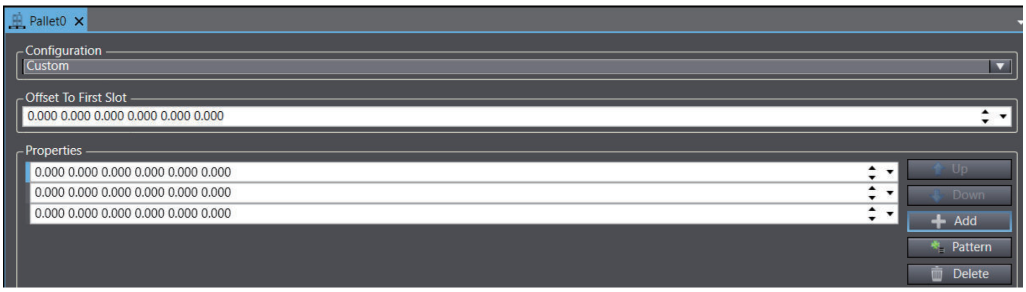


Items on the Edit Pane are as follows.

Item		Description
Configuration		The drop-down list box is used to specify the type of pallet being used. The following types are available. **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		The Offset to First Slot setting defines the offset from the pallet corner to the first slot position of the pallet.
Properties	Access Order	The Access Order property defines how the robot will access the pallet.
	X Count	This property defines the number of slots on the X-axis.
	X Spacing	This property defines the slot spacing for the X-axis. The unit is mm.
	Y Count	This property defines the number of slots on the Y-axis.
	Y Spacing	This property defines the slot spacing for the Y-axis. The unit is mm.
	Z Count	This property defines the number of slots on the Z-axis.
	Z Spacing	This property defines the slot spacing for the Z-axis. The unit is mm.
Properties (for Custom Pallet)		This property defines any slot position. Refer to *Custom Pallet Configuration Settings* on page 5-178 for details.

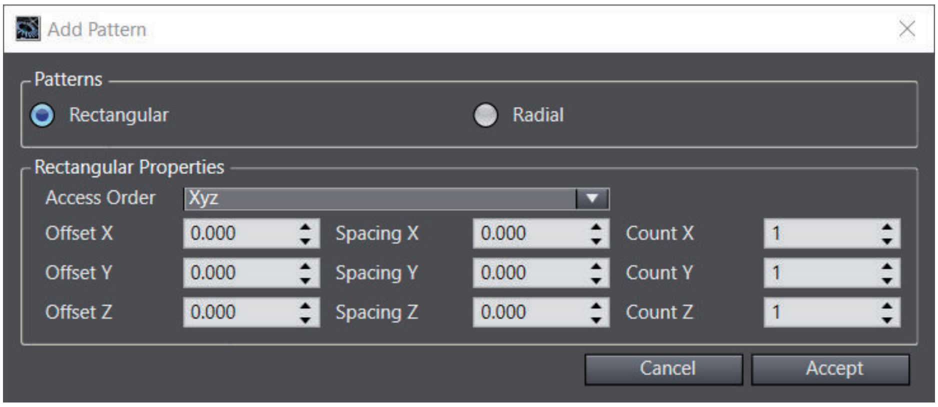
● Custom Pallet Configuration Settings

A custom Pallet is typically used for irregular slot arrangements. The custom Pallet con- figuration 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 position.

● Adding a Custom Pattern



Click the **Pattern** button to define the custom Pallet using the **Add Pattern** dialog box and then select **Rectangular** or **Radial**.

Rectangular Properties: Set the X,Y and Z Offsets, Spacings, and Counts 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* on page 5-113 and *5-9-2 Part Target* on page 5-107 for details.

5-9-8 Vision Refinement Station

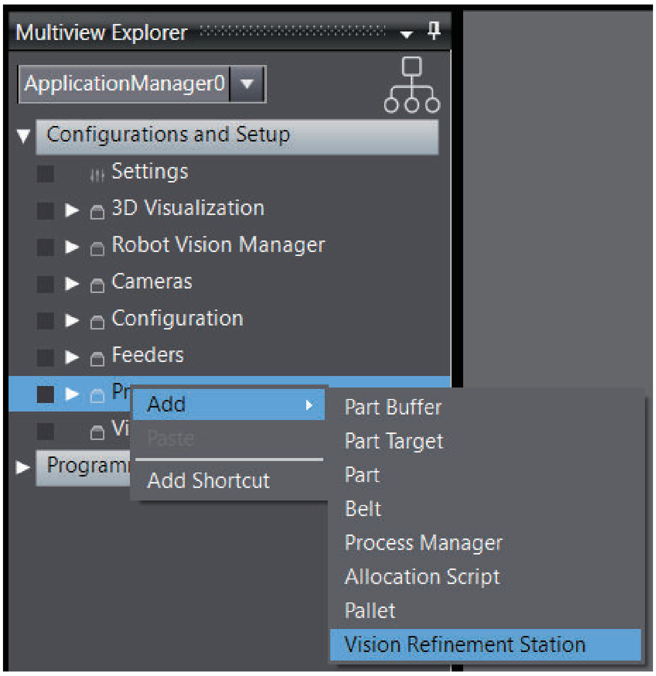
The Vision Refinement Station object defines a location with an upward-mounted camera, and determines the part-to-gripper orientation for more accurate placement of the part.

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.

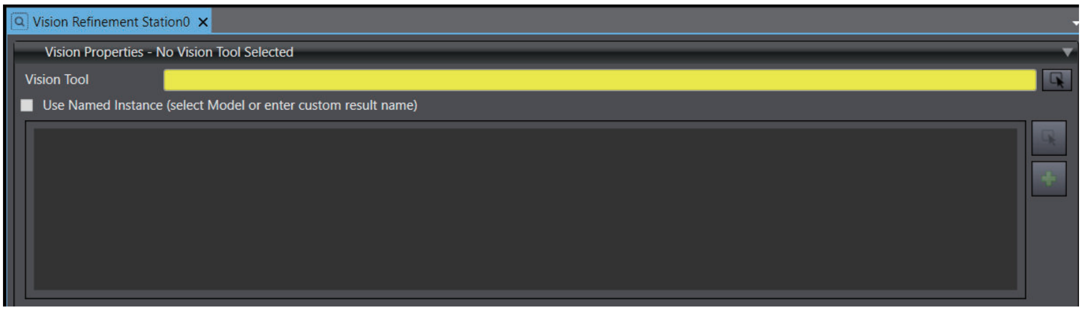
Adding a Vision Refinement Station

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



A new **Vision Refinement Station** object will be added to the **Process**.

2 Right-click **Vision Refinement Station** and select **Edit**.
The editor window will be displayed.



Items on the editor window are as follows.

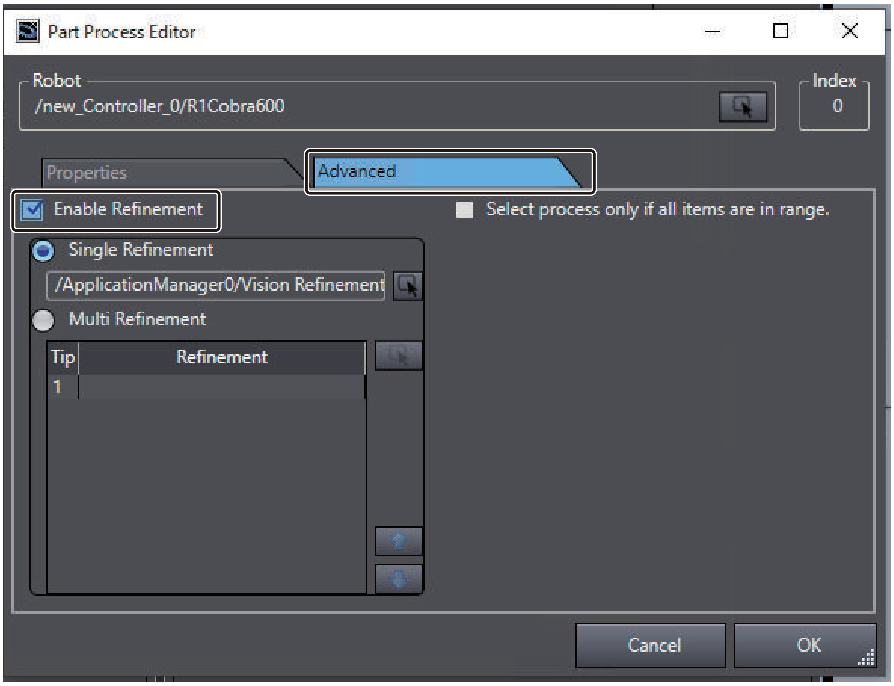
Item		Description
Vision Proper-ties	Vision Tool	This drop-down specifies the vision tool that will be used to detect a part position.
	Use Named In-stance	Checking this box displays detection results with name . Generally a model name.

● 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 exiting 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* on page 5-140 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* on page 5-151 for more information.

5-10 Vision Tools

Adds tools for image processing use in the Robot Vision Manager.
Vision Tools allow you to modify images captured through a camera, Emulation Camera, or Visualizer Capture Device for inspections.
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.

Tools	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 Sysmac Studio, their respective categories, and a brief summary of their functions. Refer to the *Automation Control Environment (ACE) Version 4 User's Manual (I633)* for details of each tool's functions and settings items.

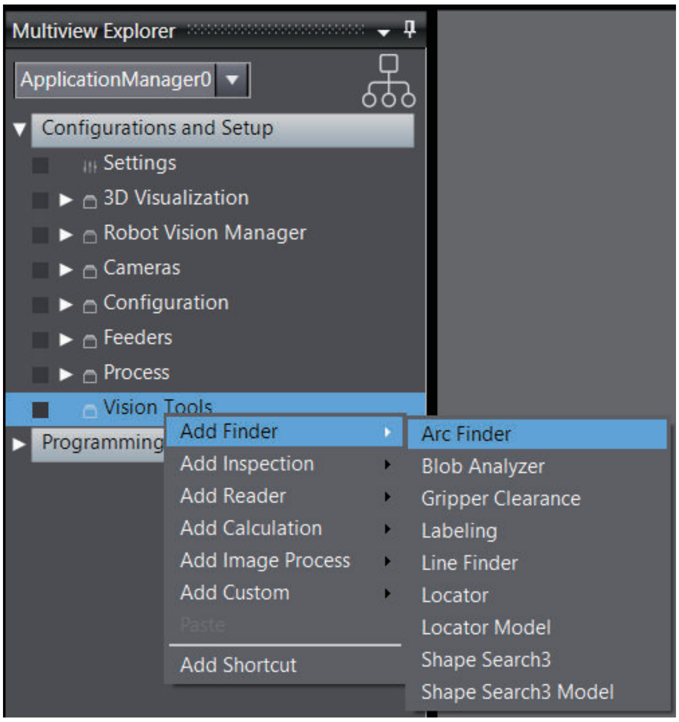
Category	Tools	Description
Finder Tools	Arc Finder	Identifies circular features on objects and returns the coordinates of the arc center, the angle of separation between the two ends, and the radius.
	Blob Analyzer	Processes information within a region of interest to identify groups of similarly-colored pixels, called blobs, and returns the position and size of each.
	Gripper Clearance	Uses histograms to identify regions of clearance around detected objects.

Category	Tools	Description
	Labeling	Searches the image for areas of a user-defined color or color range and returns the results. Multiple colors or ranges can be searched.
	Line Finder	Identifies linear features on and returns the endpoint coordinates and line angle.
	Locator	Identifies instances of objects defined by a Locator Model within a region of interest. This tool is customizable and accurate, but slower than Shape Search3.
	Locator Model	Defines the geometry of a model used by a Locator tool to identify instances.
	Shape Search3	Identifies instances of objects defined by a Shape Search3 Model within regions of interest. This tool is fast, but cannot be customized as much as the Locator tool.
	Shape Search3 Model	Defines the geometry of the model used by a Shape Search3 tool to identify instances.
Inspection Tools	Arc Caliper	Identifies one or more edge pairs in an arc-shaped or circular area and measures distance between the edges of each pair.
	Arc Edge Locator	Identifies an edge or set of edges that meet user-defined criteria within an arc-shaped or circular area.
	Caliper	Identifies one or more parallel edge pairs and measures distance between the edges of each pair.
	Color Data	Finds the average color within a region and analyzes its color variation and deviation from a specified reference color.
	Edge Locator	Identifies an edge or set of edges that meet user-defined criteria within a rectangular region.
	Feeder Histogram	Calculates product density in specified regions of interest. This tool is often used with the AnyFeeder object in regions associated with the dispense, pick, and front zones.
	Image Histogram	Computes gray-level statistics within a specified region of interest.
	Inspection	Judges the results of a tool and determines if they fall within a series of user-defined categories and filters.
	Precise Defect Path	Performs differential processing on an image to detect defects and contamination on a line or along an edge.
	Precise Defect Region	Performs differential processing on an image to detect defects and contamination within an area.
Reader Tools	Barcode	Reads barcodes in an image and acquires text string data.
	Data Matrix	Reads data matrices in an image and acquires text string data.
	OCR	Identifies text characters in image and returns them as text data. Custom characters can also be registered to an OCR Dictionary for future identification.
	OCR Dictionary	Stores dictionary data that OCR can use to identify text characters.
	QR Code	Reads QR or Micro QR code in an image and acquires text string data.
Calculation Tools	Calculated Arc	Calculates an arc based on a specified calculation mode and returns the encompassing circle.
	Calculated Frame	Calculates a frame based on a specified calculation mode and returns coordinates and orientation.

Category	Tools	Description
	Calculated Line	Calculates a line based on a specified calculation mode and returns coordinates and dimensions.
	Calculated Point	Calculates a point based on a specified calculation mode and returns coordinates.
Image Process Tools	Advanced Filter	Filters or alters an image using one of a variety of filter libraries, such as Background Suppression, Erosion/Dilation, and Color Gray Filter. This tool can be combined with other tools (including other Advanced Filters) to modify an image to the necessary specifications.
	Color Matching	Filters and analyzes areas of a specified color or color range in images.
	Image Processing	Filters or alters a gray scale image using one of a variety of filters, including logical and arithmetic calculations. This tool can be combined with other tools (including other Image Processing tools) to modify an image to the necessary specifications.
	Image Sampling	Extracts an area of an image and outputs it as a new image source.
	Position Compensation	Converts all pixels outside a rectangular region to black and outputs the result as a new image.
Custom Tools	Custom Vision Tool	Allows you to specify a program to be called when executed. This tool can execute other tools and return a custom set of results.
	Remote Vision Tool	Reserved for future use.

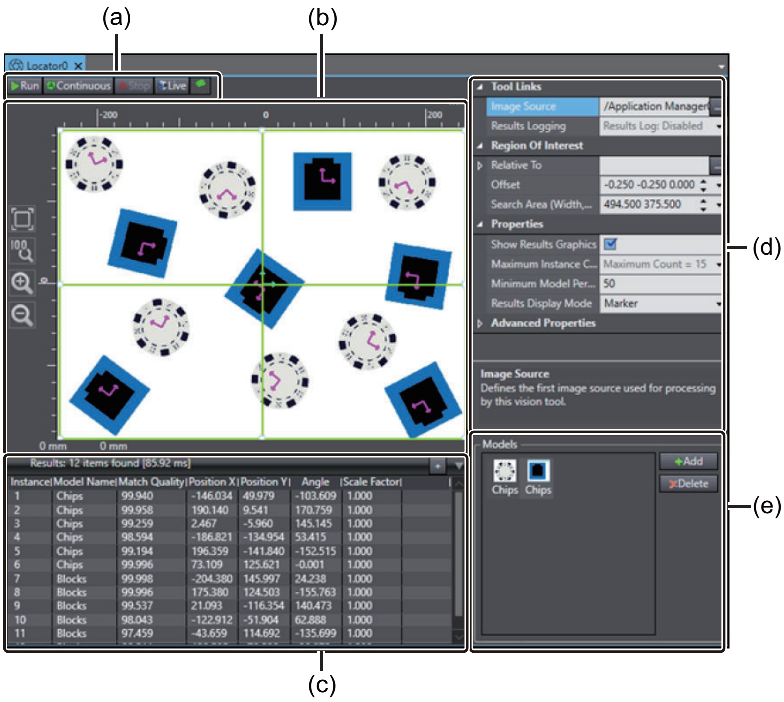
5-10-1 Adding Vision Tools

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.

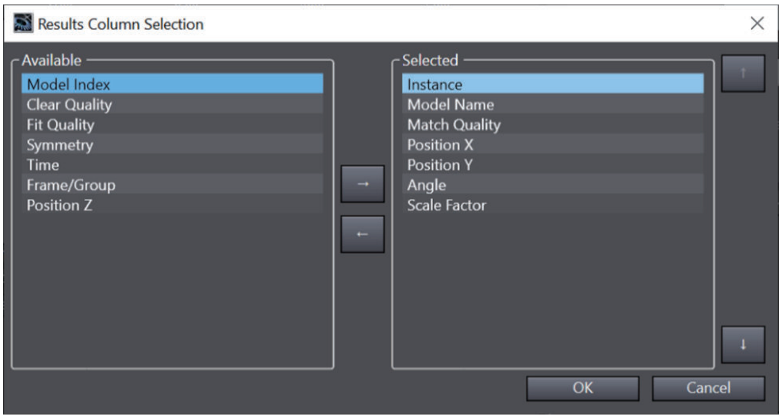


5-10-2 Vision Tool Editor

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	This area provides direct control of the tool with the following buttons. **Run**: Run the tool once. This is available with most vision tools. **Continuous**: Run the tool continuously. This is available with most vision tools. **Stop**: Stop the tool or live operation. Can only be selected if **Live** or **Continuous** has been activated. This is available with most vision tools. **Live**: Send images without running the tool. This is available with most vision tools. **Auto Tuning**: Automatically sets parameters based on pixel data. This functions differently depending on the tool. See those tools for more information.Refer to for more information. This is available with the Barcode, Color Data, Data Matrix, OCR, and QR Code tools. **New Image** button: Loads a new image into the tool. The model will need to be retrained. This replaces the **Run** button in the respective tools. This is available with the Locator Model and Shape Search 3 Model tools. **Register to OCR Dictionary** button: Registers a character to the OCR Dictionary. This is available with the OCR Dictionary tool.
(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 Figure 8-174 displays markers for each identified instance within the green region of interest, which can be modified as needed.

	Item	Description
(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 Plus button (+) which will display the **Results Column Selection** Dialog Box. Use this dialog box to select and arrange the data displayed in this area.  If the tool is in the Image Process category, it returns an image instead of data. The **Results** section of the tool editor will either show the result image, such as in **Advanced Filter**, or be **nonexistent**, such as in Color Matching. In the latter case, the Vision Window must be opened to view the resultant image.
(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 out- lined 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.

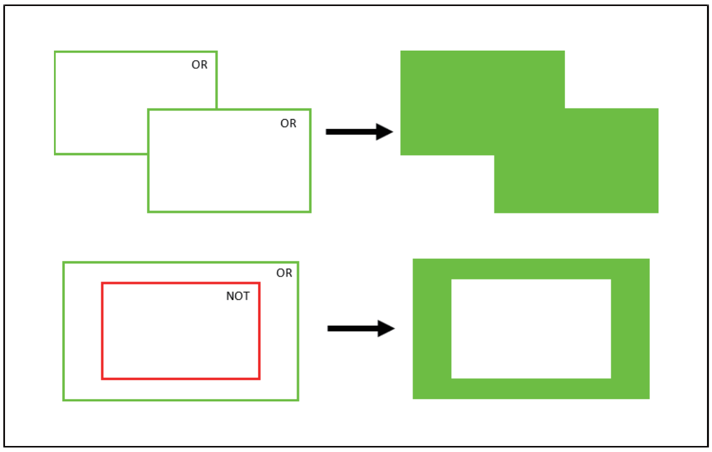
The region orientation of some tools can only be controlled with the **Offset** property.

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.

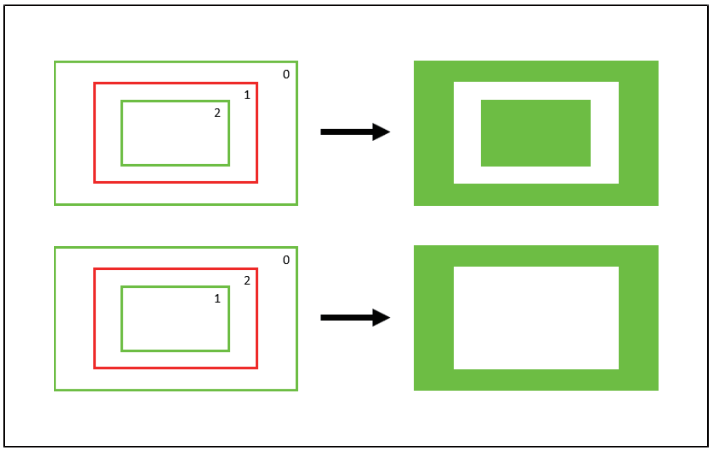
Overlap

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 following figure shows how individual regions are registered in the figure. The original regions are displayed on the left and the resulting processed region is displayed on the right, where the green area shows what parts are read. For example, in the second set of regions, the NOT region eliminates a section of the full region, resulting in a rectangular section that is not read.



Z-Order

This defines the order in which regions are processed. Each region has its own property that defines its order in the Z-direction of an image. Since the image is two-dimensional, this value is used to determine which regions are processed first. Regions are processed in ascending order. This can be seen in the regions in 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.

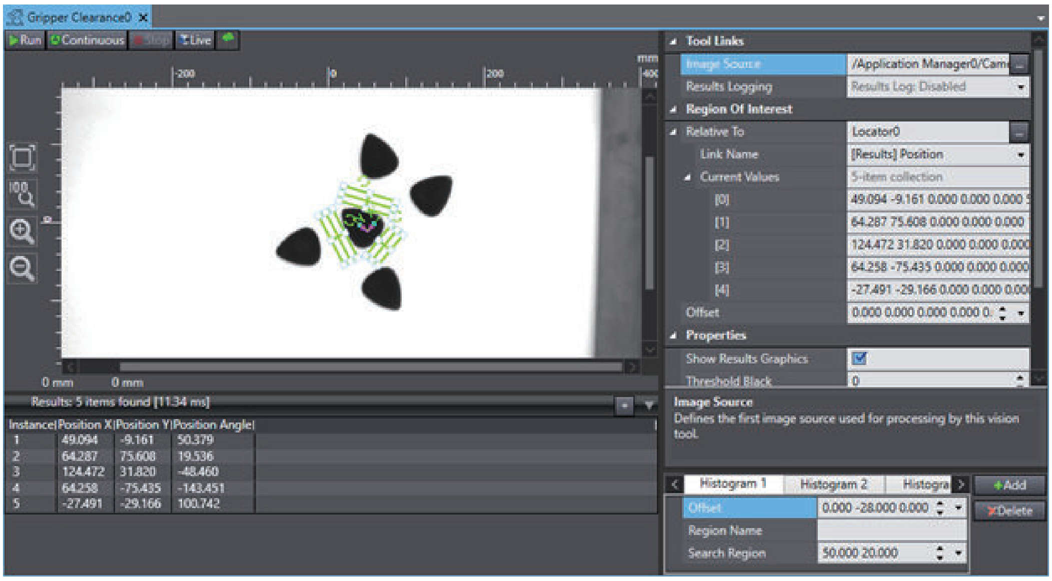


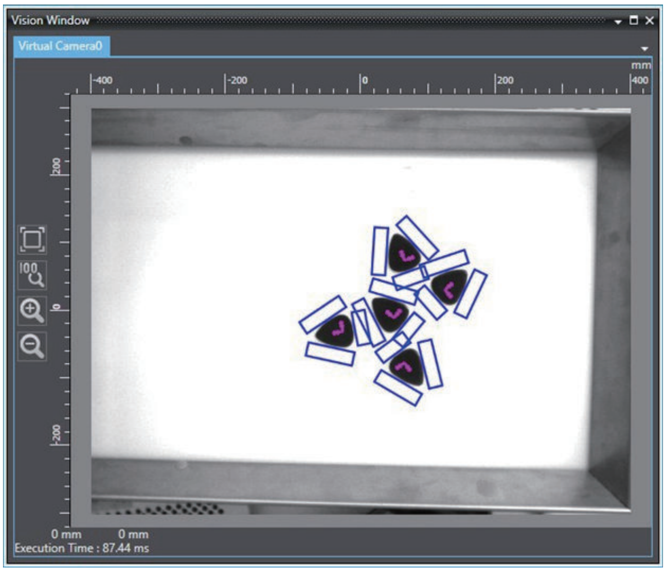
Relative to Parameter Details

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. See 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* on page 5-37 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.

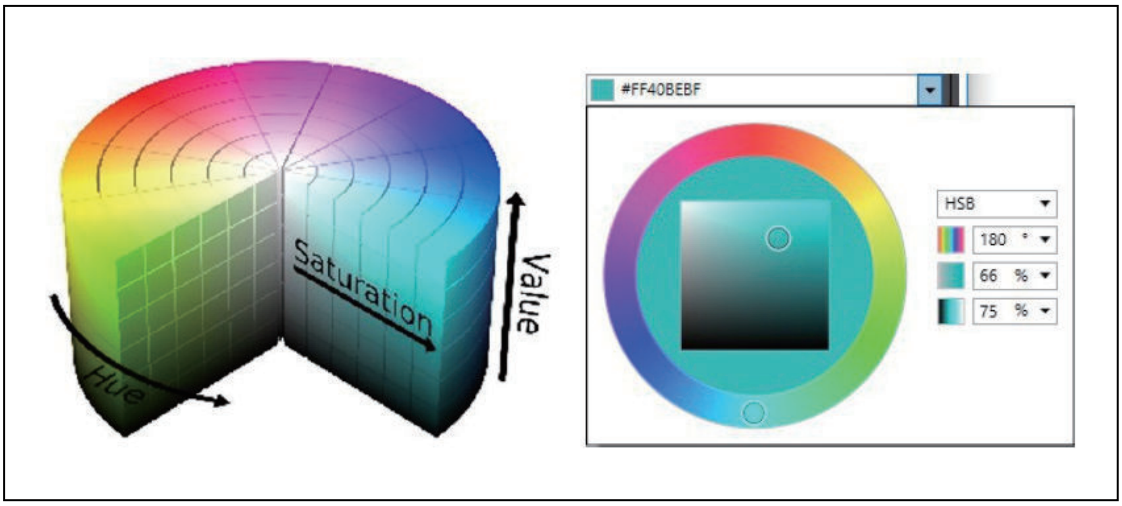
HSB and RGB Color Spaces

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

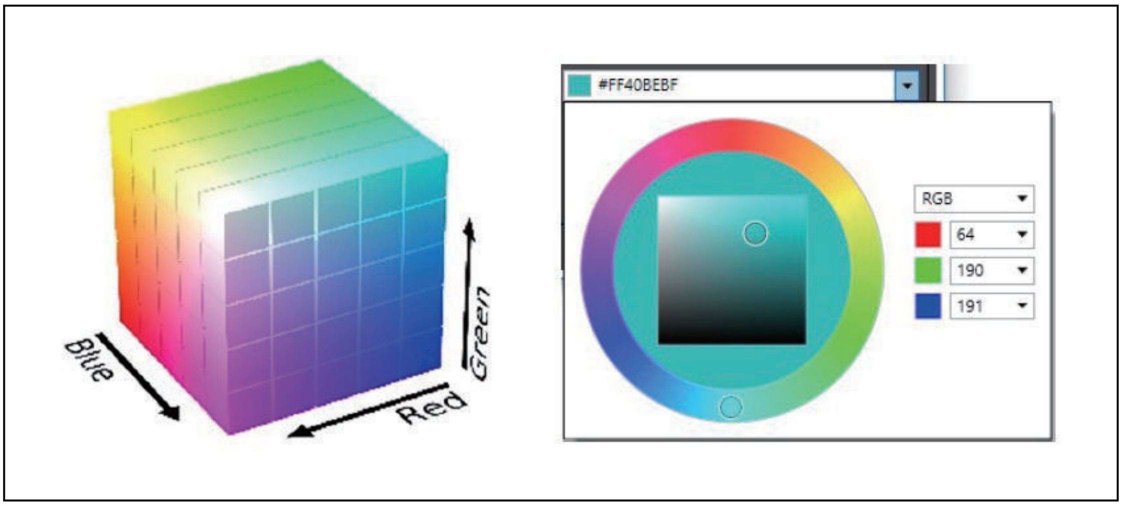
The HSB color space arranges colors of each hue radially around a central axis of basic colors, from white at the top to black at the bottom. Hue values are set in degrees from 0 to 360. Saturation and brightness are set in percentages from 0 to 100%.
Hue is the quality of color perceived as the color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

Some parts of Robot Integrated System applications, such as Application Manager or Dynamic Pick-and-place System, can be programmed in C#.
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 C#.
For the language specifications of C#, refer to the references and guides provided by Microsoft, or commercially available technical books for C#. Refer to the document below for the API specifications of the Application Manager, and sample C# programs.

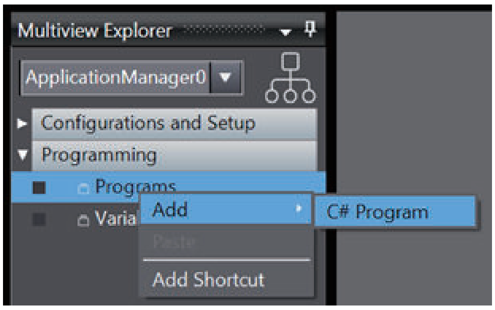
- ACE Reference Guide

5-11-1 C# Program Editor

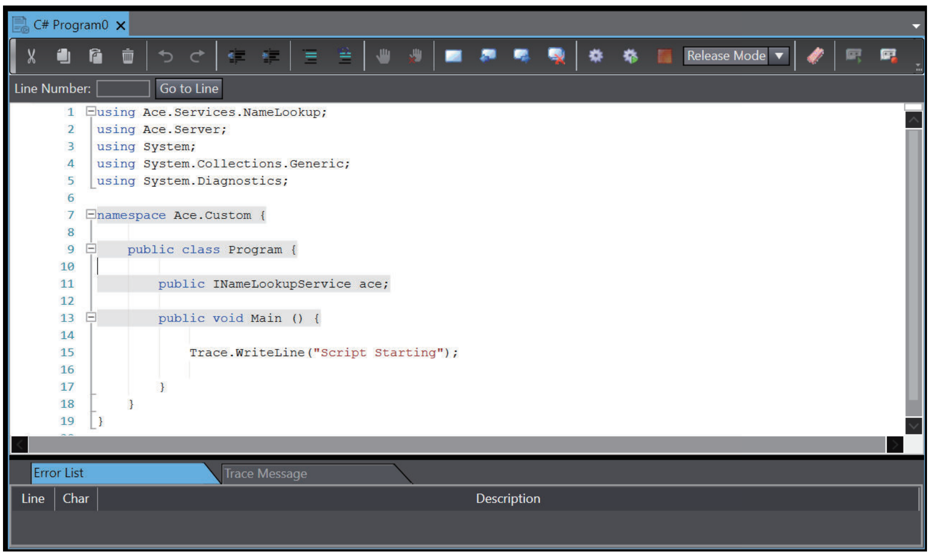
This section provides the information about how to use the C# program editor to create and debug a C# program.

Creating a New C# Program

1 To create a new C# program, right-click **Programs** and select **Add** and then **C# Program**.



A new C# program will be created.



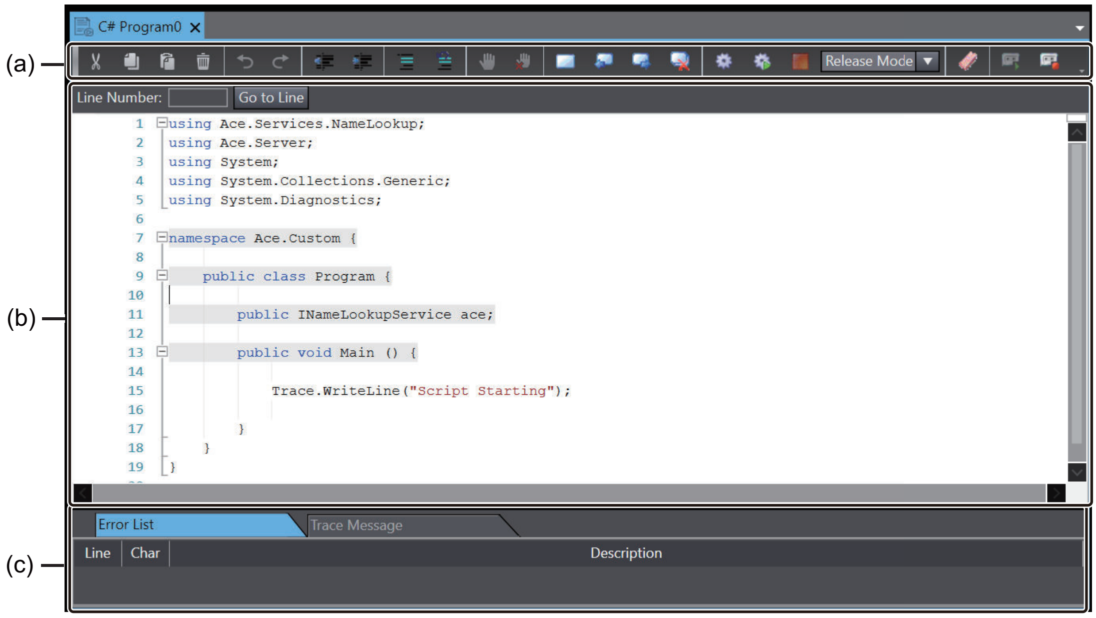
New C# programs are created with minimum program structure in place as well as a basic trace message to indicate when a script has started.

- **C# Program Names**

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

C# Program Editor

This section describes functionalities of the C# program editor.

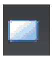
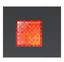
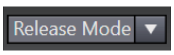


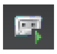
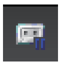
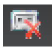
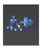
	Item	Description
(a)	Toolbar	The function buttons used to create scripts are displayed. Refer to *Toolbar Items* on page 5-193 for details.
(b)	Editor Pane	You can create scripts in this editor. Refer to *C# Programming* on page 5-195 for details.
(c)	Tab Pane	Errors and trace messages are displayed. Refer to *C# Program Editor Tabs* on page 5-195 for details.

Toolbar Items

Buttons on the toolbar can perform the operations in the table below.

Button	Operation (Short-cut)	Description
	Cut (Ctrl + X)	Cut the selection and store on the clipboard.
	Copy (Ctrl + C)	Copy the selection to the clipboard.

Button	Operation (Short-cut)	Description
	Paste (Ctrl + V)	Paste an item from the clipboard to where the cursor is.
	Delete (Del)	Delete the selection.
	Undo (Ctrl + Z)	Undo the last action.
	Redo (Ctrl + Y)	Redo the last action.
	Outdent (Shift + Tab)	Outdents the selected line.
	Indent (Tab)	Indents the selected line.
	Comment section	Comments out the selected line.
	Uncomment section	Uncomments the selected line.
	Toggle breakpoint	Adds/removes a breakpoint where a script stops to the current cursor position only in the debug mode.
	Clear all breakpoints	Removes all breakpoints only in the debug modes.
	Toggle bookmark	Adds/removes a bookmark to the selected line.
	Previous bookmark	Moves the cursor to the previous bookmark in the program.
	Next bookmark	Move the cursor to the next bookmark in the program.
	Clear bookmark	Removes all bookmarks.
	Compile	Compiles a C# program to check an error in the program. An error detail will be shown in the Error List tab.
	Compile and Run	Compiles the C# program then runs it.
	Stop	Stops the running C# program.
	Editor mode selection	Selects **Release**/**Debug** mode. Debug mode is used to debug the program.
	Erase trace	Clears messages from the Trace Message tab.

Button	Operation (Short-cut)	Description
	Run recorded macro	Runs a recorded macro. Each time you press the button, the keystrokes recorded in the macro will be performed.
	Records macro	A macro is a record of keystrokes on the C# program editor. Press this button to start recording, and press it again to stop.
	Pause recording	Pauses a macro recording. Press this button again to start recording.
	Cancel recording	Cancels a macro recording.
	Display an object member list	Displays a list of the available object members. Press the button while placing the cursor at the end of the C# object you have entered.
	Display parameter info	Displays the parameter information of the C# object that the cursor points.
	Display quick info	Displays the error information for the cursor location in a tooltip.
	Display word completion	Displays a list of the available C# object members based on the text you are typing.
	Go to Line	Moves the cursor to the line of the number you have entered.
	Step into	Displayed only in the Debug mode. Single step operation that will enter a function.
	Step over	Displayed only in the Debug mode. Single step operation that skips a function through the C# program.
	Go	Displayed only in the Debug mode. Runs the paused C# program.

C# Program Editor Tabs

The following tabs are available.

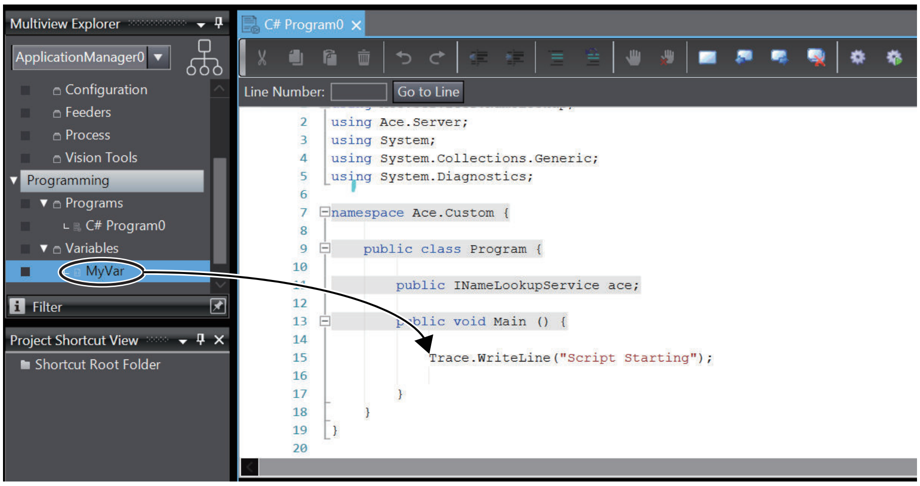
Tab	Description
Error List	Program compile errors are shown in the Error List tab. Line number, error location, and message are displayed for each error.
Trace Message	Trace messages, texts included in any Trace.WriteLine() call of a script, and warning information are displayed.

C# Programming

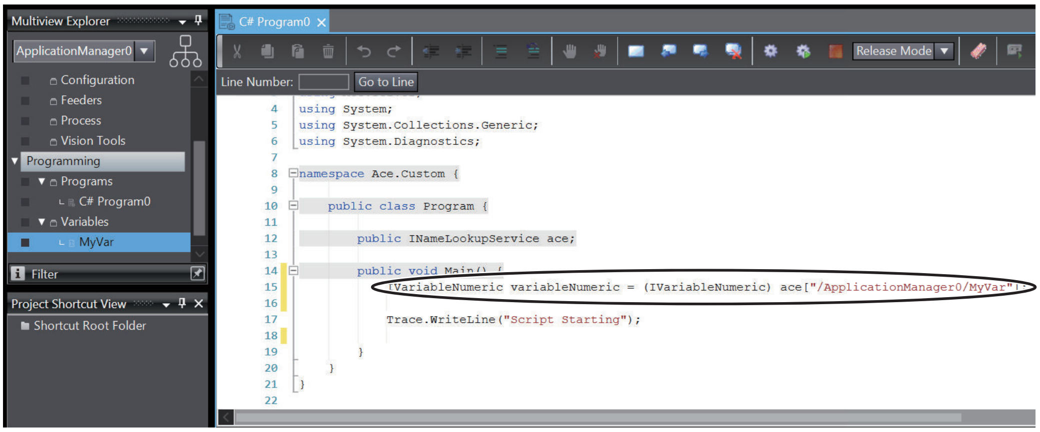
The operating procedure for programming with the C# Editor is described below.
Buttons on the toolbar help you to write a program efficiently.
Additionally, you can drag and drop objects in the Multiview Explorer to the C# Editor to declare variables.

In this section, learn the editor operation taking an example of variable declaration necessary to use C# variable objects in a C# program.

1 Click **Programming** - **Variables** in the Multiview Explorer, then drag and drop **MyVar** to the point you want to insert it.



The variable declaration for a part is inserted there.

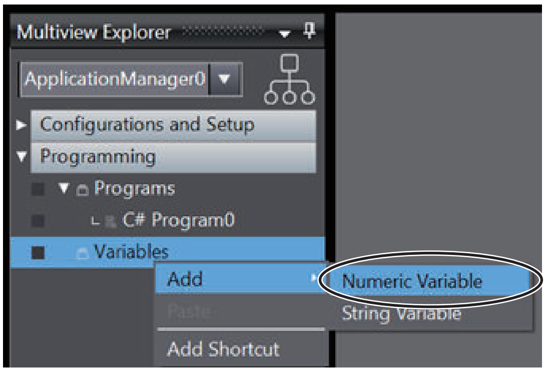


5-11-2 C# Variable Objects

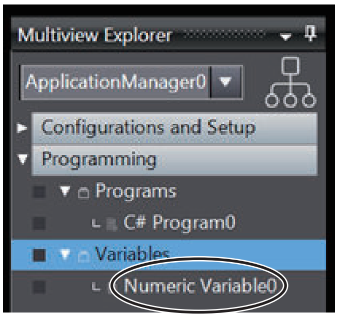
Numeric and string C# Variable objects can be used to share data between C# programs that run separately. Numeric types hold numerical values and string types hold text-based values used to represent characters. Valid string characters include 128 8-bit ASCII characters.

Creating a New C# Variable Object

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

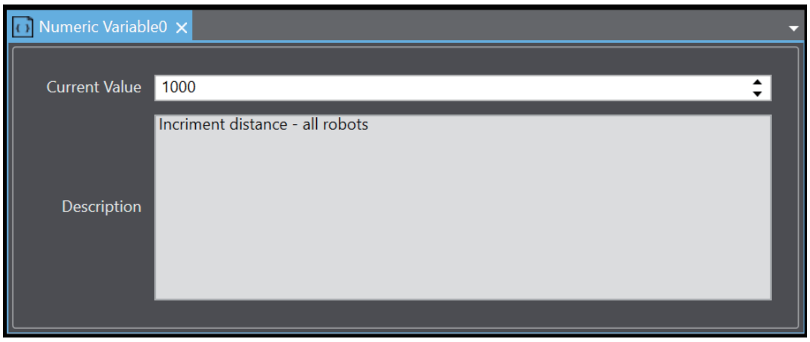


The new variable will be added to the Variables list.



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 C# Variable Object Functions

Several functions are available for C# Variables objects. Right-click a C# Variable object in the Multi-view Explorer to display a menu with function selections described below.

- 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 V+ Watch Tab Page* on page 7-46 for more information.

6

Online Connections to IPC Application Controllers

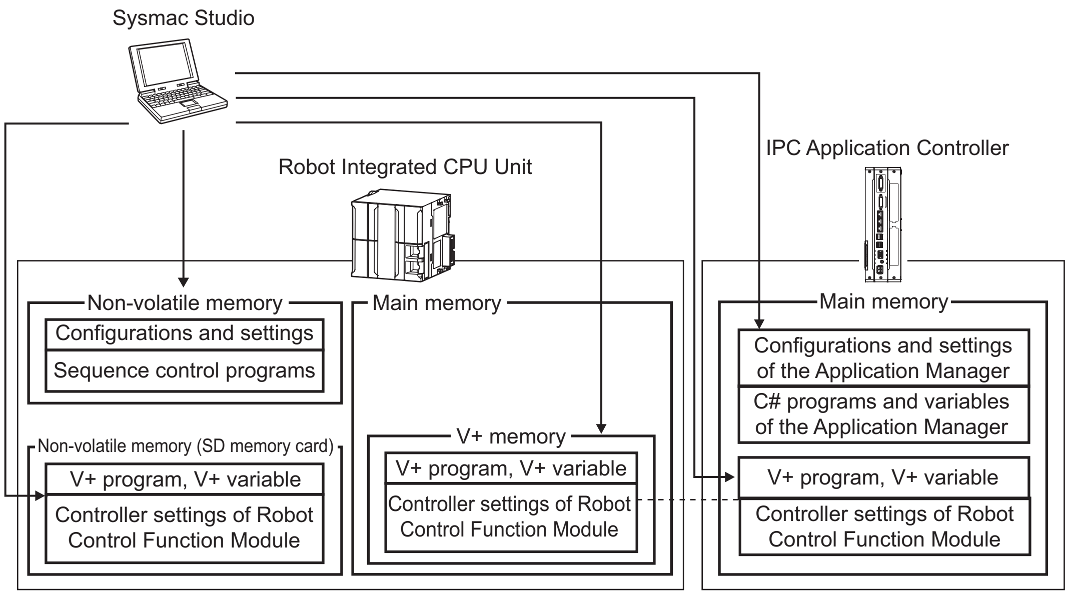
This section provides the information about online connections to the Robot Integrated CPU Units and IPC Application Controllers, and the data transfer method.

6-1 Data Locations of the Robot Integrated System with IPC Application Controller in Sysmac Studio

You need to transfer the settings and programs of Application Managers running on the IPC Application Controller, in addition to the settings and programs of the Robot Integrated CPU Unit to operate the Robot Integrated System with the IPC Application Controller.
Devices in the Robot Integrated System with IPC Application Controller have data as described below.

Devices	Data	
	Device in Project	Description
Robot Integrated CPU Unit	Controller	Controller configurations, settings, and sequence control programs
Robot Integrated CPU Unit / IPC Application Controller	RobotControlSettings	Controller settings of the Robot Control Function Module, V+ programs, V+ variables
IPC Application Controller	Application Manager	Application Manager configurations, settings, and #C programs

The IPC Application Controller manages part processes and sends robot operating instructions to the Robot Integrated CPU Unit. It also processes the image data acquired from the connected camera as a vision sensor, and sends signals to the Robot Integrated CPU Unit. In order to achieve them, it is necessary to transfer the controller settings of the Robot Control Function Module, V+ programs, and V+ variables in the Robot Integrated CPU Unit to the IPC Application Controller.



To locate the data as illustrated above, you must transfer settings and programs of the Robot Integrated System with the IPC Application Controller through an online connection to the respective devices. Establish an connection and transfer the data following the procedure described below.

Procedure	Devices	Description	Reference
1	Robot Integrated CPU Unit	Online connection	*Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*
2		Transfer of settings and programs	
3	IPC Application Controller	Online connection	*6-2 Online Connection* on page 6-4
4		Transfer of settings and programs	*6-3 Project Synchronization* on page 6-6

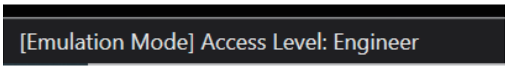
6-2 Online Connection

Establish a online connection to use online functions connecting with the IPC Application Controller.

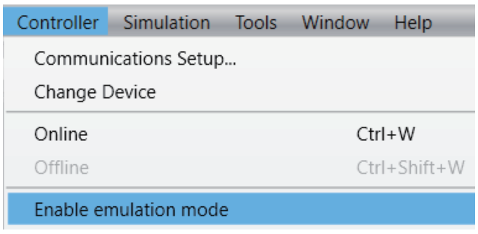
Precautions for Correct Use

The Sysmac Studio's Application Manager project version must be the same as the Application Manager version of the ACE software package running on the IPC Application Controller. If not, you cannot go online with the IPC Application Controller.
Refer to the *Automation Control Environment (ACE) Version 4 User's Manual (I633)* for details on the connectable Application Manager versions.

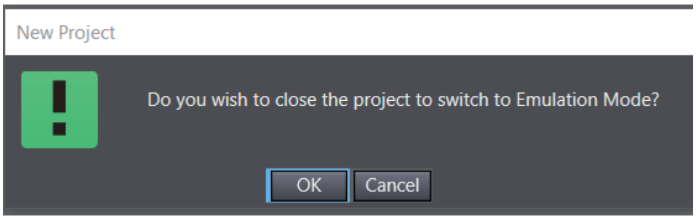
1 See the status bar to confirm that Sysmac Studio is not in the Emulation mode.



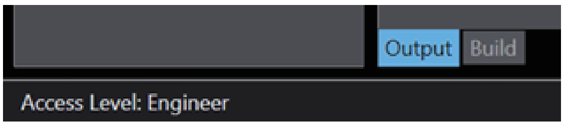
2 If in the Emulation mode, select **Controller** – **Disable emulation mode** in the main menu.



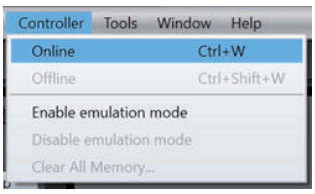
3 Click the **OK** button.



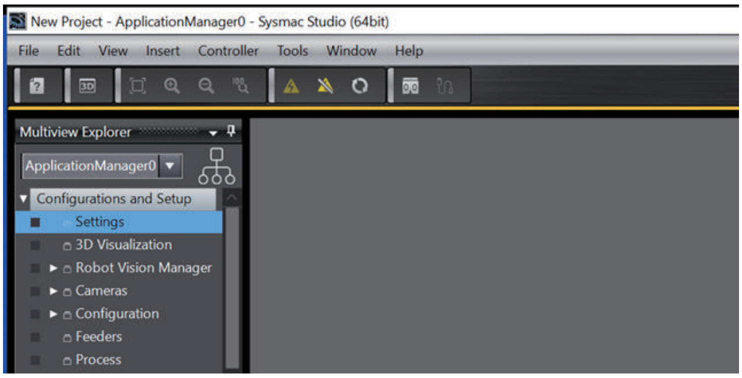
The project exits, then re-opens. Confirm not in the Emulation mode.



4 Select **Controller** – **Online** in the main menu. Otherwise click the **Online** button.



The yellow bar that indicates a online state is shown and the**Synchronize** button is enabled.



6-3 Project Synchronization

Compare the following data in the Sysmac Studio project and in the IPC Application Controller's Application Manager and list differences.

- The data of configurations, settings, and programming (#C programs, variables)
- Controller settings of the Robot Control Function Module in RobotControlSettings, V+ programs, and V+ variables

You can download and upload the project data to/from the Application Manager at once by specifying a transfer destination and direction.

⚠ WARNING

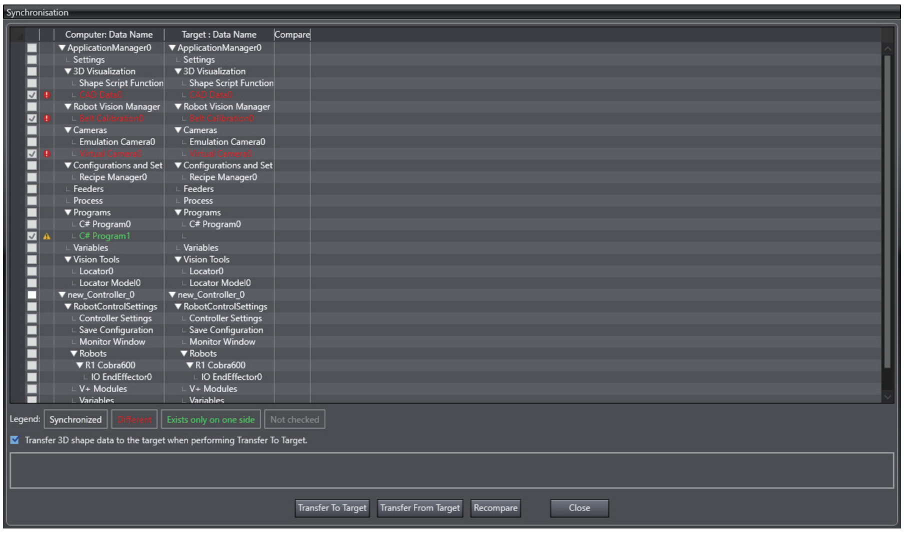
Check operations of the created user programs, data, and setting values carefully before proceeding to normal operation.

⚠ WARNING

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

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

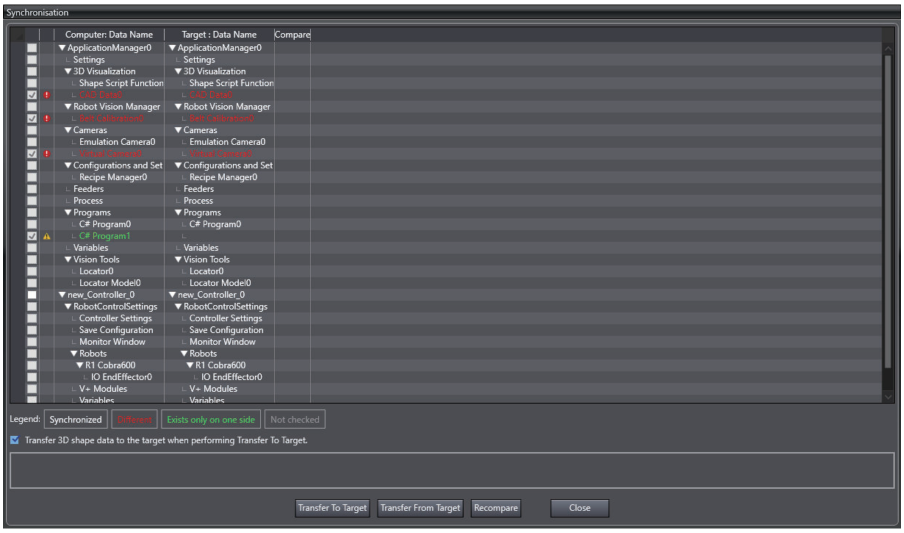
A list of the followings is shown: configurations, settings, and programs in the Application Manager devices; settings, V+ programs, and V+ variables in the robot subdevices that is to be downloaded to the associated Robot Integrated CPU Unit.



The results of the comparison are displayed as shown below. For V+ Programs, you can display the detailed comparison window.

Legend	Description	Detailed comparison
Synchronized (No differences)	The contents of the project match those of the Application Manager.	-
Different	There are differences between the project and the Application Manager in settings, programs, and/or variables.	You cannot display the detailed comparison window.
Exists only on one side	The data exist in either the project or the Application Manager.	You cannot display the detailed comparison window.

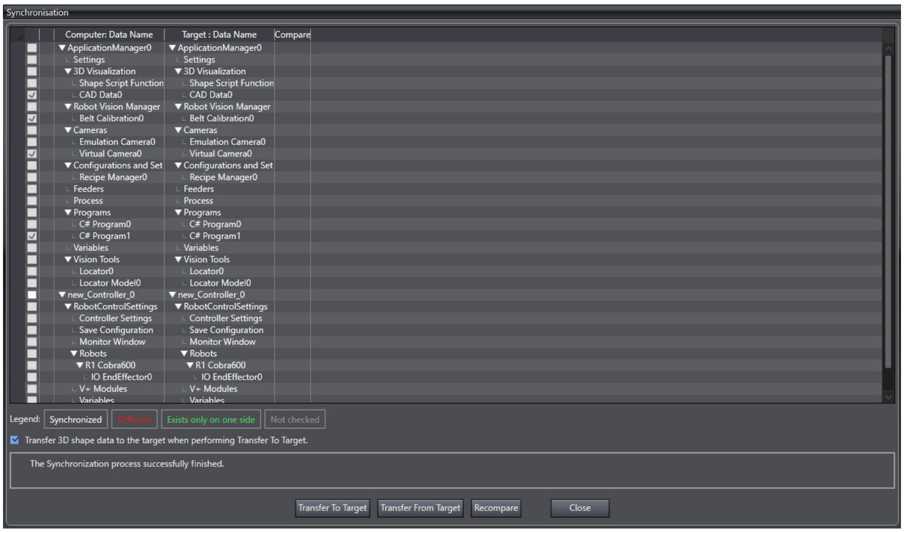
2 Select the data to transfer.



3 Select the items to check for synchronization.
The transfer process depends on which of these items are selected. Details are as follows.

Option	Default	Enabled	Description
Transfer 3D shape data to the target when performing *Transfer To Target*.	Selected	Transfer To Target	Transfers 3D shape data to the target. If no 3D shape data is transferred to the target, a simulation and a collision judgment based on the 3D shape data cannot be performed. Refer to *5-4-3 Exporting/Importing 3D Shape Data* on page 5-11 for the operation when you want to perform a simulation and a collision judgment in the project that includes no 3D shape data.

4 Click the **Transfer to Target** or **Transfer from Target** button. Here, click the **Transfer to Target** button.



The selected project data will be transferred to the Application Manager, then the synchronization is completed.

5 Click the **Close** button.

7

Debugging the Robot Integrated System

This section provides the information about functions used for a debug of the Robot Integrated System with IPC Application Controllers.

7-1 Overviewing a Debug of the Robot Integrated System with IPC Application Controllers

There are two methods to debug the Robot Integrated System: the offline debug with Sysmac Studio 3D simulation function, and the online debug connecting to an actual machine. As for the Sysmac Studio debug functions, this section describes the common functions among the offline/online debug, and unique functions for the offline and online debugs respectively.

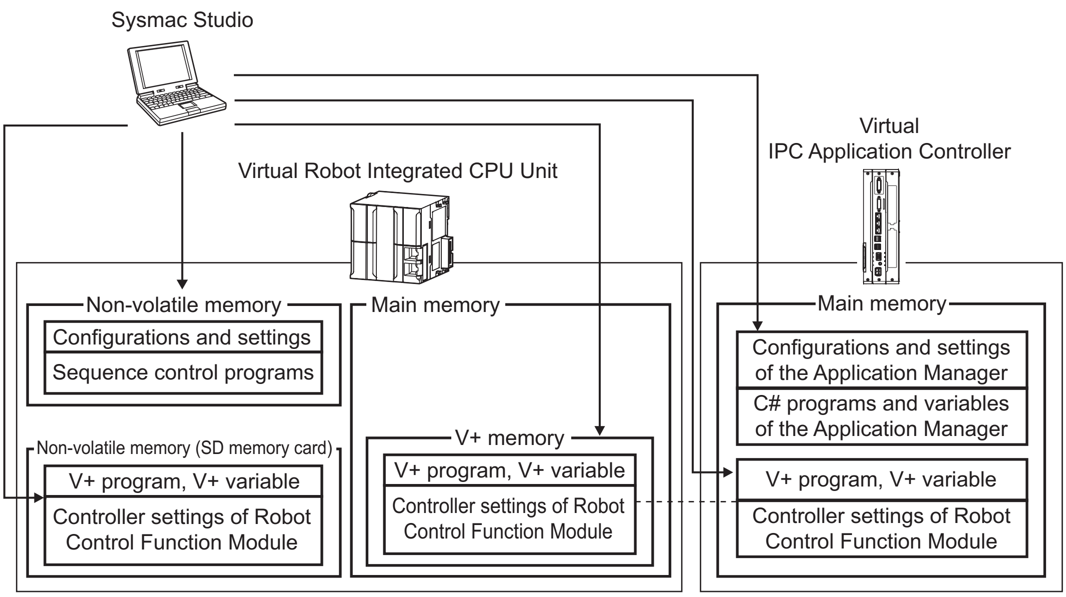
Refer to *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the debug function for sequence control programs.

⚠ 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.

7-1-1 Schema of Offline Debug Procedure

This section describes the offline debug by simulation.
Run the virtual IPC Application Controller and Robot Integrated CPU Unit on Sysmac Studio in order to debug the Robot Integrated System with the IPC Application Controller.



Following the procedure described below to boot up the virtual IPC Application Controller and the Robot Integrated CPU Unit.

Procedure	Operation on Sysmac Studio		Reference
Booting up the virtual IPC Application Controller and the Robot Integrated CPU Unit	Open a project in the Emulation mode		*4-6 Using Online and Offline Modes of the Sysmac Studio* on page 4-17
	Boot up the Simulator		*Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*
Teaching in simulation	Perform a teaching using the 3D Visualizer or V + Jog Control pane.		*7-2 3D Visualizer* on page 7-47*-4 V+ Jog Control* on page 7-22
Offline debug	Use Pack Manager	Boot up a Process	*7-5-9 Application Manager Control* on page 7-41
	Use Robot Vision Manager	Run a V+ program	*7-5-4 Task Manager* on page 7-33

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 3D Visualizer window displays the following information.

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

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 the Emulation mode
- Measurement of distance between 3D objects
- Robot jogging
- Location visualization and editing
- Integration of workcell objects such as belts, parts, end effectors, and cameras

The following sections describe operations and functions of the 3D Visualizer.

7-2-2 Creating 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.

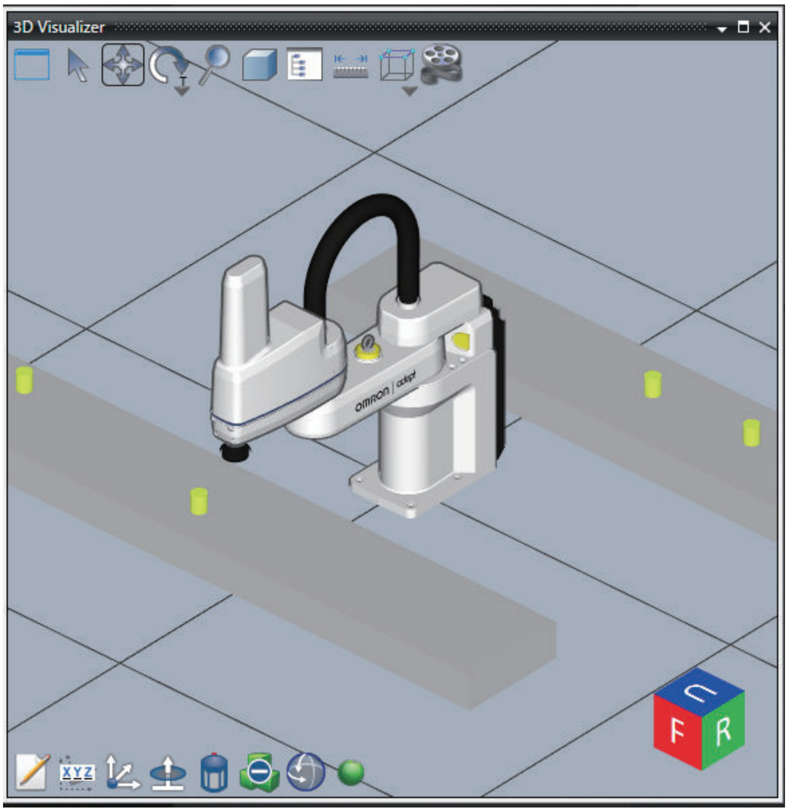
1. Add a Omron robot in the EtherCAT Configuration Tab Page. Refer to *4-2-1 Creating a New Project* on page 4-4 for details.
2. Add 3D Shape Data such as box, cylinder, and imported CAD shape, if necessary.
3. Add cameras if necessary.
4. Add feeders if necessary.
5. Add Process Manager items if necessary (part buffer, part target, part, belt, pallet etc.).

Additional Information

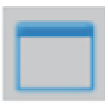
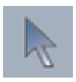
Register the boxes, cylinders, and imported CAD shapes in the Application Manager. Refer to the *Sysmac Studio 3D Simulation Function Operation Manual (Cat. No. W618)* for registration method.

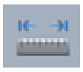
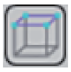
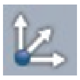
7-2-3 Description of Icons

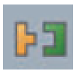
Use the main toolbar icon, or select 3D Visualizer from the **View** menu.to access the **3D Visualizer**. The 3D Visualizer has the following control icons.



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 *Split Window* on page 7-7 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 *Rotate* on page 7-7 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.

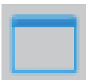
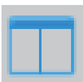
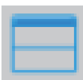
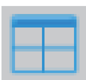
Icon	Name	Description
	Projection Mode	Changes the projection modes in the 3D Visualizer: parallel projection or perspective projection. Refer to *Projection Mode* on page 7-8 for details.
	Scene Graph	Opens the Scene Graph dialog box to configure the visibility and collision detection settings for 3D shape data. Refer to *Scene Graph* on page 7-8 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 **Select point 1** and **Select point 2** 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 *Snap* on page 7-10 for details.
	Record	Captures a simulation executed in the 3D Visualizer on video. Refer to *Record* on page 7-12 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 *Editing 3D Shape Data Simply* on page 7-14 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 **Offset From Parent** 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 **Offset From Parent** 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 *Jog Mode* on page 7-16 for details.
	Show Obstacles	Shows or hides obstacles. Refer to *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for details. This icon is displayed only when a robot is selected.

Icon	Name	Description
	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 *Sysmac Studio 3D Simulation Function Operation Manual (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 *3D View Switching Tool* on page 7-18 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 ±90° 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

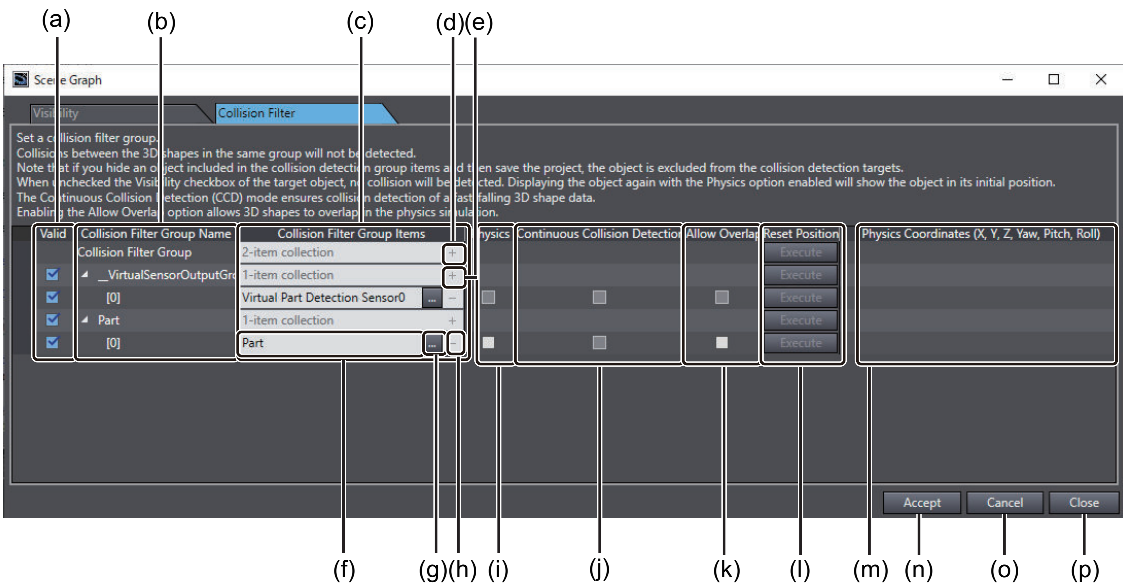
In the list displayed on the tab page, set whether to show or hide the tiles, ruler, and 3D shape data in the 3D Visualizer.

To display an item, select the corresponding check box.

	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.



	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 as a collision detection target.	Checked or unchecked	Checked
(b)	Collision Filter Group Name list	Manage the items in the Collision Filter Group list. The text box displays **-itemcollection (* is the collision filter group number)*. Click the **Add Collision Filter Group** button to add a collision filter group. The group name to be added is *Group* (* is the number of collision filter groups)*.	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. **-itemcollection (* is the number of collision filter group items)* is displayed in the list. Click the **Add Collision Filter Group Items** button to add a collision filter group item.	Text string	0-item collection (No collision filter group item in the initial status)

	Item	Description	Set value	Initial value
(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 Collision Detection Settings* on page 7-19 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)	Physics	Selecting this check box and clicking the **Accept** button enables the physics simulation of the target 3D shape data. This allows you to check how the falling 3D shape data collides with other 3D shape data and how the movable parts of custom mechanics move.	Checked or unchecked	Unchecked
(j)	Continuous Collision Detection	Enabling Continuous Collision Detection (CCD) makes it possible to certainly detect the collision of 3D shape data falling at high speed. Enabling this function makes sure of detection of collisions but affects the drawing performance of 3D Visualizer.	Checked or unchecked	Unchecked
(k)	Allow Overlap	Enabling Allow Overlap allows the target 3D shape data of physics simulation to overlap with each other.	Checked or unchecked	Unchecked
(l)	Reset Position	Resets the target 3D shape data of physics simulation to the original position. When the **Physics** check box is selected, the target 3D shape data falls from the original position or the movable parts of a custom mechanics move from the original position each time you click the **Execute** button.	---	---
(m)	Physics Coordinates (X, Y, Z, Yaw, Pitch, Roll)	Displays the coordinate values of the 3D shape data during physics simulation.	---	---
(n)	**Accept** button	Accepts the changes.	---	---
(o)	**Cancel** button	Cancels the changes and then, closes the Scene Graph dialog box.	---	---
(p)	**Close** button	Closes the Scene Graph dialog box.	---	---

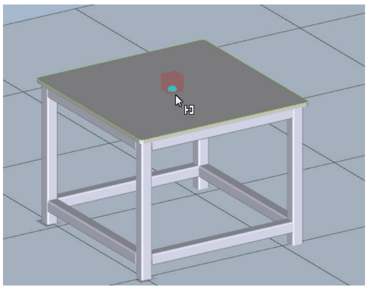
Refer to *7-2-4 Collision Detection Settings* on page 7-19 for details on how to configure the collision detection settings.

Snap

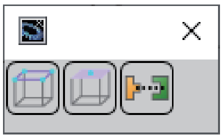
Use this icon to set an offset for a mount point or a link point of 3D shape data.

1 Select the target 3D shape data, and then click the **Snap** icon or press the T key.

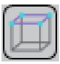
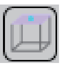
2 Move the cursor to where you want to snap it to display candidate points of the snap destination. Click one of the candidate points to snap it to that position.

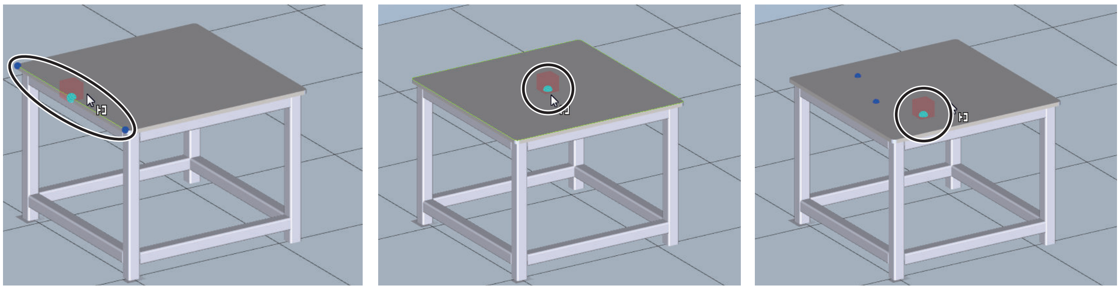


To change the mode of snapping, click the icon and select its function in the following window. Or, press the T + number keys. To close the window, click ×.



The functions of icons that you can select in the window are as follows.

Icon	Name	Function	Shortcut keys
	Snap to Edge	Moving the cursor near a mount point or link point that is set in the 3D shape data displays a preview of the snap position. 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.	T + 1
	Snap to Face	Moving the cursor near a mount point or link point that is set in the 3D shape data displays a preview of the snap position. The center of gravity of the highlighted face is emphasized, and the mount point or the link point can be snapped to the position.	T + 2
	Snap to Link	Moving the cursor near a mount point that is set in the 3D shape data displays a preview of the snap position. Any link point that is present on the highlighted face is displayed, and the mount point can be snapped to the position.	T + 5



Snap to Edge Snap to Face Snap to Link

Additional Information

- A preview of the snap position will be displayed for the following objects.
  a) Box
  b) Cylinder
  c) CAD data
  d) Mechanical component
  e) Custom mechanics
  f) Parallel link model
  When an end-effector mounted on a robot is snapped, a preview of the above objects a) to f) that have the end-effector set as the parent will be displayed.
- You can change the highlight color of the snap position. Change it in the option settings. Refer to the *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for details.

Record

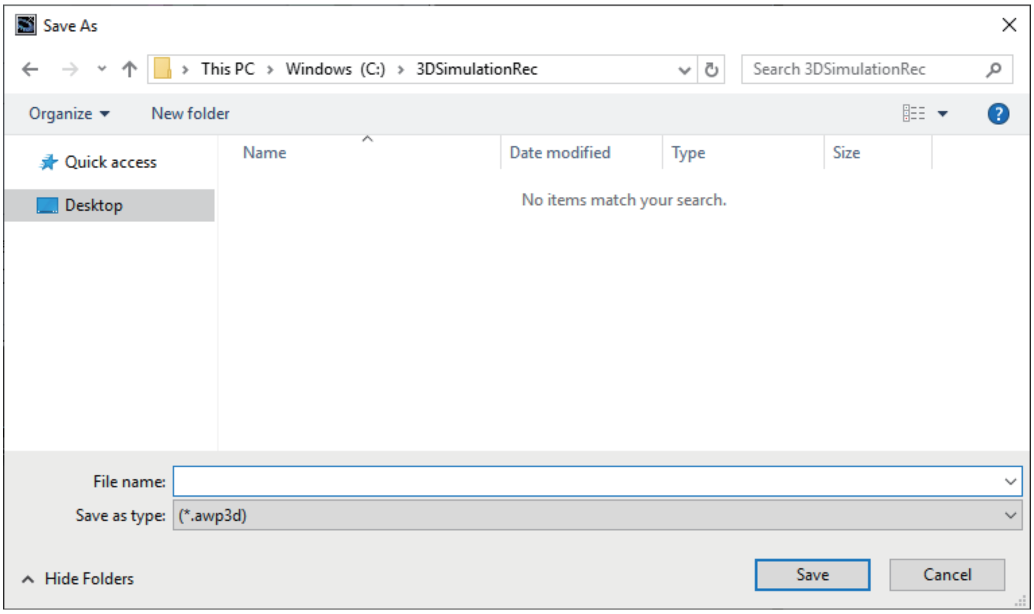
Use this icon to capture a simulation executed in the 3D Visualizer on video.

● 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.

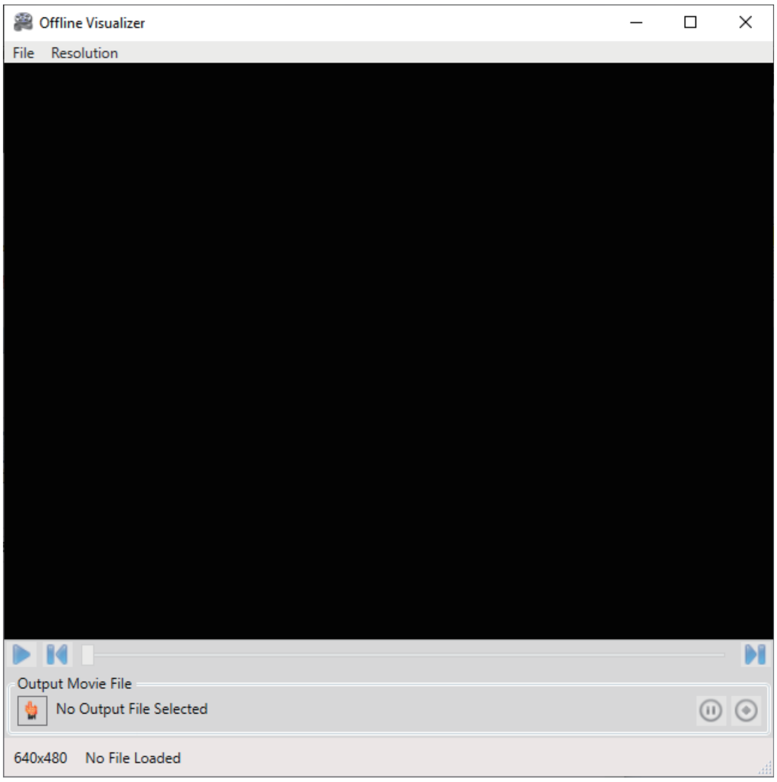


3 Enter the file name, and then click the **Save** button.
The video is saved to a file.

● Playing Back Video

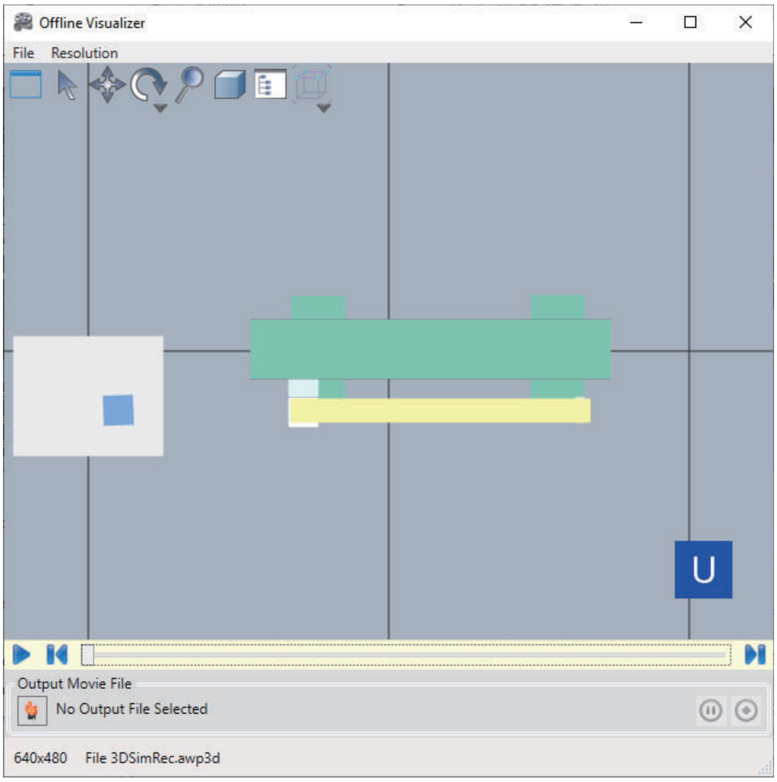
Use the Offline Visualizer to play back video.

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** 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.

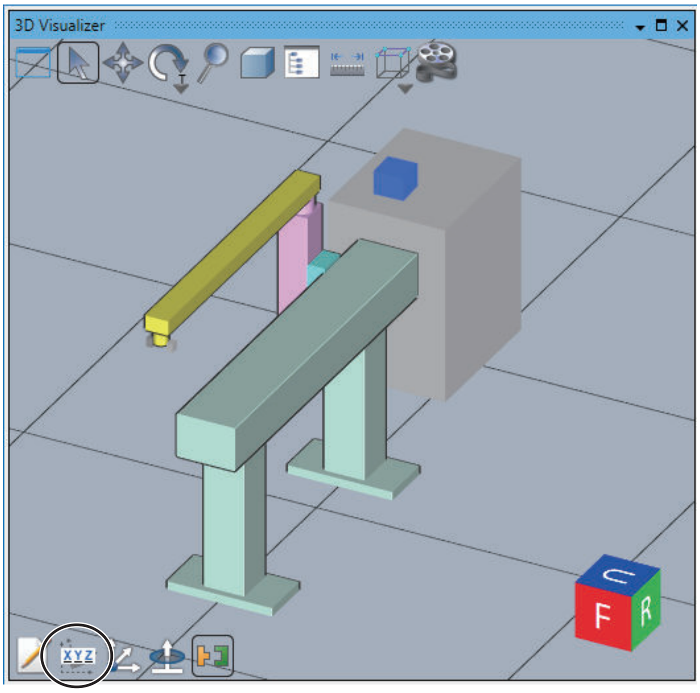


4 Click the **Play** button.
The video is played back.

Editing 3D Shape Data Simply

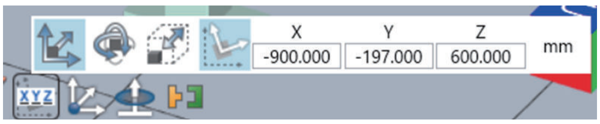
The 3D Visualizer allows you to edit 3D shape data by directly entering values while you are checking the positional relationship between the 3D shape data.

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.

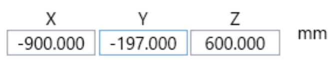
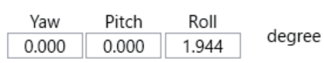


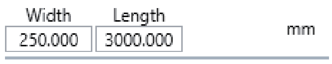
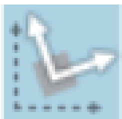
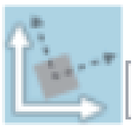
2 Click the **Direct Position Edit** icon.

The value input fields for the 3D shape data are displayed in the 3D Visualizer.



3 Select an icon in the **Direct Position Edit** to enable the value input fields for the 3D shape data.

Icon	Name	Function
	Edit Workspace Position	Enter X, Y, and Z values that specify the position of the 3D shape data. 
	Edit Workspace Orientation	Enter Yaw, Pitch, and Roll values that specify the orientation of the 3D shape data. 

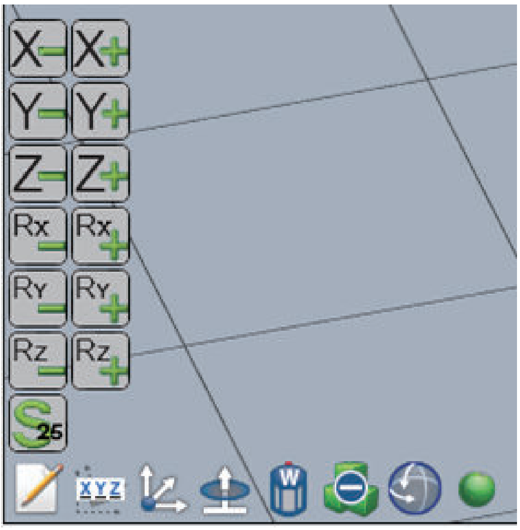
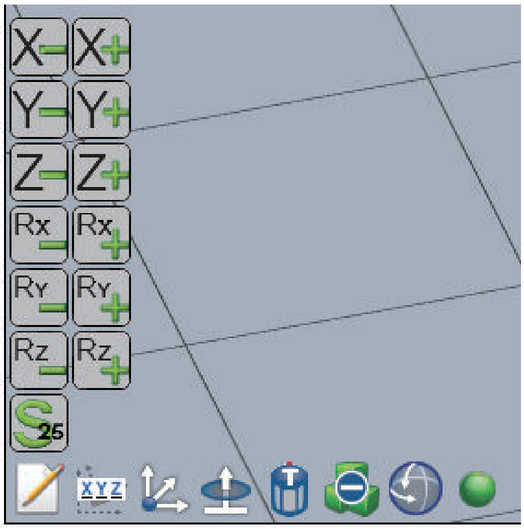
Icon	Name	Function
	Edit Size	Enter a value that specifies the size of the 3D shape data. The values that you can enter change depending on the 3D shape data. **Box**  **Cylinder**  **Virtual Part Detection Sensor**  **Belt** 
	Local Coordinate System	Select the coordinate system used for **Edit Workspace Position** or **Edit Workspace Orientation**. The coordinate system toggles every time you click. The initial setting is the local coordinate system.
	World Coordinate System	

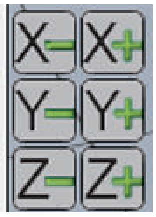
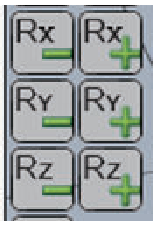
Jog Mode

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.

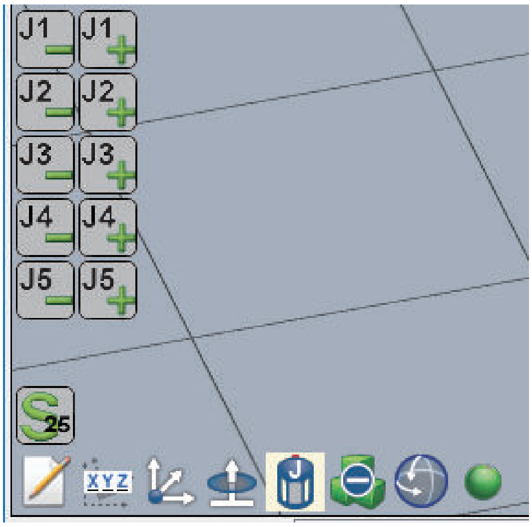
Icon	Mode
	World
	Tool
	Joint

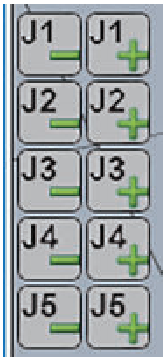
- **Jog Mode: World or Tool**

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.

Configuration element	Name	Description	Shortcut keys
	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 yz plane	Ctrl + 1
		B (Back): The face parallel to the F face	Ctrl + 2
		U (Up): The upper orthogonal face to the F face	Ctrl + 3
		D (Down): The lower orthogonal face to the F face	Ctrl + 4
		L (Left): The left side face to the F face	Ctrl + 5
		R (Right): The right side face to the F face	Ctrl + 6
	Corner	Represents a corner.	---
	Edge	Represents an edge.	Ctrl + 7 (The edge between Up and Front)

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

Collision Detection differs from 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* on page 7-8 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 (Cat. No. W618)* for details.

7-3 Event Log

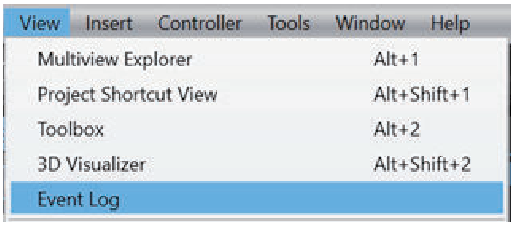
The Event Log displays a log of events that have occurred since the Robot Integrated System was started. Events are categorized as Error, Warning, and Information and can be used for troubleshooting and diagnostics. Event Log messages can be displayed by event type, time stamp, and message.

Precautions for Correct Use

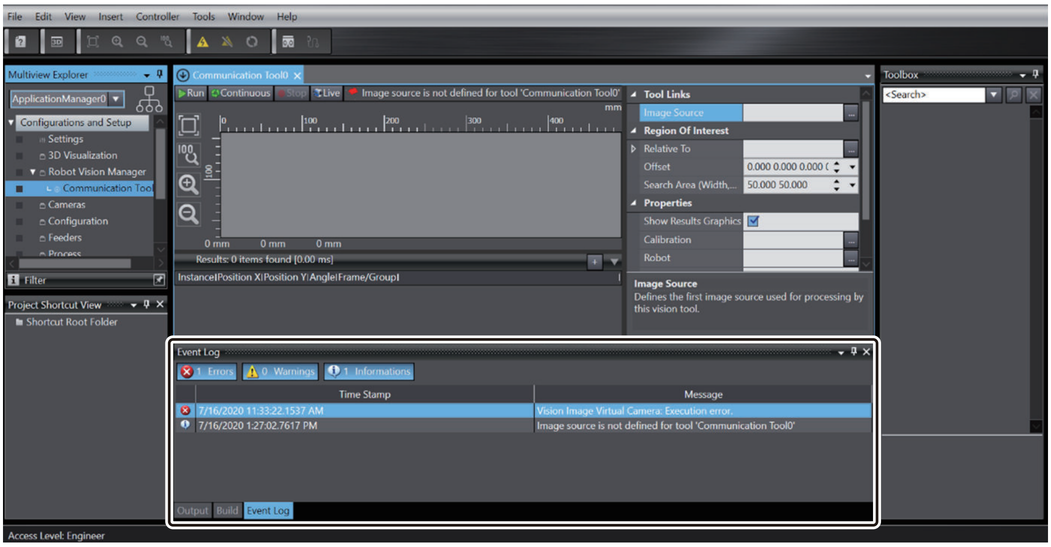
Past information that is displayed in the logs may not reflect the current status of the controller or programs.

Starting the Event Log

1 Select *RobotControlSettings* from the device list in the Multiview Explorer or *Application Manager*, then select **View** – **Event Log** from the main menu.

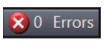


The **Event Log** pane is displayed.



Event Log Items

The following information is shown in the Event Log pane.

Icon	Event Type	Function
	Error	Click this button to display all error-type events.

Icon	Event Type	Function
	Warning	Click this button to display all warning-type events.
	Information	Click this button to display all information-type 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. Select **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 select **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 check 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.

⚠ WARNING

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

⚠ WARNING

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

⚠ WARNING

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 be-fore operating the robot.

⚠ WARNING

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

⚠ WARNING

When more than one software application included in the Sysmac Studio or IPC Appli-cation Controller is simultaneously connected online to the CPU Unit that controls a robot integrated system, do not perform the following operations at the same time.
The robots controlled by the CPU Unit may not operate as intended.

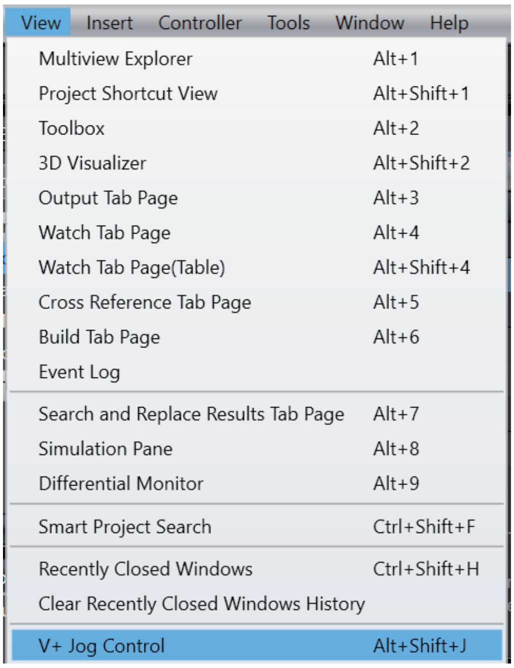
- Changing the settings of devices
- Online debug
- Teaching robots

Additional Information

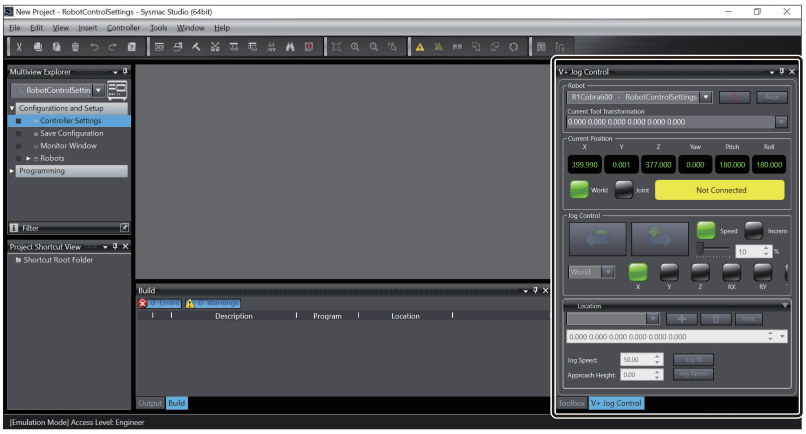
V+ Jog Control is effective on both the robot in the Emulation mode and the actual robot.
Most of jog commands and settings are disabled when the robot is controlled by Program Control. Refer to *Current Position Section* on page 7-25 for details.

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.



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

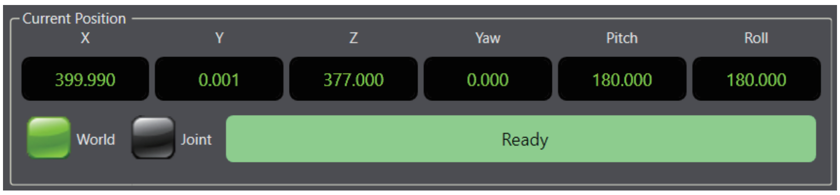
The **Align** button aligns the robot tool Z-axis with the nearest World axis (six-axis robots only).

● 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

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



Additional Information

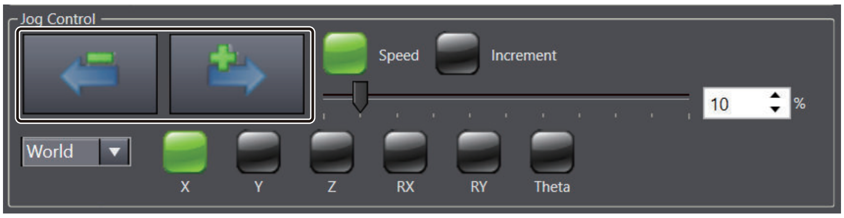
Jogging is only possible when **Ready** is displayed in the status area.
If a robot is under program control, *Robot under program control* will be displayed in this area and jogging is not possible. The task controlling the robot must be stopped before jogging functions are enabled.

Jog Control Section

You can manually position a robot in the Jog Control section.

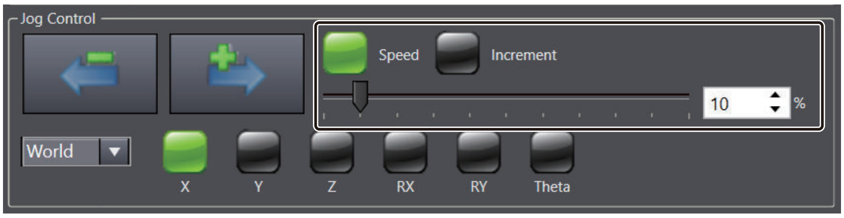
● Move Axis Buttons

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



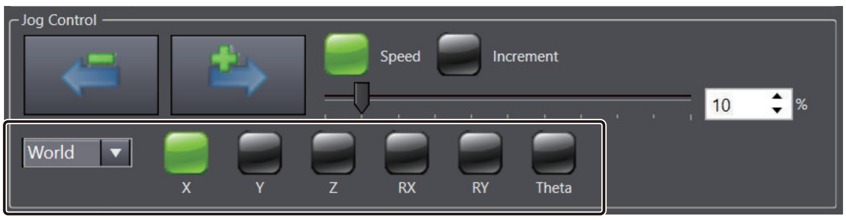
● Speed/Increment Selection Buttons

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

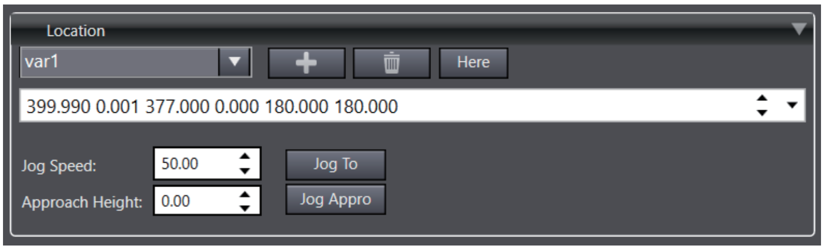
Select world, joint or tool for the JOG CONTROL mode.



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.

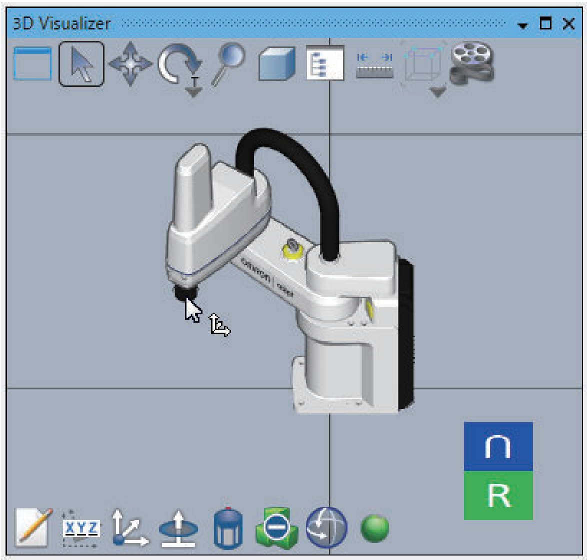
Precautions for Correct Use

- 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.

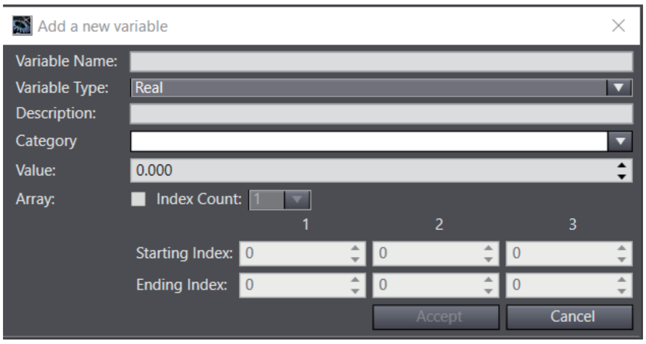
Clicking the **Here** button will put the robot's current axis positions into the display field for use in the following teach procedure.



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

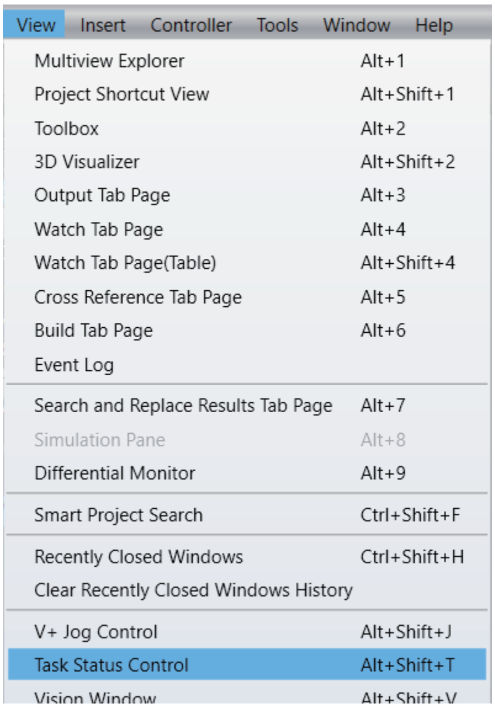
To remove an existing robot location, select the location from the drop-down menu and then click the **Delete** button. A confirmation dialog box is displayed. Click **Yes** to remove the robot location variable.

7-5 Task Status Control

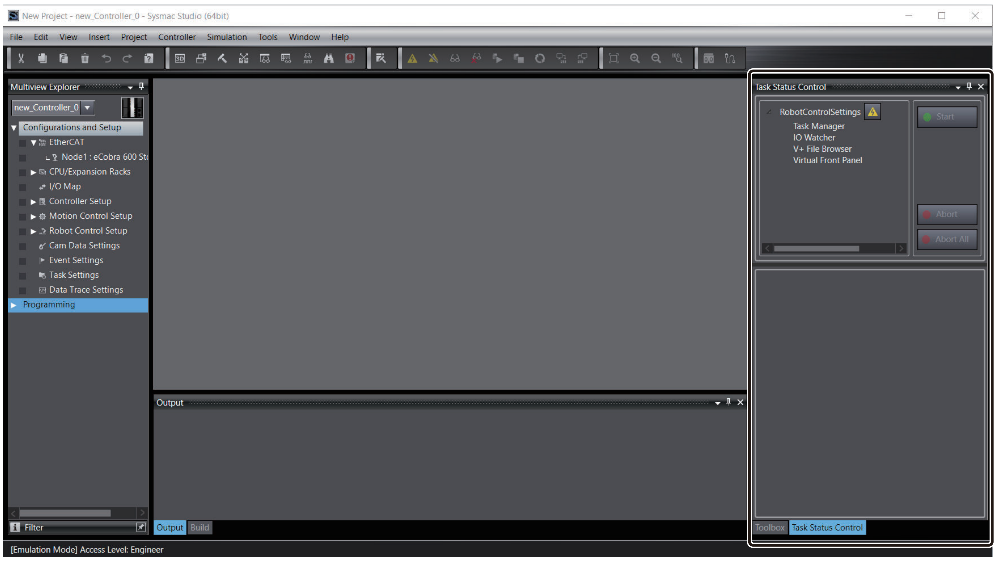
The Task Status Control provides a multi-device monitoring interface for all robot related activity in the project.
This allows you to quickly view and access all controller connection and power statuses, monitor speeds, and execution status of Vision Sequences, C# programs, Process Managers, and more.
All controllers in the project are shown in the Task Status Control Interface.
If an Application Manager device exists 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.

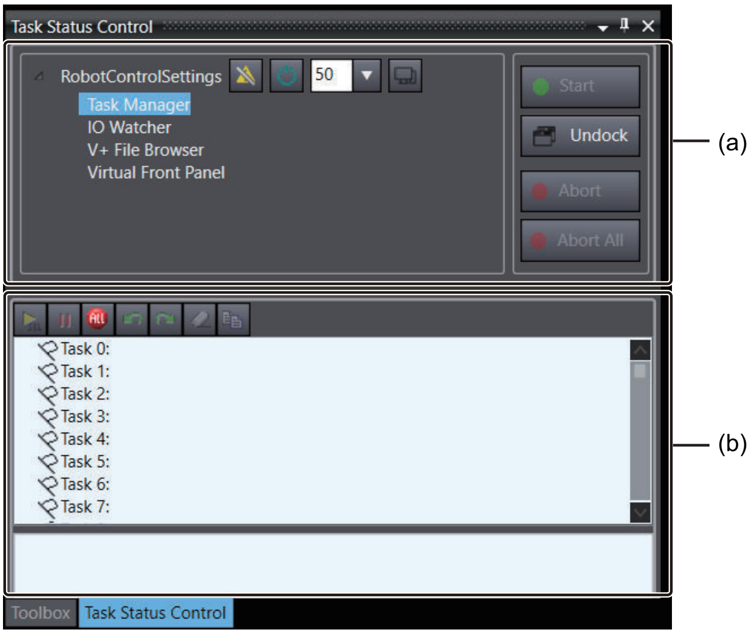


Task Status Control Tab Page is displayed.



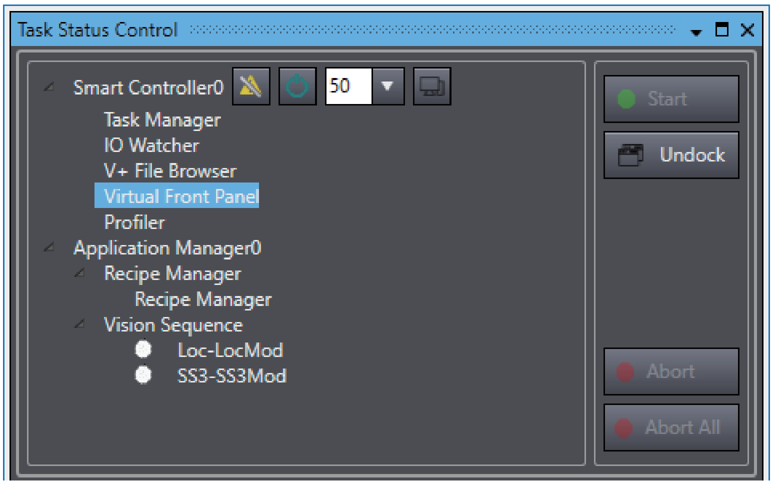
7-5-2 Task Status Control Items

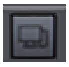
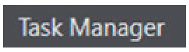
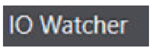
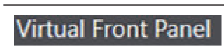
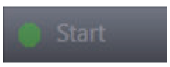
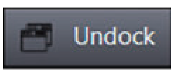
The following items are displayed in Task Status Control.

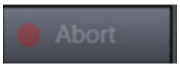
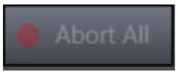


	Item	Descriptions
(a)	Item list area	Items controlled with Task Status Control Tab Page are displayed.
(b)	Description display area	Details on the Item selected in the item list area is displayed. Press the **Undock** button to show contents in this area in a different window.

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).	*Monitor Speed Setting* on page 7-33
	**Open the Monitor Window** button	Monitor Window Tab Page is displayed.	*Open the Monitor Window* on page 7-33
	Task Manager	Task Manager is displayed at the bottom	*7-5-4 Task Manager* on page 7-33
	IO Watcher	IO Watcher is displayed at the bottom.	*7-5-5 IO Watcher* on page 7-36
	V+ File Browser	V+ File Browser is displayed at the bottom.	*7-5-6 V+ File Browser* on page 7-36
	Virtual Front Panel	Virtual Front Panel is displayed at the bottom.	*7-5-7 Virtual Front Panel* on page 7-38
	Profiler	Profiler is displayed at the bottom.	*7-5-8 Profiler* on page 7-40
	Application Manager	Control the Application Manager functions. This item is shown when the Application Manager functions you want to control have been registered.	*7-5-9 Application Manager Control* on page 7-41
	Start	Execute the selected Application Manager function.	-
	Undock	Show the currently displayed view in a different window.	-

	Item	Description	Reference
	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 *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* 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.
This includes programs started by a Process Manager as shown below.

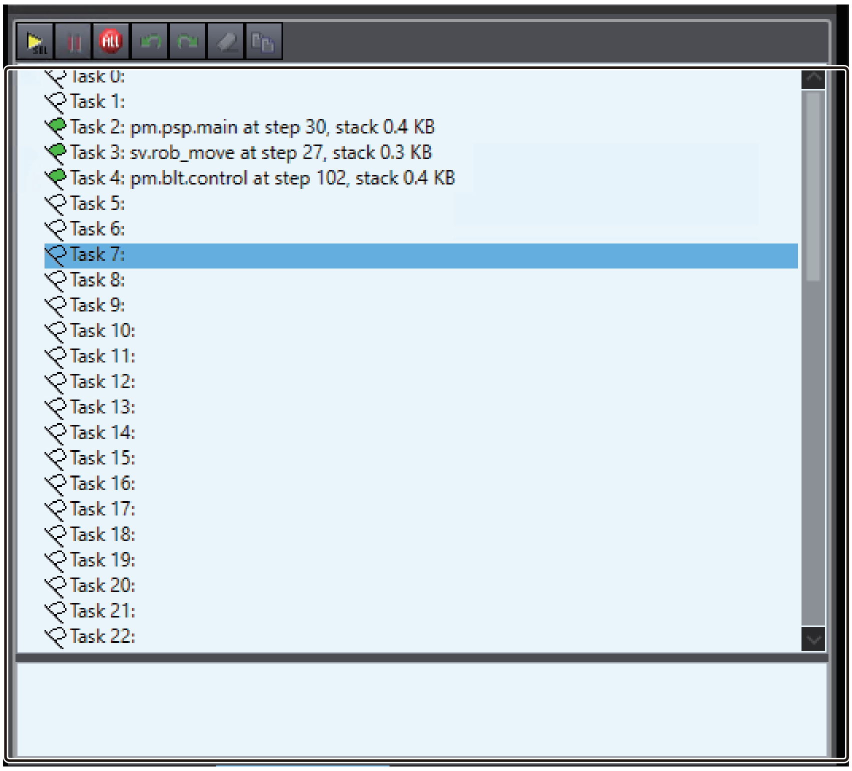
Additional Information

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 (Cat. No. 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.

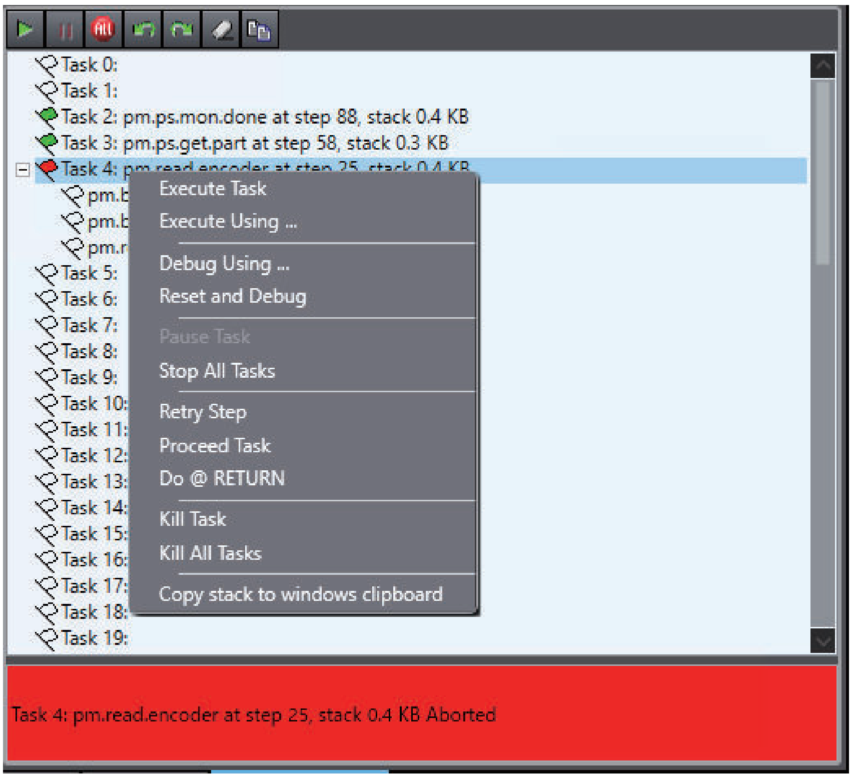
Icon	Item	Description
	Execute Task	If a stopped program is selected, this button will execute that program on the task.
		If a task is selected with no program, this button will open a dialog box for program selection. Selecting a program name and clicking **OK** will execute the program on the selected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

Other Functions

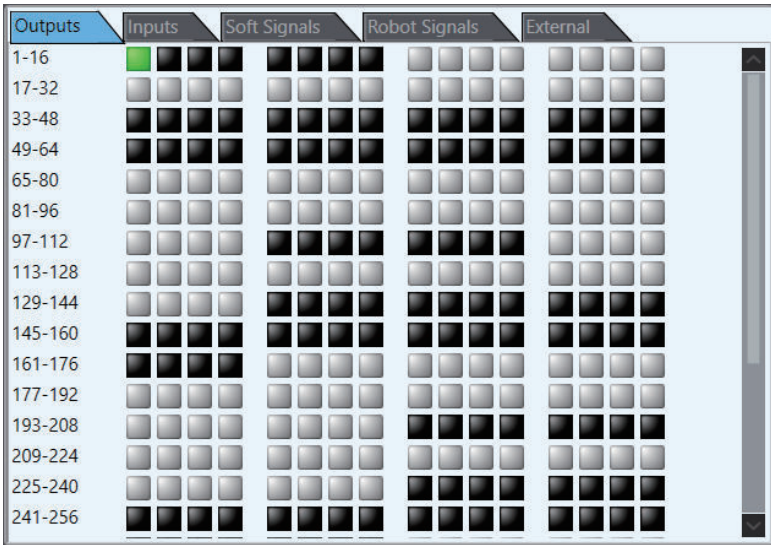
This section provides the information about other functions regarding to task execution. Right-click a task to select a function from the menu.



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 **Return** key is pressed.
Kill All Tasks	Clears all tasks.

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( ).

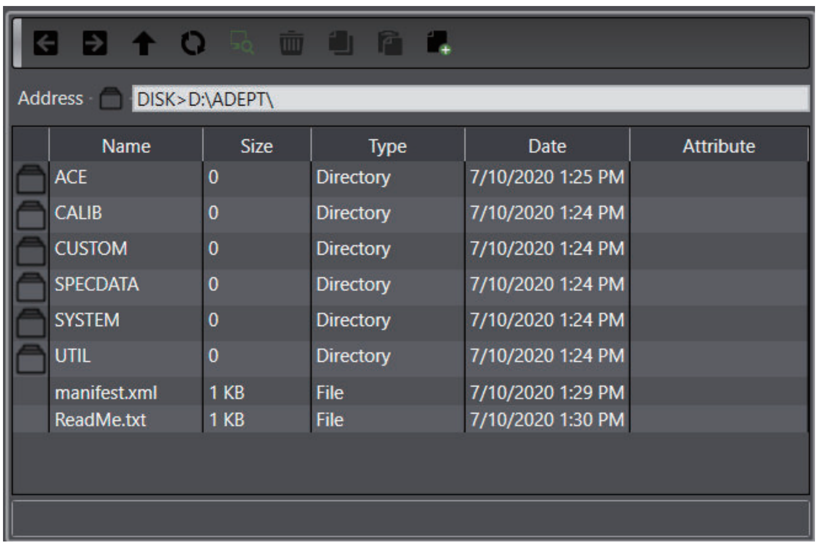


Additional Information

When the EMULATION mode is enabled, digital input signals can be manipulated.

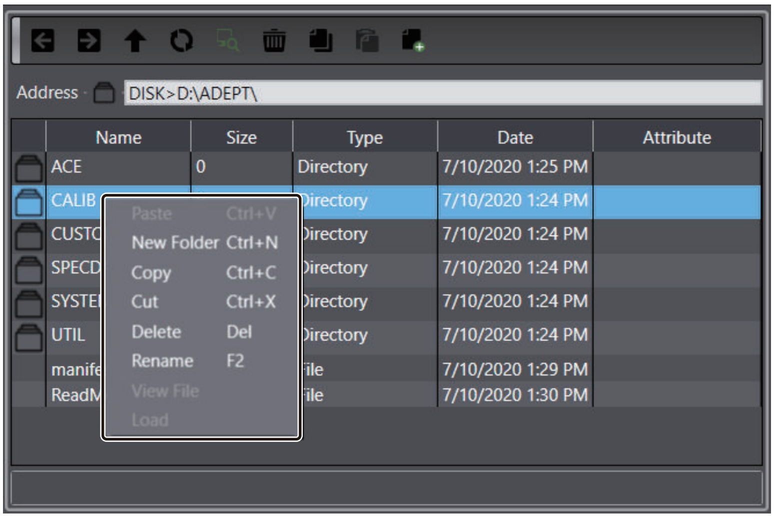
7-5-6 V+ File Browser

The V+ File Browser allows you to browse files and folders in the controller. Only available while online with the controller.



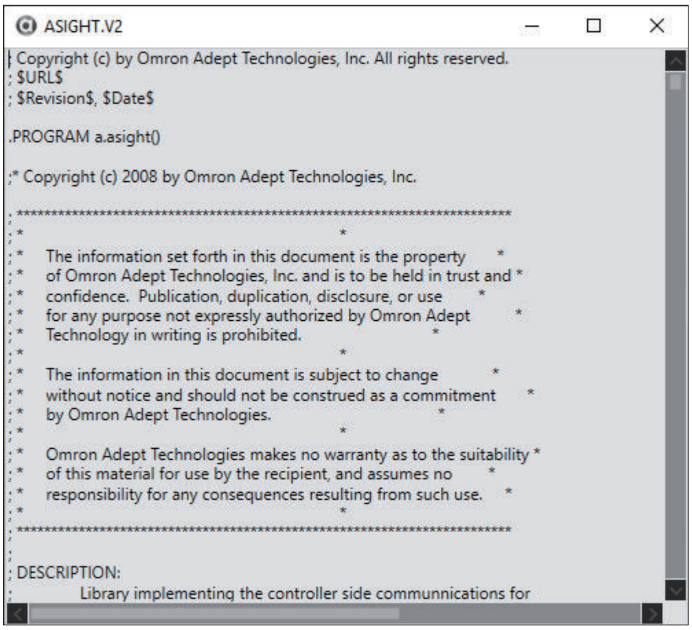
The V+ File Browser works with the Windows clipboard to enable easy transferring of files to and from controllers.

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.

⚠ 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 (Cat. No. I590)* for details.

Virtual Front Panel functions are described below.

Icon	Item
	**MODE SELECTION** switch
	MANUAL mode
	AUTOMATIC mode

Icon	Item
	**ROBOT POWER** button
	**E-Stop** 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 (Cat. No. I652)* for details.

7-5-8 Profiler

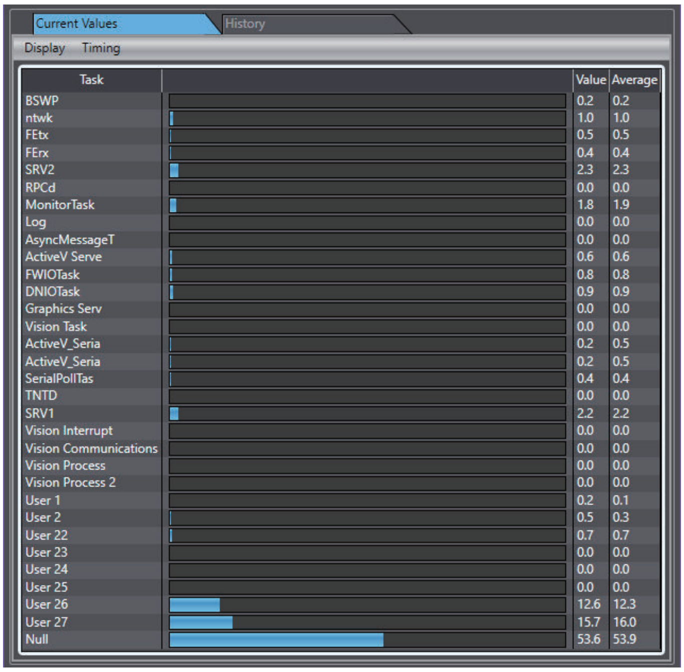
The Profiler is used to provide a graphical dis- play of controller processor usage for diagnostic purposes. It is available for each controller in the project.

Precautions for Correct Use

The Profiler function is not available in the Emulation mode.

Current Values Tab

The **Current Values** tab shows a list of tasks and their respective processor usage.
Use the **Display** and **Timing** menu items to adjust the listed items and the update rate.

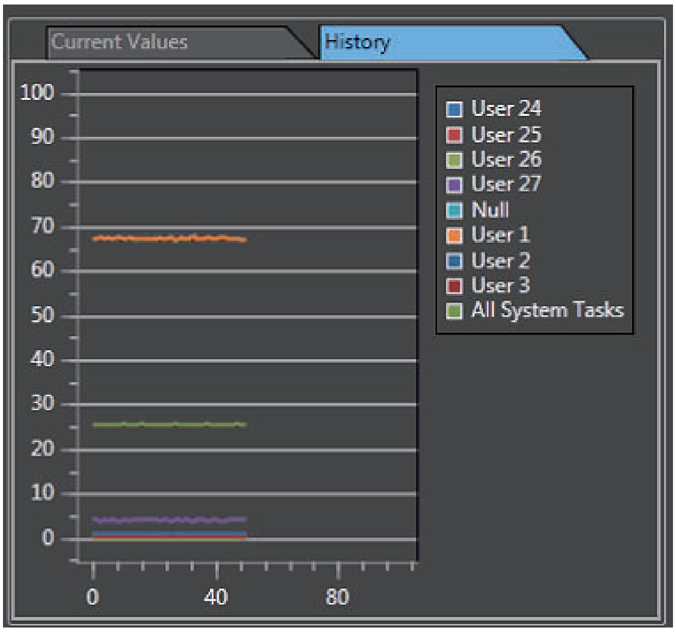


Additional Information

All User Tasks displays all the user tasks available to your system. If **All User Tasks** is not selected, only tasks with a program on the execution stack are displayed.

History Tab

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



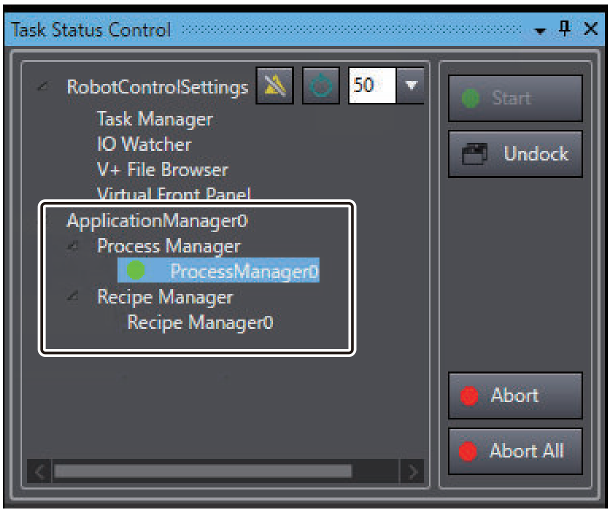
7-5-9 Application Manager Control

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 (Cat. No. 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 manual.

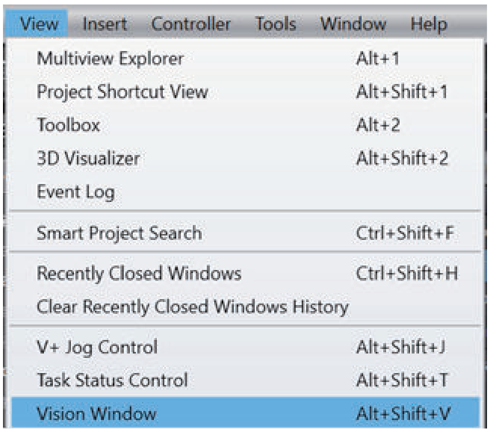

Task Status Control
RobotControlSettings
50
Task Manager
IO Watcher
V+ File Browser
Virtual Front Panel
ApplicationManager0
Process Manager
ProcessManager0
Recipe Manager
Recipe Manager0
Start
Undock
Abort
Abort All

7-6 Vision Window

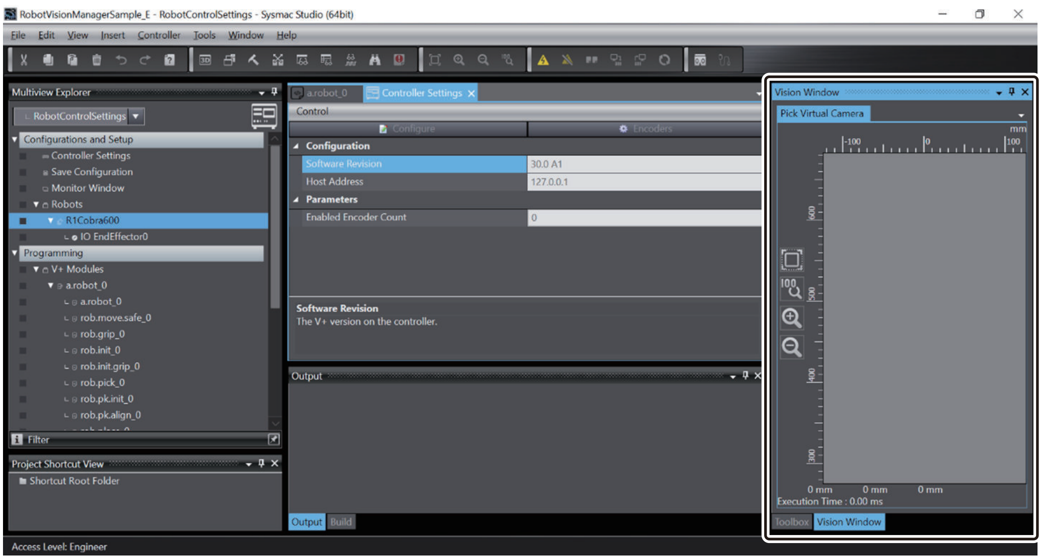
The Vision Window displays the input for each image source defined in the system. Using a tab for each source, you can view various cameras and tools configured in the project. Most vision tools can display their results in this window, allowing you to troubleshoot, test, or monitor the system performance 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* on page 5-182 for more information.

7-6-1 Showing the Vision Window

1 Select *RobotControlSettings* from the device list in the Multiview Explorer or *Application Manager*, then select **View** – **Vision Window** from the main menu.

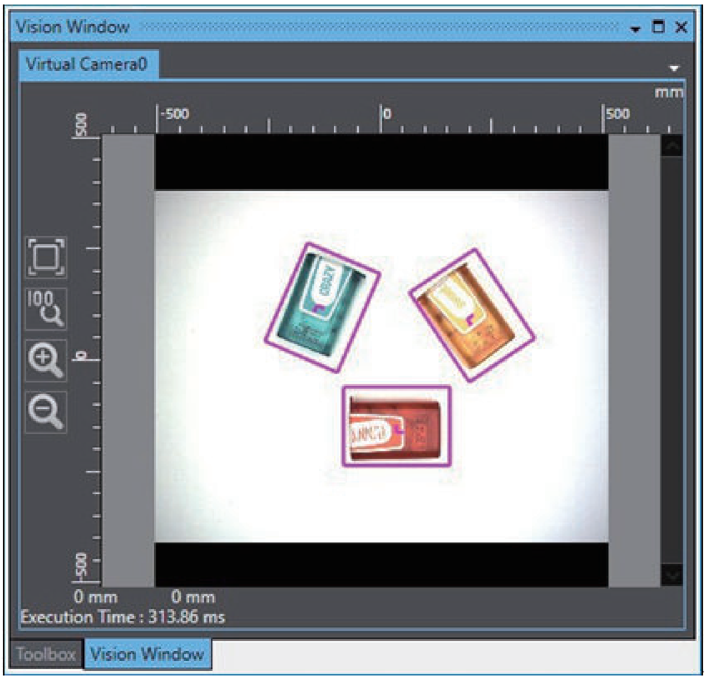


The **Vision Window** is shown.



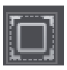
7-6-2 Vision Window Functions

This section describes functions of the Vision Window.



Zoom Level

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

	Icon	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

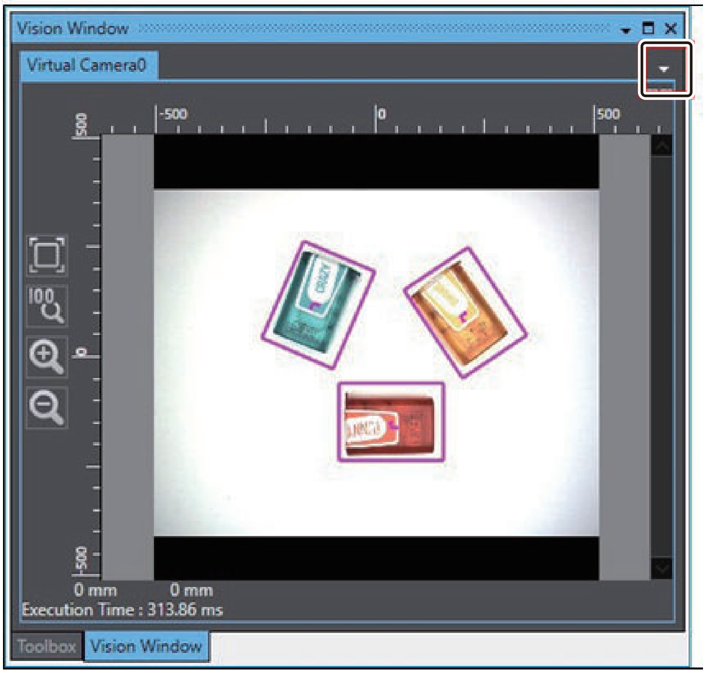
The left and top axis values represent the calibration scale setting in mm/pixel that is present in the Virtual Camera settings. Refer to *Virtual Camera Calibration* on page 5-49 for more information.

Execution Time

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

Camera Selection

Use the drop-down arrow to select from all available Image Sources defined in the system.

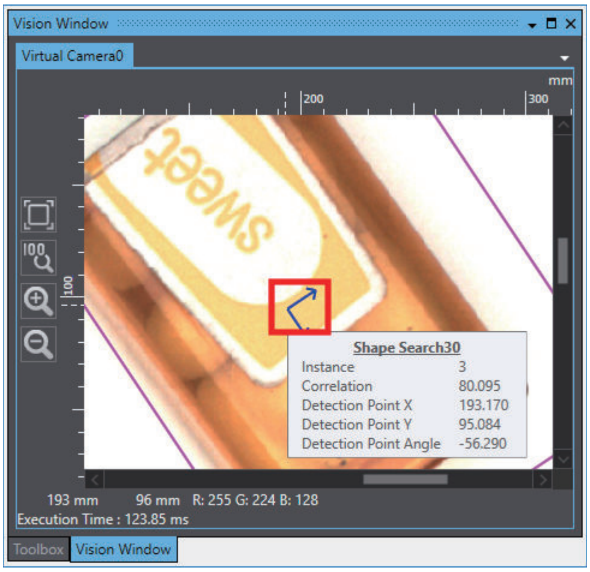


Cursor Information

Moving the cursor in the field of view portion of the Vision Window will reveal additional information about the inspection results.

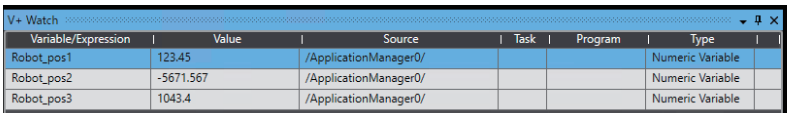
The X-Y coordinates are displayed at the bottom of the Vision Window for the current cursor position. Color/gray scale values are also displayed when applicable.

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 function to monitor C# variable objects.
You can use the V+ Watch to monitor specific variables during developing or debugging a C# program. Variables can be added to V+ watch tab page in different ways as described in the following. Contents in the V+ Watch tab page will be saved together with a project.


V+ Watch

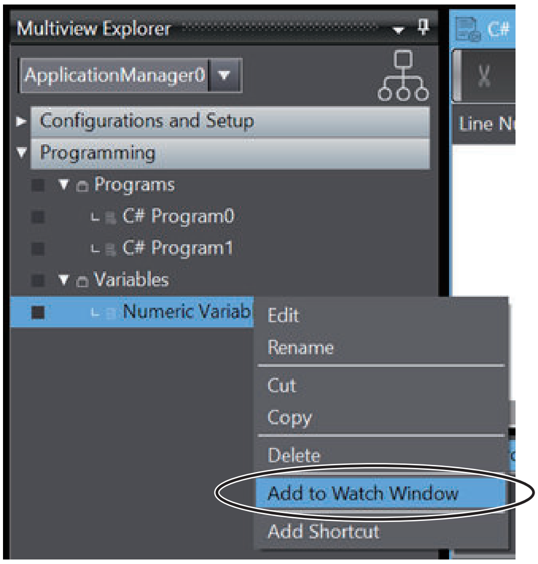
Variable/Expression	Value	Source	Task	Program	Type
Robot_pos1	123.45	/ApplicationManager0/			Numeric Variable
Robot_pos2	-5671.567	/ApplicationManager0/			Numeric Variable
Robot_pos3	1043.4	/ApplicationManager0/			Numeric Variable

⚠ 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 V+ Watch Tab Page

Use the following method to add C# variable objects to the V+ Watch tab page.
Right-click a C# variable object in the Multiview Explorer. Selecting **Add to Watch Window** will add these variables to the V+ Watch tab page.



8

Application Sample Wizard

This section describes how to build a robot integrated system that includes the IPC Application Controller, by using the Application Sample Wizard.

8-1 Overview of the Application Sample Wizard

The Application Sample Wizard creates the pick-and place type Robot Integrated System templates including Robot Integrated CPU Units and IPC Application Controllers.
You can easily create templates only following the wizard: select peripherals for the system and configure settings.

The Application sample wizard consists of the following wizards.

Name	Description
Pack Manager Sample	Pick-and place system templates for packaging machines can be created with a Pack Manager running on the IPC Application Controller. You do not need V+ language because the created Application Samples consist of only the settings of a Pack Manager object, not of codes. The Pack Manager on the IPC Application Controller and the Robot Integrated CPU Unit Integrated CPU Unit control a robot together.
Robot Vision Manager Sample	Pick-and place system templates for packaging machines can be created with a Robot Vision Manager on the IPC Application Controller and V+ program running in 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 system templates consist of various part detection methods and pick-and-place methods by using Application Sample Wizards for the Pack Manager Sample and the Robot Vision Manager Sample. An Application Sample is created combining the following system elements.

- Part detection methods
- Necessity for refining a picked part position in the gripper
- Methods to determine part placement position

For example, items of the Pack Manager Application Sample Wizard are provided in this section.
Items of the Robot Vision Manager Application Sample Wizard are the same, though details differ.

Part Detection Methods

The following types of part detection methods are available.

Type	Description
At a fixed position	Parts at a fixed position are detected.
At a fixed pallet	Parts on pallets at a fixed position are detected.
On a belt located with a camera	A fixed camera detects parts on the conveyor belt.
With an arm- or fixed- mounted camera	A camera mounted on the robot arm detects parts.
With a belt latch sensor	A latch sensor on the conveyor belt detects parts on the conveyor belt.
At a pallet located by a belt latch sensor	A latch sensor on the conveyor belt detects pallets carrying parts on the conveyor belt.

How to Determine the Part Placement Position

The following methods are available to determine a part placement position.

Type	Description
At a fixed position	Parts are placed at a fixed position.
At a fixed pallet	Parts are placed on a pallet at a fixed position.
On a belt located with a camera	A fixed camera detects a position on the conveyor belt where a part will be placed.
With an arm- or fixed- mounted camera	A camera mounted on the robot arm detects a position where a part will be placed.
With a belt latch sensor	A latch sensor on the conveyor belt detects a part placement position on the belt.
At a pallet located by a belt latch sensor	A latch sensor on the conveyor belt detects a pallet position on the belt where a part will be placed.

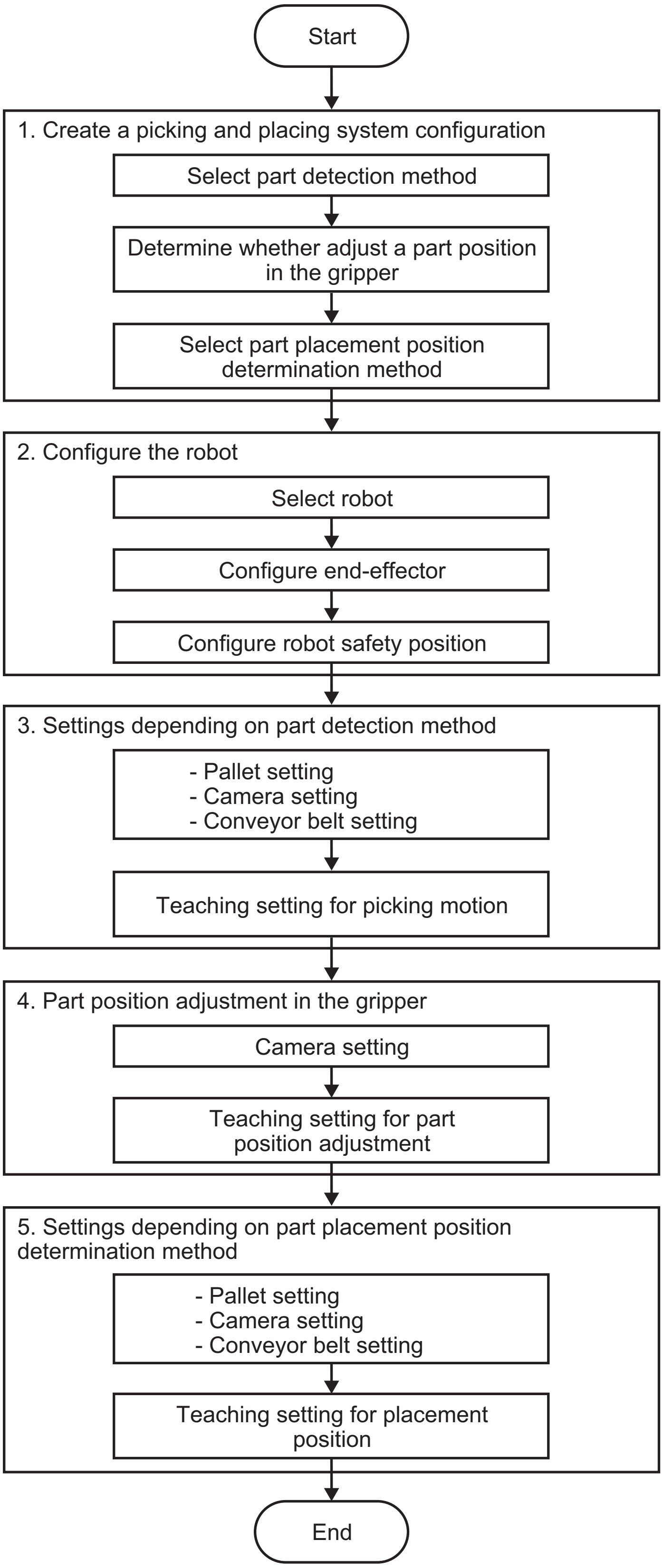
Refining a Part Position in the Gripper

You can select whether to refine a position of the picked up part in the gripper.

Type	Description
Not refine the part position	A part is held in the position as it was picked up.
Refine the part position using a camera	A picked up part is shot by a camera from underneath, and with that image, the position with respect to the robot arm tip is refined.

8-2-2 Workflow with the Application Sample Wizard

This section shows a workflow taking an example of the Pack Manager Application Wizard. A workflow of the Robot Vision Manager Application Sample is the same.



1 Creation of a pick-and-place system configuration

To build a pick-and place system configuration, select elements such as part detection method, necessity of part position refinement in the gripper, and method to determine part placement position.

Work	Description	Reference
Selecting a part detection method	Select a part detection method when the robot picks up	*Selecting the Part Detection Method* on page 8-7
Selecting whether to refine the part position in the gripper	Determine whether to refine a position of the picked up part in the gripper.	*Correcting the Part Position in the Gripper* on page 8-10
Selecting a method to determine part placement position	Select a method to determine a position where the robot places the part.	*Selecting a Method to Determine Part Placement Position* on page 8-10

2 Setup of the robot

Select the robot to use and set the safe position of the robot.

Work	Description	Reference
Selecting the robot	Select the robot to use in the Application Sample.	*Selecting the Robot* on page 8-12
Setting up the End-Effector	Set the properties of the End-Effector tips.	*Setting Up the End-Effector* on page 8-13
Setting the safe position of the robot	Set the safe position of the robot.	*Setting a Safe Position for the Robot* on page 8-14

3 Settings for the selected part detection method

Teach the robot to pick up a part.

Work	Description	Reference
• Setting up the pallet • Setting up the camera • Setting up the conveyor belt	Make the following settings for the selected part detection method in *Selecting the Part Detection Method* on page 8-7: pallet, camera, and conveyor belt.	*Teaching the Pick-up Operation* on page 8-15
Teaching the pick-up operation	Teach the robot to pick up a part.	

4 Part position refinement in the gripper

In *Correcting the Part Position in the Gripper* on page 8-10, teach the robot to refine the part position if you have selected to adjust the position of the picked-up part in the gripper.

Work	Description	Reference
Setting up the camera	Set up the camera.	*Correcting the Part Position in the Gripper* on page 8-10
Teaching part position refinement	Teach the robot to refine the part position.	

5 Settings for the selected part placement determination method

Teach the robot where to place the part.

Work	Description	Reference
• Setting up the pallet • Setting up the camera • Setting up the conveyor belt	Make the following settings for the selected part placement determination method in *Selecting a Method to Determine Part Placement Position* on page 8-10: pallet, camera, and conveyor belt.	*Teaching a Placing Position* on page 8-22
Teaching the placement position	Teach the robot where to place the part.	

8-2-3 Starting the Application Sample Wizard

With an example of picking a part in a belt-camera system through a SCARA Robot (Cobra 600), the creation procedure for a Pack Manager sample application is described.
For a use of the Application Sample Wizard, Sysmac Studio must be onlined to the Simulator or a physical Robot Integrated CPU Unit.
In the following sections, the procedure will be described assuming that you are going online with the Simulator.

Additional Information

- For preparation, register a robot you use to the EtherCAT network for the Robot Integrated CPU Unit. Refer to the *4-2-1 Creating a New Project* on page 4-4 for the registration procedure.
- To create a Robot Vision Manager sample application, select **Insert** from the menu, then select **Application Sample** - **Robot Vision Manager Sample**.

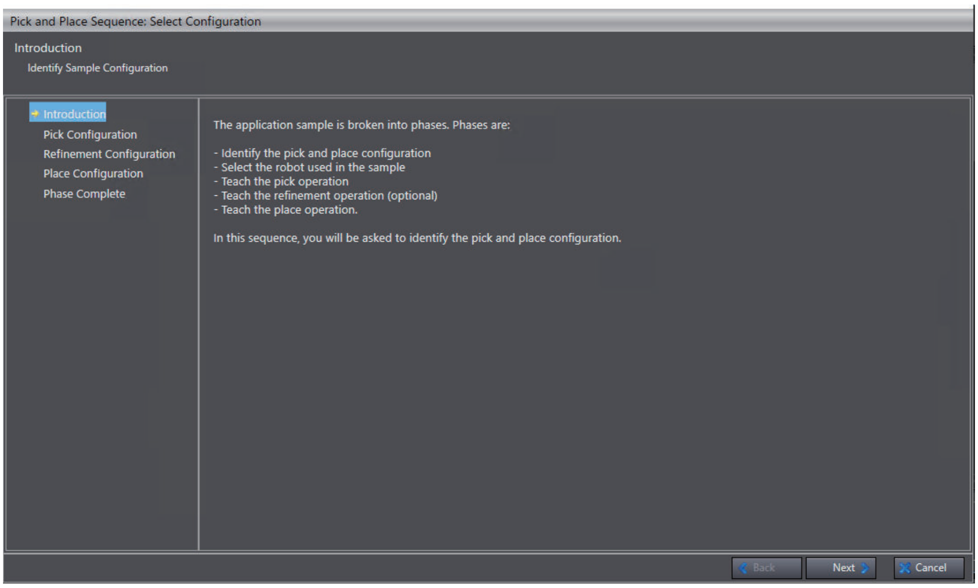
Creating a Pick-and-place System Configuration

This section describes how to create a pick-and-place system configuration by selecting system elements, such as how to detect the workpiece, whether or not to correct the position of the picked-up workpiece in the gripper, and how to determine the workpiece placement position.

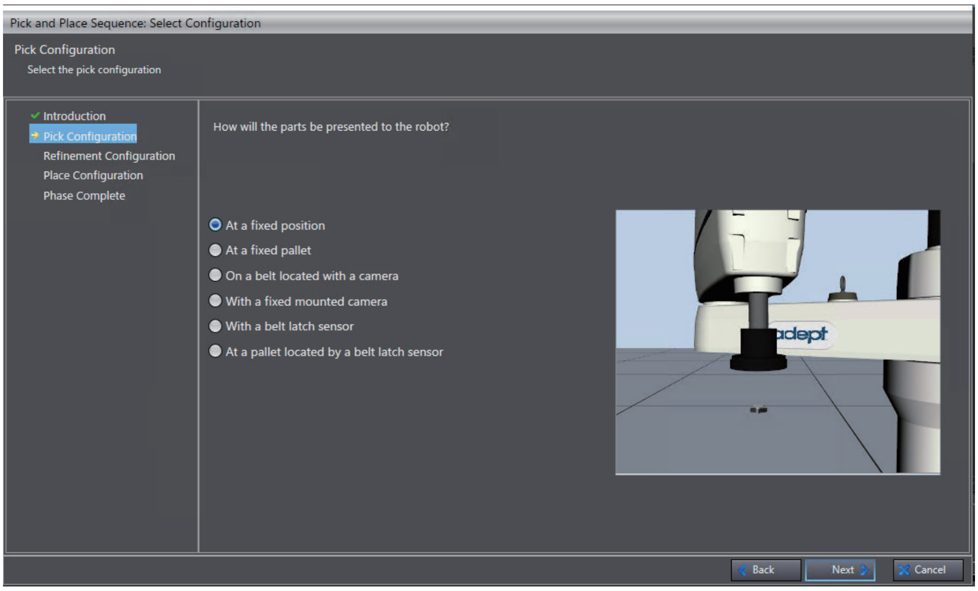
● Selecting the Part Detection Method

Select a part detection method when the robot picks up

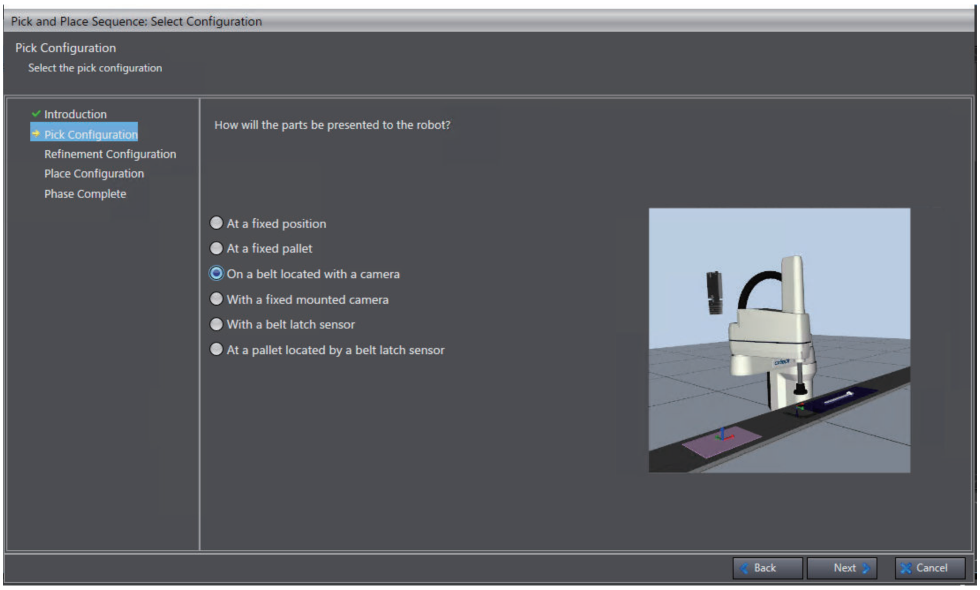
1 Select **Insert** from the menu, then select **Application Sample** - **Pack Manager Sample**.
The Application sample wizard starts and the **Introduction** page of the wizard appears.



2 Click the **Next** button.
The **Pick Configuration** page appears.



3 Select **On a belt located with a camera** to detect parts on the conveyor belt with a fixed camera.

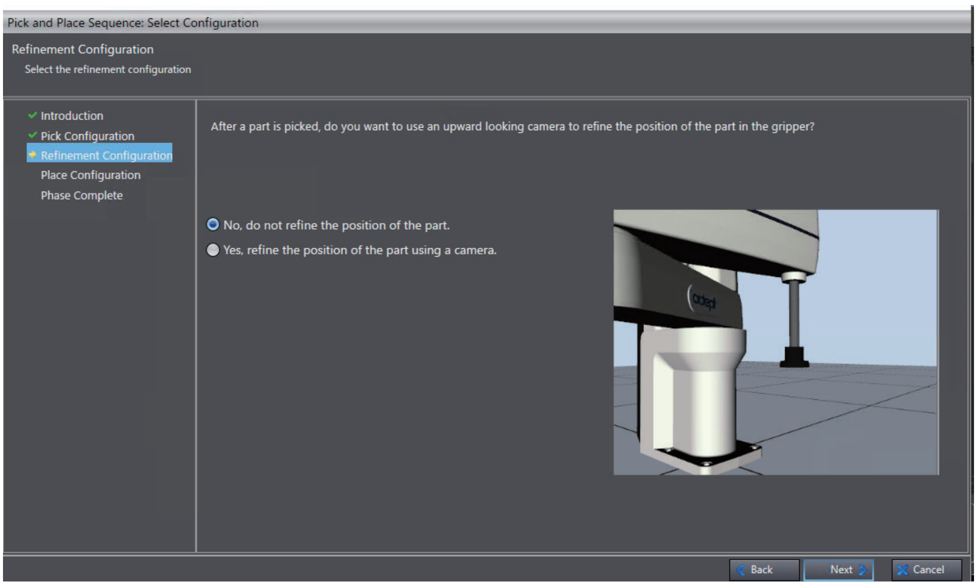


The following types of part detection methods are available.

Type	Description
At a fixed position	Parts at a fixed position are detected.
At a fixed pallet	Parts on pallets at a fixed position are detected.
On a belt located with a camera	A fixed camera detects parts on the conveyor belt.
With a fixed mounted camera	A camera mounted on the robot arm detects parts.
With a belt latch sensor	A latch sensor on the conveyor belt detects parts on the conveyor belt.
At a pallet located by a belt latch sensor	A latch sensor on the conveyor belt detects pallets carrying parts on the conveyor belt.

4 Click the **Next** button.

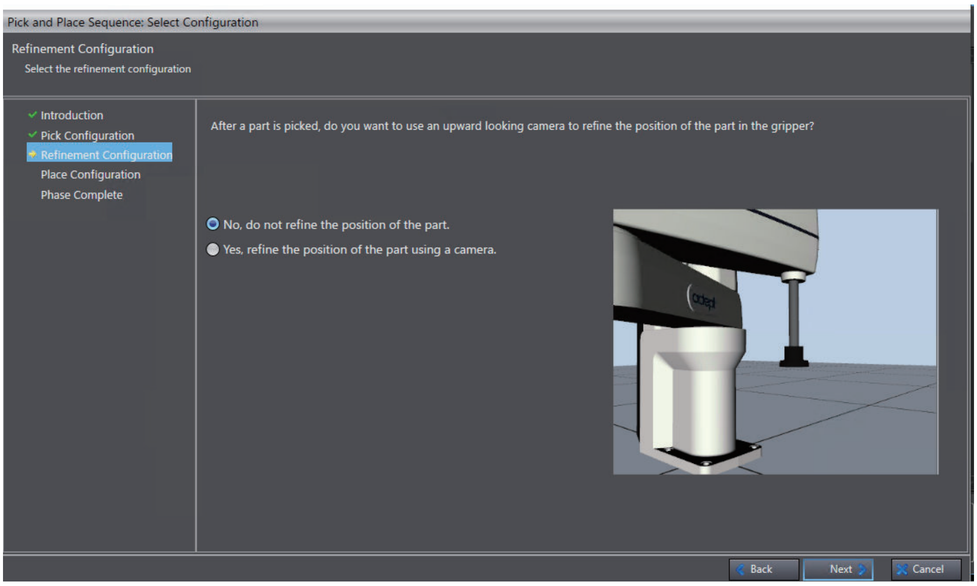
The **Refinement Configuration** pane is displayed. Select whether to refine the position of the picked-up part in the gripper. Refer to *Correcting the Part Position in the Gripper* on page 8-10 for part position refinement in the gripper.



● Correcting the Part Position in the Gripper

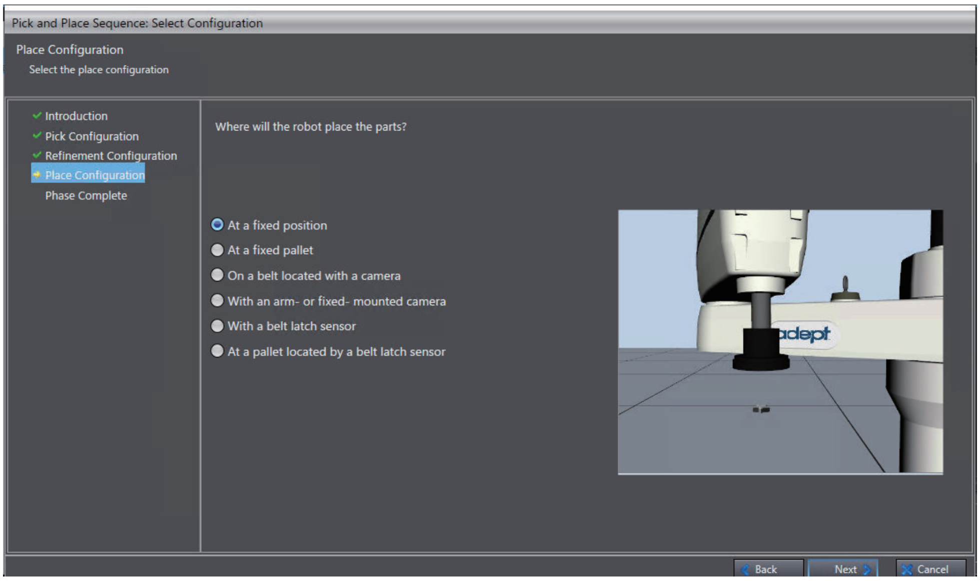
You can select whether to refine a position of the picked-up part in the gripper.

1 Select **No, do not refine the position of the part**, the option that makes no position refinement in the gripper.



If you want to refine the part position in the gripper with a camera, select the option **Yes, refine the position of the part using a camera**.

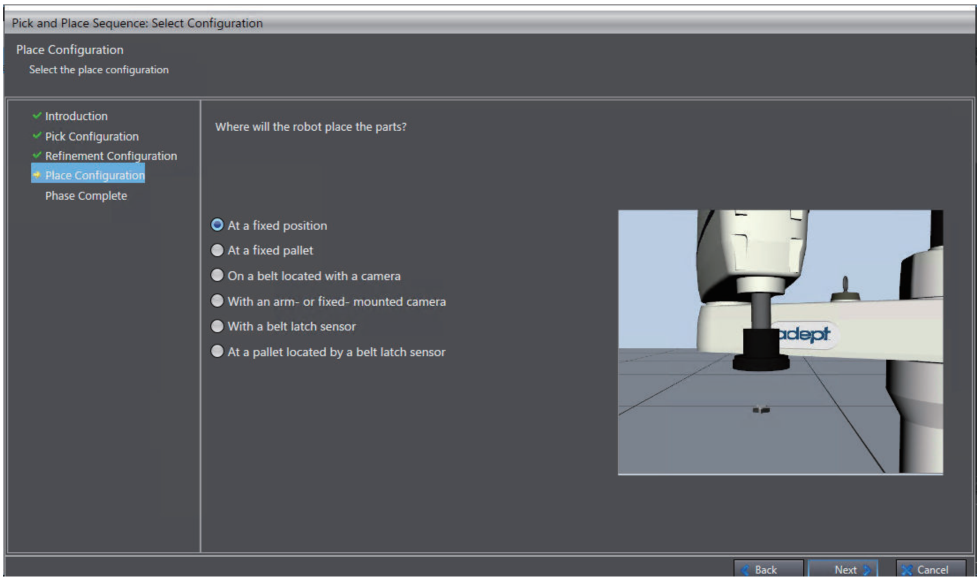
2 Click the **Next** button.
The **Place Configuration** pane is displayed. Select a method to determine a position where the robot places the part. Refer to *Selecting a Method to Determine Part Placement Position* on page 8-10 for determining a part placement position.



● Selecting a Method to Determine Part Placement Position

Select a method to determine a position where the robot places a part.

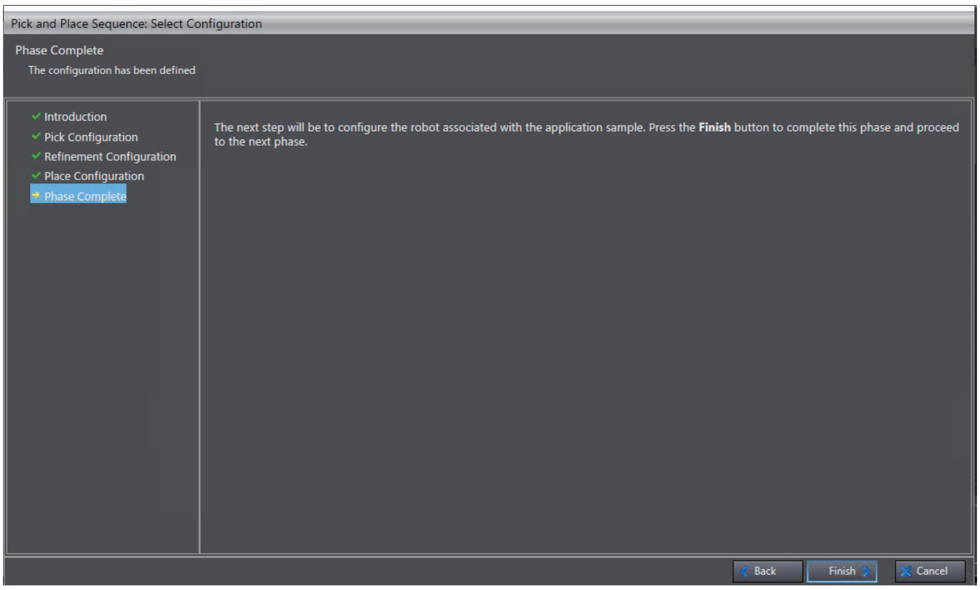
1 Select **At a fixed position** to always place the part at a fixed position.



The following methods are available to determine a part placement position.

Type	Description
At a fixed position	Parts are placed at a fixed position.
At a fixed pallet	Parts are placed on a pallet at a fixed position.
On a belt located with a camera	A fixed camera detects a position on the conveyor belt where a part will be placed.
With a fixed mounted camera	A camera mounted on the robot arm detects a position where a part will be placed.
With a belt latch sensor	A latch sensor on the conveyor belt detects a part placement position on the belt.
At a pallet located by a belt latch sensor	A latch sensor on the conveyor belt detects a pallet position on the belt where a part will be placed.

2 Click the **Next** button.
The **Phase Complete** pane appears.



This completes the creation of a pick-and-place system configuration.
Click the **Finish** button to proceed to the setup of the robot to use. Refer to *Setting UP a Robot* on page 8-12 for setting up the robot.

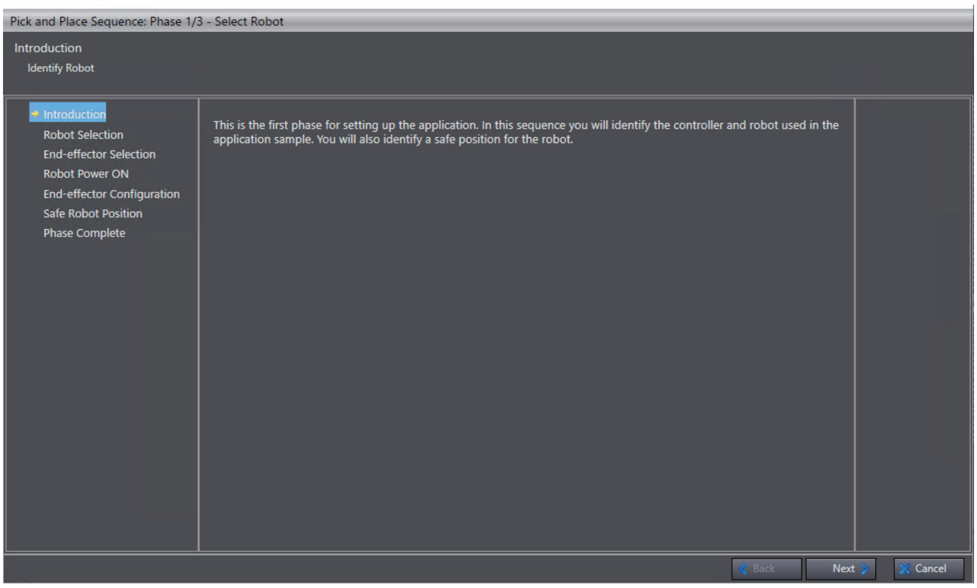
Setting UP a Robot

This section describes how to select the robot to use and set the Safe Robot Position.

● Selecting the Robot

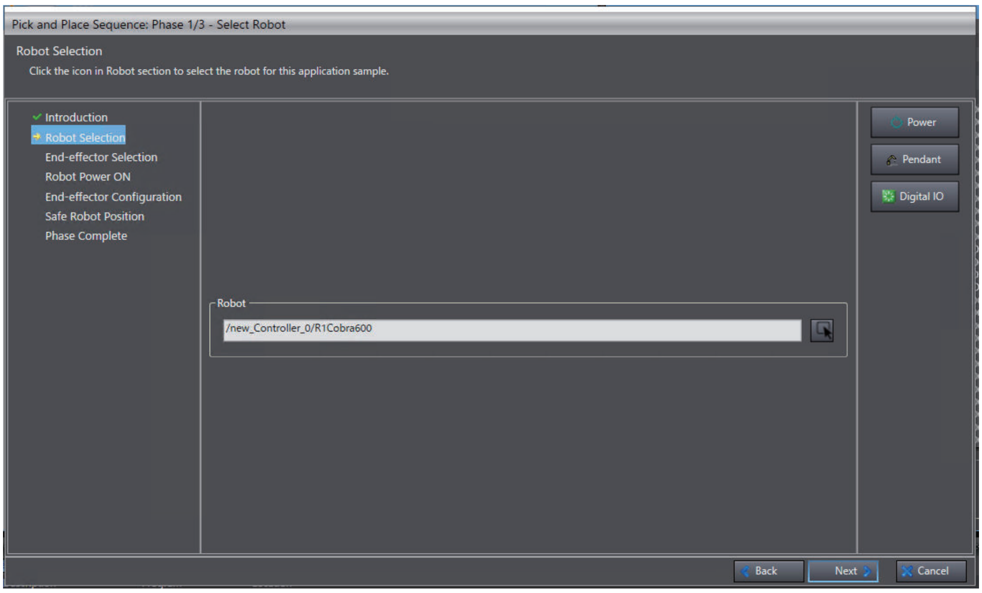
Select the robot to use in the Application Sample.

1 On the **Introduction** page of the Robot Setup wizard, click the **Next** button.

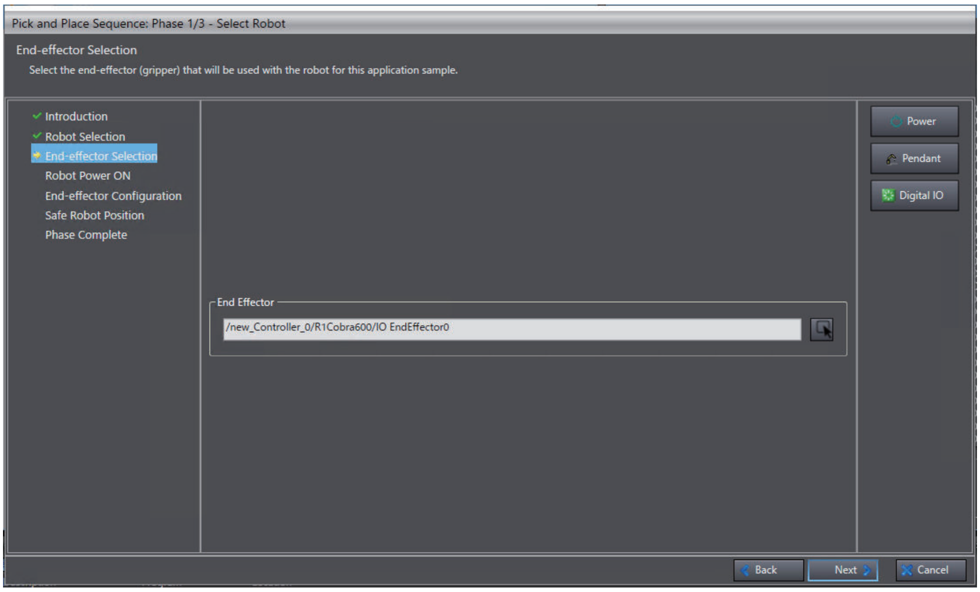


The **Robot Selection** page is displayed.

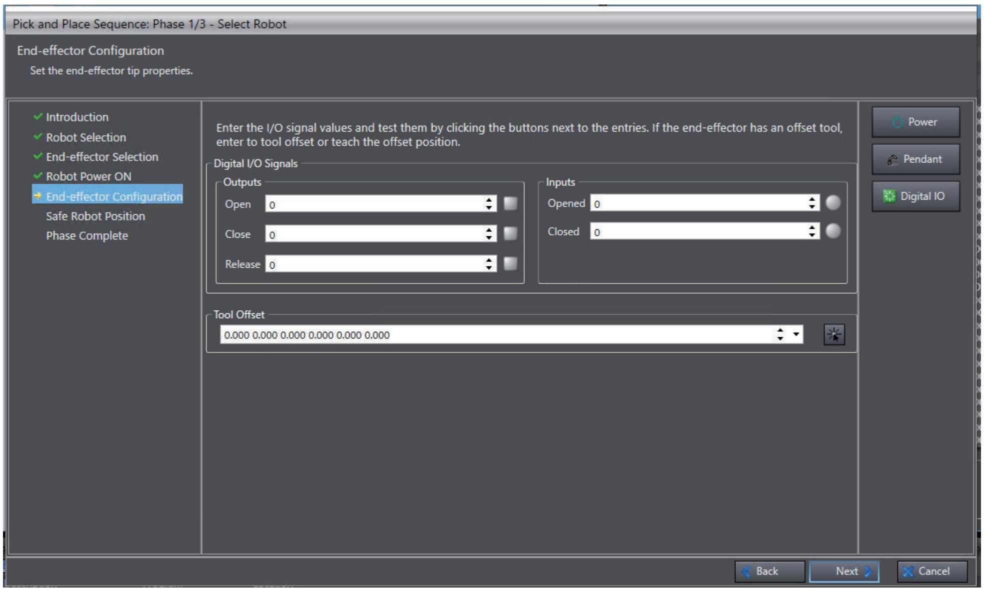
2 Specify the robot to use, and then click the **Next** button.



The **Select End-Effector** dialog is displayed.



3 Select an end effector, then click the **Next** button.
The **End-effector Configuration** dialog appears to let you configure the end effector.

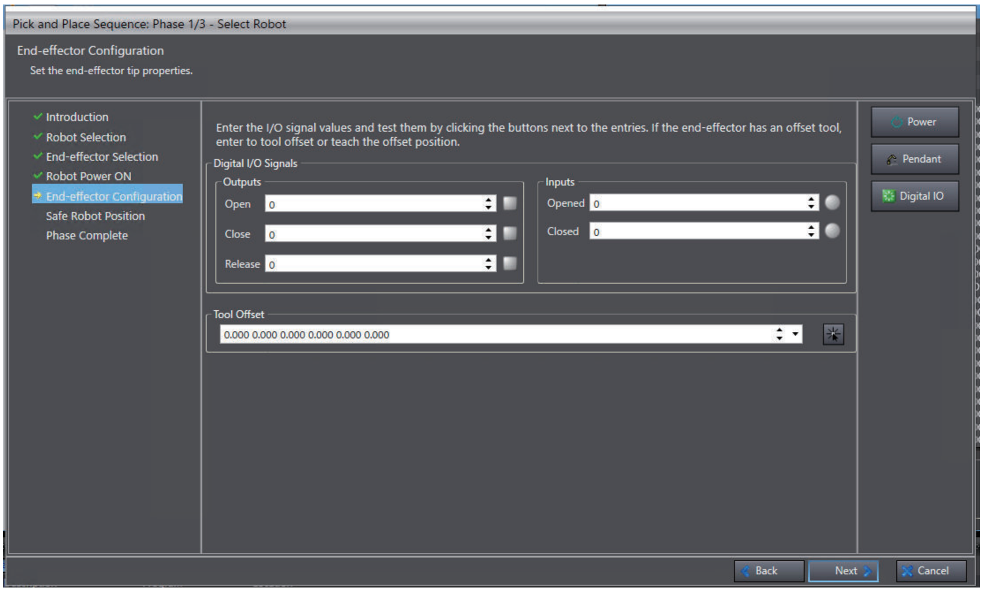


Refer to *Setting Up the End-Effector* on page 8-13 for more information about end effector configuration.

● Setting Up the End-Effector

Set the properties of the End-Effector tips.

1 Set the digital I/O signals to operate the End-Effector.



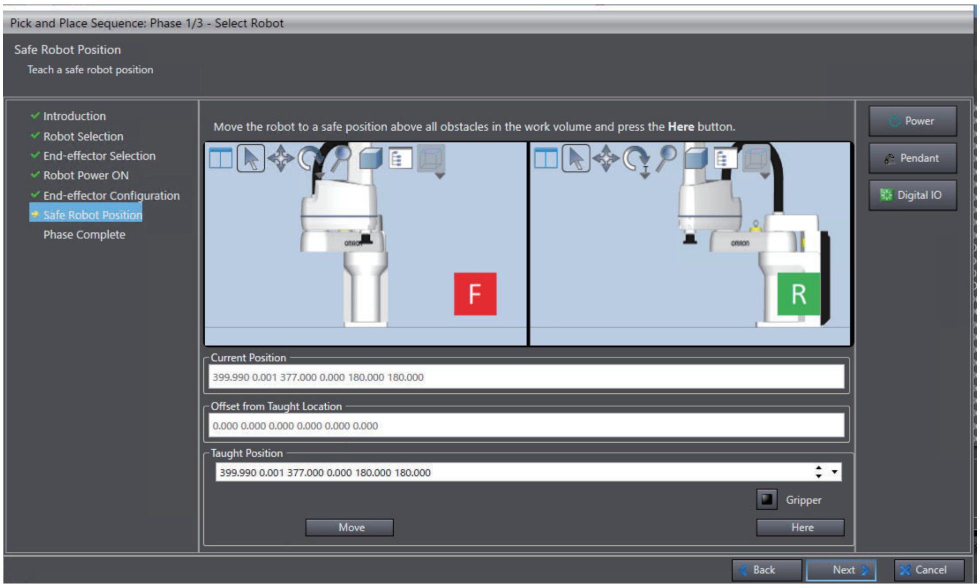
Here, leave the default settings as they are because we do not set up the End-Effector.

2 Click the **Next** button.
The **Safe Robot Position** page appears, which allows you to set up the safe position of robot arms. Refer to *Setting a Safe Position for the Robot* on page 8-14 for setting up the safe position of robot arms.

● Setting a Safe Position for the Robot

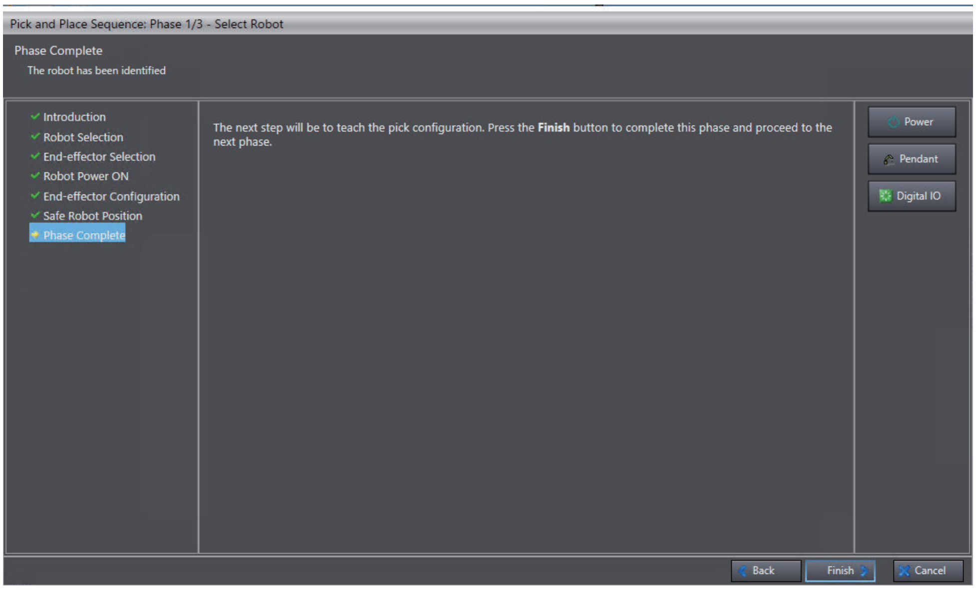
Set the safe position of the robot.

1 Set the safe position of the robot arm tips.

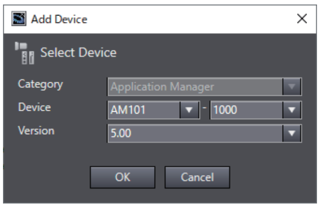


You can modify a robot position by clicking and dragging the robot in the robot view area.

2 After moving the robot to a safe position, click the **Here** button to determine the position, and then click the **Next** button.
The **Phase Complete** pane appears.



Clicking the **Finish** button will display the **Add Device** dialog box.

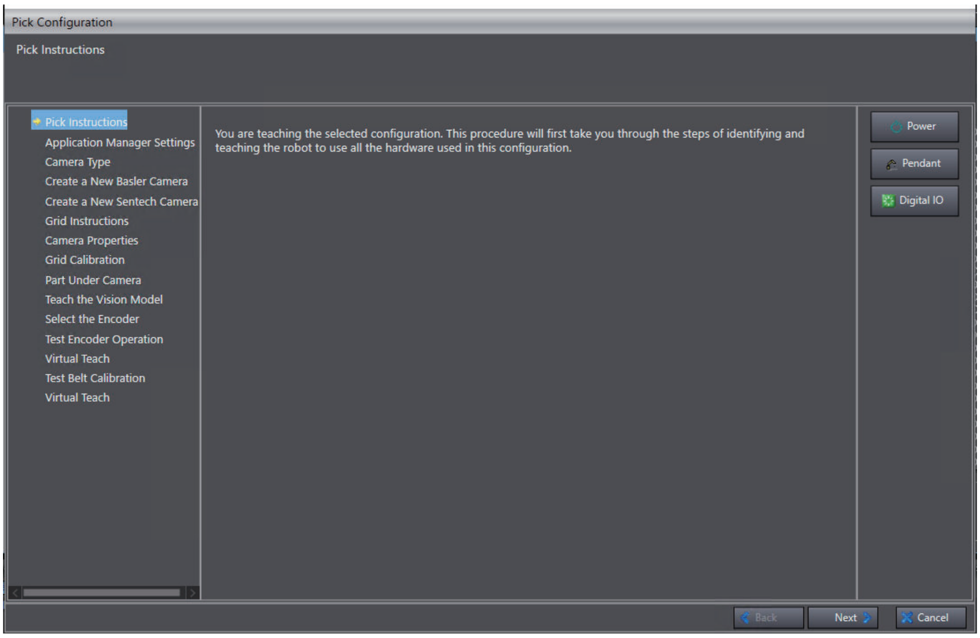


Select the device and version of the Application Manager in **Device** and **Version**, and then click the **OK** button.
Now you have finished configuring the basic robot settings.
In the next procedure, you will teach the robot to pick up a part. Refer to *Teaching the Pick-up Operation* on page 8-15 for teaching the pick-up operation.

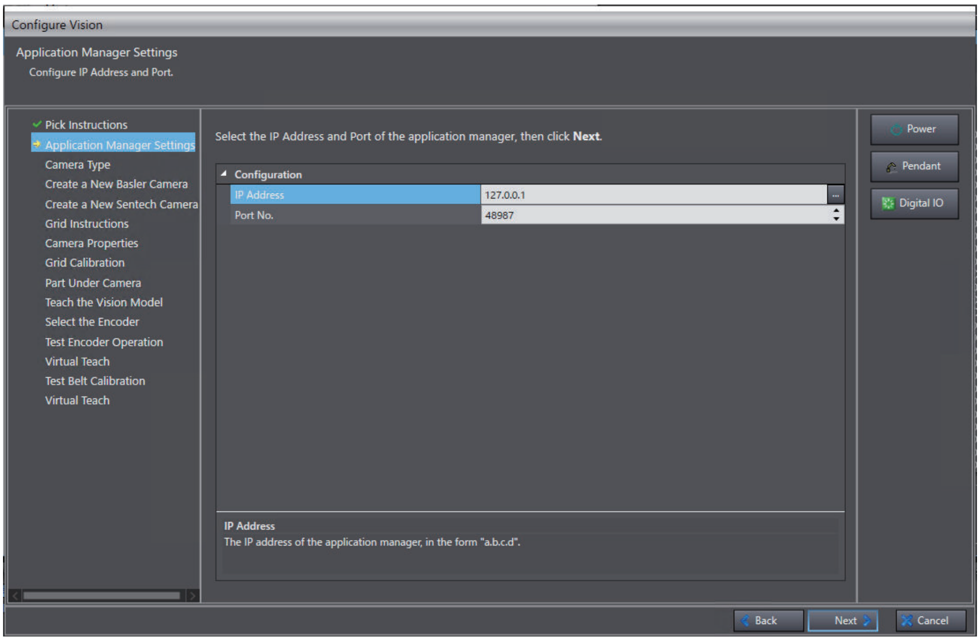
Teaching the Pick-up Operation

Teach the robot to pick up a part.

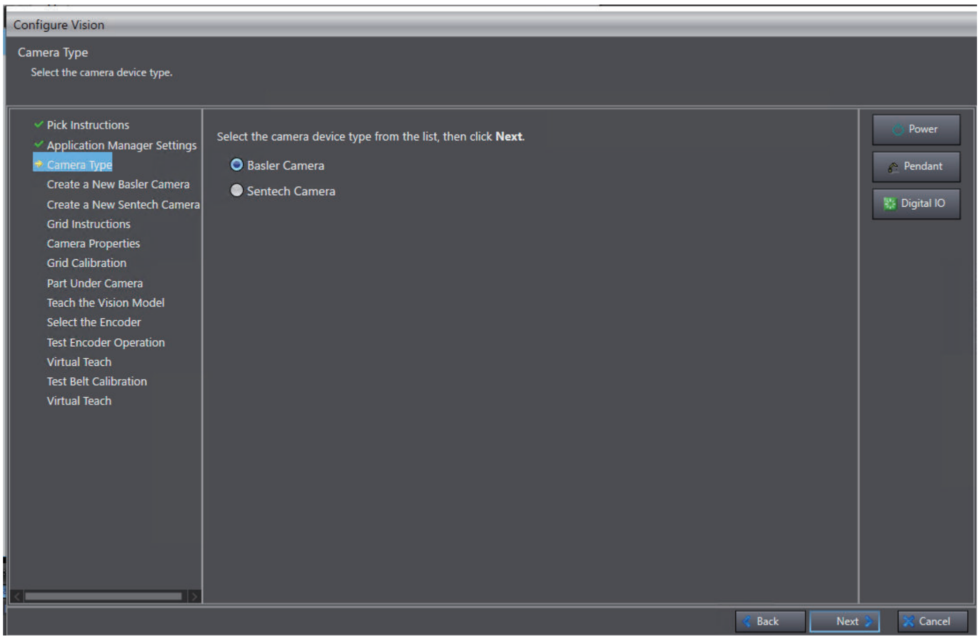
1. On the **Introduction** page of the Pick Configuration wizard, click the **Next** button.



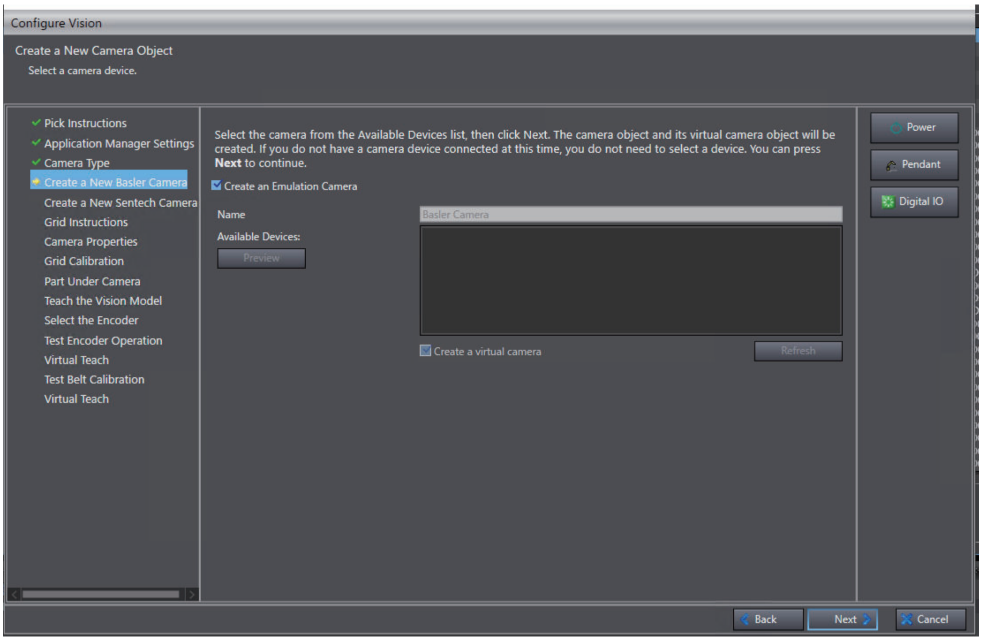
The **Application Manager Settings** dialog is shown.



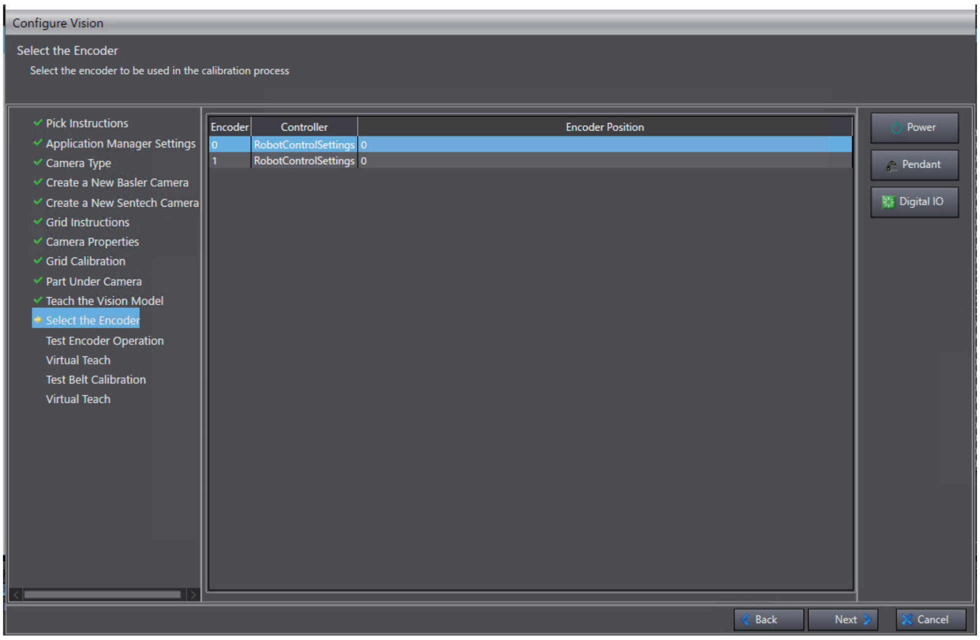
2 Click the **Next** button.
The **Camera Type** dialog is displayed.



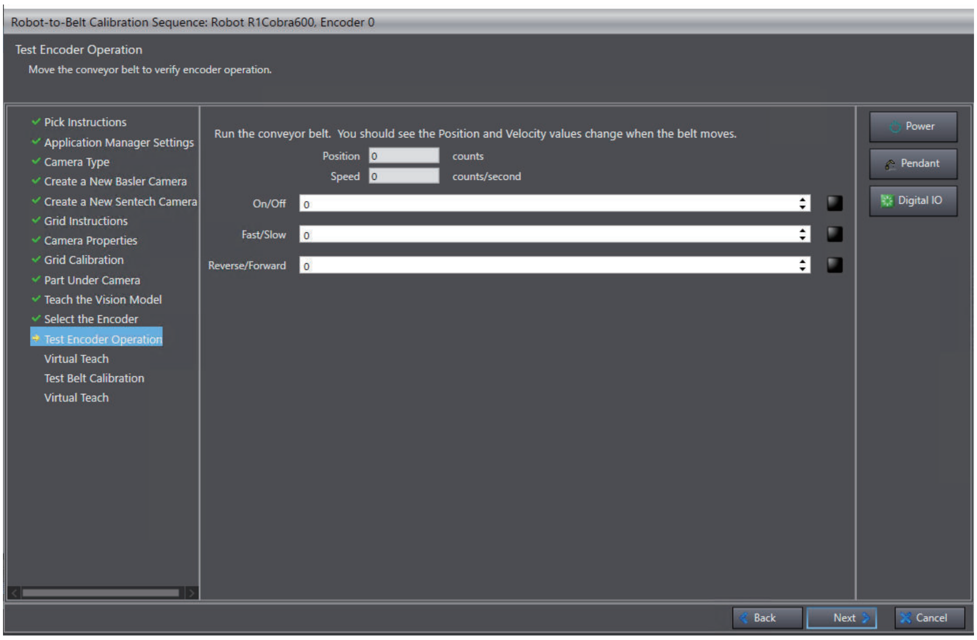
3 Select the camera to use, then click the **Next** button.
The **Create a New Camera Object** window will open.



4 Confirm that the **Create an Emulation Camera** box is checked, then click the **Next** button.
The **Select Encoder** dialog is displayed.

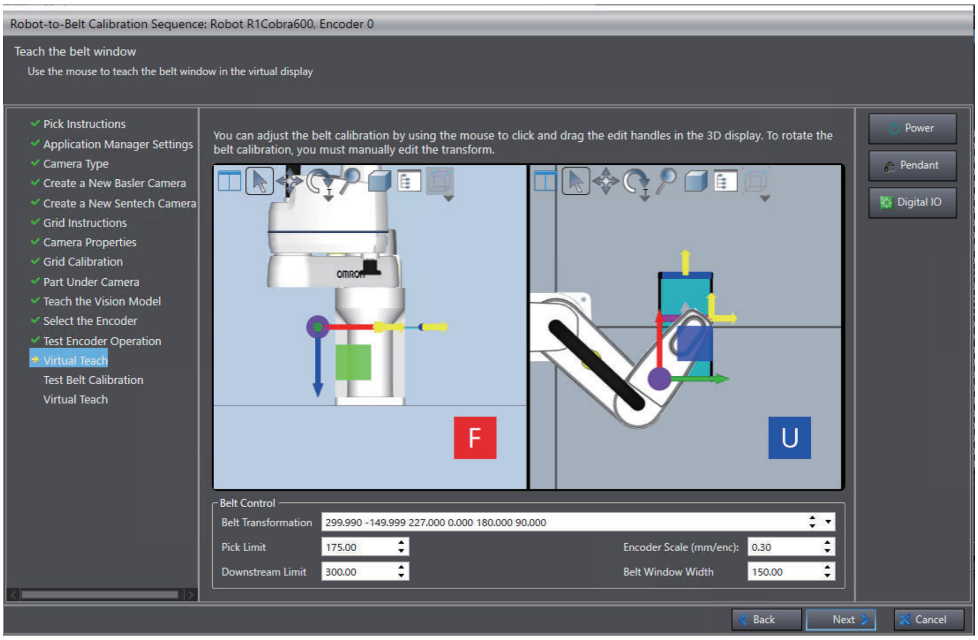


5 Select the encoder to use, then click the **Next** button.
The **Test Encoder Operation** dialog is displayed.



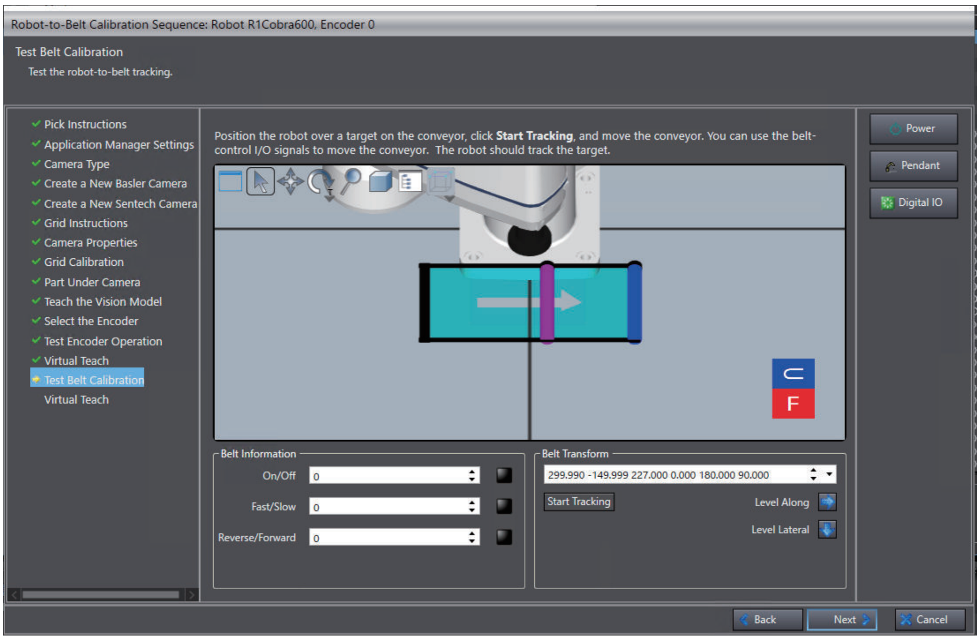
6 Move the conveyor in the view by clicking the **ON/OFF**, **Fast/Slow**, and **Forward/Reverse** buttons, and check the encoder's operation.

7 After checking the encoder operation, click the **Next** button.
The **Teach the belt** window is displayed.



8 Associate the belt travel direction with the robot workspace.
In the left window, click and drag the edit handles for a belt calibration. Refer to *5-5-3 Belt Calibration* on page 5-21 for details of belt calibration.

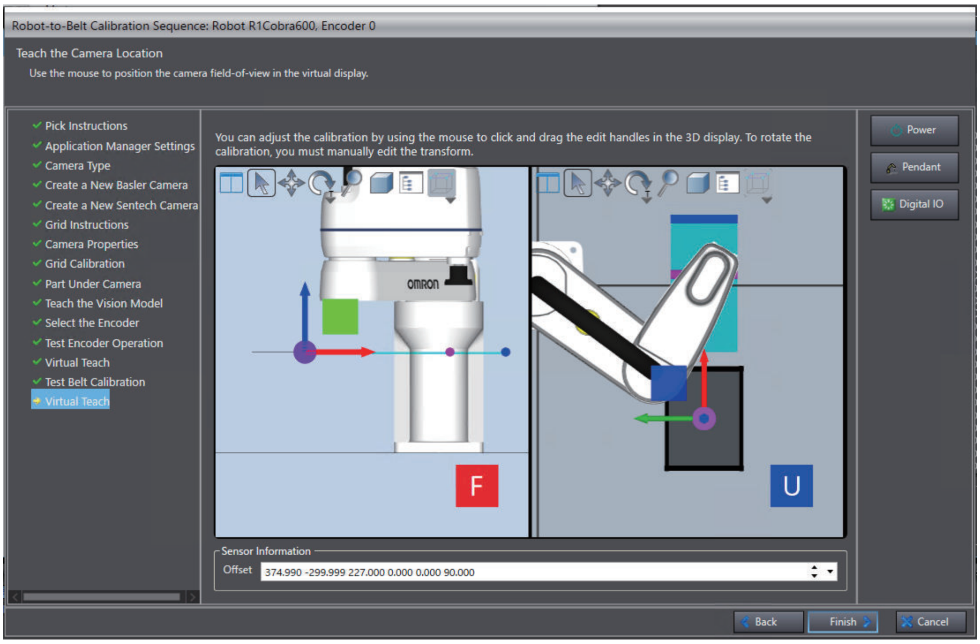
9 After finishing associating the belt travel direction with the robot workspace, click the **Next** button.
The **Test Belt Calibration** window is displayed.



10 Perform belt calibration.
Click and drag the robot's tip to directly above the part on the conveyor belt. Then click the **Start Tracking** button.
Click the **On/Off**, **Fast/Slow**, and **Reverse/Forward** buttons to check that the robot is tracking the target by simulating the movement of the conveyor belt.

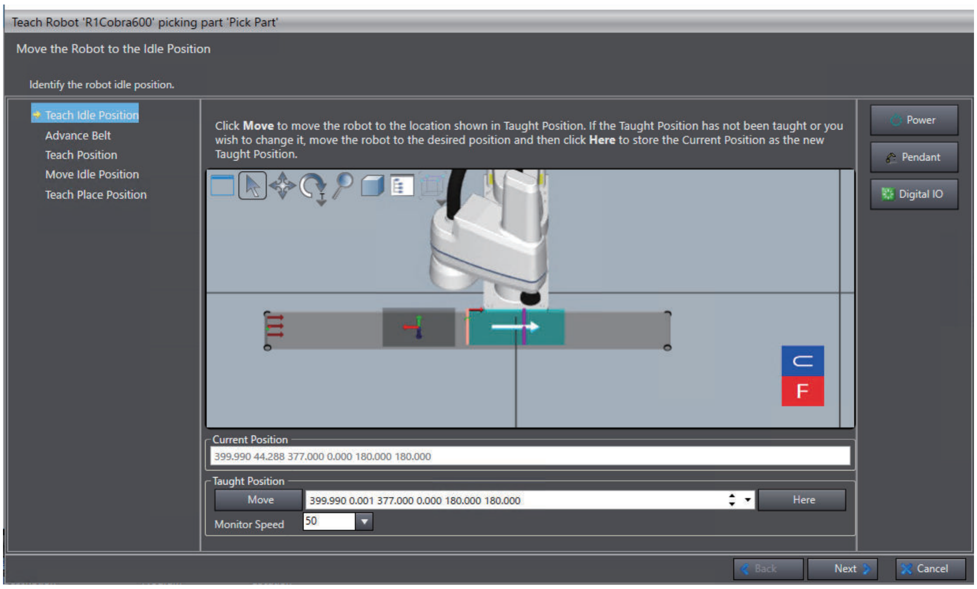
Refer to *Testing the Calibration* on page 5-27 for details of belt calibration test.

11 After completion of the Test Belt Calibration, click the **Next** button.
The **Teach the Camera Location** page appears.



12 Specify the position of the camera.
Click the edit handle in the left half of the area to set the field of vision of the camera.
Refer to *5-5-4 Camera Calibration* on page 5-27 to teach a camera position.

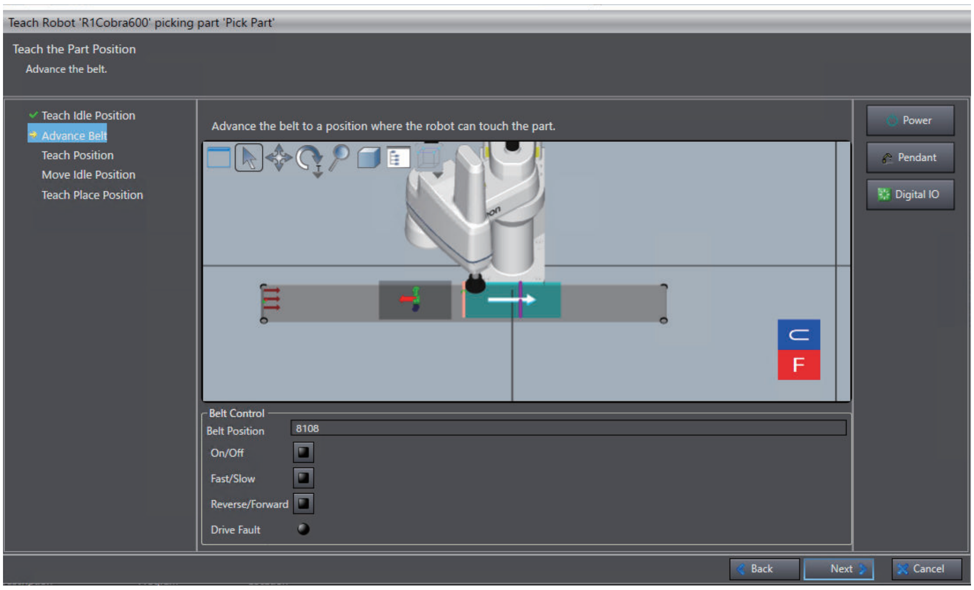
13 After specifying the position of the camera, click the **Finish** button.
The **Move the Robot to the Idle Position** page appears.



14 Specify a position where the robot stands by until a processing of the part or the part target begins.

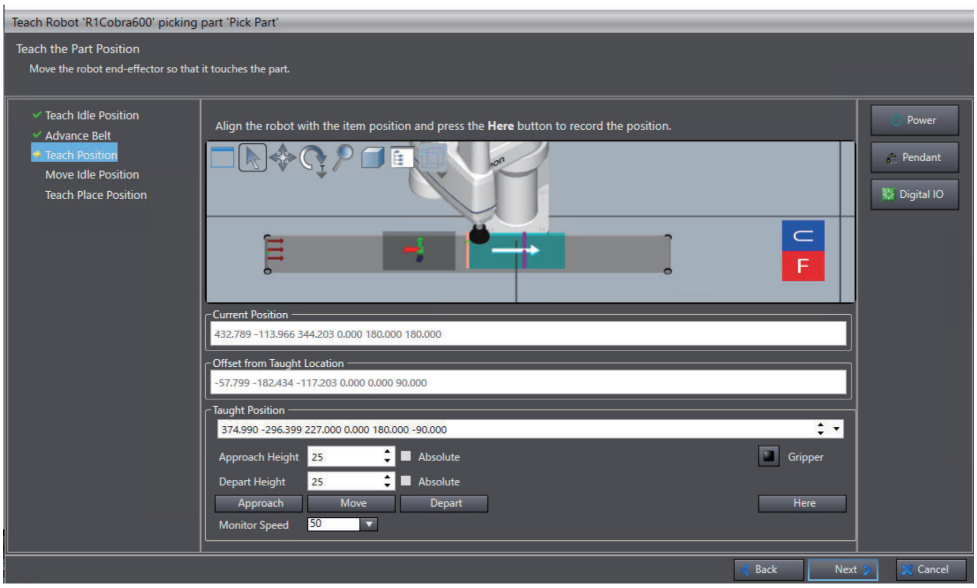
You can modify the position by clicking and dragging the robot in the robot view area. The standby position should be the upper area between the picking and placing points.

15 After moving the robot to a standby position, click the **Here** button to determine the position, and then click the **Next** button.
The **Teach the Part Position** (moving the belt forward) page appears.



Move the belt as the part approaches to the robot's moving range by clicking the **ON/OFF**, **Fast/Slow**, and **Forward/Reverse** buttons.

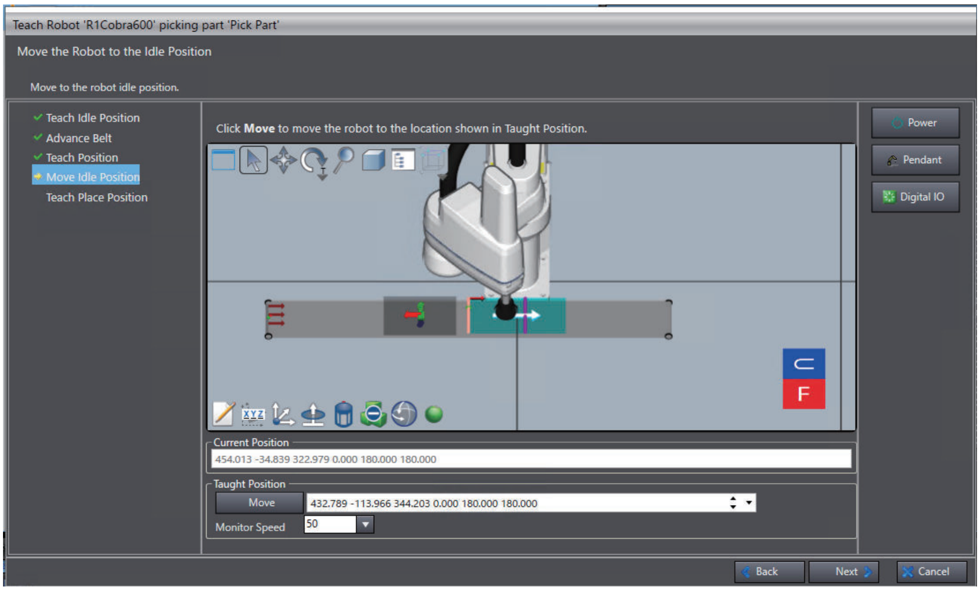
16 After confirming the robot is in the moving range, click the **Next** button.
The **Teach the Part Position** (teaching the position) page appears.



17 Move the robot's end effector to the part.
Drag the end effector to touch the part. Then click the **Here** button.
Click the **Approach** button to move the robot to the part position measured by the camera.
Clicking the **Move** button lets the end effector to pick up the part.

Clicking the **Depart** button makes the robot's end effector move to the upward.

18 After completion of the Teach the Part Position, click the **Next** button.
The **Move the Robot to the Idle Position** page appears.

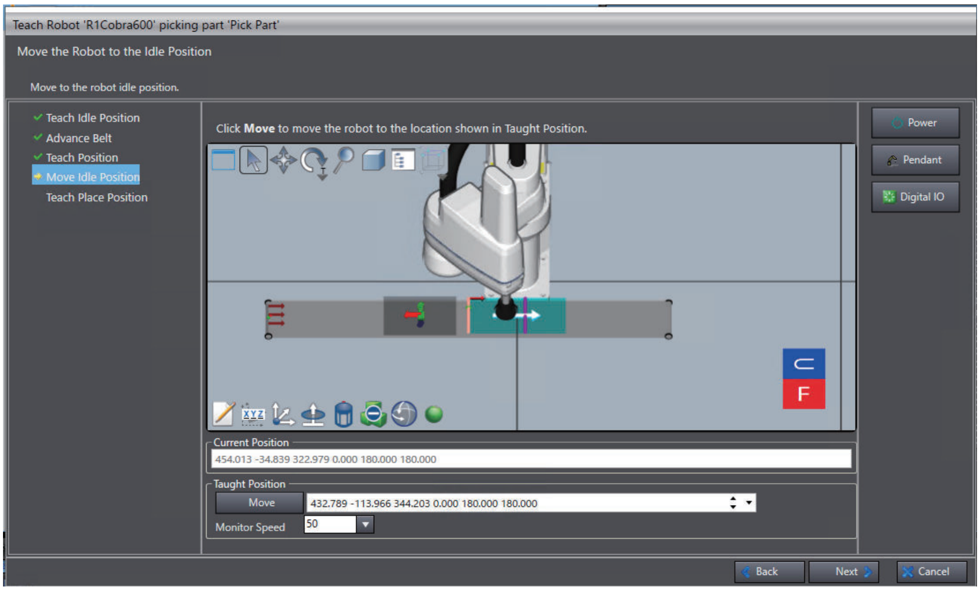


This completes the teaching of the pick-up operation.
In the next procedure, teach the robot a part placement position. Refer to *Teaching a Placing Position* on page 8-22 for teaching of part placement position.

Teaching a Placing Position

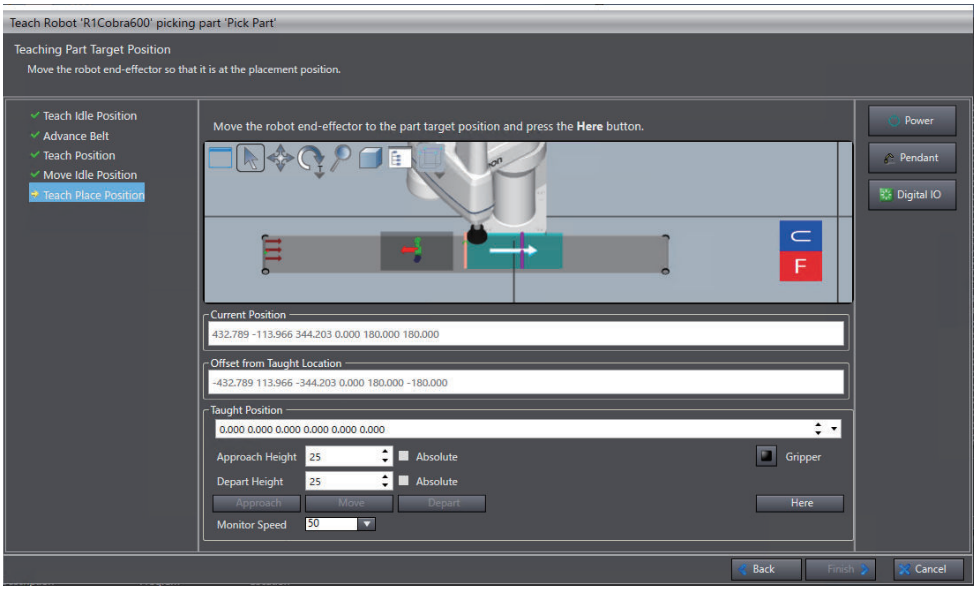
Teach the robot a part placement position.

1 Click the **Move** button to move the robot to the registered standby position.



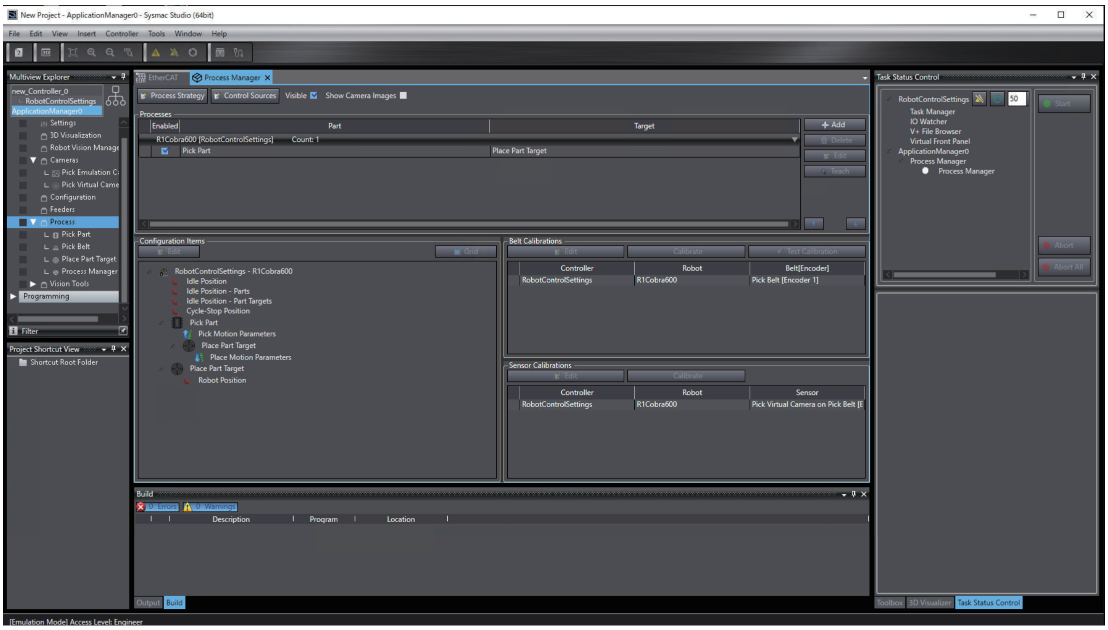
2 After moving the robot to the registered standby position, click the **Next** button.

The **Teaching Part Target Position** dialog is displayed.



Move the robot's end effector to the part placing point.
You can modify the robot position by clicking and dragging the end effector in the robot view area.

3 Move the robot's end effector to the part placement point, then click the **Here** button to establish location. Click the **Finish** button.
The Application sample wizard is completed.
Now you have finished creating the Pack Manager sample for pick-and-place operation.
The **Application Manager** is shown in the Multiview Explorer device list, and the **Process Manager** tab page is added to the Edit Pane.



8-3 Pack Manager

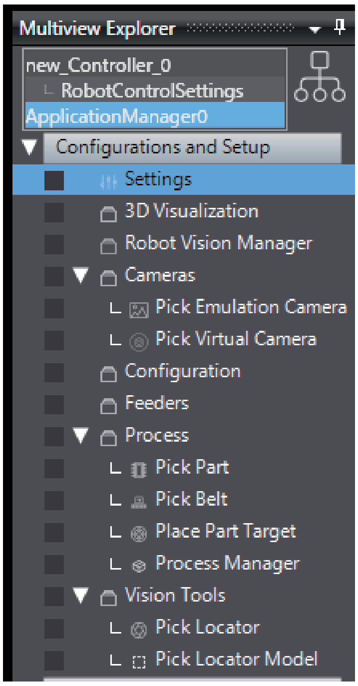
This section provides the information about configuration and offline debug for the Pack Manager.

8-3-1 Pack Manager Settings

Elements of the Pack manager are automatically created after you complete the Pack Manager Application Sample Wizard. Modify the settings for elements according to your system.

Elements of the Application Sample

Completing the Pack Manager Application Sample Wizard registers the following items.

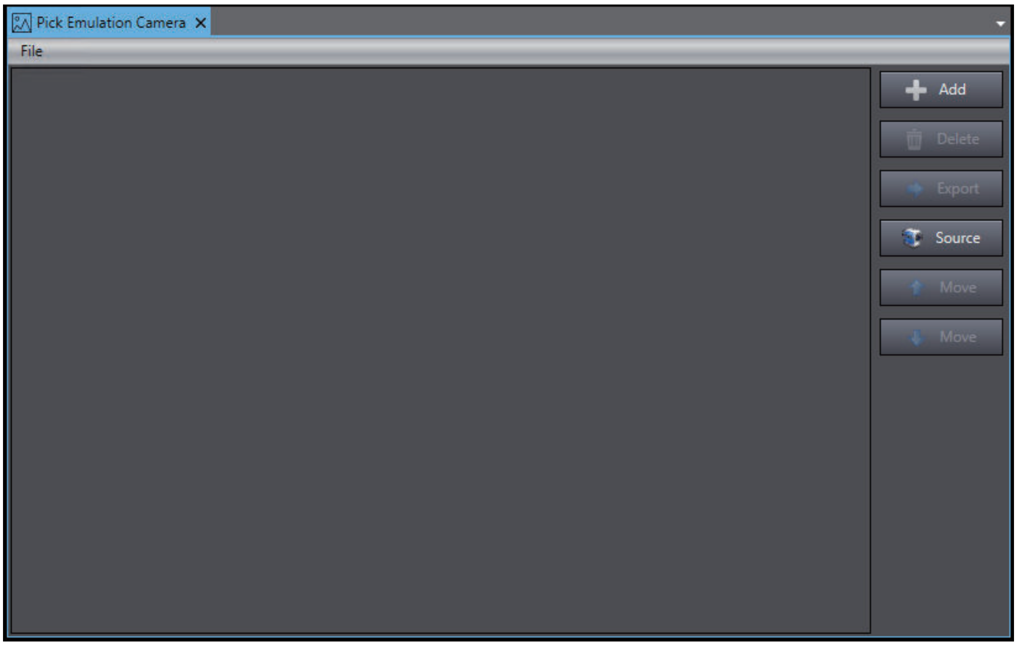


Item	Reference
Process Manager	*5-9-5 Process Manager* on page 5-127
Recipe Manager	*5-7-6 Recipe Manager* on page 5-81
Data Mapper	*5-7-2 Data Mapper* on page 5-74
System Startup	*5-7-5 Program System Startup* on page 5-80
Pick Part	*5-9-3 Part* on page 5-113
Pick Belt	*5-9-4 Belt* on page 5-115
Pick Emulation Camera	*5-6-2 Emulation Cameras* on page 5-53
Pick Locator	*5-10 Vision Tools* on page 5-182
Pick Locator Model	*5-10 Vision Tools* on page 5-182
Pick Part Target	*5-9-2 Part Target* on page 5-107
Pick Virtual Camera	*5-6-1 Virtual Camera* on page 5-47

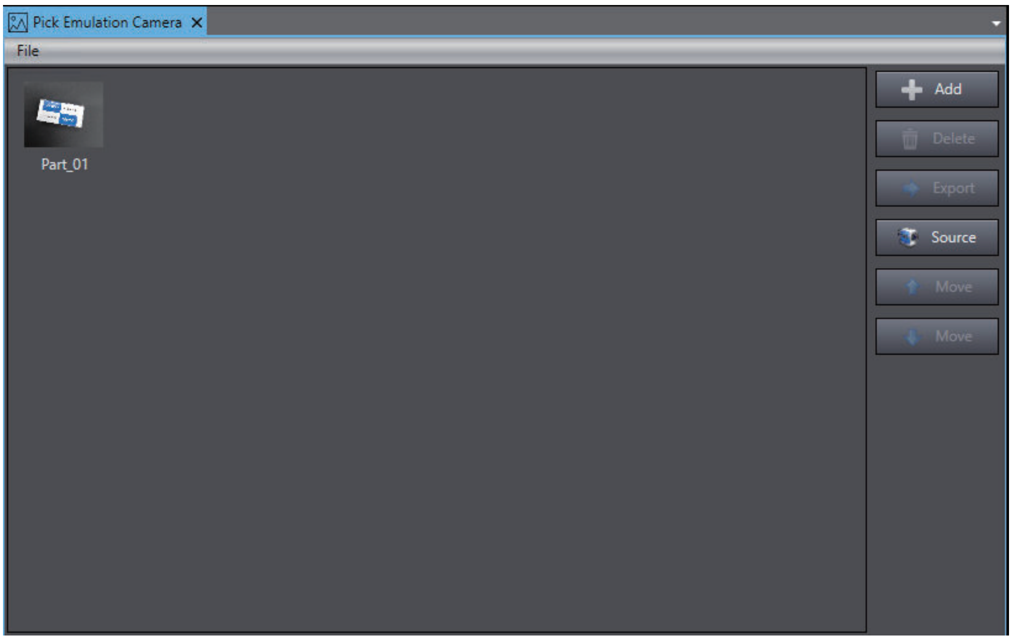
Registering a Part Image

For an offline-simulation of image discrimination, add an image of the part that the Robot Integrated System handles, if necessary.
Register the added image to the Locator Model to configure the detection setting.

1. Select **Cameras** in **Configurations and Setup** in the **Multiview Explorer**. Double-click **Pick Emulation Camera**.
   The **Pick Emulation Camera** tab page is displayed.

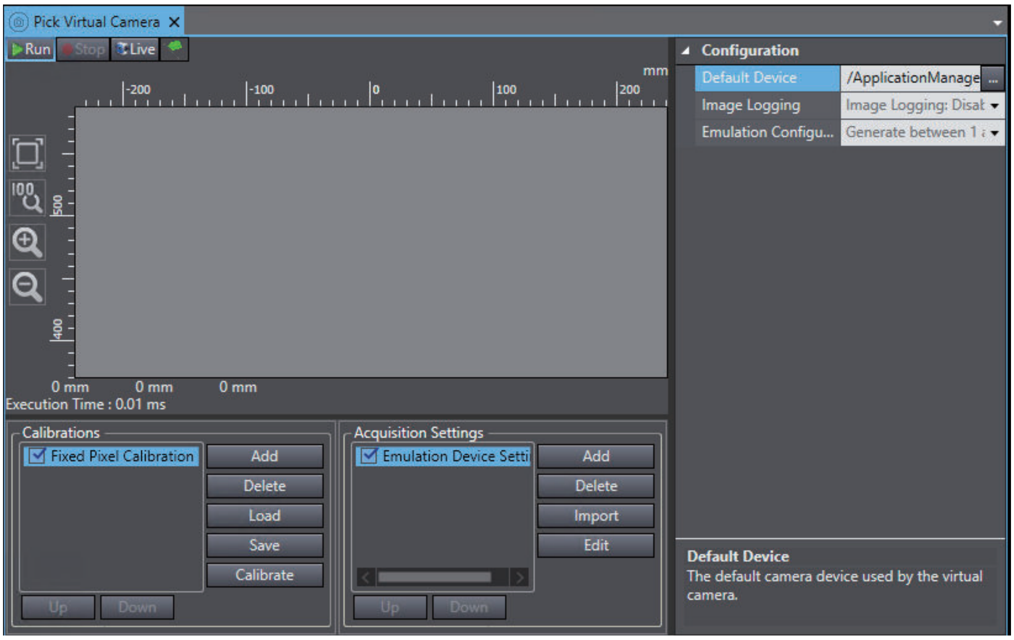


2. Click the **Add** button to select an image to add. Then click the **Finish** button.
   The image will be added.

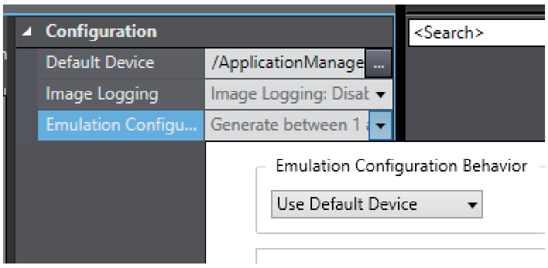


3 Select **Cameras** in **Configurations and Setup** in the **Multiview Explorer**. Double-click **Pick Virtual Camera**.

The **Pick Virtual Camera** tab page is displayed.

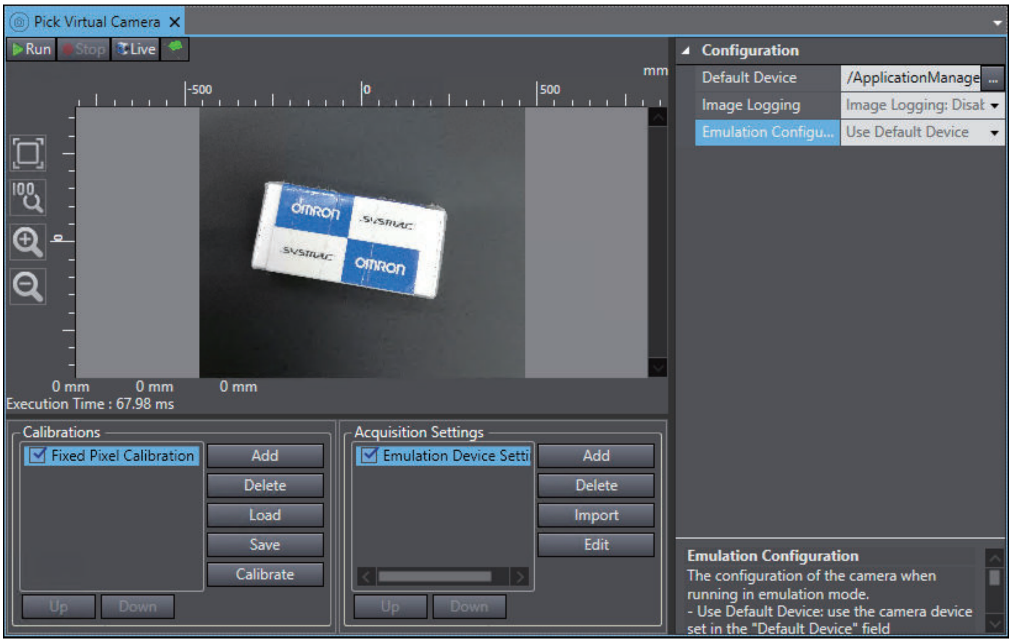


4 Select **Configuration** - **Emulation Configuration** - **Use Default Device**.



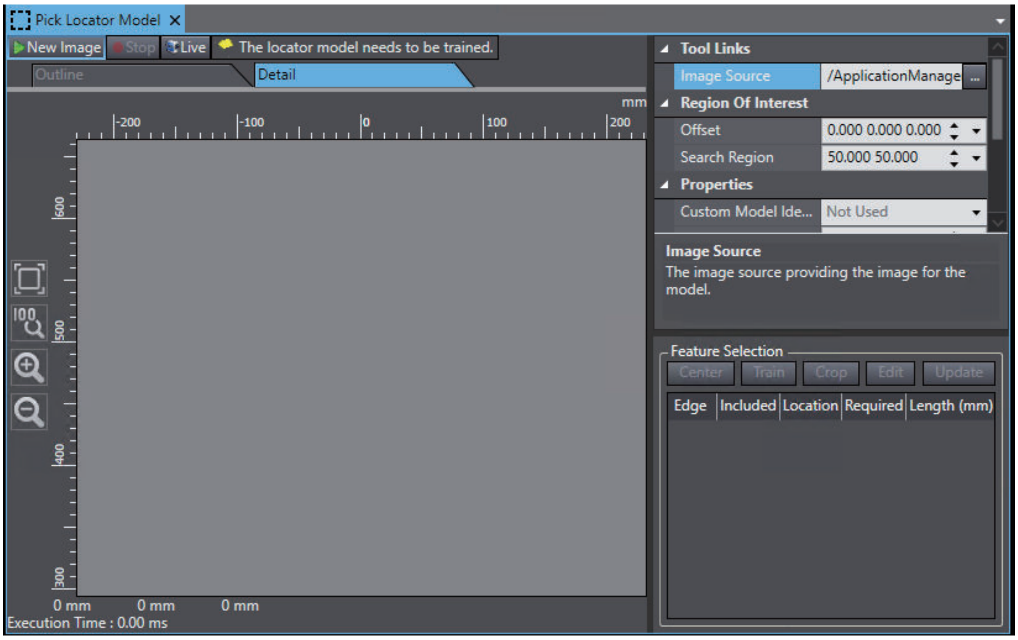
5 Click the **Run** button.

The image registered to the **Pick Emulation Camera** appears.



6 Select **Configurations and Setup** – **Vision Tools** in the Multiview Explorer. Double-click **Pick Locator Model**.

The **Pick Locator Model** tab page is displayed.

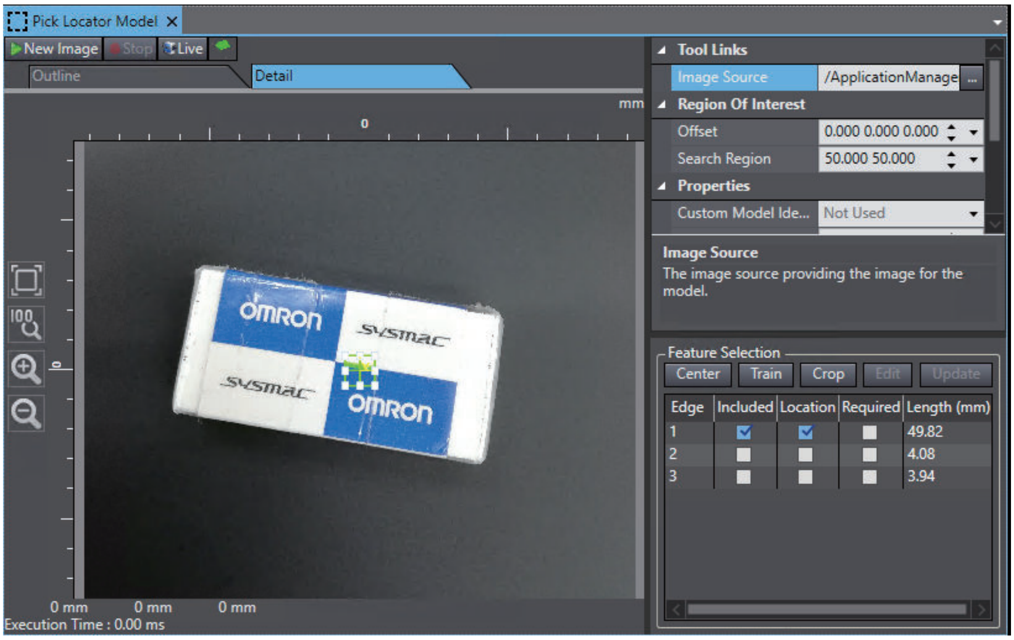


7 Click the the **New Image** button to display the image acquired by the **Pick Virtual Camera**.

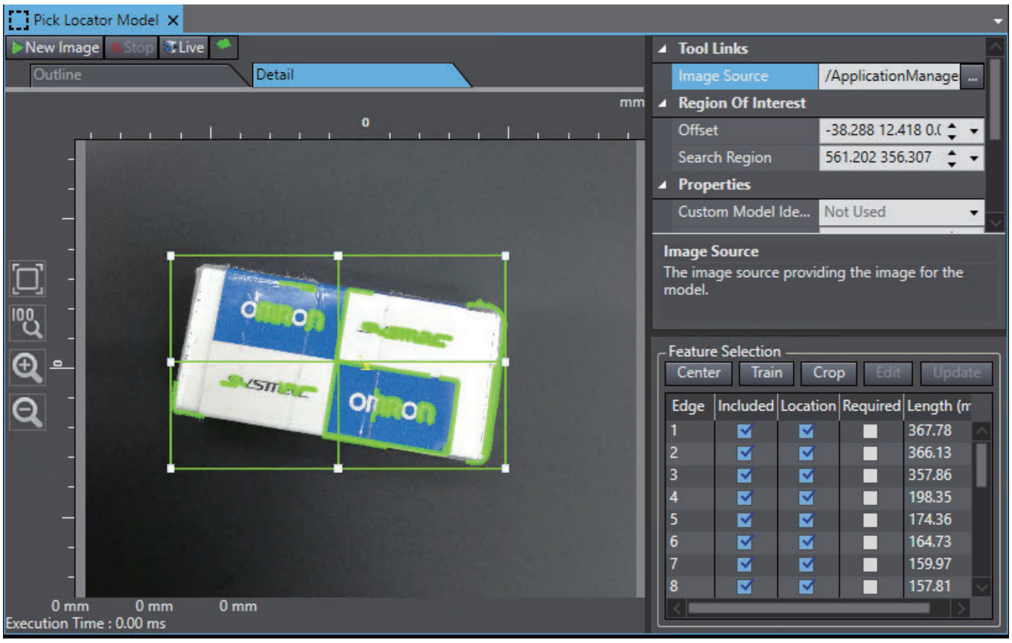
8-3 Pack Manager

8

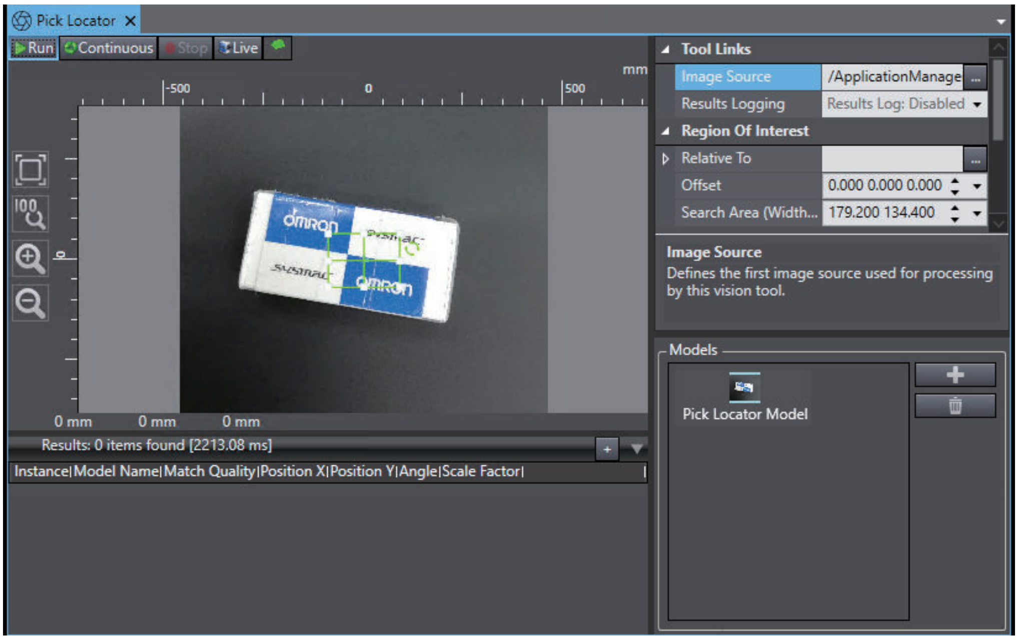
8-3-1 Pack Manager Settings



8 Set the area and center, then click the **Train** button in the **Feature Selection**.

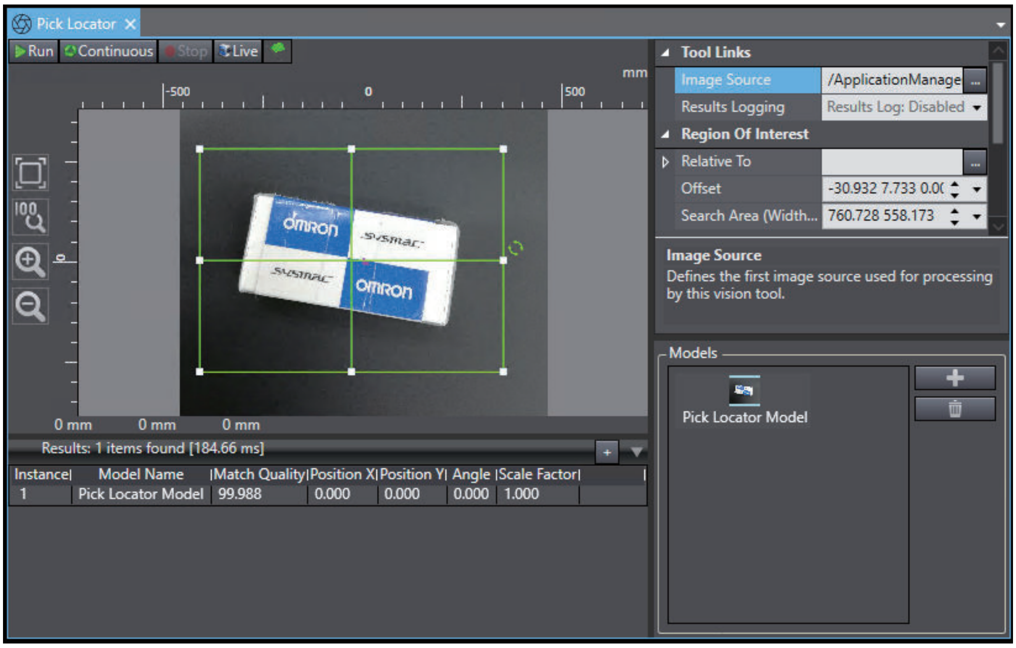


9 Double-click **Pick Locator** under **Configurations and Setup** in the Multiview Explorer.
The **Pick Locator** tab page is displayed.



If a Locator Model is not shown, click the add button in **Model**, then select the **Pick Locator Model**.

10 Confirm that the registered Locator Model is shown in **Model** under **Configuration**. Then move and adjust the green frame to specify the area where the locator will be detected. Click the **Run** button.
If the message "**1 item found**" is shown in **Results**, you have succeeded in detecting the locator.

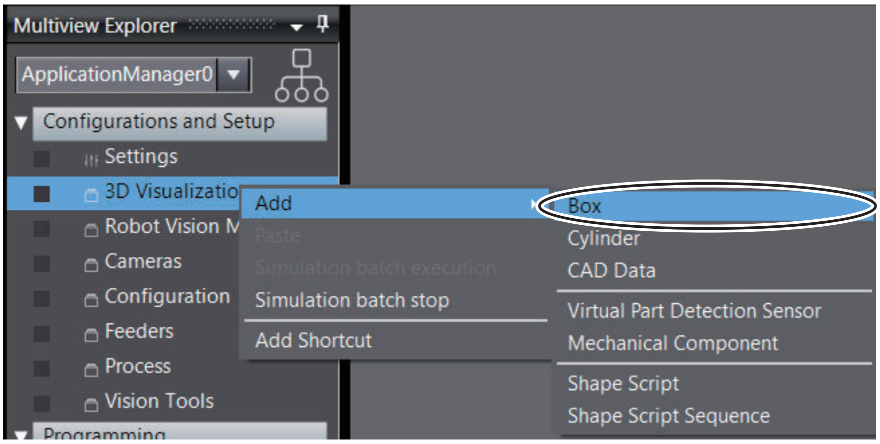


In the same way, detect the locator in other areas and images to test detections.

Adding a Peripheral Object

If you need to check a collision between the robot and a peripheral device, add a 3D shape data to the Robot Integrated System, if necessary.

1 Select **Box** or **Cylinder** from the **3D Visualization** menu to add a 3D shape data.



Place the 3D shape data on the possible point where the robot arm may contact, e.g., by the conveyor belt. Refer to *5-4 3D Visualization* on page 5-7 for the details of adding a 3D shape data.

8-3-2 Offline Debugging the Pack Manager

To perform an offline-debug of the Pack Manager, operate the Application Manager and Robot Integrated CPU Unit with a virtual controller. Refer to *7-1-1 Schema of Offline Debug Procedure* on page 7-2 for details.

8-4 Robot Vision Manager

This section describes configuration and offline debugging of the Robot Vision Manager.

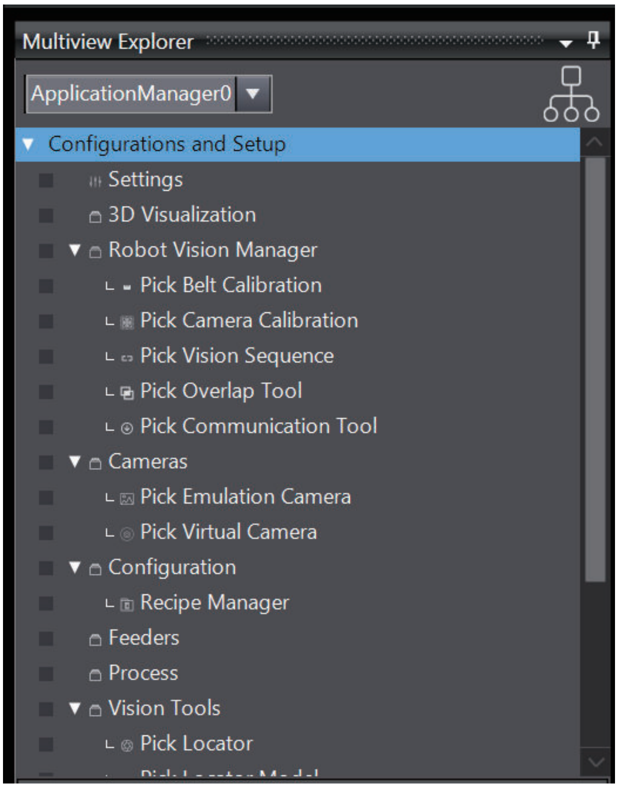
8-4-1 Robot Vision Manager Settings

Elements of the Robot Vision Manager are automatically created after you complete the Robot Vision Manager Application Sample Wizard. Modify the settings for elements according to your system.

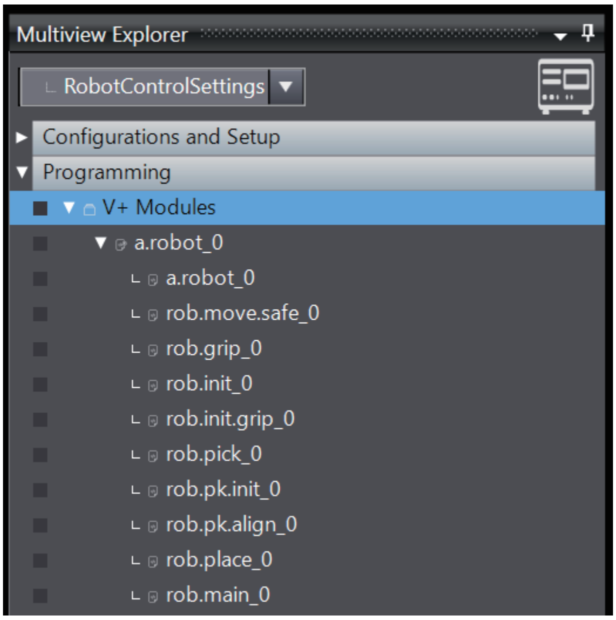
Elements of the Robot Vision Manager Application Sample

Completing the Robot Vision Manager Application Sample Wizard registers the following items.

Application Manager



RobotCotrolSettings



Item	Reference
Pick Vision Sequence	*5-5-6 Vision Sequence* on page 5-37
Pick Belt Calibration	*5-5-3 Belt Calibration* on page 5-21
Pick Camera Calibration	*5-5-4 Camera Calibration* on page 5-27
Pick Overlap Tool	*5-5-7 Overlap Tool* on page 5-39
Pick Communication Tool	*5-5-8 Communication Tool* on page 5-41
Pick Emulation Camera	*5-6-2 Emulation Cameras* on page 5-53
Pick Virtual Camera	*5-6-1 Virtual Camera* on page 5-47
Pick Locator	*5-10 Vision Tools* on page 5-182
Pick Locator Model	*5-10 Vision Tools* on page 5-182
V+ Modules	*Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*
V+ Variables	*Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*

Registering a Part Image

For an offline-simulation of image discrimination, add an image of the part that the Robot Integrated System handles, if necessary.

Refer to *Registering a Part Image* on page 8-25, *8-3-1 Pack Manager Settings* on page 8-24.

Adding a Peripheral Object

If you need to check a collision between the robot and a peripheral device, add a 3D shape data to the Robot Integrated System, if necessary.

Refer to *Adding a Peripheral Object* on page 8-30, *8-3-1 Pack Manager Settings* on page 8-24.

8-4-2 Offline Debugging the Robot Vision Manager.

To perform an offline-debug of the Robot Vision Manager, operate the Application Manager and Robot Integrated CPU Unit with a virtual controller. Refer to *7-1-1 Schema of Offline Debug Procedure* on page 7-2 for details.

9

Maintenance

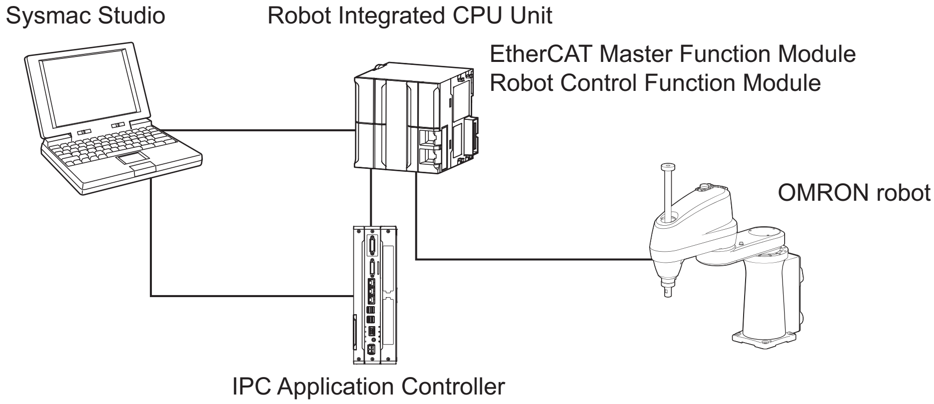
This section describes methods to check errors and events that may occur in the Robot Integrated System.

9-1 Troubleshooting

This section provides an outline of the Sysmac Studio functions that can be used for troubleshooting and maintenance of a Robot Integrated System.

9-1-1 Outline of Troubleshooting and Maintenance Functions

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.



Purpose	Function	Target	Description	Reference
Checking for errors in 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 Robot Control Function Module.	*Sysmac Studio Version 1 Operation Manual (Cat. No. W504)*
Finding the causes of troubles, adjustment	System Monitor	Application Manager Robot Control Function Module	You can monitor the parameters of the Application Managers and Robot Control Function Modules in real time.	*10-2 System Monitor* on page 10-4
	View eV+ Log	Robot Control Function Module	Display the history of processing in the Robot Control Function Module.	*Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)*
	Robot Hardware Diagnostics	OMRON robot	You can check the condition of the robot motor.	*10-3 Hardware Diagnostics* on page 10-8
	Robot Data Collection	OMRON robot	You can display and save the Robot Integrated System data.	*10-4 Data Collection* on page 10-10

Purpose	Function	Target	Description	Reference
	Robot Motor Tuning	OMRON robot	You can send a square wave positioning command to a specified motor and observe the response.	*10-5 Motor Tuning* on page 10-12

10

Other Functions

This section describes other functions as printing and hardware diagnostics.

10-1 Printing

This section describes the printing functions of the Application Manager.
Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the basic functions and specific procedures for printing.

10-1-1 Items You Can Print

You can print any of the following items.

Category	Item
Settings	IP Address
	Port Number
Adding 3D Objects	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
	Overlap Tool
	Communication Tool
Camera Settings	Virtual Camera
	Emulation Camera
	Visualizer Capture Device
	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

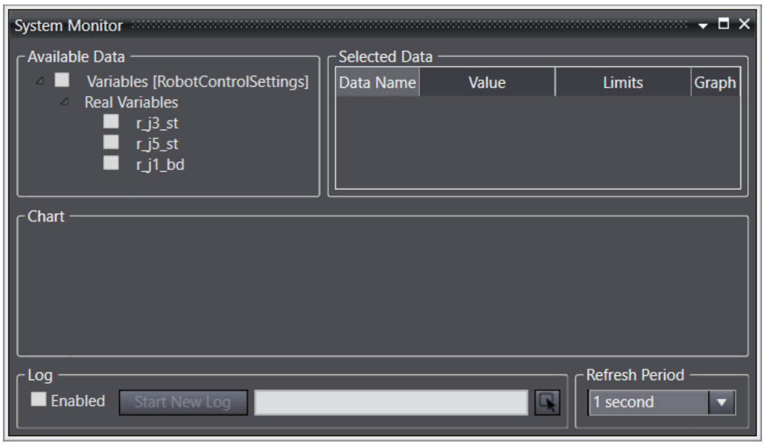
Category	Item
Process	Part Buffer
	Part Target
	Part
	Belt
	Process Manager
	Allocation Script
	Pallet
	Visual Refinement Station
Vision Tools	Finder Tools
	Inspection Tools
	Reader Tools
	Calculation Tools
	Image Process Tools
	Custom Tools
Program Creation	C# Program
	Variables

10-2 System Monitor

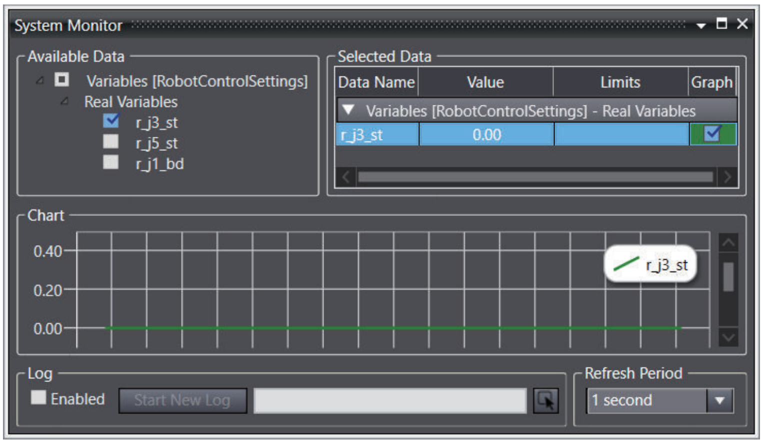
System Monitor allows you to graph the status of a process running in the Application Manager, robot parameters, V+ variable values of the Robot Integrated CPU Unit and output those values to a log file. This section provides the procedure to display V+ variables on the System Monitor during a simulation.

1 Select *RobotControlSettings* from a Multiview Explorer's device list.

2 Select **View**– **System Monitor** from the main menu.
System Monitor Tab Page appears to display registered V+ variables.



3 Check a variable you want to monitor in the **Available Data** area. Then, check the **Graph** box in the **Selected Data** area.
Values of the variable are plotted on a graph.



4 Select the folder in which to save a log in the **Log** area and check the Enabled box. Then click the **Start New Log** button.
A text file that contains variable values will be created in the designated folder.

System Monitor Data

The following tables show the detailed information about the system monitor data.

● Robot Parameters

Parameter Name	Description
Amplifier bus voltage (V)	The current amplifier bus voltage for the robot. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars). If the value drops below the range minimum, this means that the motion is too hard or the AC input voltage is too low; if the value exceeds the range maximum, this means that the motion is too hard or the AC input voltage is too high. Lowering the motion speed (more than the acceleration) can help correct these issues.
AC input (V)	The current AC input voltage (220 VAC) for the robot. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars). Running outside or close the limits may create envelope errors.
DC input (V)	The current DC input voltage (24 VDC) for the robot. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars).
Base board temperature (°C)	The current temperature (°C) for the amp-in-base processor board. This should operate within the specified min/max warning limits (yellow bars), and never reach the min/max error limits (red bars).
Encoder temperature (°C)	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 Parameters

Parameter	Description
Instance Count	The number of instances that have been available since the last restart or reset.
Belt Velocity	The average velocity of the conveyor belt.
Instantaneous Instances	The instantaneous instances since the last restart or reset. This is calculated over the update period selected in the **System Diagnostics** settings. If the graph is updated 500 ms, it will tell you the instantaneous instances/minute over each 500 ms time segment.
Instances Per Minute	The average number of instances per minute.
Active Instances	The number of active instances on the belt.
Latch Faults	The number of latch faults since the last restart or reset.

● Process Manager Robot and Process Parameters

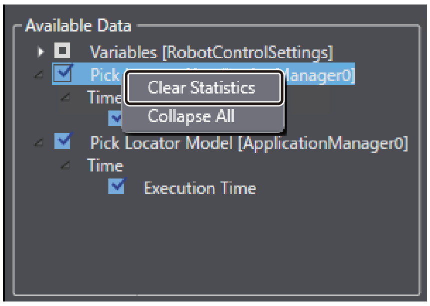
Parameter	Description
Idle Time (%)	The average idle time percentage of the total run time since the last restart or reset. ("*Idle*" is when the Process Manager is waiting on part or part target instances to process.)
Processing Time (%)	The average processing time percentage of the total run time since the last restart or reset. ("*Processing*" is when the Process Manager is actively processing part or part target instances.)
Average Total Time (ms)	For the Process Manager group only. The average total time for all robots. Other fields, such as **Parts Processed/Not Processed** and **Targets Processed/Not Processed**, show a summation for all robots.
Parts Per Minute	The average number of parts per minute. When viewing the Process Manager group, this is a summation for all robots.
Targets Per Minute	The number of targets per minute. When viewing the Process Manager group, this is a summation for all robots.
Parts Not Processed	The number of parts not processed since the last restart or reset. When viewing the Process Manager group, this is a summation for all robots.
Targets Not Processed	The number of targets not processed since the last restart or reset. When viewing the Process Manager group, this is a summation for all robots.
Parts Processed	The number of parts processed since the last restart or reset. When viewing the Process Manager group, this is a summation for all robots.
Targets Processed	The number of targets processed since the last restart or reset. When viewing the Process Manager group, this is a summation for all robots.

● Process Manager Process Strategy Parameters

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

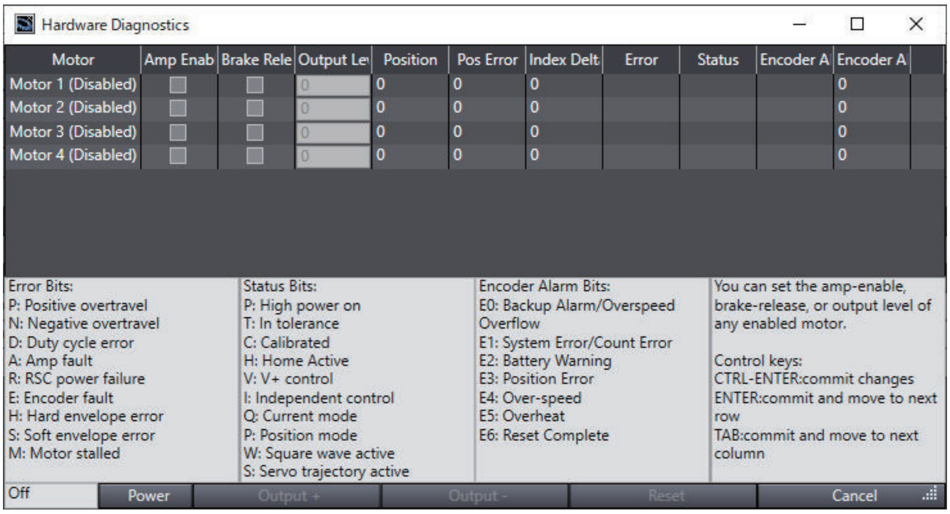
You can clear all Process Manager General Statistics. To clear statistics, right click the Process Manager item and select **Clear Statistics**.


Available Data
Variables [RobotControlSettings]
Pick
Manager0]
Clear Statistics
Collapse All
Time
Pick Locator Model [ApplicationManager0]
Time
Execution Time

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.



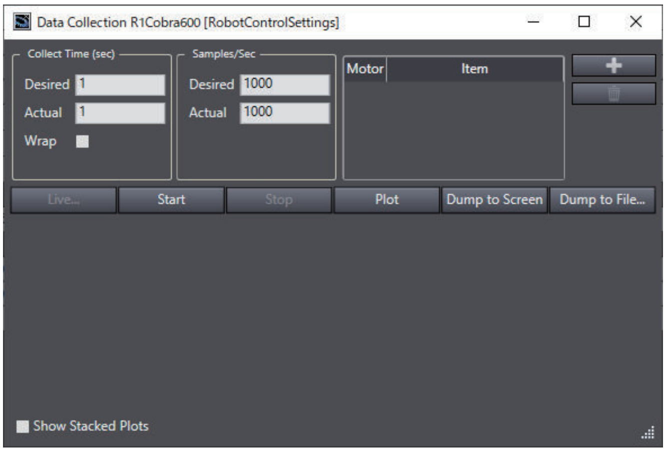
Item	Description
Amp Enable	Enables/disables an amplifier for the selected motor.
Brake Release	Enables/disables a brake release for the selected motor.
Output Level	Specifies a commanded torque, which is used to test the operation of the selected motor. The range is from -32767 to 32767.
Position	Displays 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

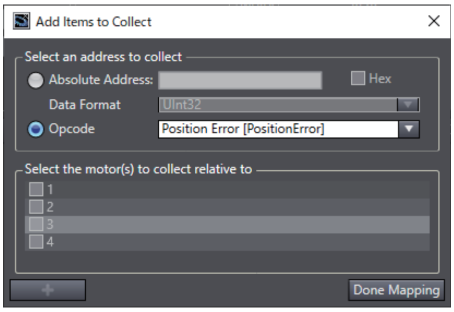
Item	Description
Status	Displays the following status codes for the selected motor. P: High power on T: In tolerance C: Calibrated H: Home sensor active V: V+ control I: Independent control Q: Current mode P: Position mode W: Square wave active S: Servo trajectory active
Reset	Resets any encoder errors for the selected motor.
Power	Toggles the high power. The current power state is shown in the status field.
Output + Output -	DAC output controls. Click the **Output +** button to increase the DAC output to the selected motor. Click the **Output -** button to decrease the DAC output to the selected motor.

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.



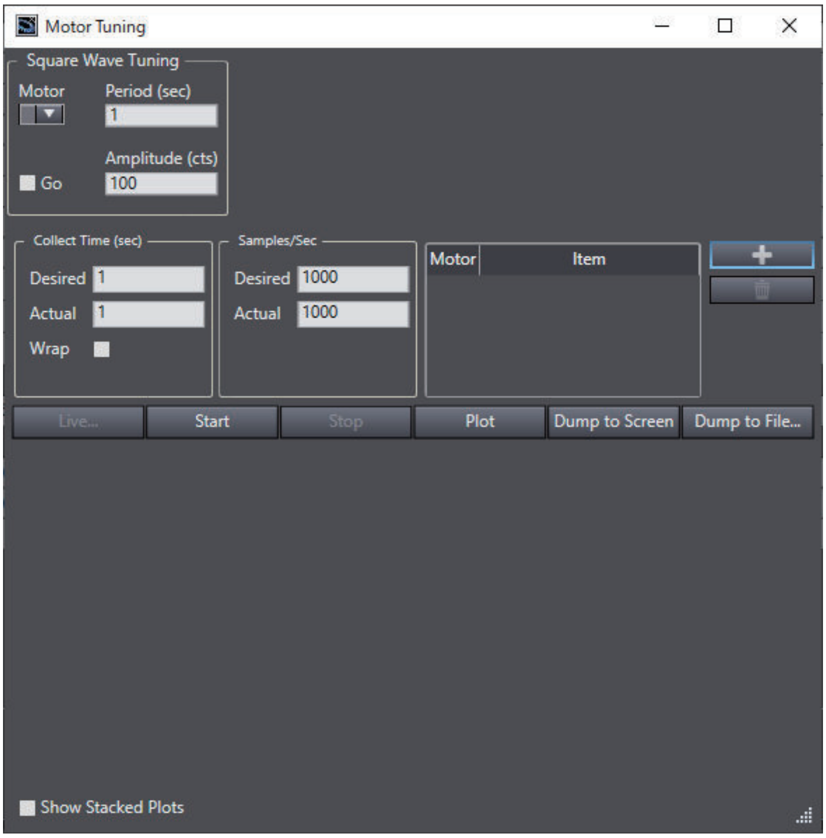
Item	Description
Collect Time (sec)	Specifies the data collection time in seconds. The default value is 1.
Samples/sec	Specifies the data collection frequency (samples/sec). The default value is 1000.
Add.../Remove	Add or remove motor items to examine with these buttons. Clicking the **Add** button will display the **Add Items to Collect** dialog box.  Enter an Absolute Address or select **Opcode** and select an available data item from the drop-down list. Then select the motor(s) from which you want to collect data.
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 **Stop** 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 **Data Collection** 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 **Save As** 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* on page 10-10 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 provides the information about the Search and Replace functions for the Application Manager.
Refer to the *Sysmac Studio Version 1 Operation Manual (Cat. No. W504)* for the basic functions and specific procedures for Search and Replace.
Refer to *Sysmac Studio Robot Integrated System Building Function with Robot Integrated CPU Unit Operation Manual (Cat. No. W595)* for behaviors of Search/Replace functions for *RobotControlSettings*, the subdevice of the Robot Integrated CPU Unit.

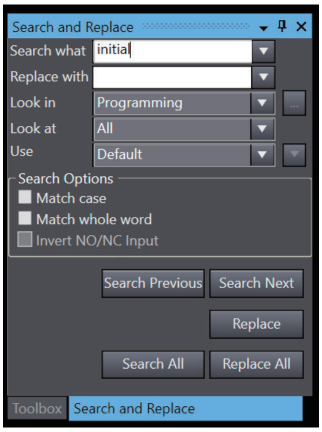
10-6-1 Scope of Searching and Replacing and Setting Items

Scope of Searching and Replacing

You can search and replace text strings that are displayed in the C# program editor.

Setting Items

The settings in the Search and Replace pane are explained below.



Setting item	Setting description
Search what	• Enter a search string. • You can select from previous search strings in the list.
Replace with	• Enter the string to replace the search string with. • 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.)
Look in	Specify the range to search. You can select from the following. **Programming**: C# program **Checked elements**: The item that is selected in the **Select search and replace scope** dialog box is searched. **Current view**: 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. **All**: Searches all text strings.

Setting item	Setting description
Use	Specify if you want to use wildcard characters. **Default**: Do not use wildcard characters. **Wildcard**: Use wildcard characters.

A

Appendices

This section provides the supplemental information for the main sections.

A-1 Version Control Function

This section describes the device-common and device-specific precautions and the displays for the safe use of project version control of Robot Integrated CPU Units and Application Managers.
Refer to the *Sysmac Studio Project Version Control Function Operation Manual (Cat. No. W589)* for precautions and specifications of other devices.

A-1-1 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 can merge changes in the following Multiview Explorer items to the data.
  a) Controller's data in **Programming** and lower-level folders
  b) HMI's data in Page and lower-level folders
  c) Programs and variables of V+ modules of RobotControlSettings
  d) The data of **User Functions** and **Shape Script** under **Configurations and Setup** – **3D Visualization** and the data of **Programming** – **Programs** in the Application Manager

  However, you cannot merge changes in other than those items: the data will always be overwritten with the contents of either the source or target of the merge.
- 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 Verification, set the user names, access levels, passwords and default user information.

A-1-2 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 **Configurations and Setup** items of the sub-device of the Robot Integrated CPU Unit, **RobotControlSettings**. This data will always be overwritten with the contents of either the source or target of the merge.
If you develop Robot Integrated CPU Unit programs with multiple developers, allow one supervisor of the project development to edit the **Configurations and Setup** items.
If you edit the **Configurations and Setup** items with multiple developers, the changes in the **Configurations and Setup** items may not be merged as intended.

Sysmac Diff Dialog Box

This section describes displays specific to the RobotControlSettings.

● Project Comparison Window

The items of the RobotControlSettings on the project comparison window are as follows.

<table>
<tr><th colspan="4">Item</th><th>Availability of Detailed Comparison</th><th>Details</th></tr>
<tr><td colspan="4">Configurations and Setup</td><td></td><td rowspan="5">When you select Configurations and Setup of the Robot Integrated CPU Unit and click the Select this to overwrite with left button, the items under that Configurations and Setup will be candidates for overwriting.</td></tr>
<tr><td></td><td colspan="3">Controller Settings</td><td></td></tr>
<tr><td></td><td colspan="3">Save Configuration</td><td></td></tr>
<tr><td></td><td colspan="3">Monitor Window</td><td></td></tr>
<tr><td></td><td colspan="3">Robots</td><td></td></tr>
<tr><td colspan="4">Programming</td><td></td><td></td></tr>
<tr><td></td><td colspan="3">V+ Modules</td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Module name</td><td></td><td>The registered program items are displayed under V+ Module.</td></tr>
<tr><td></td><td></td><td></td><td>Program name</td><td>○</td><td rowspan="2">When you select an item and click the Select this to overwrite with left button, the item will be candidates for overwriting.</td></tr>
<tr><td></td><td></td><td colspan="2">Variables</td><td>○</td></tr>
</table>

Refer to the *Sysmac Studio Project Version Control Function Operation Manual (Cat. No. W589)* for usage of project comparison window.

● Detailed Comparison Window

The RobotControlSettings-specific comparison items on the **Detailed Comparison** window are as follows.

<table>
<tr><th>Item</th><th>Description</th><th>Detailed Comparison window</th></tr>
<tr><td>V+ Module Programs</td><td>Compares V+ module programs. You can overwrite the contents of the source of comparison with the target of comparison for each line.</td><td rowspan="2">Refer to the Sysmac Studio Project Version Control Function Operation Manual (Cat. No. W589) for usage of Detailed Comparison window.</td></tr>
<tr><td>V+ Module Variables</td><td>Compares V+ module variables. You can overwrite the contents of the source of comparison with the target of comparison for each variable.</td></tr>
</table>

A-1-3 Application Manager

Observe the following precautions when you control the versions of projects that include the application manager.

Use

You can merge the following items of the Application Manager. The other data will be always overwritten with the contents of either the source or target of merge.

- **Configurations and Setup** – **3D Visualization** – **Shape Script Functions** – **User Functions**
- **Shape Scripts** under **3D Visualization** in **Configurations and Setup**
- C# program under **Programs** in **Programming**

If you develop Controller programs with multiple developers, allow one supervisor of the project development to edit anything other than the items that can be merged. Otherwise, the changes may not be merged as intended.

Sysmac Diff Dialog Box

This section describes displays specific to the Application Manager.

● Project Comparison Window

The items of the Application Manager device on the project comparison window are as follows.

Item				Availability of Detailed Comparison	Details
Configurations and Setup					
	Settings				
	3D Visualization				
		Shape Script Functions			• The registered function items are displayed under **Shape Script Functions**. • When you select an item and click the **Select this to overwrite with left** button, the item will be candidates for overwriting.
			User Functions name	○	
			Default Functions name		
		Box name			
		Cylinder name			
		CAD Data name			
		Virtual Part Detection Sensor name			
		Mechanical Component name			
		Custom Mechanics name			
		Parallel Link Model name			
		Shape Script name		○	
		Shape Script Sequence name			
	Robot Vision Manager				The registered Robot Vision Manager setting items are displayed under **Robot Vision Manager**.
		Robot Vision Manager setting items			
	Cameras				The registered camera names are displayed under **Cameras**.
		Camera name			
	Configuration				The registered setting items that the Application Manager has are displayed under **Configuration**.
		Application Manager setting items			
	Feeders				The registered feeder names are displayed under **Feeders**.
		Feeder name			

Item			Availability of Detailed Comparison	Details
	Process			The registered process-related setting items are displayed under **Processes**.
		Process-related setting items		
	Vision Tools			The registered tool names for image processing used in the Robot Vision Manager are displayed under **Vision Tools**.
		Image processing tool names in the Robot Vision Manager		
Programming				
	Programs			The registered C# program names are displayed under **Programs**.
		C# Program name	○	
	Variables			
		Numeric Variable name		When you select an item and click the **Select this to overwrite with left** button, the item will be candidates for overwriting.
		String Variable name		

Refer to the *Sysmac Studio Project Version Control Function Operation Manual (Cat. No. W589)* for usage of project comparison window.

● Detailed Comparison Window

The Application Manager-specific comparison items on the **Detailed Comparison** window are as follows.

Item	Description	Detailed Comparison window
User Functions	Compares User Functions. You can overwrite the contents of the source of comparison with the target of comparison for each line.	Refer to the *Sysmac Studio Project Version Control Function Operation Manual (Cat. No. W589)* for usage of Detailed Comparison window.
Shape Scripts	Compares Shape Scripts. You can overwrite the contents of the source of comparison with the target of comparison for each line.	
C# Programs	Compares C# programs. You can overwrite the contents of the source of comparison with the target of comparison for each line.	

I

Index

Index

S

OMRON Corporation **Industrial Automation Company**
Kyoto, JAPAN **Contact : www.ia.omron.com**

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

OMRON ASIA PACIFIC PTE. LTD.
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

Authorized Distributor:

©OMRON Corporation 2020-2025 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. W621-E1-07 0425