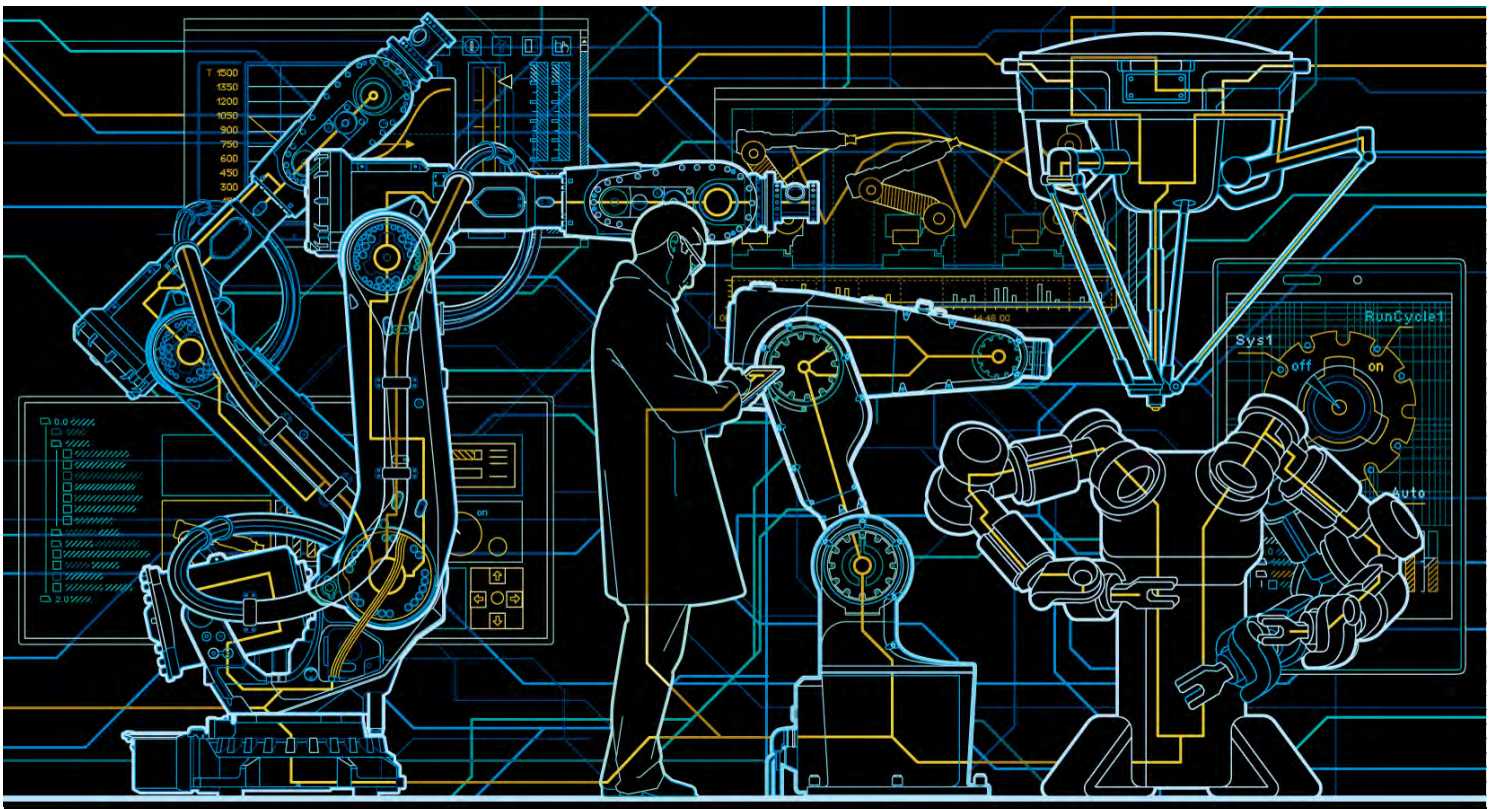


Application manual
Picking PowerPac for PickMaster™3

Power and productivity
for a better world™

Trace back information:
Workspace R16-2 version a5
Checked in 2016-09-08
Skribenta version 4.6.318

Application manual

Picking PowerPac for PickMaster™3

RobotStudio 6.04

Document ID: 3HAC048042-001

Revision: F

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damages to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

Additional copies of this manual may be obtained from ABB.

The original language for this publication is English. Any other languages that are supplied have been translated from English.

ABB AB
Robotics Products
Se-721 68 Västerås
Sweden

Table of contents

Overview of the manual 7
Safety 9

1 Overview of Picking PowerPac 11

1.1 About Picking PowerPac 11
1.2 Terms and concepts 12

2 Installation 15

2.1 System requirements 15
2.2 Installing the Picking PowerPac 16
2.3 Installing a license 17

3 Navigating the user interface 19

3.1 Introduction 19
3.2 Ribbon tab 20
3.2.1 Introduction 20
3.2.2 Line group 22
3.2.2.1 Add picking controller 22
3.2.2.2 Add Picking Tool 24
3.2.2.3 Add Conveyor 31
3.2.2.4 Add Workarea 34
3.2.2.5 Add Sensor 39
3.2.2.6 Add Flow Handler 43
3.2.3 Jobs group 48
3.2.3.1 Item/Container 48
3.2.3.2 Container Patterns 54
3.2.3.3 Job 59
3.2.4 Simulation group 74
3.2.4.1 Calibrate 74
3.2.4.2 Simulation controls 75
3.2.4.3 Flows 83
3.2.5 Online group 86
3.2.5.1 Export 86
3.2.6 3D layout group 87
3.2.7 Miscellaneous group 88
3.2.7.1 Library 88
3.3 Tree view browser 89
3.3.1 Introduction 89
3.3.2 Line tab 90
3.3.3 Jobs tab 100

4 Workflow of the Picking PowerPac 101

4.1 Workflow 101
4.2 Station 102
4.3 Creating a basic application 103

5 RAPID program 105

5.1 RAPID programs 105
5.2 Variables 110
5.3 Routines 113
5.4 Service routines 117

6 Advanced programming examples 119

6.1 Example: Mixing one pick work area and two place work areas 119
6.2 Example: Mixing two pick work areas and one place work area 120

6.3 Example: Mixing with one pick and one place work area 121
6.4 Example: Double pick single place 122
6.5 Example: Selecting item depending on clearance zone 125
6.6 Example: Sorting in negative y-direction 128
6.7 Controlling conveyor speed using an analog signal 131

7 RAPID reference information 133

7.1 Instructions 133
7.1.1 AckItmTgt - Acknowledge an item target 133
7.1.2 FlushItmSrc - Flush an item source 135
7.1.3 GetItmTgt - Get the next item target 136
7.1.4 NextItmTgtType - Get the type of the next item target 142
7.1.5 QStartItmSrc - Start queue in item source 144
7.1.6 QStopItmSrc - Stop queue in item source 145
7.1.7 ResetFlowCount - Reset flow counter 146
7.2 Functions 147
7.2.1 GetQueueLevel - Get queue level 147
7.2.2 GetQueueTopLevel - Get queue top level 149
7.2.3 GetFlowCount - Get number of passed items 150
7.3 Data types 151
7.3.1 itmtgt - Item target data 151
7.3.2 selectiondata - Selection data 154
7.3.3 sortdata - Sort data 157

Index 159

Overview of the manual

About this manual

This manual contains information and instructions for installing, and using Picking PowerPac.

Usage

This manual describes Picking PowerPac and includes step-by-step instructions on performing the various tasks that the software offers.

Who should read this manual?

This manual is mainly intended for:

- System integrators
- Machine builders
- ABB sales and product support

Prerequisites

A reader on a beginner level should have:

- Some basic experience with RobotStudio
- Some basic knowledge of robot picking applications

The advanced user should also have:

- Good skills in PickMaster, the IRC5 robot controller, and RAPID programming

References

References	Document ID
Application manual - PickMaster 3	*3HAC031978-001*
Operating manual - RobotStudio	*3HAC032104-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC050917-001*
Technical reference manual - RAPID overview	*3HAC050947-001*
Technical reference manual - RAPID kernel	*3HAC050946-001*
Technical reference manual - System parameters	*3HAC050948-001*

Revisions

Revision	Description
-	Released with RobotStudio 5.60 First edition.
A	Released with RobotStudio 5.61 Following are the updates: • Updated the section *Cameras on page 40*. • Updated the section *Controlling conveyor speed using an analog signal on page 131*.

Continues on next page

Continued

Revision	Description
B	Released with RobotStudio 6.0 Following are the updates: • Updated the section *Add Picking Tool on page 24*. • Updated the section *Flow control on page 80*.
C	Released with RobotStudio 6.01 Following are the updates: • Updated the section *Add Picking Tool on page 24*. • Updated the section *Add Sensor on page 39*. • Updated the section *Example: Double pick single place on page 122*. • Updated the section *Container Patterns on page 54*.
D	Released with RobotStudio 6.02 • Updated the section *The Activators tab on page 28*.
E	Released with RobotStudio 6.03 • Updated the section *Statistics on page 79*. • Updated the section *Advanced programming examples on page 119*.
F	Released with RobotStudio 6.04 • Updated the section *Add Conveyor on page 31*. • Updated the section *Add Workarea on page 34*. • Updated the section *Job on page 59*.

Safety

Safety of personnel

A robot is heavy and extremely powerful regardless of its speed. A pause or long stop in movement can be followed by a fast hazardous movement. Even if a pattern of movement is predicted, a change in operation can be triggered by an external signal resulting in an unexpected movement.

Therefore, it is important that all safety regulations are followed when entering safeguarded space.

Safety regulations

Before beginning work with the robot, make sure you are familiar with the safety regulations described in the manual *Operating manual - General safety information*.

This page is intentionally left blank

1 Overview of Picking PowerPac

1.1 About Picking PowerPac

Overview

The Picking PowerPac is a RobotStudio add-in, extending RobotStudio with picking functionality, which enables you to do offline programming and simulation of PickMaster controlled picking lines in RobotStudio.

Some of the features of the Picking PowerPac are:

- Enables realistic simulation of picking lines in the 3D environment of RobotStudio.
- Offers a simple but a powerful programming environment in RobotStudio that requires less skill and experience of the user.
- Allows you to configure a PickMaster controlled picking line in a few steps and simulate the production.
- Allows you to configure and simulate large scale customized PickMaster systems including realistic flows.
- Allows you to monitor the performance of the PickMaster system including detailed statistics on how different items and containers are handled by the system, for example, measurements on completed and incomplete containers.
- Allows you to try out and optimize the best line setup.
- Allows you to verify that the system is able to handle different flows with random variations.
- Allows you to import actual flows, recorded from the real PickMaster application, and use them in the simulation.
- Allows you to export PickMaster configuration files and import them into the real PickMaster application. In PickMaster, the imported configuration will reduce the time required to complete the online system configuration.

The Picking PowerPac consists of two software parts:

- A PowerPac in RobotStudio. This is a programming and simulation environment for PickMaster.
- Virtual PickMaster. This is a virtual version of the real PickMaster application that runs in the background during simulation.

The Picking PowerPac has a programming environment tailored for the programming of PickMaster applications. While the application is programmed, a virtual environment is automatically built up in the station view with robot, cameras, conveyors, items, and so on. The virtual environment is used to visualize the picking line layout and simulations. Simulation can be started as soon as a line and a picking job has been created.

The product is not mandatory for PickMaster 3 users. But it is a complementary for those who want to simulate their systems before going into the commissioning phase.

1.2 Terms and concepts

Terms list

Some terms have a specific meaning when used in this manual. Following are some terms with their description.

Term	Definition
Work area	A work area is a dedicated area where a robot picks or places objects.
Item source	RAPID data type that represents a work area. Sometimes referred to as a queue.
Conveyor work area	A work area placed on a conveyor. Picking or placing are performed using robot movements coordinated with the conveyor, that is, conveyor tracking.
Indexed work area	A work area placed anywhere near the robot. Picking or placing are performed using uncoordinated robot movements, that is, without conveyor tracking.
Sensor	A sensor is used to detect the location of objects and their types.
Camera	A camera is a sensor that can detect the exact locations of multiple objects. A camera can differentiate between different types of objects. It can be used for inspection, that is, to decide if an object is “good” (accepted) or “bad” (rejected).
I/O sensor	An I/O sensor, for example, a photoelectric sensor, detects the objects that are moved by a conveyor into a plane. For a photoelectric sensor, the plane will represent a light beam. An I/O sensor has limited functionality compared to a camera. It can only be used to locate objects of the same type, moving along the same path, having the same orientation, for example, a row of containers. The objects must also be separated from each other in order to be individually detected.
Flow	A flow represents incoming objects that appear on a specific location, for example, on a conveyor or on an indexed work area.
Flow Handler	A flow handler defines how an incoming flow of objects shall be handled by PickMaster. It selects a sensor for object detection and one or more work areas to pick or place the objects.
Item	An item is an object that can be picked and placed.
Item target	An item target is a RAPID data type representing the target position of an item.
Container	A container is an object that can be filled with items.
Container Pattern	A container pattern is a predefined pattern of items in a container.
Job	A job defines what a picking application should do with a selection of items and container patterns.
Distribution	A distribution defines how a flow handler should allocate detected items (or place locations for items) on different work areas.
Load balancing	Load balancing is a distribution type. Load balancing is used to share the work load between the robots. The detected objects are distributed to different work areas.

Continues on next page

Term	Definition
ATC	ATC is a distribution type. ATC is used to maximize the efficiency of the robots. Each detected object is distributed with redundancy to multiple work areas. This means that the robots always have more objects available to handle compared to load balancing.
Scene	A scene represents all objects that are detected at a single recording of a sensor, that is, all the objects visible in one captured image. In production, a sensor generates a sequence of scenes.

This page is intentionally left blank

2 Installation

2.1 System requirements

Hardware and software requirements

Hardware requirements

Following are the hardware requirements

- CPU: 2.0 GHz or faster processor. Multicore processor is recommended.
- Memory: 8 GB RAM or more
- Free disk space: 5 GB or more recommended
- Graphics card: High performance DirectX 11 graphics card with support for feature level 10_1 or higher.
- Display settings: 1920 x 1200 pixels recommended
- DPI: Normal size (100% / 96 dpi) up to Larger size (150% /144 DPI)
- Mouse: Three-button mouse
- 3D Mouse (optional): Any 3D mouse from 3DConnexion

  See http://www.3dconnexion.com

Software requirements

Following are the software requirements:

- Windows 7 or Windows 8 operating system
- RobotStudio 5.60 or above
- RobotWare 5.60 or above

2.2 Installing the Picking PowerPac

Procedure

Installing Picking PowerPac

> **Note**
>
> Make sure that you have installed RobotStudio, RobotWare, and Virtual PickMaster on your computer before installing Picking PowerPac. For the installation procedure of RobotStudio, see *Operating manual - RobotStudio*.

Use the following procedure to install the Picking PowerPac:

1. Browse to the Picking PowerPac installation package and double-click `Setup.exe`.

   The installation starts.
2. Read the **license agreement** and accept the terms.
3. Click **Install**.
4. When the installation is complete, click **Finish**.

Installing Virtual PickMaster

Use the following procedure to install the Virtual PickMaster:

1. Browse to the Virtual PickMaster installation package and double click `Setup.exe`.

   The installation starts.
2. Read the **license agreement** and accept the terms.
3. Click **Install**.
4. When the installation is complete, click **Finish**.

   The Virtual PickMaster is installed.

Installing RobotWare

Use the following procedure to install the RobotWare:

1. Browse to the RobotWare installation package and double click `Setup.exe`.

   The installation starts.
2. Read the **license agreement** and accept the terms.
3. Click **Install**.
4. When the installation is complete, click **Finish**.

   The RobotWare is installed.

2.3 Installing a license

License installation

For the license installation procedure of RobotStudio and PowerPac, see *Operating manual - RobotStudio*.

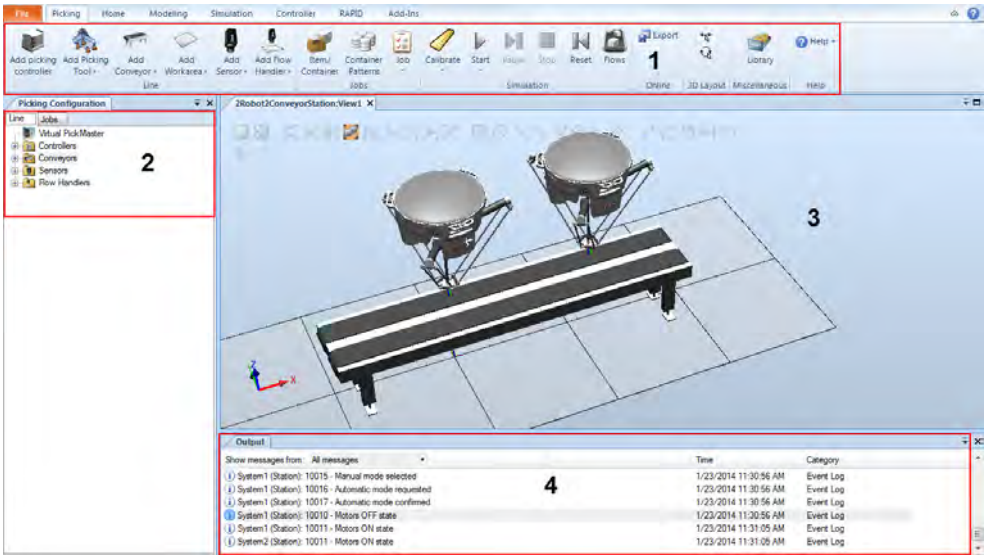
This page is intentionally left blank

3 Navigating the user interface

3.1 Introduction

Overview

This chapter describes about the user interface of the Picking PowerPac. The following figure and table provides information regarding the major elements in the user interface.



xx1300002150

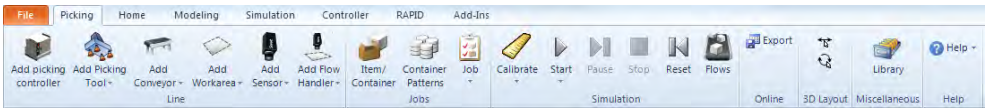
	Item	Description
1	Ribbon tab	Contains the general functions for Picking PowerPac. When creating a new picking application, the work flow is usually from left to right. For more details, see the section *Ribbon tab on page 20*.
2	Tree view browser	Organizes the programmable objects (for example, robots, sensors, and conveyors) of the picking application in a tree structure. It is separated into line and job tabs. For more details, see the section *Tree view browser on page 89*.
3	Station view	Realistic 3D display of the picking application. The objects in the station view are highlighted when selected or edited using the tree view browser.
4	Logs	Displays the log of the user actions when configuring the Picking PowerPac.

3.2 Ribbon tab

3.2.1 Introduction

Overview

The Picking PowerPac ribbon contains elements arranged in various groups. The following figure and table provides more information regarding the elements in the Picking PowerPac ribbon.



xx1300002151

Group	Button	Description
Line	**Add picking controller**	Allows you to create a picking controller and associate it with a robot system in the station view.
	Add Picking Tool	Allows you to create a picking tool.
	Add Conveyor	Allows you to create a conveyor.
	Add Workarea	Allows you to create a work area.
	Add Sensor	Allows you to create a sensor.
	Add Flow Handler	Allows you to create a flow handler.
Jobs	**Item/Container**	Allows you to create items and containers.
	Container Patterns	Allows you to create container patterns.
	Add Job	Allows you to create a job.
Simulation	**Calibrate**	Allows you to calibrate the conveyor areas.
	Start	Allows you start a simulation.
	Pause	Allows you pause the simulation.
	Stop	Allows you stop the simulation.
	Reset	Allows you reset the station view from objects temporarily created in the previously run simulation.
	Flows	Allows you to create flows of items and container patterns.
Online	**Export**	Allows you to export the PickMaster configuration to a folder.
3D Layout	**Move**	Allows you to drag an item relative to the active reference coordinate system.
	Rotate	Allows you to enable rotation around the various axes of an object determined by the reference coordinate system.
Miscellaneous	**Library**	Displays the contents of the library. You can save items, containers, and layouts in the library for reuse in other RobotStudio stations.

Continues on next page

Group	Button	Description
Help	**Help**	Provides the user manual of the Picking PowerPac
	About	Provides the general information about Picking PowerPac.

3.2.2 Line group

3.2.2.1 Add picking controller

Adding a picking controller

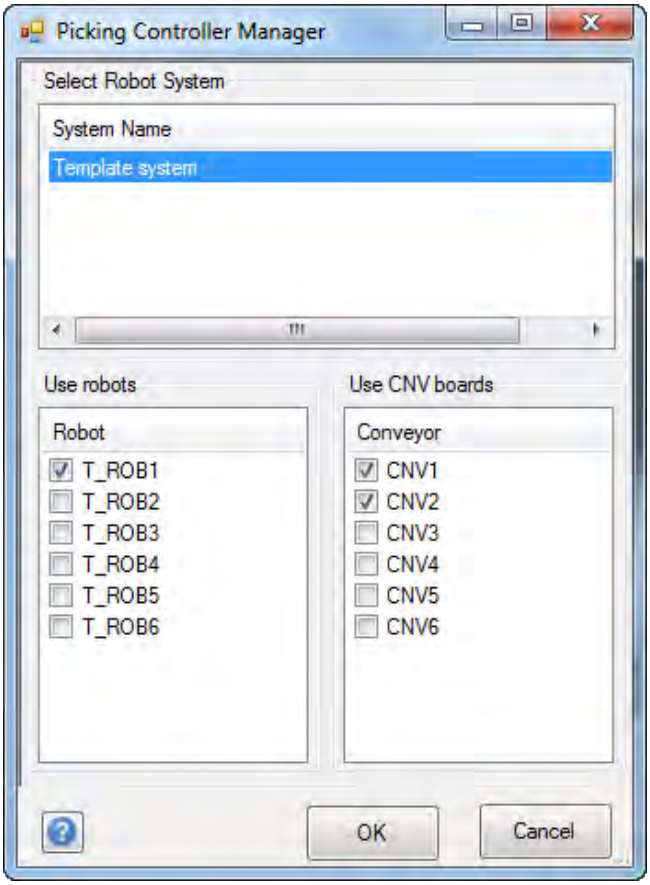
The **Add picking controller** button allows you to create a picking controller and associate it with a robot system in the station view. If there is no robot system available, a template robot system can be selected. Later, the picking controller can be associated with another robot system.

xx1300002152

To add a picking controller:

1 In the **Line** group, click **Add picking controller**.

The **Picking Controller Manager** window is displayed.



xx1300002162

2 Select a robot system from the **Select Robot System** list.

The robots and conveyor boards available with the selected robot system are displayed in the **Use robots** and **Use CNV boards** list.

Continues on next page

3 Select the required robots from the **Use robots** list.

4 Select the required conveyor boards from the **Use CNV boards** list.

One conveyor board is needed for every conveyor work area that shall be used with the picking controller.

5 Click **OK**.

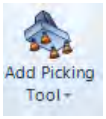
The picking controller is added.

3.2.2.2 Add Picking Tool

Adding a picking tool

Procedure

The **Add Picking Tool** button allows you to create a picking tool to be attached to a robot.

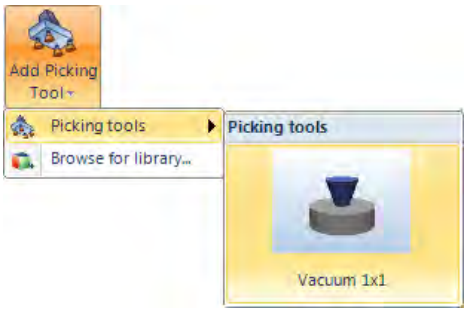


xx1300002153

To add a picking tool:

1 In the **Line** group, click **Add Picking Tool**.

The available tools are displayed.



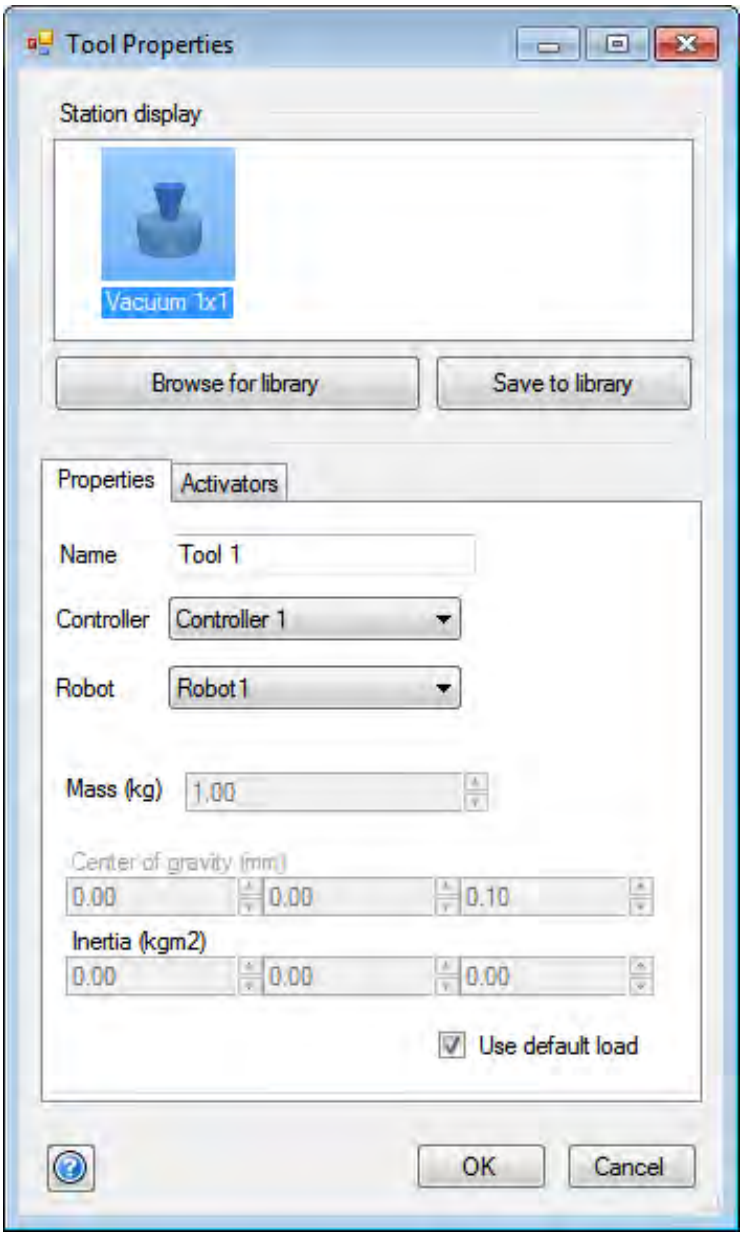
xx1300002163

2 Click **Picking tools** > **Picking tools** and select a tool. This selection can be changed later.

> **Note**
>
> You can also use the **Browse for library** option to select a tool. Using the library files allows you to use the predefined values or customized graphics.

Continues on next page

The **Tool properties** window is displayed.



xx1300002164

3 Configure the settings in the **Properties** and the **Activators** tab.

For more details, see the sections *The Properties tab on page 26* and *The Activators tab on page 28*.

Note

You can change the tool to a customized tool using the **Browse for library** button.

4 Click **OK**.

Continues on next page

3.2.2.2 Add Picking Tool
Continued

The selected tool is added.

Note

You can also save the tool to the library using the **Save to library** button.

Export and import the picking tool

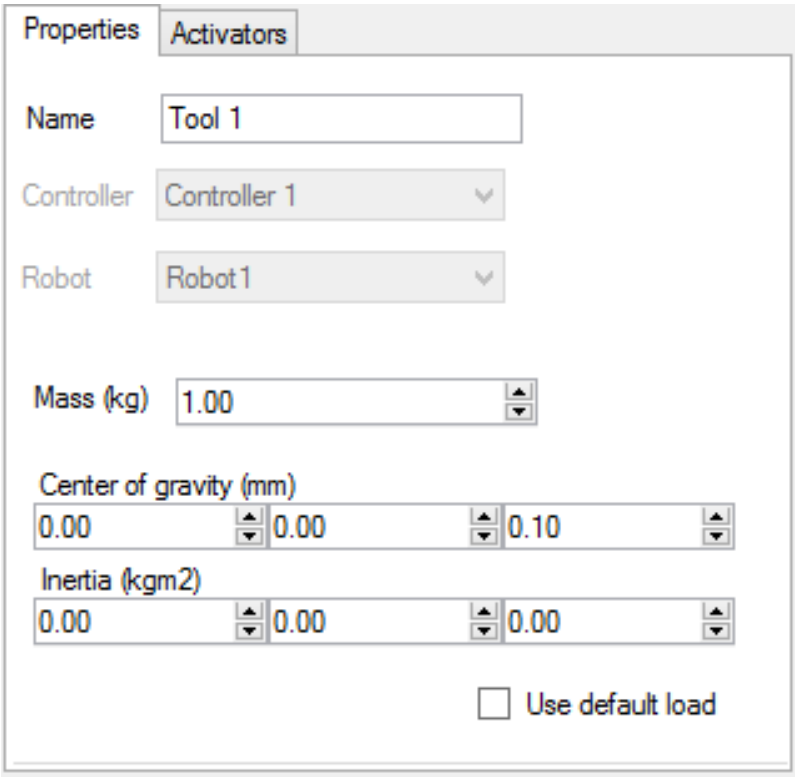
To export a picking tool for later reuse, click **Save to library**, type a filename, and select a folder. Two files with the extensions `*.xml` and `*.rslib` are created in the selected folder. The `*.xml` file contains all data and the `*.rslib` file contains the graphics.

To import a picking tool, click **Browse for library**, browse to the folder, and select the `*.xml` file. The graphics is automatically included if the `*.rslib` file is available in the same folder.

Customize the tool graphics

To import the tool graphics (and keep the other data), click **Browse for library** and select an `*.rslib` file. Any library file can be used.

The **Properties** tab



xx1300002297

	Description
Name	Type a name for the tool in the **Name** field.
Controller	Select a controller from the **Controller** list.
Robot	Select a robot from the **Robot** list.
Mass (kg)	Type the mass of the tool in the **Mass (kg)** field.

Continues on next page

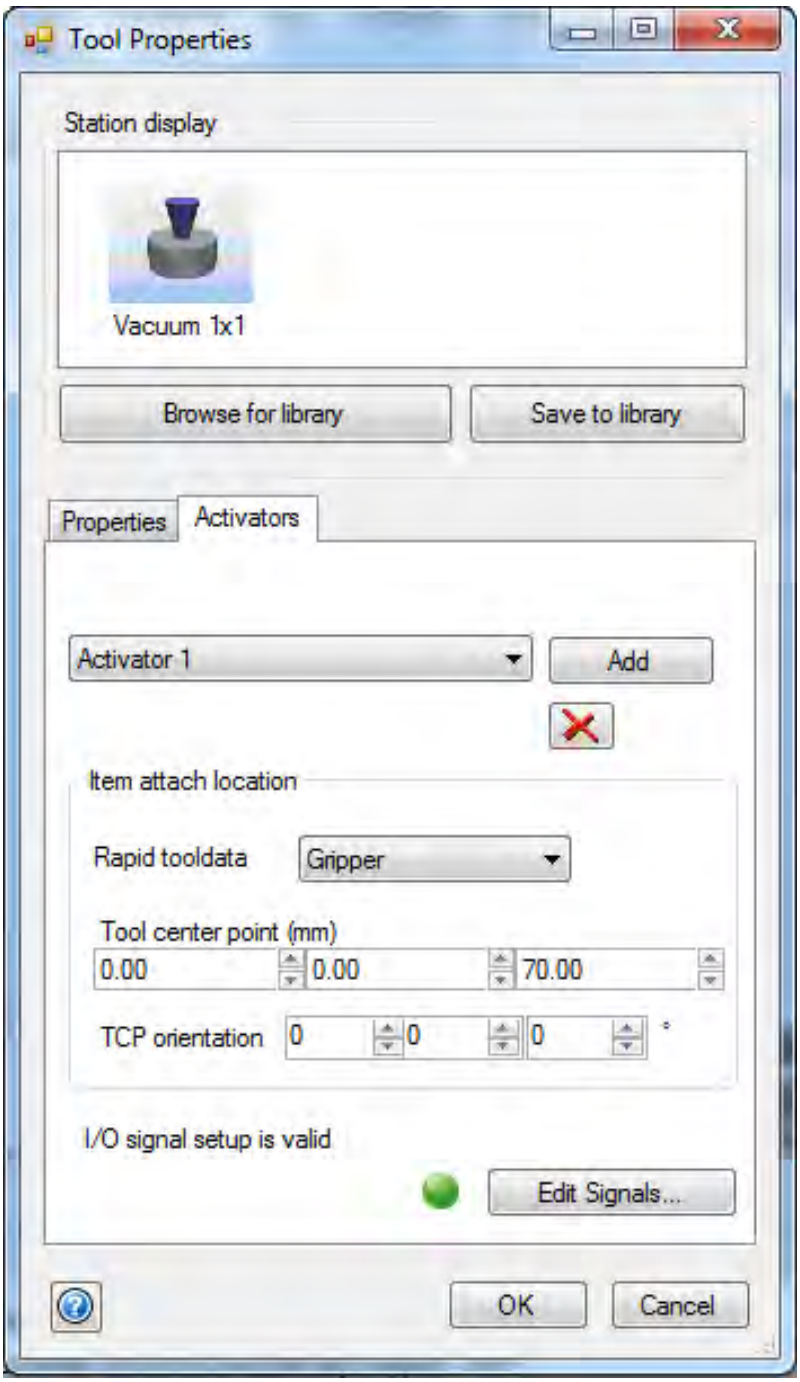
	Description
Center of gravity (mm)	Type the coordinates of the center of gravity.
Inertia (kgm2)	Type the values of the inertia in **Inertia (kgm2)**.
Use default load	Select the **Use default load** check box to use the default load configuration.

Continues on next page

3.2.2.2 Add Picking Tool
Continued

The **Activators** tab

The **Activators** tab allows you to configure the tool activators. An activator is a location on the tool where a picked up item is attached. To pick up single items, one activator is enough. Picking up multiple items requires one activator for each item to be picked up simultaneously.



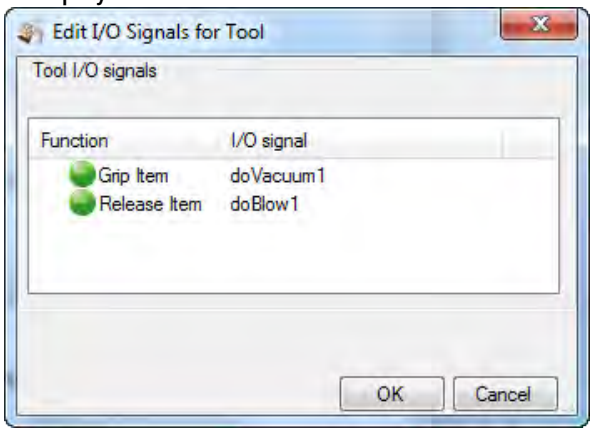
xx1300002296

Continues on next page

	Description
Activators	Select an activator from the Activators list. Click the **Add** button to add a new activator. Click the delete button to delete a selected activator.
Tool data	Select a RAPID tooldata from **Tool data**. The selected tooldata shall be used by the RAPID program when picking with this activator. **Note** The RAPID program needs to be updated if more than one activator is used. For more details see, *Example: Double pick single place on page 122*.
Tool center point (mm)	Type the coordinates of the tool center point. The tool center point defines the location on the tool where an item is attached. **Note** The coordinates are applied to the selected tooldata during the simulation.
TCP orientation	Type the orientation of the tool center point. The TCP orientation defines the desired orientation of the tool while picking up an item. The orientation shall be specified as Euler XYZ angles (degrees). **Note** The orientation is applied to the selected tooldata during the simulation.

Continues on next page

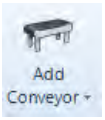
3.2.2.2 Add Picking Tool
Continued

	Description
Edit Signals	Allows you to manage the I/O signals. Click the **Edit Signals** button. The **Edit I/O Signals for Tool** window is displayed.  xx1300002298 For a selected **Function**, click on the **I/O signal** column to select a different signal from the list. Following are the details of the I/O signal configuration signal. • **Grip**: The output signal is set when an item shall be attached to the tool. **Note** The Grip signal is controlled from the RAPID program. In simulation, the RAPID triggdata SimAttachX controls when the signal is set. On a real robot, the RAPID triggdata `VacuumActX` controls when the signal is set. • **Release**: The output signal is pulsed when an item shall be detached from the tool. **Note** The Release signal is controlled from the RAPID program. In simulation, the RAPID triggdata SimDetachX controls when the signal is set. On a real robot, the RAPID triggdata `VacuumRevX` and `VacuumOffX` controls when the signal is set/pulsed.

3.2.2.3 Add Conveyor

Adding a conveyor

The **Add Conveyor** button allows you to create a conveyor.

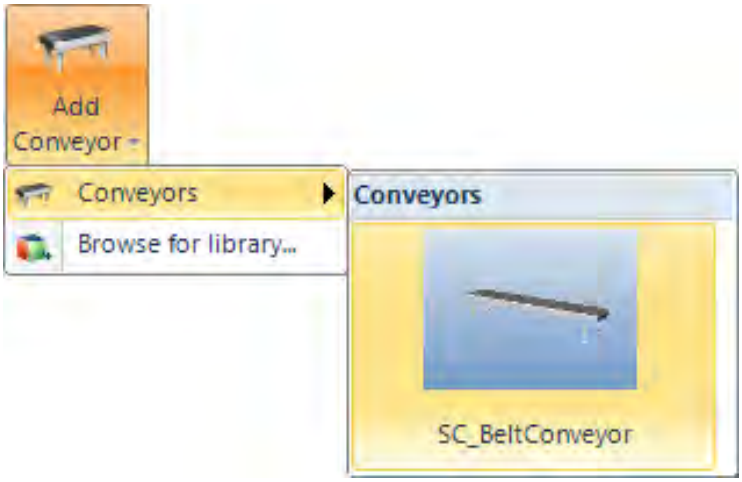


xx1300002154

To add a conveyor:

1 In the **Line** group, click **Add Conveyor**.

The available conveyor types are displayed.



xx1300002165

Note

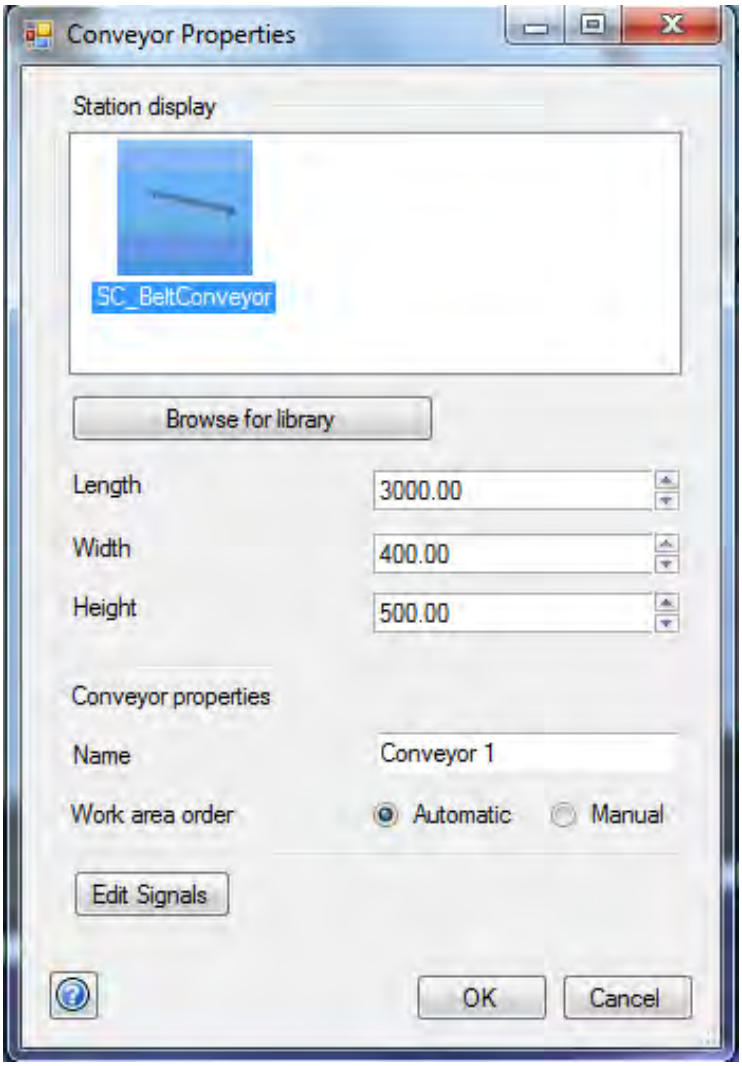
You can create a conveyor with customized graphics by using the **Browse Library** option.

2 Select a conveyor type from the options available in **ABB Conveyors** or use the **Browse Library** option to select a customized conveyor type.

Continues on next page

3.2.2.3 Add Conveyor
Continued

The **Conveyor Properties** window is displayed.



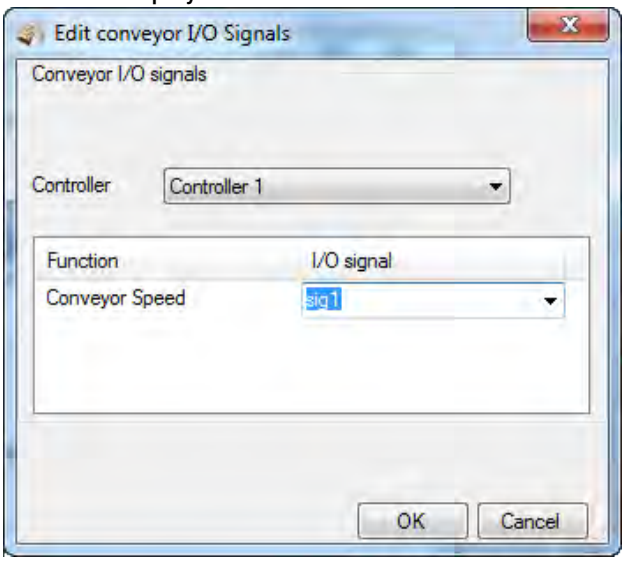
xx1300002166

3 Configure the required settings.

The following options are available in the **Conveyor Properties** window.

Configuration options	Description
Length	Type the length of the conveyor. This field also defines the end point of the conveyor where items and containers are removed during simulation.
Width	Type the width of the conveyor.
Height	Type the height of the conveyor.
Name	Type a name for the conveyor.

Continues on next page

Configuration options	Description
Work area order	Decides in which work area order an incoming flow will be handled. Automatic means, the order is auto generated based on the location of the robots along the conveyor. Manual means, the order is set manually for each work area. **Note** Using automatic is recommended. For example, if a manual selection is incorrect, the robot may try to place at a location that already was placed by an upstream robot.
Edit Signals	Allows you to manage the I/O signals. Click the **Edit Signals** button. The **Edit conveyor I/O Signals** window is displayed.  xx1300002442

4 Click **OK**.

The conveyor is added.

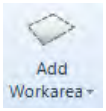
Customize the conveyor graphics

To import conveyor graphics (and keep the other data), click **Browse for library**, and select an `*.rslib` file. Any library file can be used.

3.2.2.4 Add Workarea

Adding a work area

The **Add Workarea** button allows you to create a work area.

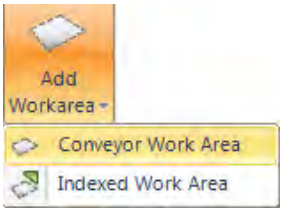


xx1300002155

To add a workarea:

1. In the **Line** group, click **Add Workarea**.

   The available work area types are displayed.



xx1300002167

   Conveyor Work Area is an area on a conveyor where the robot picks or places items. For more information see, *Conveyor work area on page 35*.

   Indexed Work Area is a fixed area (without conveyor tracking) where a robot picks or places items. For more information see, *Indexed work area on page 37*.

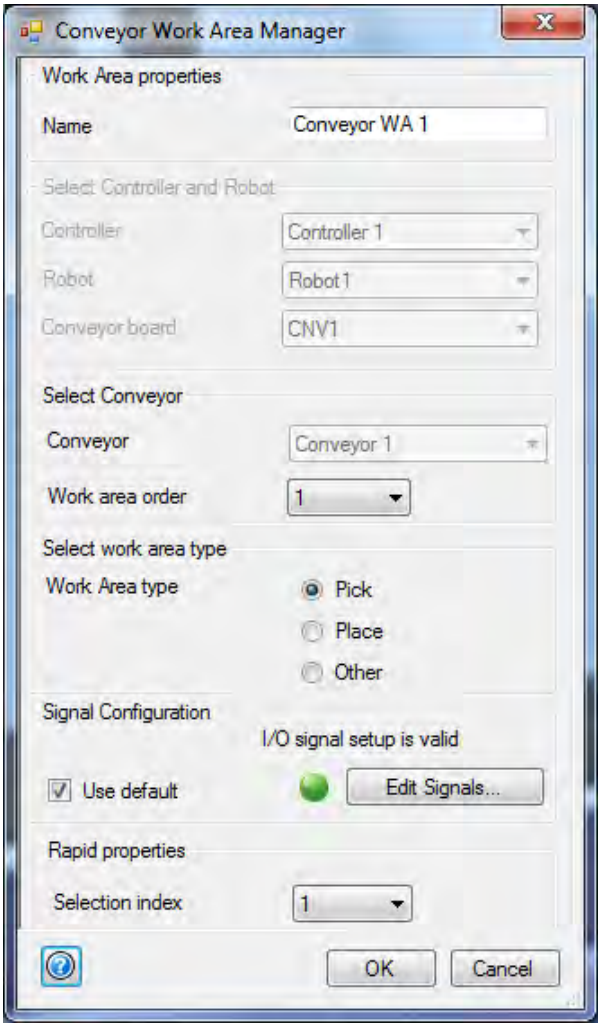
2. Select the required work area type and configure the settings.
3. Click **OK**.

Continues on next page

Conveyor work area

The conveyor work area is an area on the conveyor where the robot picks or places items. One conveyor board is required for each conveyor work area. A robot usually has only one conveyor work area on each related conveyor, but there is no restriction.

When you select **Conveyor Work Area** the **Conveyor Work Area Manager** window is displayed. The following figure and table provide information regarding managing the conveyor work area.



xx1300002170

	Description
Name	Type a name for the conveyor work area.
Controller	Select a picking controller from the list.
Robot	Select a robot from the list.
Conveyor board	Select a conveyor board from the list.
Conveyor	Select a conveyor from the list.

Continues on next page

3.2.2.4 Add Workarea
Continued

	Description
Work area order	Indicates the upstream order of this work area among other work areas on this conveyor. For example, “2” indicates that this is the second most upstream work area on this conveyor.
Work Area Type	Select work area type from the available options. • **Pick**: Select this if the work area is a picking area. • **Place**: Select this if the work area is a placing area. • **Other**: Select this if the work area is not used for pick or place with robot, for example, post inspection (For more details see, *Application manual - PickMaster 3*).
Signal Configuration	Configure the signals. Use the **Edit Signals** options to manage the signals. For more information regarding Conveyor work area signals see the following section. Select the **Use default** check box to use the default signal configuration.
Selection Index	Select an index to specify the pick or place order in the RAPID program when using more than one pick work area and one place work area with the selected robot

Conveyor work area signals

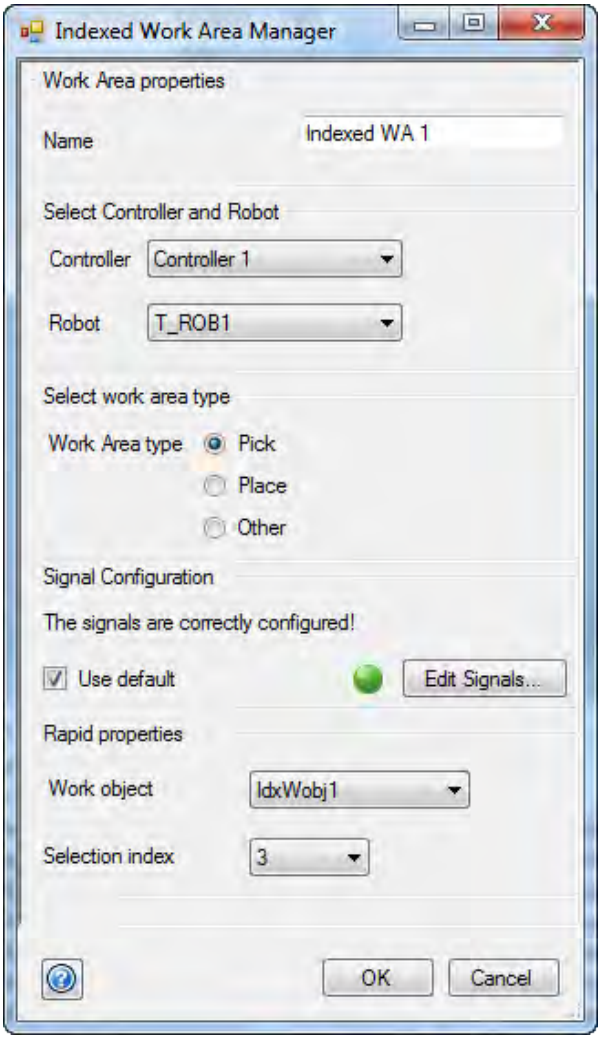
Signal	Description
Conveyor start/stop	Digital output signal. This signal is used if an overflow shall be avoided by letting the conveyor movement be controlled by the work area. The signal goes high when the conveyor shall start moving and goes low when the conveyor shall stop to avoid an overflow.
Queue idle	Digital output signal. This signal is high when the queue for this work area is empty. The signal goes high when the last item is retrieved from the queue.
Position available	Digital output signal. This signal is high when there is one or more items between the enter and exit limits for the work area.

Continues on next page

Indexed work area

An indexed work area is an area for picking or placing without conveyor tracking.

When you select the **Indexed Work Area** option the **Indexed Work Area Manager** window is displayed. The following figure and table provide information regarding managing the indexed work area.



xx1300002171

	Description
Name	Type a name for the indexed work area.
Controller	Select a controller from the list.
Robot	Select a robot from the list.
Work Area Type	Select work area type from the available options. • **Pick**: Select this if the indexed work area is a picking area. • **Place**: Select this if the indexed work area is a placing area. • **Other**: Select this if the work area is not used for pick or place with robot.

Continues on next page

3.2.2.4 Add Workarea
Continued

	Description
Signal Configuration	Configure the signals. Use the **Edit Signals...** options to manage the signals. For more information regarding indexed work area signals see the following section. Select the **Use default** check box to use the default signal configuration.
Work object	Select a RAPID work object data (`wobjdata`). The associated `wobjdata` is automatically used with the indexed work area. **Note** No work object calibration is needed. The selected `wobjdata` is automatically updated when a simulation is started.
Selection Index	Select an index to specify the pick or place order in the RAPID program when using more than one pick work area and one place work area with the selected robot.

Indexed work area signals

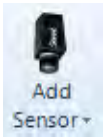
Signal	Description
Robot execution	This optional digital input I/O signal is used to indicate that it is allowed for the robot to execute an item target in the RAPID program. Execution starts when the signal is high and stops when the signal goes low. If the signal goes low, all remaining items in the currently executing scene is dropped, so when the signal goes high again, the item targets for the next scene is executed. The signal must also go low after one scene is finished and then go high again to start executing item targets for the next scene.
Queue idle	This output signal is high when the queue for this work area is empty. The signal goes high when the last item is retrieved from the queue.
Position available	This output signal is high when there are one or more items when the *Robot execution* signal is high for the work area. If no *Robot Execution* signal is used the `Position Available` signal will go high as soon as there are any items in the queue.

3.2.2.5 Add Sensor

Sensor configuration

Adding a sensor

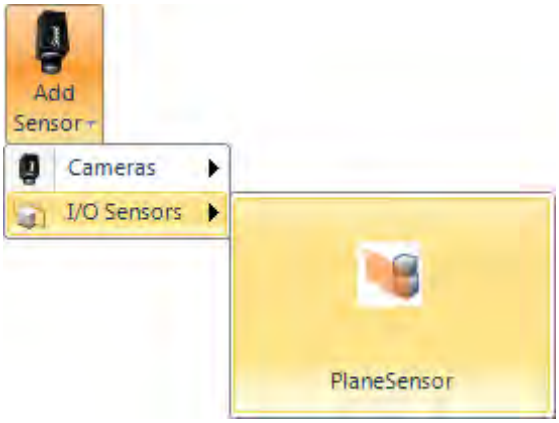
A sensor can either be a camera or an I/O sensor. The **Add Sensor** button allows you to create a sensor.



xx1300002156

To add a sensor:

1. In the **Line** group, click **Add Sensor**.
   The available sensor types are displayed.



xx1300002168

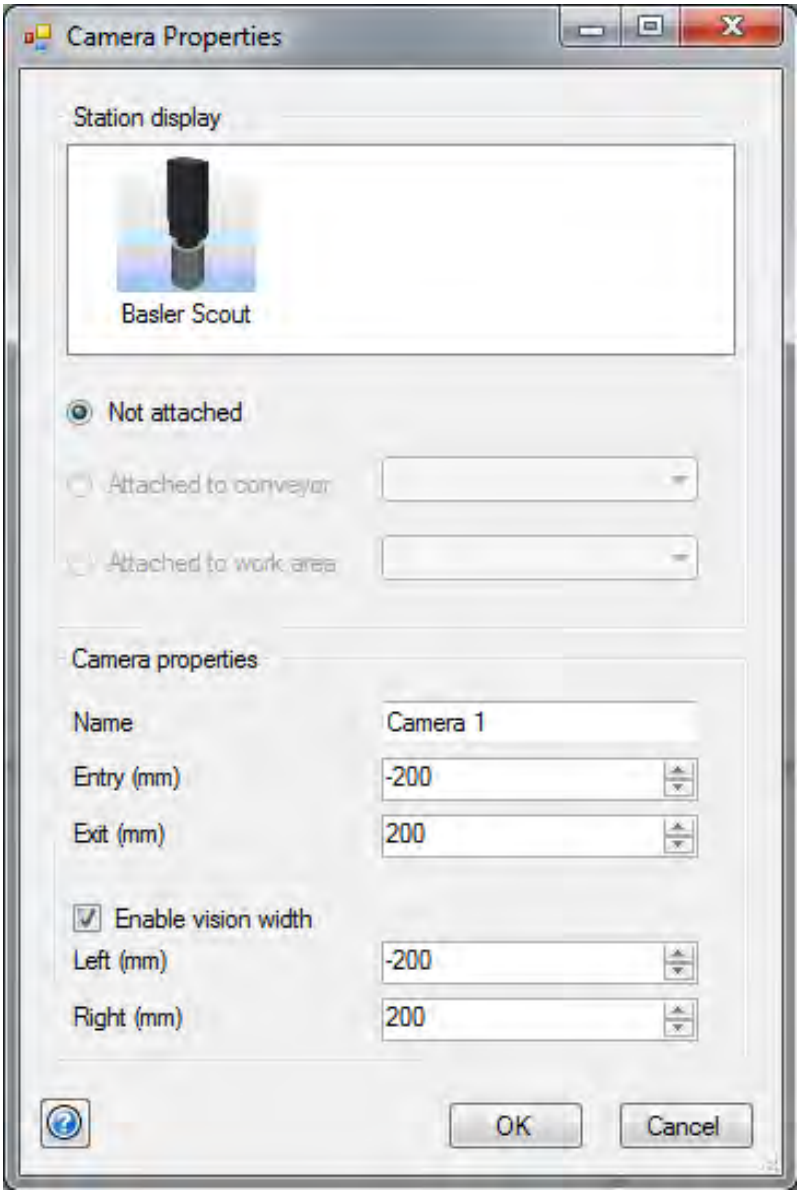
2. Select the required sensor type from the available options.
   The configuration window is displayed.
   For more information regarding **Cameras**, see *Cameras on page 40*.
   For more information regarding **I/O Sensors**, see *I/O sensors on page 42*.
3. Select the required option and configure the settings.
4. Click **OK**.
   The configured sensor is created.

Continues on next page

3.2.2.5 Add Sensor
Continued

Cameras

When you select the **Cameras** option the **Camera Properties** window is displayed. The following figure and table provide information about configuring a camera.



xx1300002172

	Description
Not attached	Select this if the camera shall not be attached to a conveyor or work area.
Attached to conveyor	Select this if the camera shall be attached to a conveyor.
Attached to work area	Select this if the camera shall be attached to an indexed work area.
Name	Type a name for the camera.
Entry (mm)	Type an entry limit for the visible area below the camera along a conveyor. A negative value is used if the visible area starts up-streams from the camera location.

Continues on next page

	Description
Exit (mm)	Type an exit limit for the visible area below the camera along a conveyor. A positive value is used if the visible area ends down-streams from the camera location.
Enable vision width	Select this to enable a width limitation of the visible area.
Left (mm)	Type a limit value for the left side of the visible area. A negative value is used if the visible area ends on the left side of the camera location (from an upstream viewpoint).
Right (mm)	Type a limit value for the right side of the visible area. A positive value is used if the visible area ends on the right side of the camera location (from an upstream viewpoint).

Note

The visible area is not limited if the camera is used with an indexed work area .

Note

The camera will not detect any objects created or placed on the other conveyors or indexed work areas.

Tip

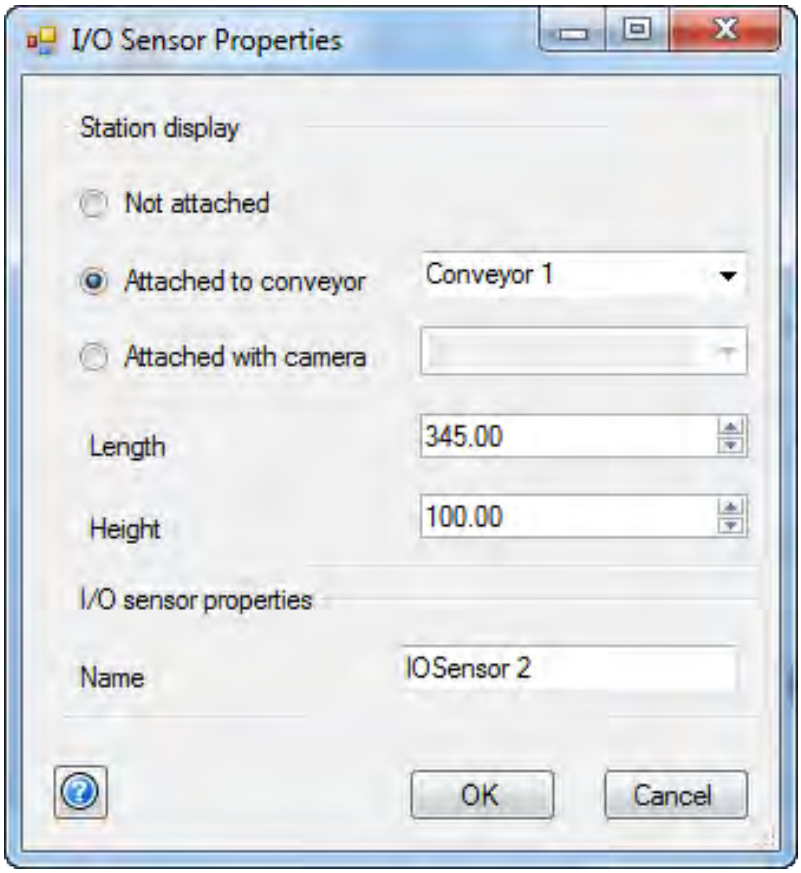
Check the built-in camera spotlight which can be used to monitor when the camera images are captured. To see the spotlight, activate **Advanced Lighting** in the **Graphic Tools** tab. The **Graphic Tools** tab can be selected in the **Home** tab. The spotlight is enabled by default but can be disabled in the **Graphics** tree view.

Continues on next page

3.2.2.5 Add Sensor
Continued

I/O sensors

When you select the **I/O Sensors** option the **I/O Sensor Properties** window is displayed. The following figure and table provide information about configuring an I/O sensor.



xx1300002173

	Description
Not attached	Select this if the I/O sensor shall not be attached to a conveyor.
Attached to conveyor	Select this if the I/O sensor shall be attached to a conveyor.
Attached with camera	Select this if the I/O sensor shall be attached with a camera to a conveyor.
Length	Type the length of the I/O sensor.
Height	Type the height of the I/O sensor.
Name	Type a name for the I/O sensor.

Note

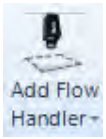
To function correctly, an I/O sensor must not be in contact with other stationary objects, for example, the conveyor.

3.2.2.6 Add Flow Handler

Flow handler configuration

Adding flow handler

The **Add Flow Handler** button allows you to create a flow handler.



xx1300002157

A flow handler consists of a sensor for object detection and one or more work areas to pick or place the detected objects. A conveyor flow handler is used to handle objects located on a conveyor while an indexed flow handler is used to handle objects on an indexed work area.

To add a flow handler.

1. In the **Line** group, click **Add Flow Handler**.

   The available options are displayed.

xx1300002169

2. Select the required flow handler type from the available options.

   The configuration window is displayed.

   For more information regarding **Conveyor Flow Handler** see, *Conveyor flow handler on page 44*.

   For more information regarding **Indexed Flow Handler** see, *Indexed flow handler on page 46*.

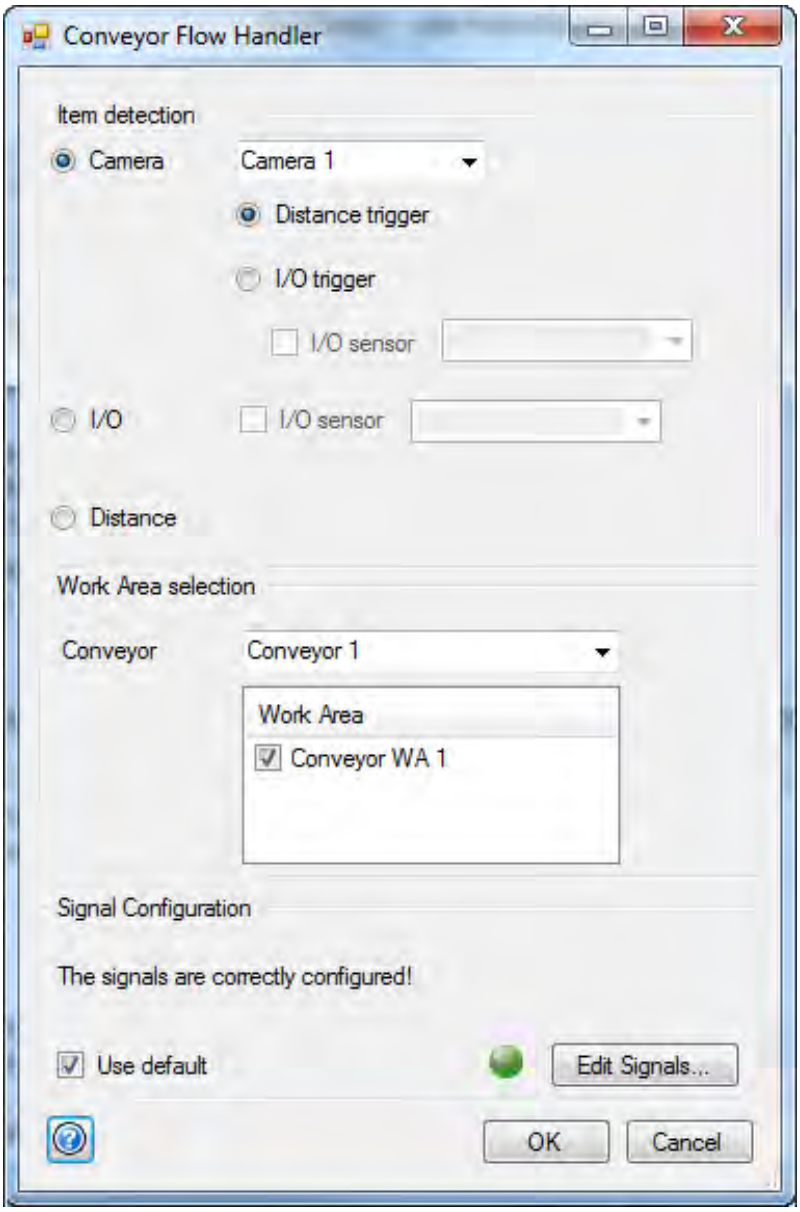
3. Select the required flow handler type and configure the settings.
4. Click **OK**.

Continues on next page

3.2.2.6 Add Flow Handler
Continued

Conveyor flow handler

The **Conveyor Flow Handler** window allows you to manage a conveyor flow handler. The following figure and table provide information about configuring the conveyor flow handler.



xx1300002174

Continues on next page

		Description
Item detection To detect items on a conveyor you can select either **Camera**, **I/O**, or **Distance**.	**Camera**	If **Camera** is selected: • Select a camera from the list. • Select a trigger type to define how the camera capture shall be triggered. You can use **Distance trigger** or **I/O trigger**. **Distance trigger** means that the camera captures a scene as the conveyor moves a certain distance, for example, 150 mm. The trigger distance is configured in the job settings for the conveyor flow handler. **I/O trigger** means that the camera captures a scene when an I/O signal (the position generator signal) is set. For this option it is recommended to use an I/O sensor to handle the triggering of the camera.
	I/O	If **I/O** is selected: • Items are detected when an I/O signal is set (for example, the position generator signal). For this option, it is recommended to use an I/O sensor to handle the item detection.
	Distance	If **Distance** is selected: • Items are generated instead of detected. New items are generated when the conveyor has travelled a certain trigger distance. The trigger distance is configured in the job settings for the conveyor flow handler.
Work Area selection	**Conveyor**	Select the conveyor in which the item detection should be enabled.
	Work Areas	Select the conveyor work areas where detected items shall be picked or placed.
Signal Configuration		Configure the signals. Use the **Edit Signals...** option to manage the signals. Select the **Use default** check box to use the default signal configuration.

Conveyor flow handler signals

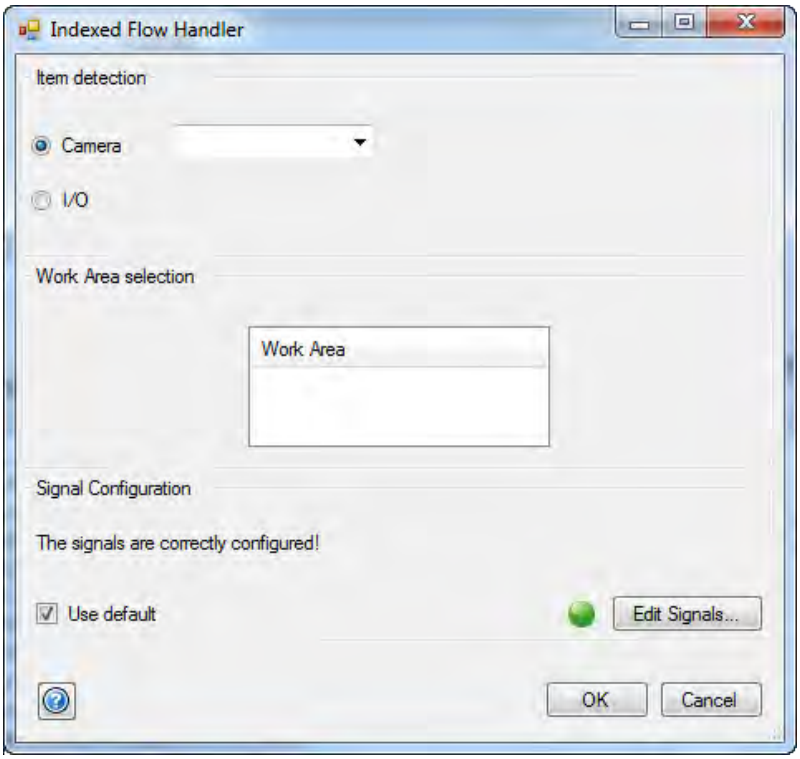
Signal	Description
Position generator	Digital input signal. This is defined for each selected work area. If an I/O sensor is used with the conveyor flow handler, the signal is used to indicate the detection of new items or containers. If an I/O triggered camera is used with the conveyor flow handler, the signal is used to generate captures of the new scenes.
Trig	Digital output signal. This is defined for each selected work area. If a camera is used with the conveyor flow handler, the signal is used to trigger captures of new scenes. If an I/O sensor is used with the conveyor flow handler, the signal is used to register sensor registrations of new items/containers.

Continues on next page

3.2.2.6 Add Flow Handler
Continued

Indexed flow handler

The **Indexed Flow Handler** window allows you to manage an indexed flow handler. The following figure and table provide information about configuring the indexed flow handler.



xx1300002175

		Description
Item detection There are two different ways to detect items on an indexed work area. You can use a **Camera** or **I/O**. Select an option from this section.	**Camera**	If Camera is selected: • Select which camera to use. The camera captures a scene when an I/O signal (for example, the position generator signal) is set. In simulation this happens as soon as new items or empty place locations for items has been generated in the indexed work area. The generation of new items is made when the robot has picked or placed all available items. The generation of new items can be delayed by setting a Load Time for the indexed work area in the job tree view, Assign Flow menu selection.
	I/O	If I/O is selected: • Items are detected when an I/O signal is set (for example, the position generator signal). In simulation this happens as soon as new items or empty place locations for items has been generated in the indexed work area. The generation of new items is made when the robot has picked or placed all available items. The generation of new items can be delayed by setting a Load Time for the indexed work area in the job tree view, Assign Flow menu selection.
Work Area selection	**Work Area selection**	Select one indexed work area in which item detection should be enabled and where items shall be picked or placed.

Continues on next page

		Description
Signal Configuration	**Signal Configuration**	Configure the signals. Use the **Edit Signals...** option to manage the signals. Select the **Use default** check box to use the default signal configuration.

Indexed flow handler signals

Signal	Description
Position generator	Digital input signal. This is defined for each selected work area. If I/O is used with the indexed flow handler, the signal is used to indicate the detection of new items or containers. If an I/O triggered camera is used with the indexed flow handler, the signal is used to trigger captures of the new scenes.
Trig	Digital output signal. This is defined for each selected work area. If a camera is used with the indexed flow handler, the signal is used to trigger the captures of the new scenes. If I/O is used with the indexed flow handler, the signal is used to register sensor registrations of new items/containers.
Strobe	Digital input signal. This is defined for each selected work area. If a camera is used with the indexed flow handler, the signal is used to synchronize the robot with the scenes captured with the camera. If an I/O sensor is used with the indexed flow handler, the signal is used to synchronize the robot with sensor registrations of new items/containers.

3.2.3 Jobs group

3.2.3.1 Item/Container

Adding an item/container

Procedure

The **Item/Container** button allows you to create items and containers.

xx1300002158

Note

Items can be picked and placed by robots but this is not possible for the containers. Items can be placed in containers or they can be picked from the containers.

Note

The configured weight of the items is not considered by the simulation. To consider the weight, the RAPID program must be manually updated. For more details, see *Public variables in PickMaster template programs on page 111*.

To add items and containers:

1. In the **Jobs** group, click **Item/Container**.

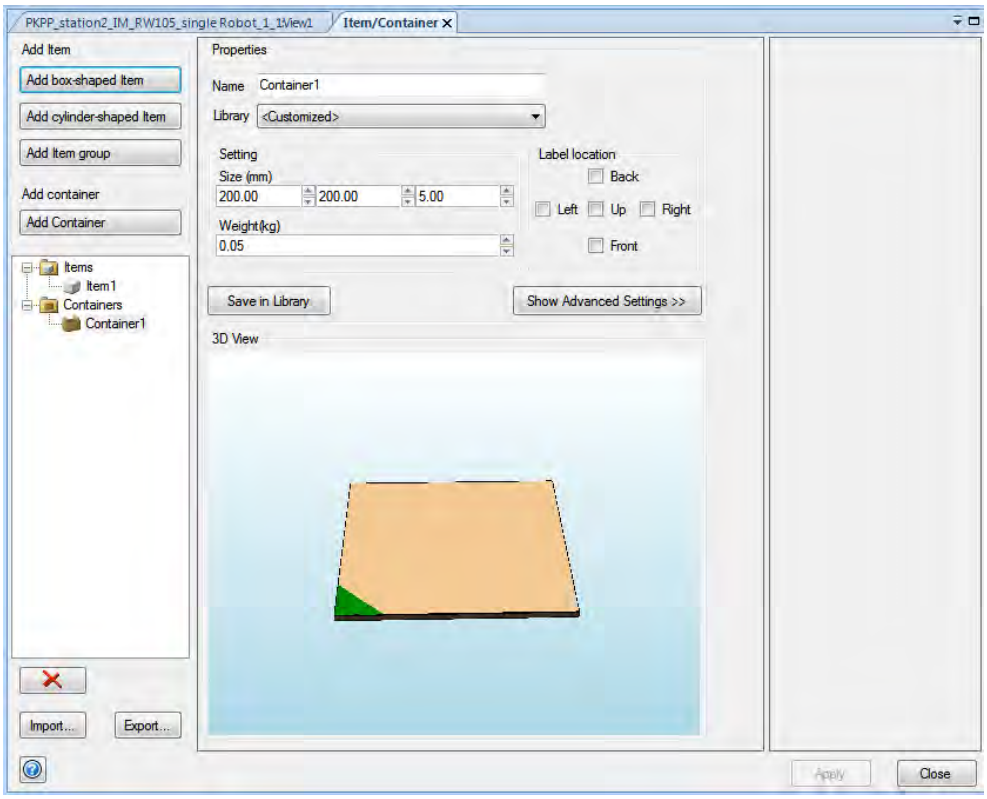
   The **Item/Container** window is displayed.
2. In the **Add Item** section, click **Add box-shaped Item** and **Add cylinder-shaped Item** to add the required items.
3. In the **Add container** section, click **Add Container** to add the containers.
4. Click **Apply**.

   The items and containers are saved.
5. Click **Close**.

Continues on next page

Item/Container properties

The following figure and table provide information regarding the **Properties** window.



xx1300002181

	Description
Add box-shaped Item	Allows you to to add a box-shaped item.
Add cylinder-shaped Item	Allows you to add a cylinder-shaped item.
Add item group	Allows you to add an item group. An item group is a special item type that is built up by a pattern with the other items. An item group can be picked and placed. It can be added to container patterns, flows, and jobs.
Add Container	Allows you to add a container.
Browser	Displays in a tree structure the items and containers that you are adding or importing. Select an item or container from the tree structure to customize its properties.
Delete button	Select an item or container in the browser window and click the delete button to delete it.
Import	Allows you to import saved items and container from the library.
Export	Allows you to export items and containers to the library.
Name	Allows you to change the name for the selected item or container.
Library	Allows you to select an item or container from the library. The properties are updated from the selection.
Setting	Allows you to configure the physical dimensions and weight for the selected item or container.

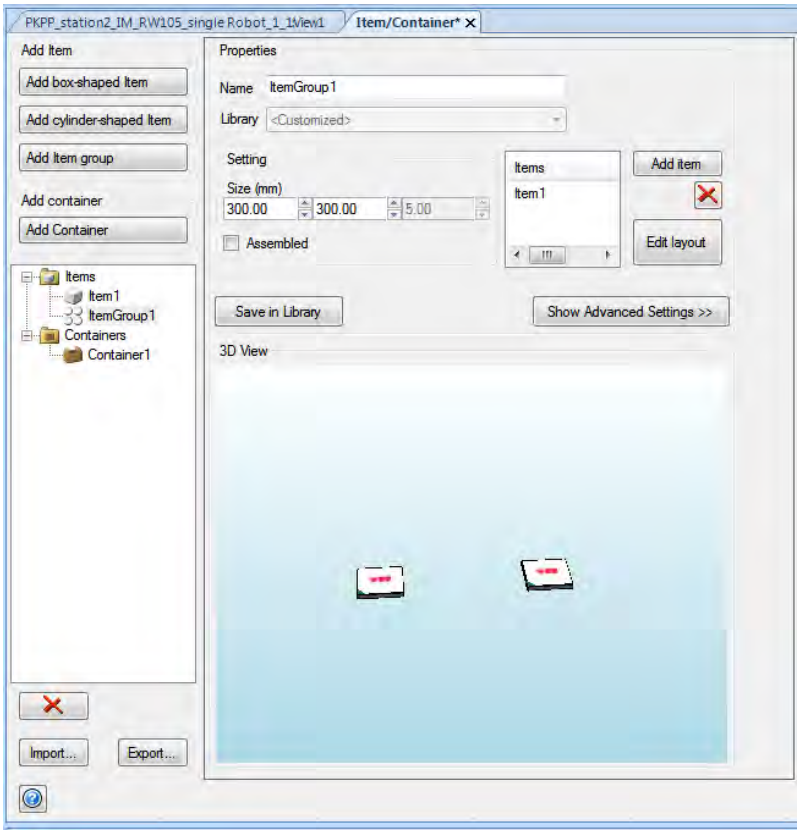
Continues on next page

3.2.3.1 Item/Container
Continued

	Description
Label location	Allows you to define the desired locations for the label picture on the selected item or container.
Save in Library	Saves the item or container in the library.
Show Advanced Settings	Displays the advanced settings.

Item group properties

The following figure and table provide information regarding the Properties window for an item group.



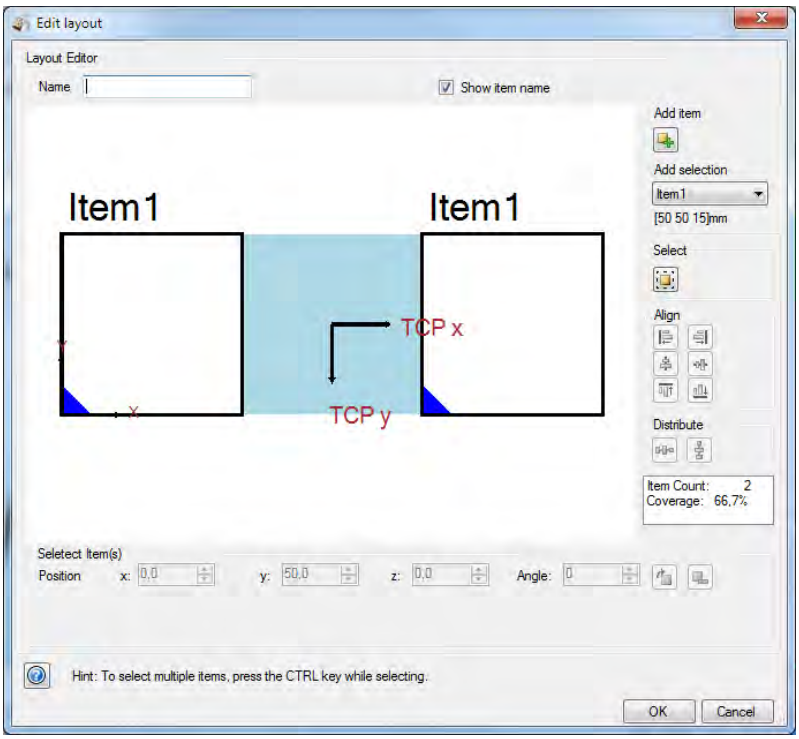
xx1500000404

	Description
Name	Allows you to change the name.
Library	Allows you to select an item group from the library. The properties are updated from the selection.
Size	Allows you to configure the size of the item group. **Note** The grip location is affected by the XY size. For proper gripping, it is recommended to use a size that is just enough to contain the layout items. By default, the item group is attached to the picking tool in the center of the rectangular XY area. The grip location can be further adjusted from the center point in the **Advanced Settings**.

Continues on next page

	Description
Assembled	Allows you to pick and place the complete group with a single tool activator. If the **Assembled** check box is not selected, items in the group are picked and placed individually.
Save in Library	Saves the item group in the library.
Add item	Adds an item that can be used in the item group.
Delete button	Removes an item from the item group.
Edit layout	Allows you to edit the item group layout. You can add, remove, or adjust the items. For more details, see *Edit layout on page 56*.
Show Advanced Settings	Displays the **Advanced Settings** tab. You can define the Pick settings for the item group.

Following is an example of an item group layout with a suitable size. Item size is 50 x 50 mm. The gap between the two items is 50 mm. The item group size is just enough to cover the two items, 150 x 50 mm.



xx1500000405

Note

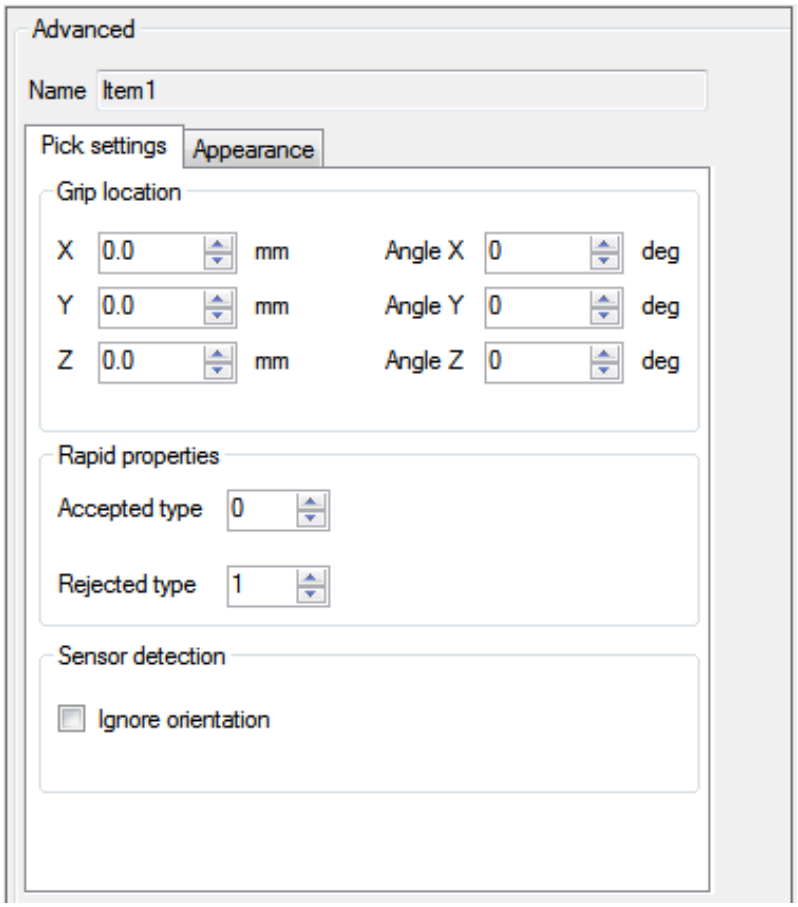
The location of the tool center point of the picking tool when handling the item group is indicated. This can be used as a guidance to define the different TCP offsets that is needed in the rapid program. For example, when programming multiple picks of items followed by a single place of an item group, a separate tcp offset is needed for each item to be picked up into the group. A tcp offset is also needed for placing the complete item group. The tcp offset used for picking the individual items must match the tcp offset of the item group.

Continues on next page

3.2.3.1 Item/Container
Continued

Advanced settings - **Pick settings** tab

> **Note**
>
> The **Pick settings** tab is available only for items, not for containers.

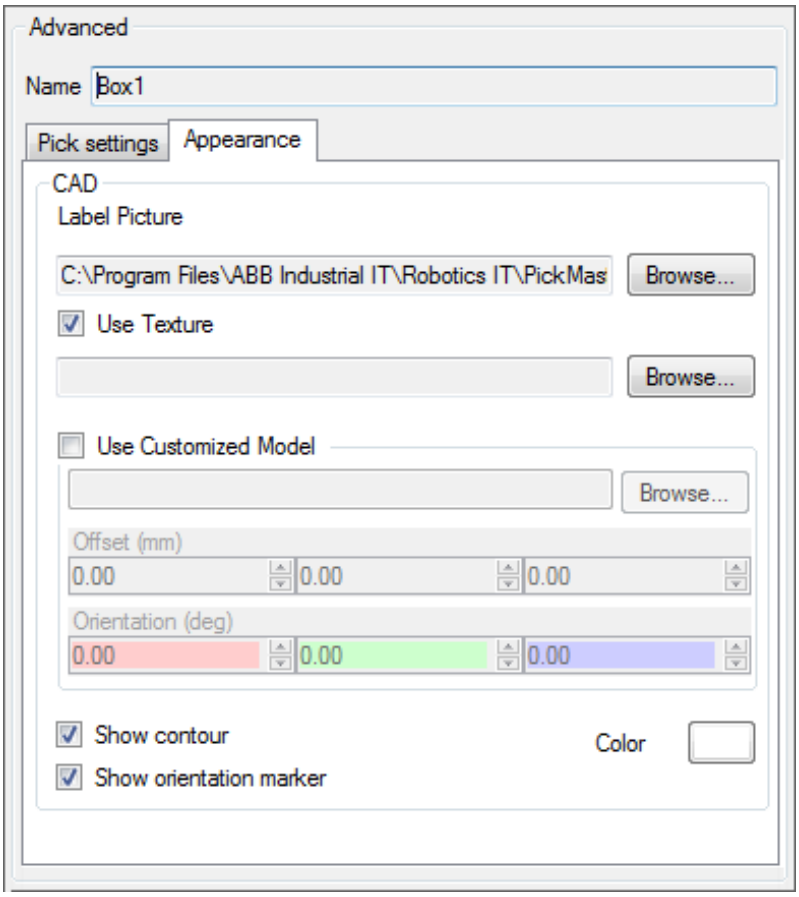


xx1300002187

	Description
Name	Displays the name of the selected item.
Grip location	Allows you to configure the location of the spot where the item is attached to the gripper. If X = 0, Y = 0, the item's center is attached to the tool center point. An item group's center is the the center of the layout.
Rapid properties	Allows you to define the values for accepted or rejected item types. The values for the accepted and rejected item type are sent to the RAPID program and are supplied with the item targets. For more details see, *GetItmTgt - Get the next item target on page 136*.
Ignore Orientation	Allows you to ignore the orientation when an item is detected with a camera. When selected, picking of the camera detected items is always be made with the same orientation of the robot tool. As a consequence, the item orientation in the gripper and at the place location is affected by the actual orientation of the item when being picked up.

Continues on next page

Advanced settings - **Appearance** tab



xx1300002188

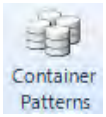
	Description
Label picture	Allows you to select an image file for the label picture.
Use Texture	Allows you to use a texture image file for the item or container.
Use Customized model	Allows you to use a customized graphic model compiled in the .rslib file format.

3.2.3.2 Container Patterns

Adding a container pattern

Procedure

The **Container Patterns** button allows you to add and manage container patterns.



xx1300002159

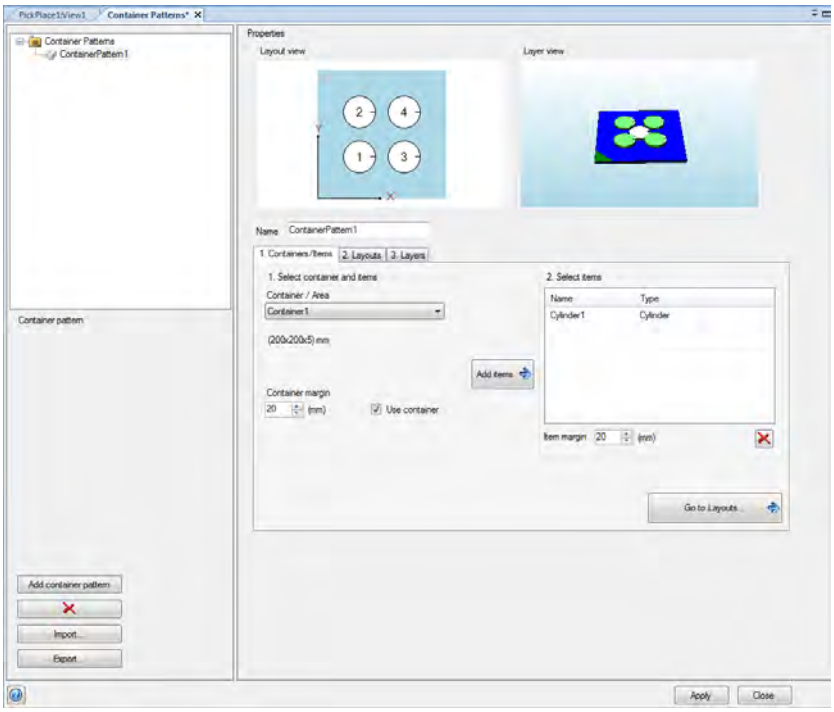
To add a container pattern:

1. In the **Job** group, click **Container Patterns**.

   The **Container Patterns** window is displayed.

2. Click **Add container pattern** in the **Container pattern** section.

   An empty container pattern is added to the browser and visualized in the Layer view. The **Containers/Items** tab is displayed.



xx1300002182

3. Select a container from the **Container/Area** list.

> **Note**
>
> It is also possible to create a container pattern without a container object by clearing the **Use Container** check box.

4. Click **Add items**.

   The **Select items** window with the available items are displayed.

Continues on next page

5 Select one or more items to be used in the container pattern and click **OK**.

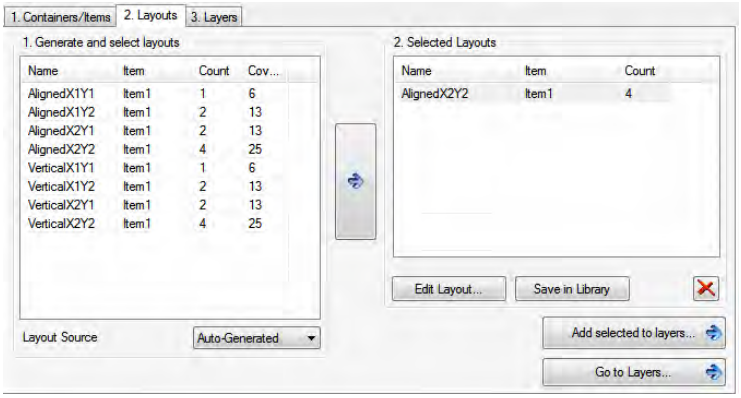
The items are added to the **2. Select items** section.

> **Note**
>
> You can remove an added item using the delete button.

> **Note**
>
> **Item margin** defines the minimum distance between different items in the auto generated layouts. **Container margin** defines the minimum distance between the items and the container border in the auto generated layouts. Layouts can also be selected from the library.

6 Click **Go to Layouts** or select the **Layouts** tab.

The **Layouts** tab is displayed. The **Generate and select layouts** list displays some auto generated layouts with the selected items.



xx1500000410

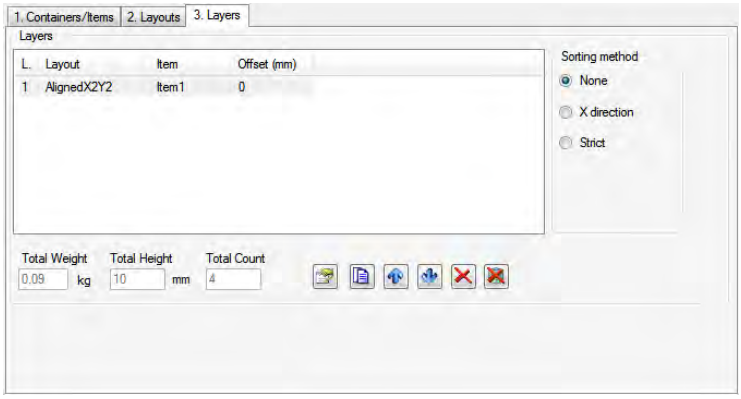
7 Select one or more suitable layouts from the **Generate and select layouts** list and click on the arrow to add them to the **Selected Layouts** list. To edit a layout see the section *Edit layout on page 56*.

8 To add a new layer, select a layout in the **Selected Layouts** list and click **Add selected to layers**.

Continues on next page

3.2.3.2 Container Patterns
Continued

The **Layers** tab is displayed.



xx1500000411

9 Open the **Layer** tab. You can change the order, delete, or rearrange the selected layers using the available options. You can adjust the vertical position of each layer by modifying the **Offset** (mm). You can also manage the sorting method. The **Sorting Method** section defines the order in which the items in the container pattern shall be handled by the robots. Following are the options available in **Sorting method**.
   - **None**: The items in the layer shall be accessed in the same order as they are defined in the layout for each layer, but if the next item cannot be reached the next one after that is used.
   - **X direction**: The items shall be accessed in the X direction for each layer, that is, in the order they travel along a conveyor.
   - **Strict**: The items shall be used in the same order as they are defined in the layout for each layer. If a robot cannot access the next item position in a layer, that robot does not use any more item positions in the container pattern.

10 Click **Apply**.
   The container pattern is saved.

11 Click **Close**.
   The window is closed.

Edit layout

The following layout types are used in Picking PowerPac:

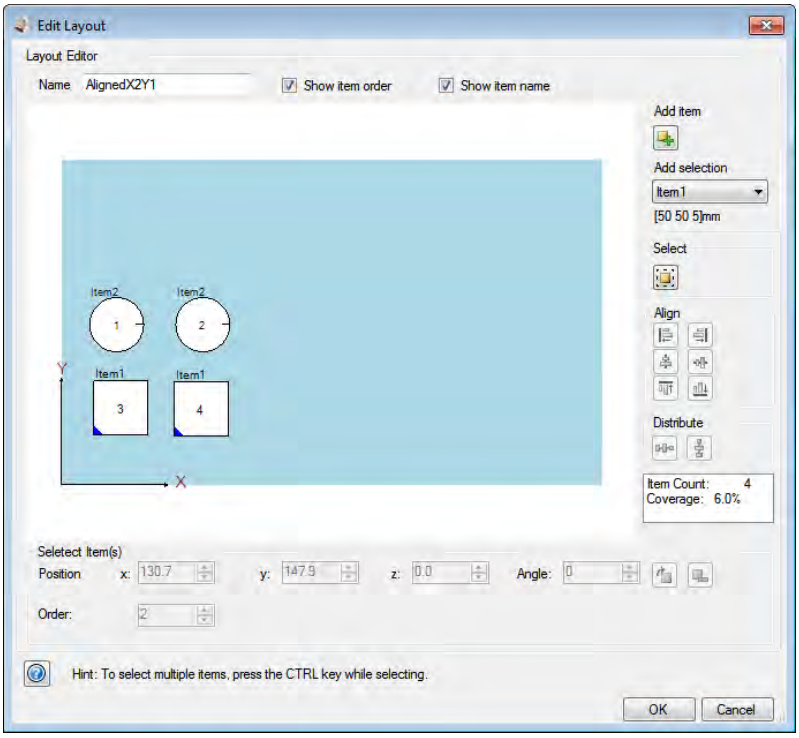
- Container pattern layouts.
- Flow layouts.
- Item group layouts.
- Library layouts.

The different layout types are edited in the same way with some minor variations. For example, to edit a Container pattern layout, select the layout in the **Selected Layouts** list and click **Edit Layout**.

You can change the position and orientation of individual or grouped items. Items can be moved using the mouse or keyboard. They can be added or deleted.

Continues on next page

The following figure and table provides information regarding the **Edit Layout** window.



xx1300002189

The TCP location (not shown in this image) is indicated for Item group layouts.

	Description
Name	Allows you to type a name for the layout.
Show item order	Select this check box to display the order of the items. The order number defines the default access order in which the items are handled by the robots. This is available only for the Container pattern layouts.
Show item name	Select this check box to display the name of the items. For Flow layouts, the names of Container patterns are also displayed.
Add item section	Select an item from the **Add selection** list and click the button to add the selected item. It is also possible to add Container patterns from this list for Flow layouts.
Select section	Click the button to select all the items. To select an item, click on it. To select multiple items, press the CTRL key while clicking on the items. The reference item (the latest selected item) in a selected group is displayed with a darker color.
Align section	Allows you to align the selected items. The items are aligned with the reference item.
Distribute section	Allows you to distribute the selected items. The items are distributed from the reference item.
Position	Displays the coordinates of the selected item. You can modify the coordinates according to your requirement.
Angle	Displays the angle of the selected item. You can modify the angle according to your requirement.

Continues on next page

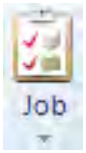
3.2.3.2 Container Patterns
Continued

	Description
Order	Displays the order of the selected item. You can modify the order according to your requirement. This is available only for the Container pattern layouts.
Rejected	Indicates if the item shall be rejected by a camera. This is available only for the Flow layouts.
Flip icon	Select the items and click the flip icon to flip the items.
Delete icon	Select an item and click the delete icon or press the DELETE key to delete the items.

3.2.3.3 Job

Introduction

The **Job** button allows you to manage the job.

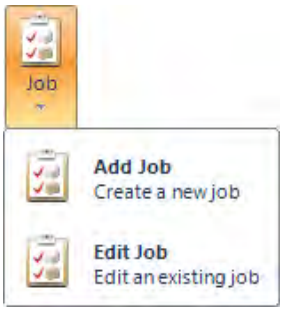


xx1300002160

To manage job:

1 In the **Job** group, click **Job**.

The job options window is displayed.



xx1300002514

2 Select the required option and configure the settings.

The **Add Job** option allows you to create a job. For more information see, *Adding a job on page 59*.

The **Edit Job** option allows you to edit an existing job. For more information see, *Editing a job on page 72*.

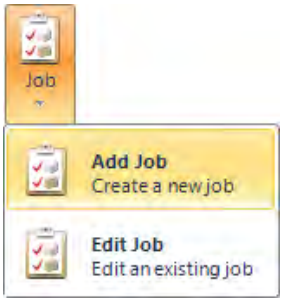
Adding a job

The **Add Job** option allows you to add a job.

To add a job:

1 In the **Job** group, click **Job**.

The job options window is displayed.



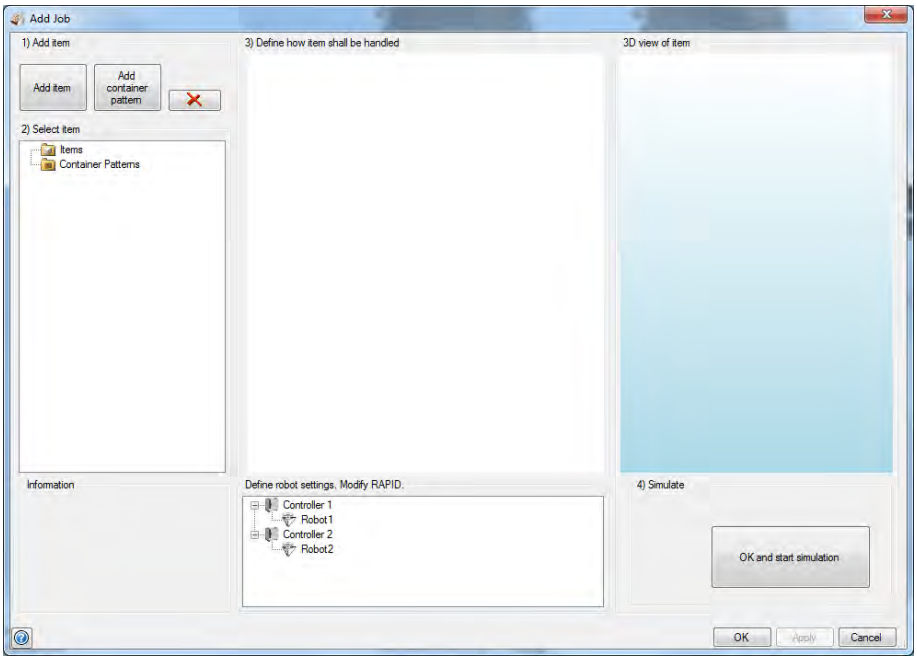
xx1300002515

2 Click **Add Job**.

Continues on next page

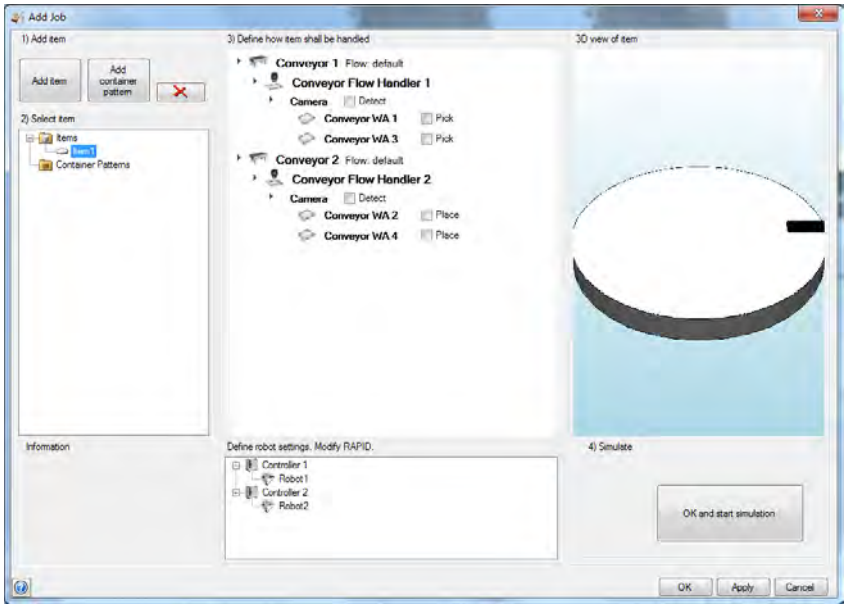
3.2.3.3 Job
Continued

The **Add Job** window is displayed.



xx1300002517

3 For each item that shall be handled by the job:
   a Click **Add item** in the **1) Add item** section.
   The **Add item** window with the defined items is displayed.
   b Select the required item from the list and click **OK**.



xx1300002518

   c Select the added item in the **2) Select item** section.

Continues on next page

d In the **3) Define how item shall be handled** section, select the **Pick** and **Place** check boxes for those work areas where this item shall be picked and placed.

e In the **3) Define how item shall be handled** section, select the **Detect** check boxes for those cameras and I/O sensors that shall detect this item.

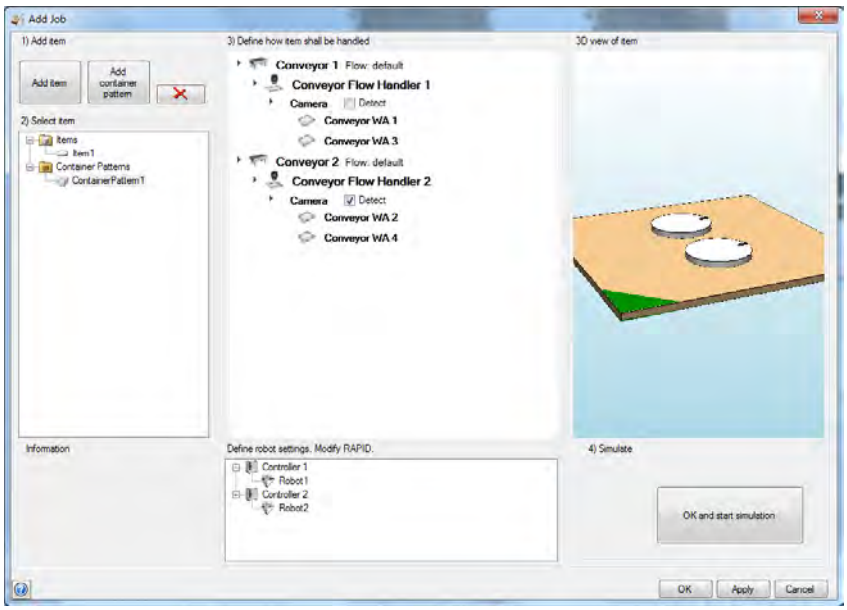
4 For each container pattern that shall be handled by the job:

a Click **Add container pattern** in the **1) Add item** section.

The **Add pattern** window with the defined container patterns is displayed.

> **Note**
>
> To pick or place items from or in the container patterns, the contained items must be individually added to the job.

b Select the required container pattern from the list and click **OK**.



xx1600001161

c Select the added container pattern in the **2) Select item** section.

d In the **3) Define how item shall be handled** section, select the **Detect** check box for those cameras and I/O sensors that shall detect this container pattern.

> **Note**
>
> There is no pick or place option for container patterns. Pick and place is defined only for the individual items. The items may be picked or placed separately, or to/from the detected container patterns.

5 Click **Apply**.

Continues on next page

3.2.3.3 Job
Continued

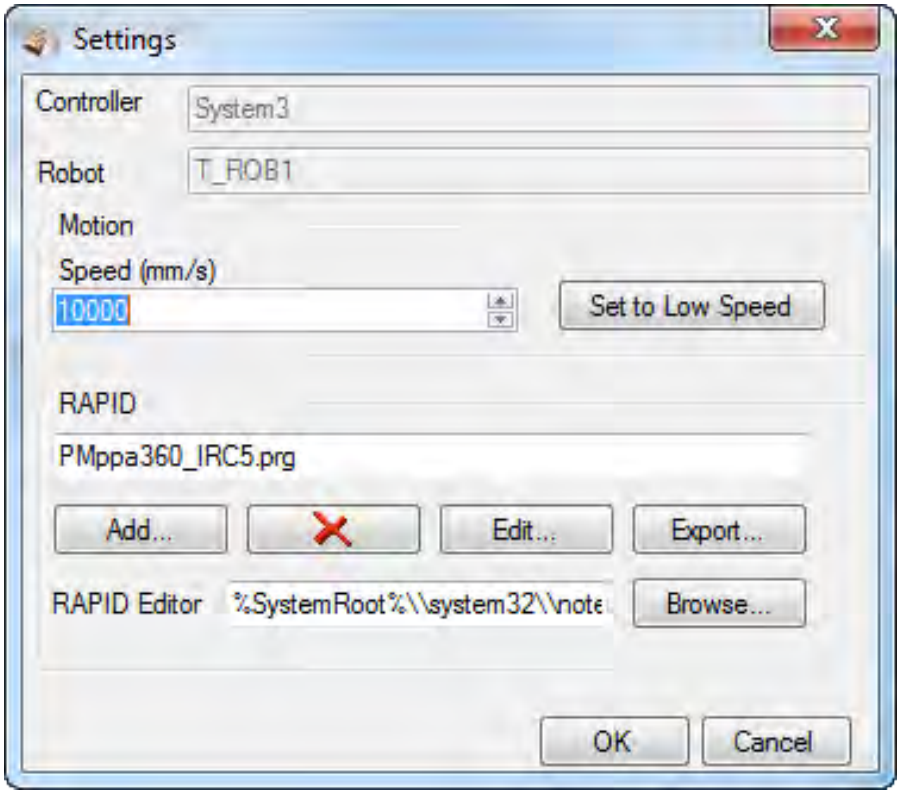
The job is saved.

6 Click **OK**.

The job is added and displayed in the **Jobs** tab in the application browser.

Define robot settings

Right-click on the robot and click **Settings** to display the settings window. You can set the speed of the robot and manage the robot's RAPID program.



xx1300002190

	Description
Controller	Displays the name of the controller.
Robot	Displays the name of the robot.
Motion	Allows you to set the speed.

Continues on next page

	Description
RAPID	Allows you to manage the RAPID program. **Note** Having a RAPID program is mandatory for the PickMaster robots. For more information on the RAPID program, see *RAPID program on page 105*. • The **Add** button allows you to select a RAPID program. The default selection is `PMppa360_IRC5.prg`, this is the template program used with 4 axes FlexPicker robots. If a 6 axes robot of articulated arm type shall be used, for example, IRB140 or IRB120, use instead the template program `PMppa_6Axes_IRC5.prg`.. The template programs are found in the RAPID folder of the Picking PowerPac installation folder. • The Delete button allows you to delete the selected RAPID program. • The **Edit** button allows you to edit the selected RAPID program. When you click the **Edit** button the selected RAPID program opens in a text editor. Make the required changes in the program, save and close the text editor. Click **OK** to save the changes. • The **RAPID Editor** section allows you to select a text editor application that shall be used for editing the RAPID programs.

Define how items shall be handled

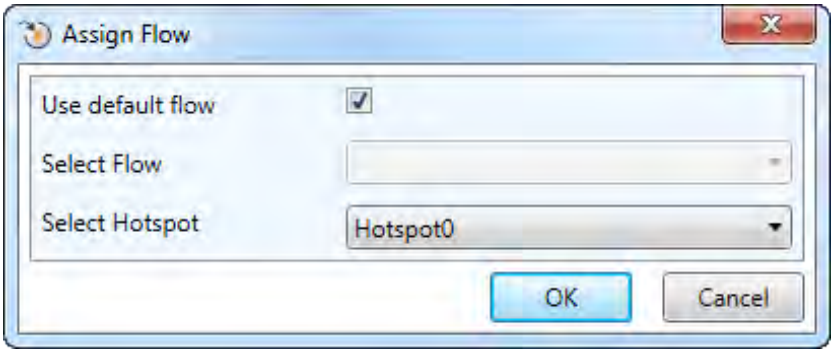
Flow settings

A flow is always assigned on conveyors and indexed work areas. The flow appears during simulation. If no selection of a customized flow is made, a default flow will be assigned instead.

To assign a flow on a conveyor:

1 Right-click on the conveyor and select **Assign flow**.

The **Assign flow** window is displayed.



xx1300002191

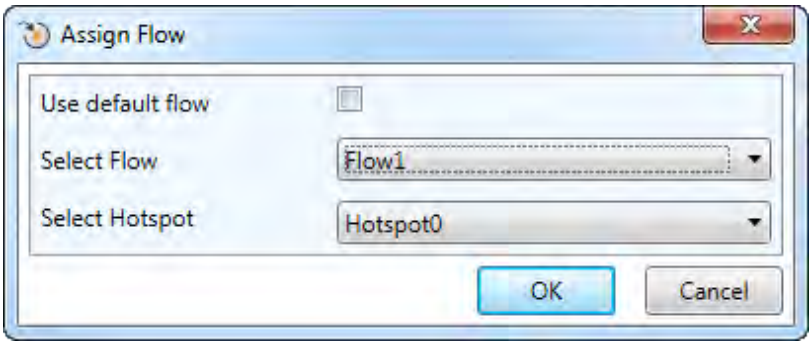
Continues on next page

3.2.3.3 Job
Continued

2 Select the **Use default flow** check box if you plan to use default product flow.

Note

In case you want to use a configured flow, clear the **Use default flow** check box. When you clear the check box the **Select flow** list is enabled.



xx1300002300

Select the required flow from the list.

3 Select a required hotspot from the **Select Hotspot** list.

Note

The hotspot location defines the start location of the flow. The hotspot can be modified to move the complete flow in a desired direction, for example, closer to one side of the conveyor belt.

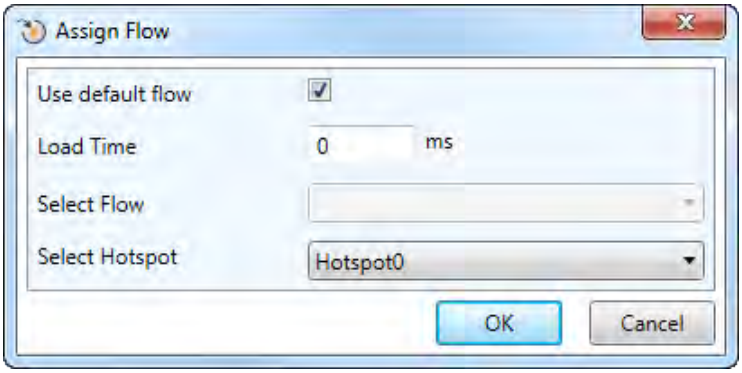
4 Click **OK**.

The flow is assigned.

To assign a flow on an indexed work area:

1 Right-click on the indexed work area and select **Assign flow**.

The **Assign flow** window is displayed.



xx1300002706

You can select a default flow or assign a configured Layout flow. A new flow layout is generated on the indexed work area at simulation start and as soon as the robot has finished picking or placing one flow layout. You can select

Continues on next page

a hotspot defining where the flow layout shall be generated. You can also select a Load time (ms) to delay the generation.

> **Note**
>
> Recorded flows cannot be used with indexed work areas.

2 Click **OK**.

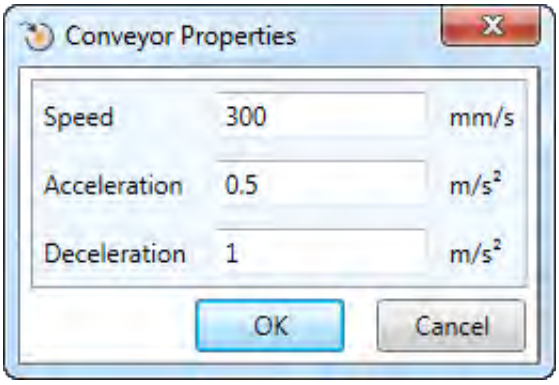
The flow is assigned on an indexed work area.

Editing conveyor properties

To edit conveyor properties:

1 Right-click on the conveyor and select **Properties**.

The **Conveyor Properties** window is displayed.



xx1300002192

2 Make the required changes in the **Speed**, **Acceleration**, and **Deceleration** field according to your requirement.

3 Click **OK**.

The changes are saved.

Manage distribution settings

For each item in a job, there are distribution settings for each flow handler. The distribution settings define to which work areas the item shall be distributed for pick or place. It also defines how the item shall be distributed, for example, if a load balance group or an ATC group shall be used.

There are four types of item distributors that can be selected for an item.

- **Work area**: The item targets are handled by a single work area.
- **ByPass**: The item targets are discarded, that is not handled by any work area. If no distributor is selected for an item type it is considered as ByPass.
- **LB group**: The item targets are handled by the work areas included in a load balance group. A load balance group is a collection of Work areas, ByPass, and ATC group distributors. Item targets are distributed among the work areas in an optimal way to avoid sending two adjacent item targets to the same work area.

Continues on next page

3.2.3.3 Job
Continued

- **ATC group**: Item targets are handled by the work areas included in an Adaptive Task Completion (ATC) group. An ATC group is a collection of ordered work areas that gets the same item targets. The first robot accesses as many item targets as possible. The other robots in the ATC group can access any missed item targets. If the last work area in the group is a conveyor work area with start and stop it is guaranteed that all the item targets can be accessed.

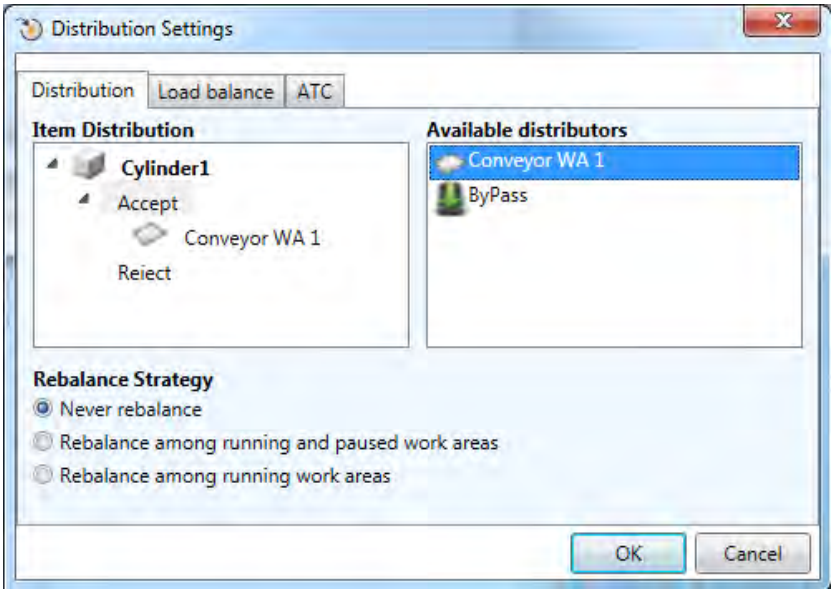
> **Note**
>
> The distribution settings can be setup different for the items that shall be rejected after the camera inspection. For example, the accepted items are handled by a single work area and rejected items will bypass.

Editing distribution settings

To edit distribution settings:

1 Select an item in the browser, right-click on the flow handler, and select **Distribution**.

The **Distribution Settings** window is displayed.



xx1300002277

2 **Distribution** tab

a Select the required distributor from the **Available distributors** list.

> **Note**
>
> New distributors can be created in the **Load balance** tab and the **ATC** tab.

b Drag and drop the selected distributor to the **Accept** or **Reject** option in the **Item Distribution** list. Only one distributor can be selected.

3 Click **OK**.

Continues on next page

The changes are saved.

Load balance tab

To create an LB Group distributor:

1. Select the **<New LbGroup>** option from the **Available distributors** list and drag and drop the selected option to the **Load balance groups** list.

   The **Enter Load balance group name** window is displayed.
2. Type a name for the load balance group in the **Name** field.
3. Click **OK**.

   The group is displayed in the **Load balance groups** list.
4. Select the required distributor from the **Available distributors** list and drag and drop the selected distributor to the selected load balance group in the **Load balance groups** list.

Continues on next page

3.2.3.3 Job
Continued

ATC tab

To create an ATC group distributor:

1. Select the **<New AtcGroup>** option from the **Available work areas** list and drag and drop the selected option to the **Adaptive Task Completion groups** list.

   The **Enter ATC group name** window is displayed.
2. Type a name for the ATC group in the **Name** field.
3. Click **OK**.

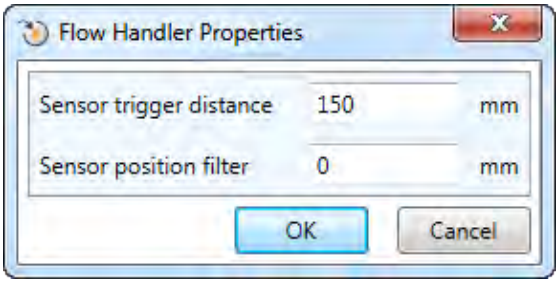
   The group is displayed in the **Adaptive Task Completion groups** list.
4. Select the required work area from the **Available work areas** list and drag and drop the selected work area to the selected ATC group in the **Adaptive Task Completion groups** list.

Managing conveyor flow handler properties

To manage conveyor flow handler properties:

1. Right-click on the conveyor flow handler, and select **Properties**.

   The **Flow Handler Properties** window is displayed.



xx1300002520

2. Make the required settings.
   - **Sensor trigger distance**: If a distance triggered camera is used, sensor trigger distance specifies the conveyor distance between consecutive recordings of the camera.

     Note

     If the sensor trigger distance is set too long, that is, longer than the view of the camera as defined by the camera entry and exit properties, objects may pass by the camera without being detected. If a sensor is not used, that is, Item detection = Distance, the value specifies the distance between generated flow objects. The Sensor trigger distance property is not used with I/O triggered cameras and I/O sensors.

   - **Sensor position filter**: If the distance between two detected objects is less than this value, PickMaster considers the object as one.
3. Click **OK**.

   The changes are saved.

Continues on next page

Note

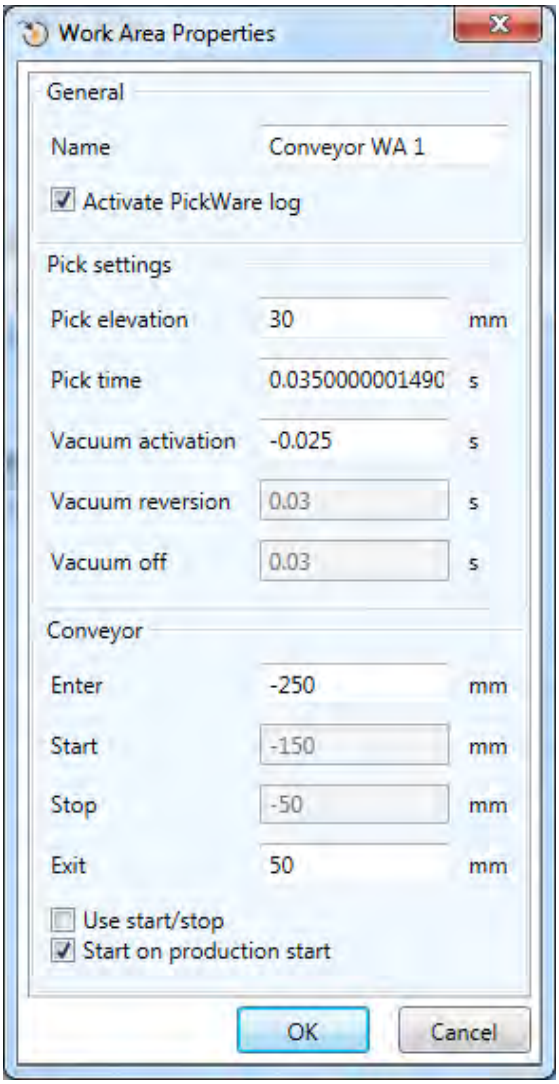
Trigger distance is used only with conveyor flow handlers. If a distance triggered camera is used as item detection method it defines the travel distance of the conveyor (mm) until a new image is captured. If distance is used as item detection method, trigger distance defines the travel distance of the conveyor until new items are generated.

Managing conveyor work area properties

To manage the conveyor work area properties:

1 Right-click on the conveyor work area, and select **Properties**.

The **Work Area Properties** window is displayed.



xx1300002278

2 Make the required changes in the **Pick settings** section.

- **Pick elevation** is the vertical distance, from where the robot approaches the item target.

Continues on next page

- **Pick time** is the time the robot is in the pick/place position. If the conveyor is moving during the pick/place time, the robot tracks along the conveyor to keep the relative position on the moving conveyor.
- **Vacuum activation** is the time in seconds before the middle of the corner path of the approaching position, when the vacuum I/O should be set. If a negative value is entered, the vacuum I/O sets the time after the middle of the corner path. This value is valid only for work areas of type pick or Other.

> **Note**
>
> Vacuum activation does not affect picking of items in simulation. Items are attached to the picking tool using `SimAttach` events, for example, in the Pick Routine.

- **Vacuum reversion** is the time in seconds before the half place time in the place position, when the blow I/O should be set. If a negative value is entered, the blow I/O sets the time after the half place time in the place position. This value is valid only for work areas of type place or other.

> **Note**
>
> Vacuum reversion does not affect placing of items in simulation. Items are detached from the picking tool using `SimDetach` events, for example, in the Place Routine.

- **Vacuum Off** is the time in seconds after the half place time in the place position, when the blow I/O should be reset. If a negative value is entered, the blow I/O resets the time before the half place time in the place position. This value is only valid for work areas of type place or other.

> **Note**
>
> Vacuum Off does not affect the placing of items in simulation. Items are detached from the picking tool using `SimDetach` events, for example, in the Place Routine.

3 Make the required changes in the **Conveyor** section.

- **Enter** is the limit from where the robot starts to execute item targets on the work area. The distance is calculated in millimeters from the center of the robot. The range is positive if the limit is downstream and beyond the center of the robot. Make sure that the enter limit can be reached by the robot.
- **Start** is when the next item to execute on the conveyor is above this limit, the conveyor is started. The distance is calculated in millimeters

Continues on next page

from the center of the robot. The range is positive if the limit is downstream and beyond the center of the robot.

- **Stop** is when an item on the conveyor reaches this limit, the conveyor is stopped. The distance is calculated in millimeters from the center of the robot. The range is positive if the limit is downstream and beyond the center of the robot, relative to the moving direction of the conveyor.
- **Exit** is the limit from where the robot considers an item target as lost on the work area. The distance is calculated in millimeters from the center of the robot. The range is positive if the limit is downstream and beyond the center of the robot. When the tracked item passes beyond this limit it is dropped. This limit must be chosen well within the maximum reach of the robot. The robot must be able to reach this position from an arbitrary position in the robot's working area before the position is out of reach.

4 Select the **Use start/stop** check box if the work area should supervise the start and stop limits.

5 Select the **Start on production start** check box if the work area should start the conveyor when the project is started, and stopped when the project is stopped.

6 Click **OK**.

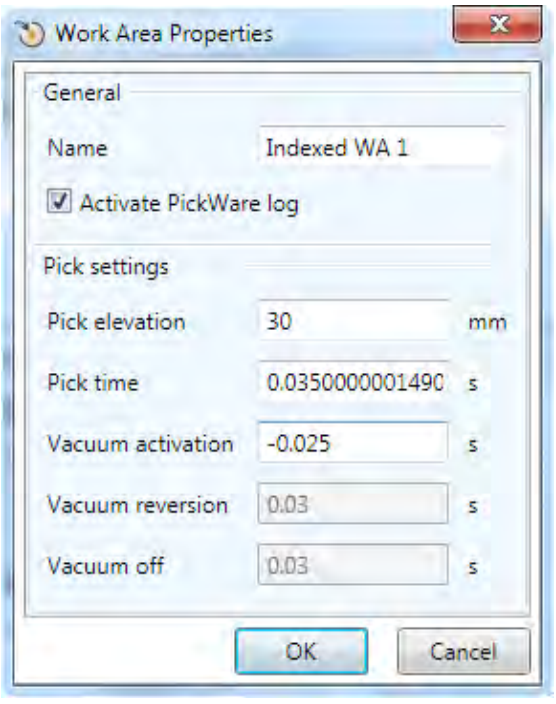
The work area properties are saved.

Managing indexed work area properties

To manage the indexed work area properties:

1 Right-click on the indexed work area, and select **Properties**.

The **Work Area Properties** window is displayed.



xx1300002299

Continues on next page

3.2.3.3 Job
Continued

2 Make the required changes in the **Pick settings** section.

For more details about the fields in the **Work Area Properties** window is available in the section *Managing conveyor work area properties on page 69*.

3 Click **OK**.

The indexed work area properties are saved.

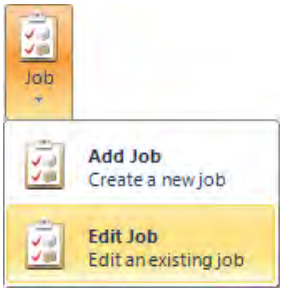
Editing a job

The **Edit Job** button allows you to edit the settings of an existing job.

To edit an existing a job:

1 In the **Job** group, click **Job**.

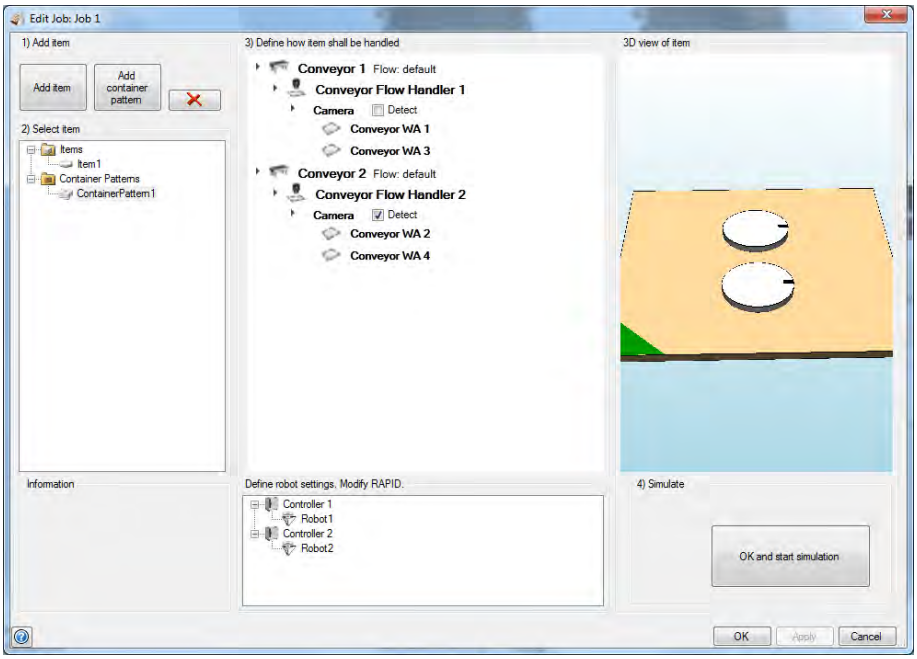
The job options window is displayed.



xx1300002516

2 Click **Edit Job**.

The **Edit Job** window is displayed.



xx1300002519

3 Edit the job settings according to your requirement. For more details see, *Adding a job on page 59*, *Define robot settings on page 62*, and *Define how items shall be handled on page 63*.

Continues on next page

4 Click **Apply**.

   The changes to the job are saved.

5 Click **OK**.

3.2.4 Simulation group

3.2.4.1 Calibrate

Quick calibration

Once you define a job, click **Quick Calibration** to automatically calibrate the conveyor work areas with base frames in need of calibration.

> **Note**
>
> All conveyor work areas in need of calibration are marked in the line tab with a warning triangle.

Edit Calibration

You can calibrate a selected conveyor work area using this option.

To calibrate a selected conveyor work area:

1. Click **Edit Calibration**.

   The **Calibration** window is displayed
2. Select the required conveyor work area that needs to be calibrated.
3. Select the **Restart** check box.
4. Click **OK**.

   The related controllers are restarted and the selected work areas become calibrated.

3.2.4.2 Simulation controls

Simulation controls

Controlling the simulation

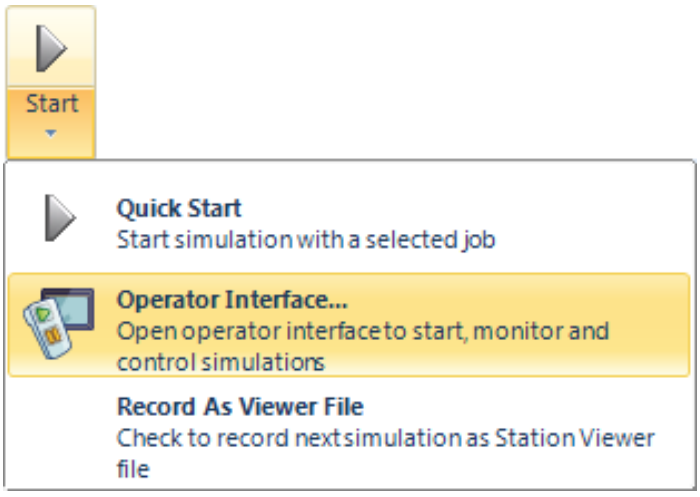
The following table provides description regarding the simulation controls.

Button name	Description
Start	Allows you to start the simulation. The **Start** button has the following sub options: • **Quick Start**: Allows you to start simulation with a selected job. • **Operators Interface**: Allows you to schedule and start the jobs. For more details, see *Operator interface on page 75*. • **Record as viewer file**: Allows you to save the next simulation as a RobotStudio Viewer file. When the simulation has stopped, a save dialogue appears.
Pause	Allows you to pause the simulation.
Stop	Allows you to stop the simulation.
Reset	Allows you to reset the simulation. This also causes all the temporary objects created in the previous simulation to be removed, for example, items and containers.

Operator interface

Overview

In the **Simulation** group, click **Start** > **Operator Interface**.



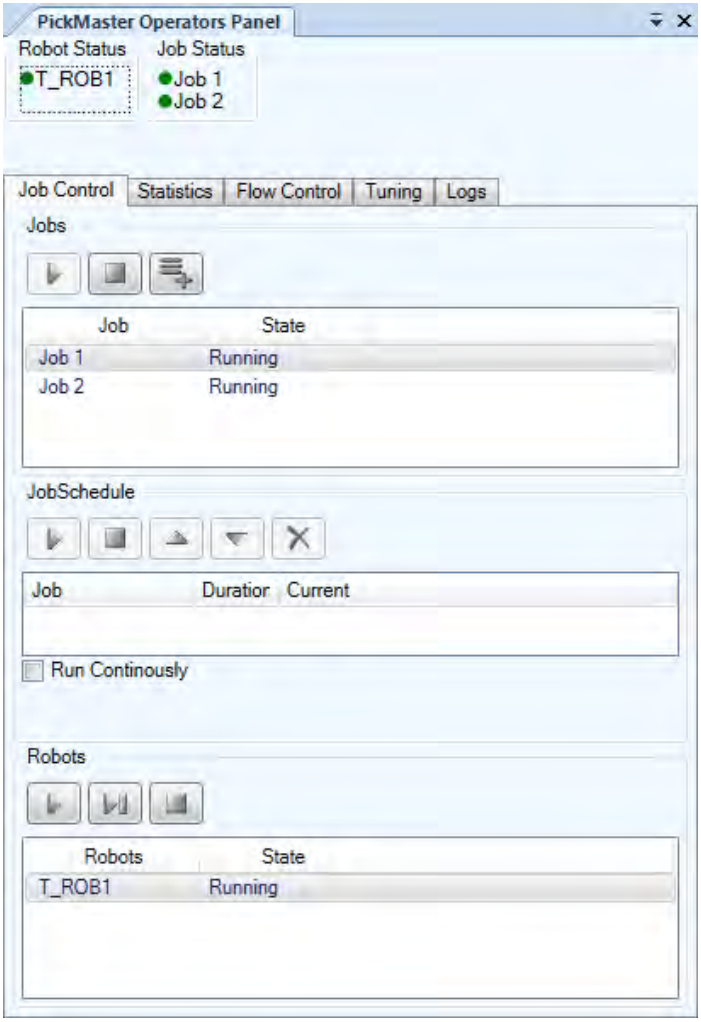
xx1300002193

The **PickMaster Operators Panel** window is displayed.

Continues on next page

3.2.4.2 Simulation controls
Continued

The following figure and table provide details about the **PickMaster Operators Panel** window.



xx1300002194

	Description
Robot Status	Displays the status of the robots. • - Indicates the robot is running. • - Indicates the robot is paused or stopped.
Job Status	Displays the status of the jobs. • - Indicates the job is running. • - Indicates the job is stopped.
Job Control	Allows you to manage the jobs. For more details, see *Job control on page 77*.
Statistics	Displays simulation statistics. For more details, see *Statistics on page 79*.
Flow Control	Allows you to manage the flow. For more details, see *Flow control on page 80*.
Tuning	Allows you to tune different properties of the application. For more details, see *Tuning on page 81*.

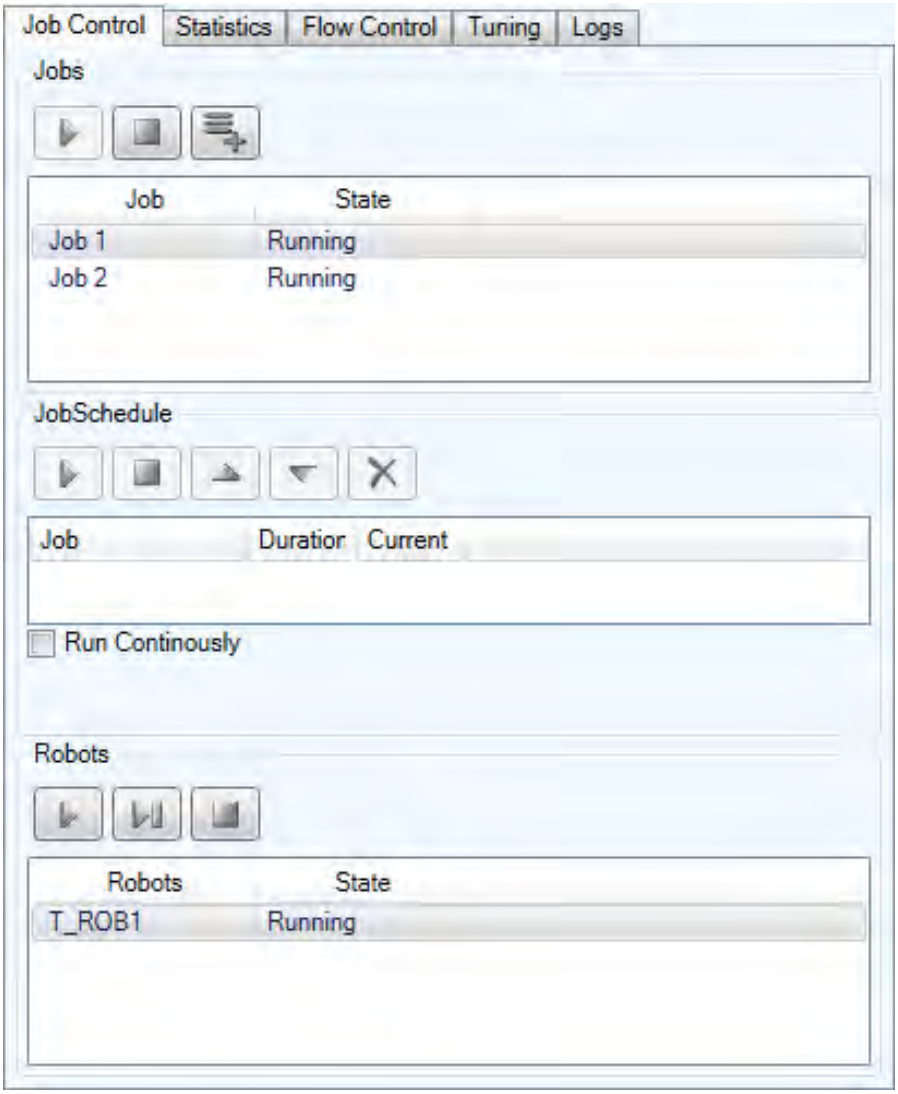
Continues on next page

	Description
Logs	Displays the logs of the Virtual PickMaster.

Job control

The **Job Control** tab allows you to manage the jobs.

The following figure and table provides the details about the **Job Control** tab.



xx1300002199

	Description
Jobs	This section allows you to manage the available jobs. •  - Starts a selected job. •  - Stops a selected job. •  - Adds a selected job to the **Job Schedule** section.

Continues on next page

3.2.4.2 Simulation controls
Continued

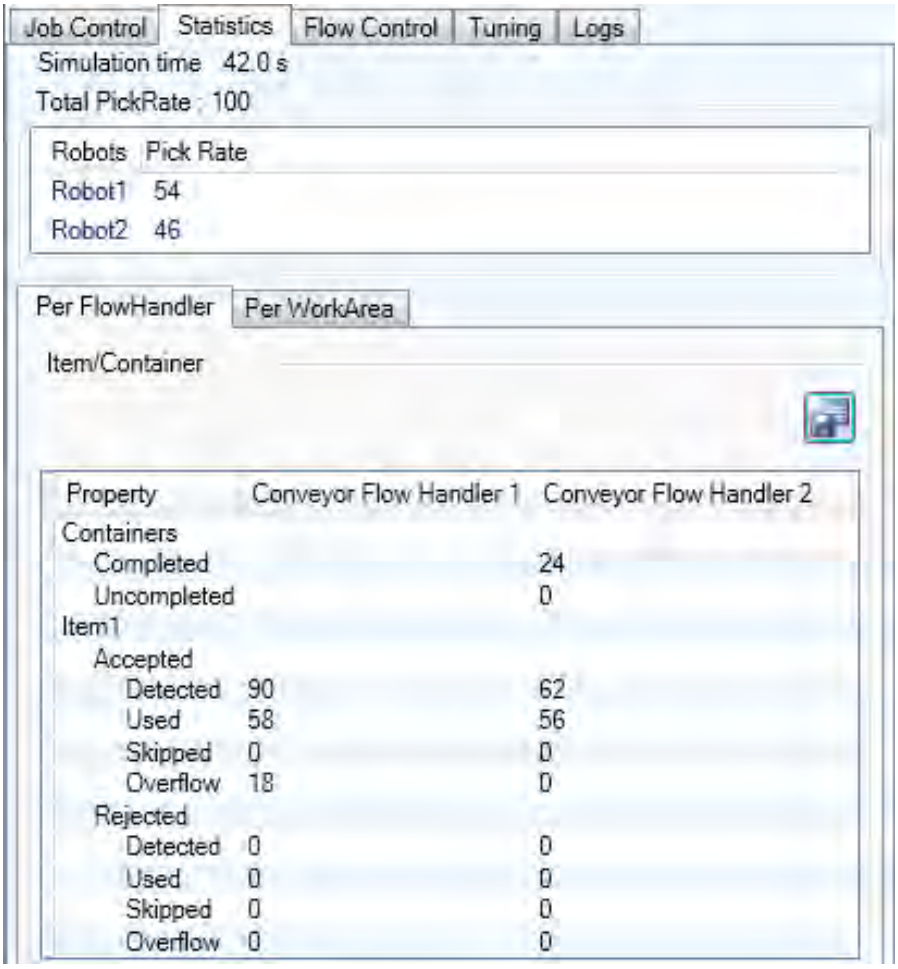
	Description
Job Schedule	This section allows you to manage the scheduled jobs. •  - Starts the scheduled jobs. •  - Stops the scheduled jobs. •  - Moves up a job in the schedule. •  - Moves down a job in the schedule. •  - Deletes a scheduled job. Select the **Run Continuously** check box to run the scheduled jobs continuously.
Robots	Allows you to manage the robots. •  - Starts the select robot. •  - Pauses the selected robot. •  - Stops the selected robot.

Continues on next page

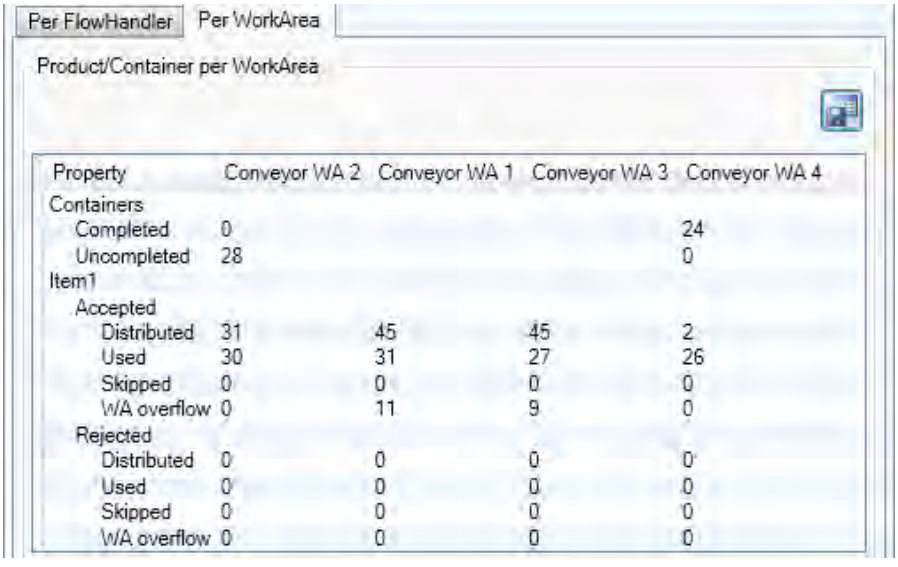
Statistics

The **Statistics** tab displays the simulation statistics information.

The following figure and table provides the details about the **Statistics** tab.



xx1300002195



xx1500003244

Continues on next page

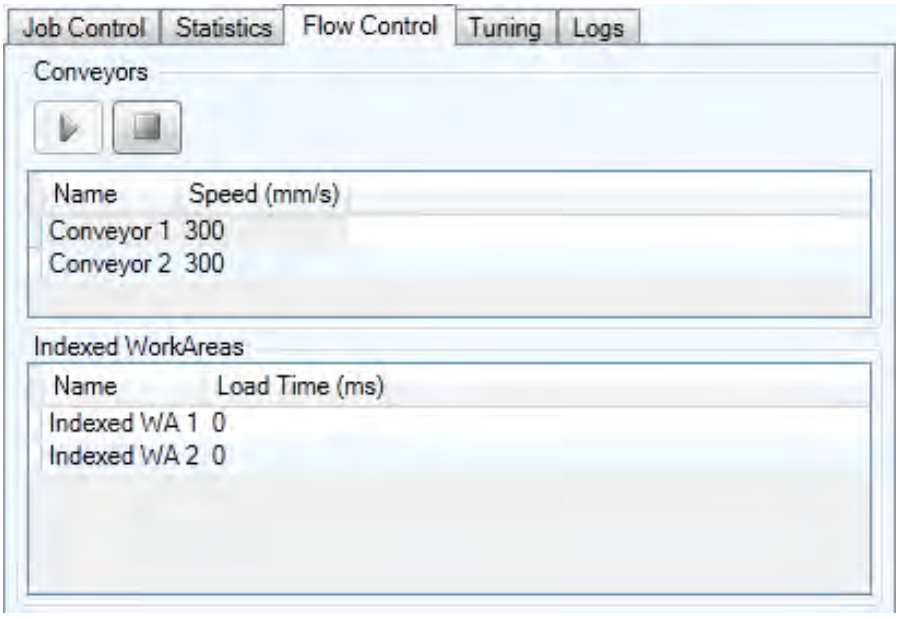
3.2.4.2 Simulation controls
Continued

	Description
Simulation Time	Displays the elapsed simulation time (in seconds) since the simulation is started.
Total PickRate	Displays the current pick rate (picks/second) for the robots.
Per Flow Handler	Displays the simulation statistics per flow handler.
Per WorkArea	Displays the simulation statistics per work area.
No of stops	Number of times the conveyor has been ordered to stop.
Completed	Number of completed containers.
Uncompleted	Number of unfinished containers.
Detected	Number of items detected by the flow handler.
Distributed	Number of items distributed to a work area.
Used	Number of used items, that is, picked, placed, or acknowledged as used in the RAPID program.
Skipped	Number of skipped items, that is, acknowledged as skipped in the RAPID program.
Overflow	Number of detected items passing through all the work areas without being picked, placed, or acknowledged as used.
WA overflow	Number of distributed items passing through a work area without being picked, placed, or acknowledged as used.

Flow control

The **Flow Control** tab allows you to manage the flows.

The following figure and table provide details about the **Flow Control** tab.



xx1300002196

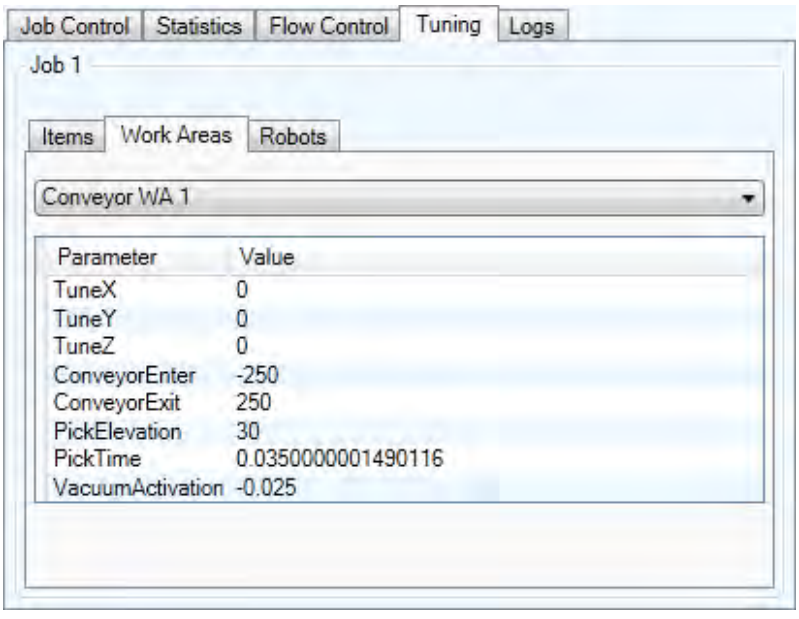
Continues on next page

	Description
Conveyors	Allows you to manage the conveyors. For example, change the conveyor speed. •  - Starts the selected conveyor. •  - Stops the selected conveyor. To change the speed of a conveyor, select the conveyor, type the new speed, and press the ENTER key. The new value comes in effect in the simulation.  **Note** The changed conveyor speed is saved when the station is saved. The speed is saved to the used flow. If a default flow is used, the speed is saved to the job properties of the conveyor.
Indexed WorkAreas	Allows you to manage the indexed work areas. For example, change the load time.  **Note** The changed load time is saved when the station is saved.

Tuning

The **Tuning** tab allows you to tune the parameters for items, work area, or robots.

The following figure and table provides the details about the options available in the **Tuning** tab.



xx1300002197

Continues on next page

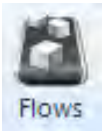
3.2.4.2 Simulation controls
Continued

	Description
Items	The **Items** tab allows you to tune the parameters of the items. To tune a parameter of an item: 1 Select the item from the list. The parameters associated with the selected item is displayed. 2 Select the parameter and type the required tuning value in the **Value** column. 3 Press the ENTER key. The tuned value is updated for the parameter. The tuned value comes in effect in the simulation. **Note** The tuned values are saved when the station is saved.
Work Areas	The **Work Areas** tab allows you to tune the parameters of the work areas. To tune a parameter of a work area: 1 Select the work area from the list. The parameters associated with the selected work area is displayed. 2 Select the parameter and type the required tuning value in the **Value** column. 3 Press the ENTER key. The tuned value is updated for the parameter. The tuned value comes in effect in the simulation. **Note** The tuned values are saved when the station is saved.
Robots	The **Robots** tab allows you to tune the parameters of the robots. To tune a parameter of a robot: 1 Select the robot from the list. The parameters associated with the selected robot is displayed. 2 Select the parameter and type the required tuning value in the **Value** column. 3 Press the ENTER key. The tuned value is updated for the parameter. The tuned value comes in effect in the simulation. **Note** The tuned values are saved when the station is saved.

3.2.4.3 Flows

Managing flow

The **Flows** button allows you to manage the flow.



xx1300002513

A flow is used to define how the items and containers are to be generated in the simulation. A flow is attached to a hot spot on a conveyor or an indexed work area in the job configuration. When attaching the flow, the hot spot becomes a source from where items and containers appear in the simulation according to the flow configuration. Following are the two types of flows:

- Layout: A Layout flow is a predefined layout that is periodically regenerated at the hot spot. The layout may have some random variation regarding the locations of items or containers and the availability of them. The layout may consists of different items or container patterns.
- Recorded: A recorded flow is a recording of a sensor from a real PickMaster system in operation. The recording is exported from PickMaster as an xml file having information of all the detected items and containers during a time interval. When the file is imported, the items detected are mapped to the configured items and container patterns.

Note

Recorded flows cannot be used with indexed work areas.

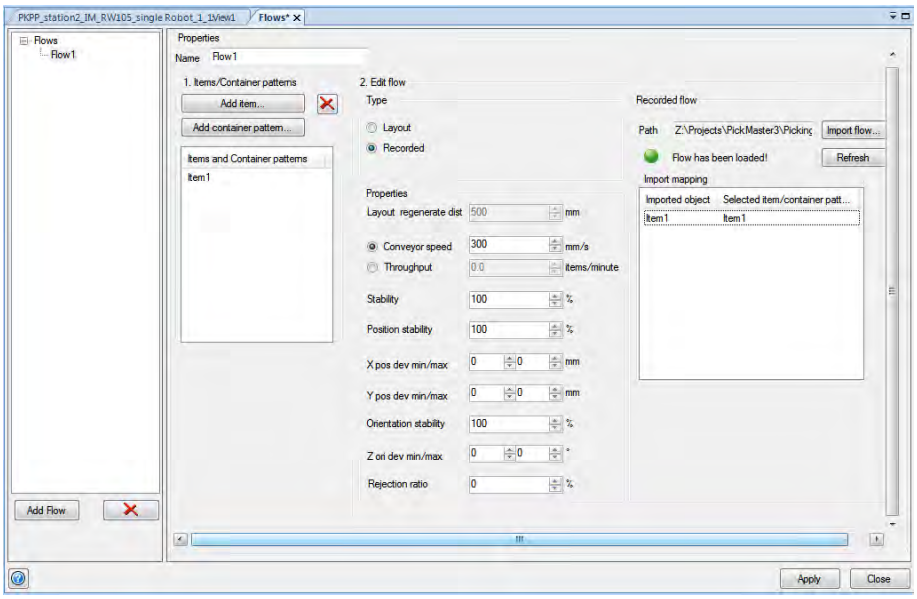
To configure a flow:

1 In the **Simulation** group, click **Flows**.

Continues on next page

3.2.4.3 Flows
Continued

The **Flows** window is displayed.



xx1300002348

2 Click **Add Flow**.

The flow is added to the browser and the properties of the flow are displayed in the **Properties** section.

3 Type a suitable name for the flow in the **Name** field.

4 In the **1. Items/Container patterns** section, click **Add item**.

The item is added to the **Item and Container patterns** list.

Note

To add a container pattern, click the **Add container pattern** button.

5 In the **2. Edit Flow** section, select the type of flow from the **Type** section.

- If the **Layout** flow is selected, type the required values for the parameters in the **Properties** section. Click **Edit Layout** to edit the layout according to your requirement. For more details see the section *Edit layout on page 56*.
- If the **Recorded** flow is selected, type the required values for the parameters in the **Properties** section, click the **Import Flow** button and select the xml file to be imported. A list of imported objects (for example, PickMaster 3 items) appears in the **Import mapping** section. For each object, select which item or container pattern that shall represent this object in the simulation.

The following table provides details regarding the **2. Edit Flow** section.

	Description
Layout regenerate dist	The distance between consecutive flow layouts generated on a conveyor. Not used with recorded flows.

Continues on next page

	Description
Conveyor speed	The desired conveyor speed when the flow is assigned to a conveyor.
Throughput	The desired item throughput when running the flow. This is an alternative way of defining the conveyor speed when the flow is assigned to a conveyor.
Stability	If set to 100%, all the items in the layout are generated on every trigger without losses. A lower value defines the probability that an item in the layout is generated.
Position stability	If set to 100%, the generated items always have correct position. A lower value defines the probability that an item gets the correct position.
Position min/max	Defines the minimum and maximum deviation of the position from the correct value.
Orientation stability	If set to 100%, the generated items always have correct orientation. A lower value defines the probability that an item gets correct orientation.
Orientation min/max	Defines the minimum and maximum deviation of the orientation from the correct value.
Layout size	Defines the size of the layout.
Rejection ratio	Defines the probability that an item becomes rejected by a camera. If set to 0%, the item setting “Rejected” in the Layout will decide if the item is rejected.  **Note** In the simulation, a rejected item is marked with a red cross when it is detected by a camera.

6 Click **Apply**.

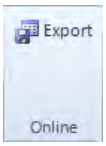
The flow is saved.

7 Click **Close**.

3.2.5 Online group

3.2.5.1 Export

Exporting line and job



xx1300002185

To export the jobs and line to corresponding PickMaster project (`.pmproj`) and line (`.pmline`) files:

1. In the **Online** group, click **Export**.

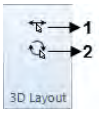
   The **Select destination folder** window is displayed.
2. Click **Browse** and select a destination folder.
3. Click **OK**

   The line and job are exported to the selected location as PickMaster line and project files.

3.2.6 3D layout group

Overview

The following figure and table provides you information regarding different elements in the **3D Layout** group.



xx1300002184

Label	Button name	Description
1	**Move**	Allows you to drag an item relative to the active reference coordinate system.
2	**Rotate**	Allows you to enable rotation around the various axes of an object determined by the reference coordinate system.

Moving an item

1 In the **Layout** browser, select the item you want to move.
2 Click **Move**.
3 In the graphics window, click one of the axes and drag the item into position.

Rotating an item

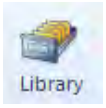
1 In the **Layout** browser, select the item you want to rotate.
2 Click **Rotate**.
3 In the graphics window, click one of the rotational rings and drag the item into position.

If you press the **ALT** key while rotating, the item will snap 10 degrees at a time.

3.2.7 Miscellaneous group

3.2.7.1 Library

Managing the library



xx1300002386

In the **Miscellaneous** group, click **Library** to display the items, containers, and layouts saved in the library.

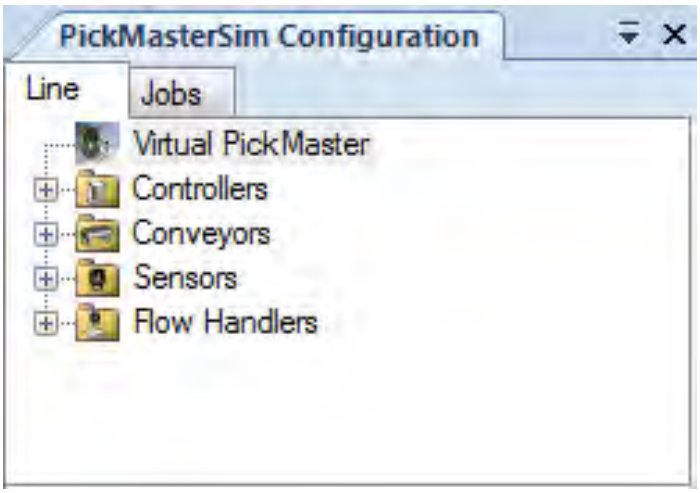
The library is saved as different xml files in the RobotStudio user document directory: for example, `C:\Users\MyUserName\Documents\RobotStudio\Picking\Library`

3.3 Tree view browser

3.3.1 Introduction

Overview

The tree view browser in the Picking PowerPac consists of a **Line** tab and a **Jobs** tab.



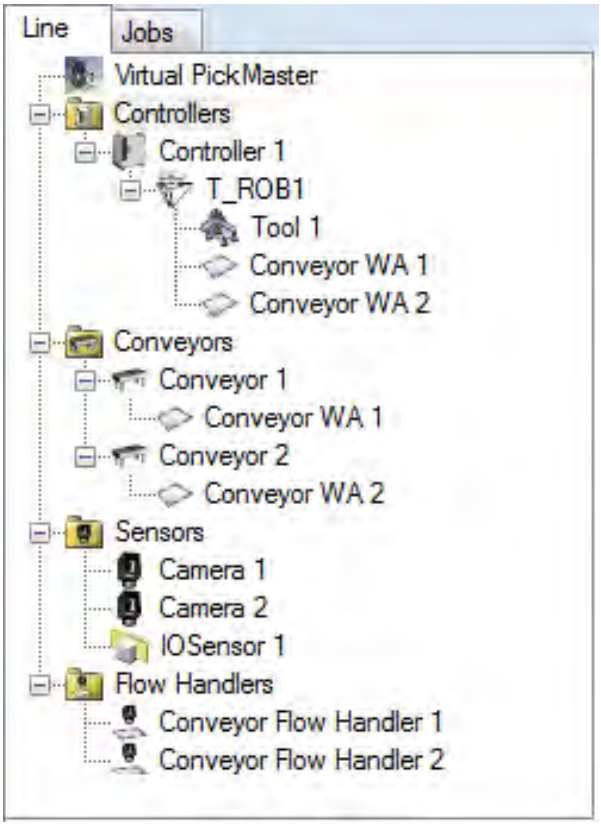
xx1300002200

The **Line** tab displays the Virtual PickMaster and the application hardware objects such as robots, cameras, conveyors, and flow handlers. The **Jobs** tab displays the saved jobs, each one defining how the application shall handle a selection of items and containers.

3.3.2 Line tab

Overview

Following are the objects and configurations saved in the **Line** tab.



xx1300002280

- Virtual PickMaster
- Robots and controllers
- Picking tool
- Conveyors
- Conveyor and indexed work areas
- Sensors
- Flow handlers

Virtual PickMaster

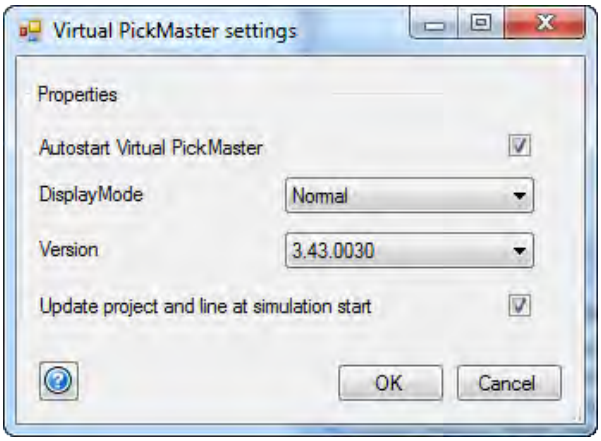
Virtual PickMaster is a special software installation of PickMaster 3 that is required for running simulations with Picking PowerPac. It is installed separately from Picking PowerPac and the version numbers follow those of PickMaster 3, for example, Virtual PickMaster 3.43. The installation is distributed with Picking PowerPac.

Virtual PickMaster is launched when Picking PowerPac is activated and runs in the background. It is closed when the Picking PowerPac is deactivated. In simulation, it acts exactly as PickMaster 3, controlling the operations of the robots.

Continues on next page

Virtual PickMaster has its own user interface. The user interface is a reduced version of PickMaster 3. However, in normal cases there is no need to use this interface since the programming of Virtual PickMaster is completely done using Picking PowerPac. For information on the Virtual PickMaster user interface, see *Application manual - PickMaster 3*. After running a simulation, it is possible to check how the project and line have been setup in Virtual PickMaster.

Right-click on **Virtual PickMaster** and select **Settings**. The **Virtual PickMaster settings** window is displayed. The following figure and table provide more details about the **Virtual PickMaster settings** window.



xx1300002381

	Description
Autostart Virtual PickMaster	Select this check box to automatically launch Virtual PickMaster when Picking PowerPac is activated. This check box is selected by default.
DisplayMode	Allows you to define the display mode of Virtual PickMaster when you activate the Picking PowerPac. Following are the options available in the **DisplayMode** list. • Normal: The Virtual PickMaster is opened in normal mode. • Hidden: The Virtual PickMaster is opened but is hidden in the back ground. • Minimized: The Virtual PickMaster is opened in minimized mode. • Maximized: The Virtual PickMaster is opened in maximized mode.
Version	Allows you to select the version of the Virtual PickMaster.
Update project and line at simulation start	Select this check box to always update the project and line configuration of Virtual PickMaster at simulation start. This ensures that Virtual PickMaster is always correctly updated during simulation. Clearing the check box turns off the configuration update. One reason for doing this is that it makes it possible to make modifications directly in the Virtual PickMaster GUI which otherwise would become over written.

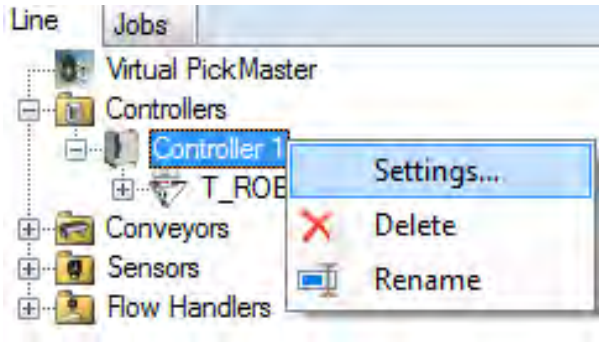
Continues on next page

3.3.2 Line tab
Continued

Controller

Managing controller

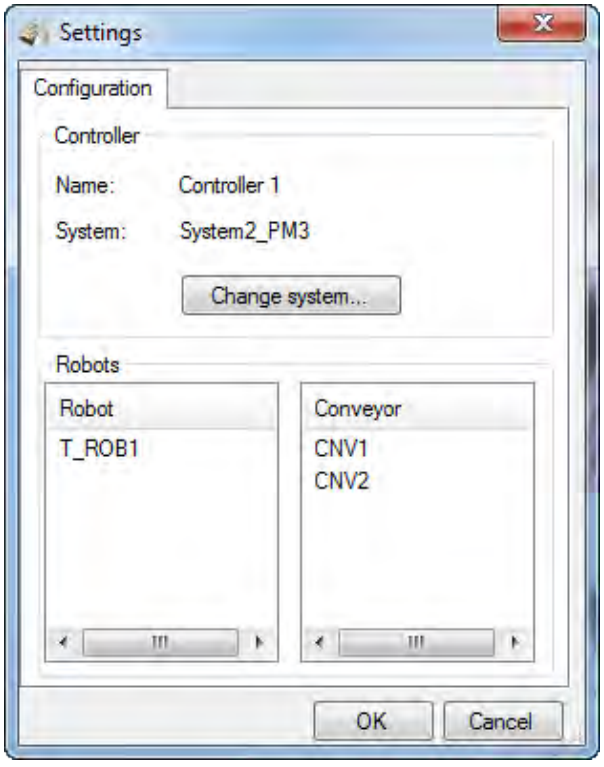
Right-click on the controller. The following window is displayed.



xx1300002281

	Description
Settings	Allows you to change the settings for the selected controller. When you right-click on a controller and select **Settings...**, the **Settings** window is displayed. See the following section for more details about managing a selected controller.
Delete	Allows you to delete the selected controller.
Rename	Allows you to change the name of the selected controller.

The following figure and table provides details about the Controller **Settings** window.



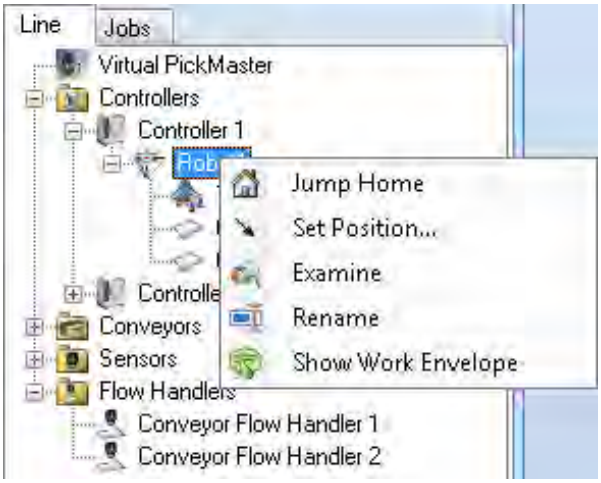
xx1300002282

Continues on next page

	Description
Name	Displays the name of the selected controller.
System	Displays the name of the system. The **Change System** button allows you to select a different system.
Robot	Displays the robots configured in the selected system.
Conveyor	Displays the conveyor boards configured in the selected system.

Managing robot

Right-click on the robot icon. The following window is displayed



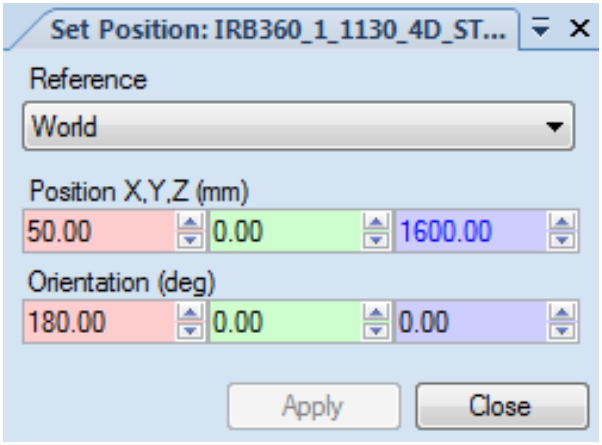
xx1300002283

	Description
Jump Home	Allows you to move the robot to the home position.
Set Position	Allows you a set a position for the selected robot. When you right-click on a robot and select **Set Position...** the **Set Position** configuration window is displayed. See the following section for more details about managing the position of a selected robot.
Examine	Allows you to examine the robot in the Station view.
Rename	Allows you to change the name of the selected robot.
Show Work Envelope	Allows you to display the robot's work envelope in the station view.

Continues on next page

3.3.2 Line tab
Continued

The following figure and table provides details about the **Set Position configuration** window.

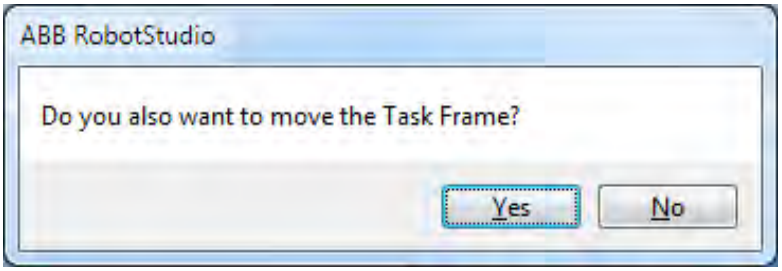


xx1300002284

	Description
Reference	Allows you to select a coordinate system.
Position X,Y,Z (mm)	Allows you to set a new position for the selected robot.
Orientation (deg)	Allows you to set a new orientation for the selected robot.

Note

When changing the robot position or orientation the following window appears.



xx1300002704

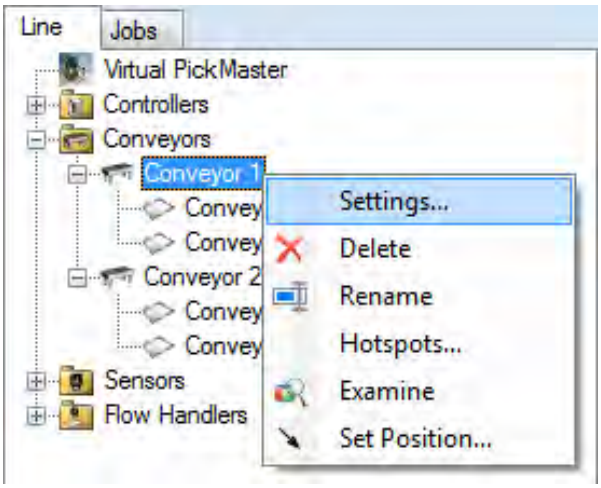
ABB recommends you to click **Yes** to move the Task Frame.

Continues on next page

Conveyor

Managing conveyor

Right-click on a conveyor. The following window is displayed.



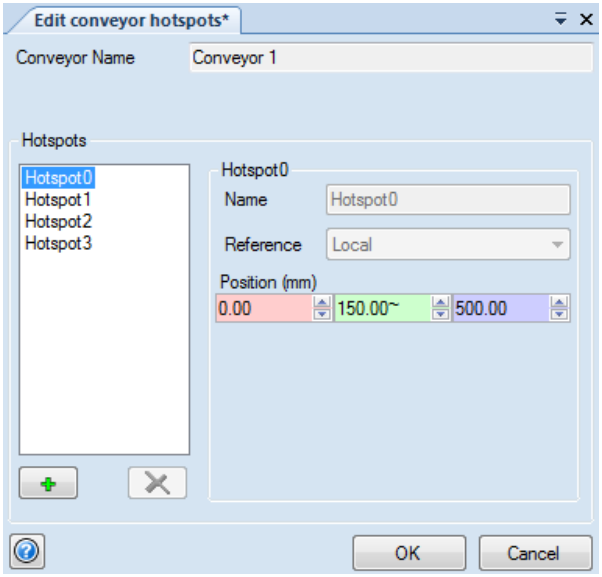
xx1300002285

	Description
Settings...	Allows you to manage the settings of the selected conveyor. When you select **Settings...**, the **Conveyor Properties** window is displayed. More details about managing a selected conveyor is available in the section *Add Conveyor on page 31*.
Delete	Allows you to delete the selected conveyor.
Rename	Allows you to change the name of the selected conveyor.
Hotspots...	Allows you to manage the hotspots. When you select **Hotspots...**, the **Edit Conveyor hotspots** window is displayed. See the following section for more details about the **Edit Conveyor hotspots** window. **Note** The hotspot is a saved location on the conveyor. A hotspot is used to define where on the conveyor the flow shall be generated. There is always a default hotspot, Hotspot0, located at the beginning of the conveyor. If the flow appears at a wrong location, modify the hotspot location to adjust it.
Examine	Allows you to examine the selected conveyor in the Station view.
Set Position...	Allows you a set a position for the selected conveyor. When you select **Set Position...** the **Set Position** configuration window is displayed from where you can configure a new position.

Continues on next page

3.3.2 Line tab
Continued

The following figure and table provides details about the **Edit Conveyor hotspots** window.



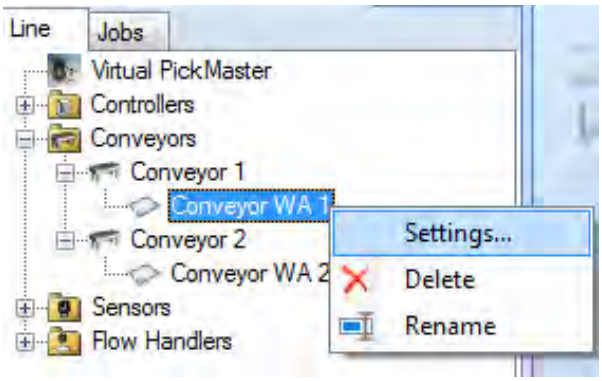
xx1300002290

	Description
Conveyor Name	Displays the name of the conveyor.
Hotspots list	Displays the available hotspots.
Add icon	Allows you to create a new hotspot. When you click the add icon, a new hotspot is added to the **Hotspots** list. You can then define the properties of the new hotspot.
Delete icon	Allows you to delete a selected hotspot.
Name	Allows you type a name for the hotspot.
Reference	Displays the name of the reference coordinate system.
Position	Allows you define the position of the hotspot.

Manage work area

Conveyor work area

Right-click on a work area. The following window is displayed.

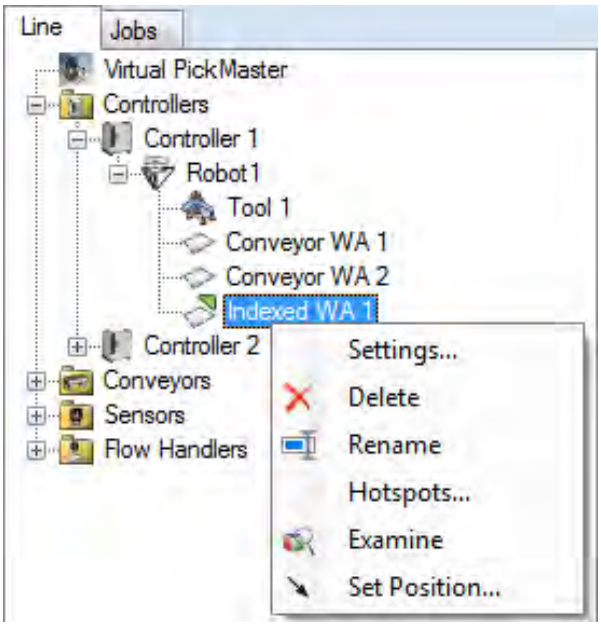


xx1300002292

Continues on next page

Indexed work area

Right-click on an indexed work area. The following window is displayed.



xx1300002707

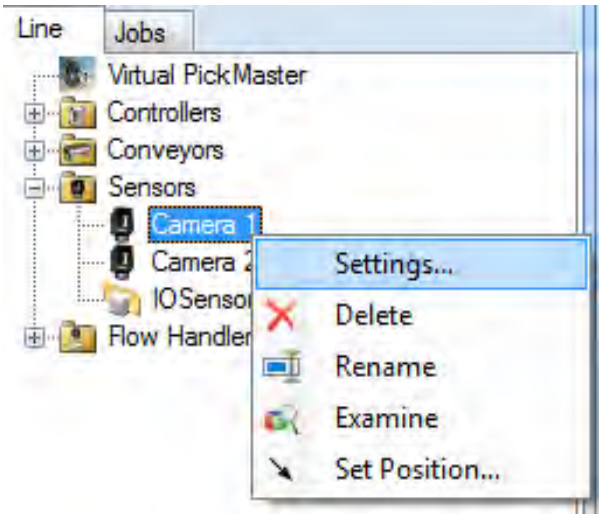
	Description
Settings...	Allows you to manage the settings of the selected work area. When you right-click on a conveyor work area and select **Settings...**, the **Conveyor Work Area manager** window is displayed. More details about managing a conveyor work area is available in the section *Conveyor work area on page 35*. When you right-click on an indexed work area and select **Settings...**, the **Indexed Work Area manager** window is displayed. More details about managing an indexed work area is available in the section *Indexed work area on page 37*.
Delete	Allows you to delete the selected conveyor work area.
Rename	Allows you to change the name of the selected conveyor work area.
Hotspots...	Allows you to manage the hotspots for indexed work areas. When you select **Hotspots...**, the **Edit Work area hotspots** window is displayed. The dialogue is similar to the **Edit Conveyor hotspots** window as described in the section *Managing conveyor on page 95*. **Note** The hotspot is a saved location on the indexed work area. A hotspot is used to define where on the indexed work area the flow shall be generated. There is always a default hotspot, Hotspot0, located at the origin of the indexed work area. If the flow appears at wrong location, modify the hotspot location to adjust.
Examine	Allows you to examine the selected indexed work area in the Station view.
Set Position...	Allows you a set a position for the selected indexed work area. When you select **Set Position...**, the **Set Position configuration** window is displayed from where you can configure a new position.

Continues on next page

3.3.2 Line tab
Continued

Sensors

Right-click on a sensor. The following window is displayed.

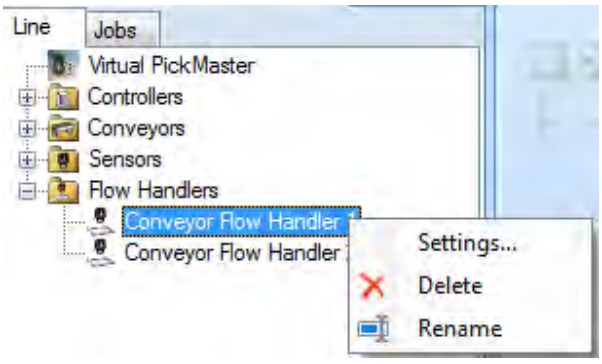


xx1300002293

	Description
Settings...	Allows you to manage the settings of the selected senor. When you right-click on a camera and select **Settings...**, the **Camera Properties** window is displayed. More details about managing a camera is available in the section *Cameras on page 40*. When you right-click on an I/O sensor and select **Settings...**, the **I/O Sensor Properties** window is displayed. More details about managing an I/O sensor is available in the section *I/O sensors on page 42*.
Delete	Allows you to delete the selected sensor.
Rename	Allows you to change the name of the selected sensor.
Examine	Allows you to examine the selected sensor in the Station view.
Set Position	Allows you to set position of the selected sensor.

Flow handler

Right-click on a flow handler. The following window is displayed.



xx1300002294

Continues on next page

	Description
Settings...	Allows you to manage the flow handler. When you right-click on a conveyor flow handler and select the **Settings...** option, the **Conveyor Flow Handler** window is displayed. More details about managing conveyor flow handler is available in the section *Conveyor flow handler on page 44*. When you right-click on an indexed flow handler and select the **Settings...** option, the **Indexed Flow Handler** window is displayed. More details about managing indexed flow handler is available in the section *Indexed flow handler on page 46*.
Delete	Allows you to delete the selected conveyor flow handler.
Rename	Allows you to change the name of the selected conveyor flow handler.

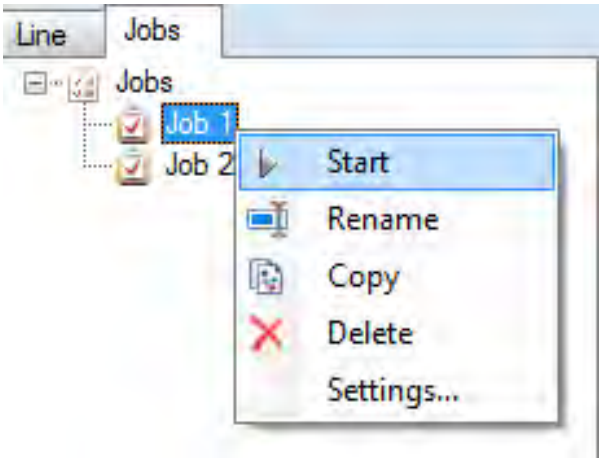
3.3.3 Jobs tab

Overview

The following objects and configurations are saved in the job.

- A selection of Items, including distribution settings on work areas and sensor detection.
- A selection of Container Patterns, including sensor detection.
- Work area settings, for example, enter and exit limits.
- Robot settings, including RAPID program.
- Conveyor settings, for example, speed, acceleration, and assigned flows.

The following figure and table provide details about managing jobs from the **Jobs** tab.



xx1300002295

	Description
Start	Allows you to start the simulation of the selected job.
Rename	Allows you to change the name of the selected job.
Copy	Allows you to make a copy of the selected job.
Delete	Allows you to delete the selected job.
Settings...	Allows you to manage the settings for the selected job. When you select the **Settings...** option, the **Edit Job** window is displayed. More details about managing job is available in the section *Job on page 59*.

4 Workflow of the Picking PowerPac

4.1 Workflow

Introduction

The following is the recommended working procedure for Picking PowerPac.

4.2 Station

Creating a station

Before starting Picking PowerPac, you must create a RobotStudio station. You must also create at least one robot system to simulate a PickMaster application.

> **Note**
>
> In a station without robot systems, you can configure a PickMaster application but not simulate it.

To create a station and a system, and start using Picking PowerPac:

1. In RobotStudio, create an empty station.
2. Select the **Home** tab, select **ABB library** and add one IRB 360 robot to the station.
3. Select **Robot System**, from Layout. Also add the following required system options:
   - 709-1 DeviceNet m/s
   - 642-1 PickMaster 3, with the following sub options:
     - Number of conveyors. This the number of conveyors that the robot shall be able to track. In this work flow example, at least one conveyor is needed.
     - Number of I/O boards. One I/O board is enough to simulate most PickMaster applications.

   > **Note**
   >
   > Since the robot is upside down, rotate it 180 degrees and place it on a suitable height. Always answer **Yes** to the message **Do you also want to move the task Frame?** when you move the robot.
4. The station is created.

> **Note**
>
> For more information on how to create a station and a system, see *Operating manual - RobotStudio*.

4.3 Creating a basic application

Procedure

Use the following procedure to create a basic application where an IRB 360 robot is picking items from a conveyor and placing them on a container on an indexed work area.

1. Open **RobotStudio** and follow the steps as described in the section *Station on page 102*.
2. On the **Add-Ins** tab in the ribbon, select **Picking** from the **PowerPacs** group.

   The **Picking PowerPac** opens and **Virtual PickMaster** is launched.

   A dedicated tab for Picking is added to the ribbon. The Picking ribbon and tree view browser opens.
3. Click **Add picking controller**. Select the created robot system. Select the robot and the conveyor board. More details about configuring the picking controller is available in the section *Add picking controller on page 22*.
4. Click **Add Picking Tool** to add a tool on the robot. More details about configuring the picking tool is available in the section *Add Picking Tool on page 24*.
5. Click **Add Conveyor**. Make sure the robot can reach the middle of the conveyor with the attached tool. More details about configuring the conveyor is available in the section *Add Conveyor on page 31*.
6. Click **Add Workarea** to define where the items are to be picked, in this example, a conveyor work area. Add another work area to define where items are to be placed, in this example, an indexed work area. Make sure the robot can reach the indexed work area. More details about configuring the work area is available in the section *Add Workarea on page 34*.
7. Click **Add Sensor->ABB cameras->Basler Scout** and attach it to the conveyor. The camera is needed to detect a flow of items entering the conveyor. More details about configuring the sensors is available in the section *Add Sensor on page 39*.
8. Click **Add Flow Handler** to handle the items that are to be picked from the conveyor. Add a conveyor flow handler. Select camera as item detection method, select the created camera, select distance trigger. Select the conveyor and the work area.
9. Click **Add Flow Handler** to handle empty place locations where items are to be placed. Add an indexed flow handler. Select I/O as item detection method. Select the indexed work area. More details about configuring the flow handler is available in the section *Add Flow Handler on page 43*.
10. Click **Item/Container** and add one item and one container. More details about configuring item/container is available in the section *Item/Container on page 48*.
11. Click **Container Patterns** and add a container pattern. Use the created container and item in the container pattern. More details about configuring container pattern is available in the section *Container Patterns on page 54*.

Continues on next page

12 Click **Add Job** and add a job. More details about configuring job is available in the section *Job on page 59*.

13 Add the item to the job. Since the item shall be detected by the camera, select camera detection on the conveyor flow handler. Select pick on the conveyor work area. Select place on the indexed work area.

14 Add the container pattern to the job. Since the container pattern shall be detected on the indexed work area, select I/O detection on the indexed flow handler.

15 Click **Calibrate** > **Quick Calibration** and calibrate the conveyor work area. The controller is automatically restarted. More details about calibration is available in the section *Calibrate on page 74*.

16 Press the play button in the **Simulation** group.

A default flow of items appear on the conveyor and an empty container appears on the indexed work area. The robot picks the first item when it comes with in its reach and places it on the container. More details about simulation controls is available in the section *Simulation controls on page 75*.

5 RAPID program

5.1 RAPID programs

Introduction

Overview

Each robot has a default RAPID program that can be edited using a normal text editor from the robot settings of the job dialog. When a job is started, the program is downloaded by PickMaster in the picking controller. The program contains the Main routine where the program execution starts.

Note

Due to the download procedure, this program cannot be modified directly on the robot system.

The installation contains the following program template files:

Template	Customized for
PMppa360_IRC5.prg	Four axes FlexPicker IRB 360.
PMppa360_IRC5_MM2.prg	Four axes FlexPicker IRB 360. To be used for a second multi move robot.
PMppa360_IRC5_DoublePick.prg	Four axes FlexPicker IRB 360. Adapted for double pick, single place.
PMppa360_IRC5_6_02.prg	Four axes FlexPicker IRB 360. To be used with RobotWare 6.02.
PMppa360_IRC5_6_02_DoublePick.prg	Four axes FlexPicker IRB 360. Adapted for double pick, single place. To be used with RobotWare 6.02.
PMppa360_IRC5_DoublePick.prg	Four axes FlexPicker IRB 360 adapted for double pick, single place.
PMppa_6Axes_IRC5.prg	Six axes robots of articulated arm type, for example, IRB 120.
PMppa_6Axes_IRC5_6_02.prg	Six axes robots of articulated arm type, for example, IRB 120. To be used with RobotWare 6.02.

Note

PMppa360_IRC5.prg is the default selection. If you have selected a 6 axis robot of articulated arm type, change the template to PMppa_6Axes_IRC5.prg.. The template files are located in the installation folder.

Program execution - General

The RAPID program is loaded and started from the Main routine by PickMaster when a new job is started.

For every cycle, the default RAPID program performs:

- a pick on a pick work area.
- a place on a place work area.

Continues on next page

If there are more than one pick work area with a robot, it uses the one having the lowest configured work area index. If there are more than one place work area with a robot, it uses the one having the lowest configured work area index. The RAPID program can be modified to implement another sequence, for example, to double pick with single place.

Program execution – Work areas

In RAPID, a work area is always associated with an item source object. The item source is sometimes referred to as a queue. The item source holds all target positions related to this work area. Target positions are continuously received in the item source, while being detected with the associated flow handler sensor.

Program execution – Target positions

For each pick, a pick target is fetched from the pick item source. The target position gives the location of the next item to be picked.

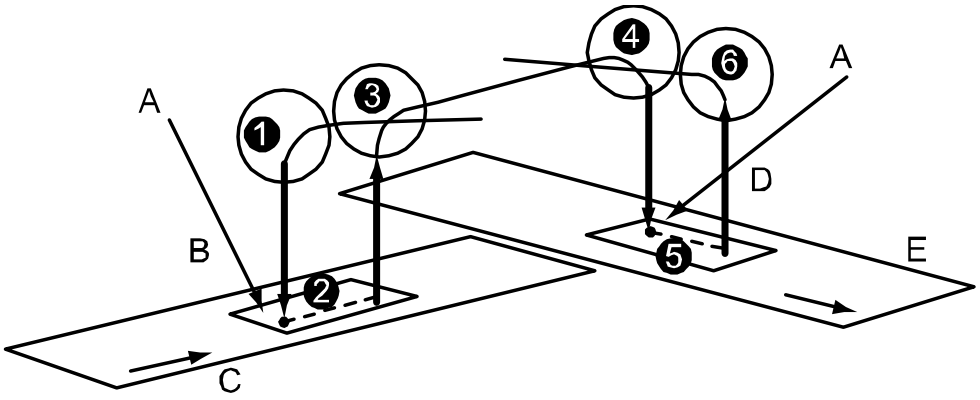
For each place, a place target is fetched from the place item source. The target position gives the location of the next empty place location for the item to be placed.

Movements

The RAPID program is built with six different movements.

For a six axis robot, the following two intermediate points must be used:

- Between position 3 and position 4.
- Between position 6 and the next loop's position 1.



xx0800000326

The following six movements are included.

	Description
1	Approach position above the pick target. The distance above the pick target is the pick elevation value, in negative z-direction of the tool, given in the **Work Area Properties** dialog in the job dialog. The target is of corner path type and the vacuum activation occasion is calculated as the time before the middle of the corner path. The time is entered in the **Work Area Properties** dialog.
2	This is the pick target. The robot TCP is coordinated relative to the conveyor during the pick time entered in the **Work Area Properties** dialog. The TCP follows the pick target during the pick time.

Continues on next page

	Description
3	Last position in the pick sequence. The distance above the pick target is calculated in the same way as the approach position. The position is coordinated to the conveyor until the middle of the corner path. Therefore the used item target must be acknowledged, so the item source can start tracking the next item target in the pick work area buffer. The target cannot be a fine point.
4	Approach position above the place target. The distance above the place target is the pick or place elevation value, in negative z-direction of the tool, given in the **Work Area Properties** dialog.
5	This is the place target. The robot TCP is coordinated relative to the conveyor during the place. The moment for the vacuum reversion event is calculated as the time before the half place time. The vacuum off moment is calculated as a time after the half place time.
6	Last position of the sequence. The position is coordinated to the conveyor until the TCP passes the middle of the corner path or goes into the fine point. Therefore the used item target must be acknowledged, so the item source can start tracking the next item target in the pick work area buffer. The target cannot be a fine point.

Note

When running a pick and place cycle over moving conveyors, the RAPID program pointer runs in advance and picks out a target long before it is going to be used. By the time the robot uses the target it may already have moved past the exit limit. RAPID moves the program pointer in advance about 100ms. In a coordinated fine point the "running in advance" is triggered at the beginning of the fine point movement as the robot locks above the conveyor. If the PickTime is long (for example, 50ms) the next target will be taken out of the queue long before (50ms) the robot is physically going to go there. If the conveyor speed is high 50ms may mean that the target to pick is already beyond the exit limit. Still the robot will try to pick it.

Program modules

The default RAPID program contains three program modules.

Module	Description
PPAMAIN	Handles the main program initiations and execution sequence. Do not edit this module for customization purpose.
PPASERVICE	Contains the service routines and service variables that can be executed and set from PickMaster.
PPAEXECUTING	Handles the pick and place operations. Customize this module for your purposes.

Continues on next page

System modules

An ABB IRC5 robot controller with the RobotWare option *Prepared for PickMaster (PickWare)* will always contain the loaded system modules *ppaBase* (crypted) and *ppaUser* (open).

Module	Description
ppaBase	Contains variables for communication with PickMaster, event routines and routines for creating, initiating, and deleting item sources.
ppaUser	Contains declarations of public data types and holds the work object data for indexed work areas. It also contains the declaration of default tool data, for example, `PickAct1` and `PickAct2`.

Public data types

Overview

The system module *ppaUser* contains two record definitions, *sourcedata* and *noncnvwobjdata*.

sourcedata

The *sourcedata* is used in the variable array *ItmSrcData*. This array holds data about every item source.

The record can be extended for other purposes, but do not change or delete any component in the structure.

Name	Alias	Description
`Used`	bool	Flag to indicate that the array index is used.
`ItemSource`	itmsrc	Descriptor to the item source.
`SourceType`	itmsrctype	Type of source, `PICK_TYPE`, `PLACE_TYPE` or `UNDEFINED_TYPE`.
`Ack`	`triggdata`	Triggdata for acknowledging the item targets.
`Nack`	`triggdata`	Triggdata for negative acknowledging the item targets.
`SimAttach1`	`triggdata`	Triggdata for attaching a nearby item to activator 1 in simulation.
`SimAttach2`	`triggdata`	Triggdata for attaching a nearby item to activator 2 in simulation.
`SimDetach1`	`triggdata`	Triggdata for detaching an item held by activator 1 in simulation.
`SimDetach2`	`triggdata`	Triggdata for detaching an item held by activator 2 in simulation.
`VacuumAct1`	`triggdata`	Triggdata for vacuum activation on real robot.
`VacuumAct2`	`triggdata`	Triggdata for vacuum activation on real robot.
`VacuumRev1`	`triggdata`	Triggdata for vacuum blow on real robot.
`VacuumRev2`	`triggdata`	Triggdata for vacuum blow on real robot.
`VacuumOff1`	`triggdata`	Triggdata for vacuum off on real robot.
`VacuumOff2`	`triggdata`	Triggdata for vacuum off on real robot.
`Wobj`	`wobjdata`	Work object data for the source

Continues on next page

Name	Alias	Description
VacActDelay	num	Vacuum activation delay
VacRevDelay	num	Vacuum reversion delay
VacOffDelay	num	Vacuum off delay
TunePos	pos	Position tuning for the work area.
TrackPoint	stoppointdata	Follow time data.
OffsZ	num	Height for the offset point above the pick or place position.

noncnvwobjdata

The *noncnvwobjdata* is used in the persistent variable array *NonCnvWOData*. This is only used for indexed work areas. The work object data is stored in this array. This data is then used when the item sources are created.

The record can be extended for other purposes, but do not change or delete any component in the structure.

Name	Alias	Description
Used	bool	Flag to indicate that the array index is used.
NonCnvWobjName	string	Name of the work area.
Wobj	wobjdata	The stored work object data.

AlwaysClearPath

Clear path

The robot path is cleared before the restart when a stop occurs during a motion that is coordinated to a moving work object. Otherwise the coordinated motion continues the stored path, but the position of the object in the conveyor may have changed to a position that is out of reach by the robot.

Unconditional path clearing

The AlwaysClearPath (bool always) routine unconditionally clears the path before the restart, if the input parameter value is set to TRUE.

5.2 Variables

Introduction to variables

The PickMaster robot controller contains many RAPID variables. The variables are declared in both `ppaBase` and `ppaUser`. Many are not used in customized programs.

Public variables in ppaUser

Overview

The following variables in `ppaUser` can be used.

VAR sourcedata ItmSrcData{MaxNoSources}

This array variable keeps information about all work areas. The index given in the work area configuration is the index of the `ItmSrcData` array.

PERS noncnvwobjdata NonCnvWOData{MaxNoSources}:=[[…

This array variable stores the work object frames for the indexed work areas. The key to find a certain work object calibration is the name, that must be same as the name in the work area configuration.

TASK PERS tooldata PickAct1:=[…]

This tooldata is used for pick and place operations.

> **Note**
>
> The direction of tool must fit the direction of items that are retrieved from the queue. The target positions of the items, which are retrieved from the queue, are rotated 180 degrees around their x-axis from the defined direction.

In an installation with a hanging IRB 360 and items lying on a horizontal conveyor, the tool's z-direction will point out from the nose and down into the conveyor, like `tool0`.

Public variables in ppaBase

The following variables in `ppaBase` can be used.

TASK PERS num Vtcp:=1000

Used for speed adjustment from PickMaster.

TASK PERS speeddata MaxSpeed:=[…]

Highest speed used for movements.

TASK PERS speeddata LowSpeed:=[…]

Low speed used for movements.

TASK PERS speeddata VeryLowSpeed:=[…]

Lowest speed used for movements.

Continues on next page

Public variables in PickMaster template programs

The following public variables are used in the PickMaster template program.

VAR num PickWorkArea{X}:=0

The PickWorkArea array is used to specify from which work area the robot will pick an item. The pick work areas are ordered with respect to selection index.

`PickWorkArea{1}` has the lowest work area selection index.

`PickWorkArea{2}` has the second lowest selection index.

VAR num PlaceWorkArea{X}:=0

The PlaceWorkArea array is used to specify on which work area the robot will place an item. The place work areas are ordered with respect to selection index.

`PlaceWorkArea{1}` has the lowest work area selection index.

`PlaceWorkArea{2}` has the second lowest selection index.

VAR num OtherWorkArea{X}:=0

The OtherWorkArea array is used to specify to which work area the robot will go for a user defined purpose. The other work areas are ordered with respect to selection index.

`OtherWorkArea{1}` has the lowest work area selection index.

`OtherWorkArea{2}` has the second lowest selection index.

VAR itmtgt PickTarget:=[...]

Used to retrieve a pick target from a pick item source.

VAR itmtgt PlaceTarget:=[...]

Used to retrieve a place target from a place item source.

TASK PERS wobjdata WObjPick:=[…]

Holds the wobjdata for the work area. The information is moved from `ItmSrcData` to `WObjPick` in the Pick routine because the motion instructions need to have the wobjdata as `PERS` type.

TASK PERS wobjdata WObjPlace:=[…]

Holds the `wobjdata` for the work area. The information is moved from `ItmSrcData` to `WObjPlace` in the Place routine because the motion instructions need to have the `wobjdata` as `PERS` type.

TASK PERS robtarget SafePos:=[…]

Defined start position for the robot. Edit this robtarget to fit the application.

TASK PERS robtarget IntPosPickX:=[…]

Defined intermediate position for every pick work area robot. Edit this robtarget to fit each work area.

TASK PERS robtarget IntPosPlaceX:=[…]

Defined intermediate position for every place work area robot. Edit this robtarget to fit each work area.

Continues on next page

TASK PERS loaddata ItemLoad:=[…]

Load data (`loaddata`) used for pick and place operations. Edit this `loaddata` to fit the picked item. If different item types are used, declare one loaddata for each type. It is important that correct loaddata is used to get the best performance of the robot.

The default loaddata is the same as `tooldataload0`.

5.3 Routines

Introduction to routines

The PickMaster RAPID modules contain many routines, some are very useful for the end user, others are only to be used internally by the PickMaster program.

Public routines in PickMaster template programs

The following public routines are available in the PickMaster template programs.

PROC main()

Start routine for the RAPID program. The program will always start from this routine.

PROC InitSafeStop()

Initiates the SafeStop trap. It must be executed at the beginning of the program execution to get a correct robot stop when the PickMaster project is paused or stopped.

PROC InitTriggs()

Sets trigger events for the vacuum activation, reversion and turning off, at the project start for every used work area index. See more at `SetTriggs`.

PROC InitPickTune()

Initiates the PickTune trap. Must be executed at the beginning of the project start so the work areas can be tuned.

PROC SetTriggs(num Index)

Sets trigger events for the vacuum activation, reversion and turning off. The default program only sets up events for one vacuum ejector on the I/O group `goVacBlow1`. If more than one vacuum ejector is used, the new vacuum ejector I/O group must be setup for the correct work area and the default routine must be edited to get the right vacuum ejector to each work area.

PROC InitSpeed()

Sets the robot speed used in the program. The instruction `VelSet` is executed in this routine, which sets the maximum allowed speed for the robot. If a six axes robot is used, this limit can be tuned to avoid motion errors.

PROC PickPlace()

Starts the item queues and initiates the final settings. The pick and place sequence is called from this routine. Do not make changes in this routine.

This routine is called when the pick and place execution is started.

PROC SafeStop()

When the project is stopped or paused this routine will be called either from the `SafeStopTrap` routine or the `PickPlace` routine. The slow motion to the safe position is called from this routine.

Continues on next page

PROC GotoRestartPos()

Runs the slow motion to the safe position and sends a negative acknowledge to all item sources. This must be done to tell the sources that the execution was interrupted.

PROC Home()

Service routine that moves the robot to the safe position.

PROC WashDown()

Wash down service routine.

PROC TestCycle()

Test service routine.

PROC Homepos()

Service routine that moves the robot to the synchronization position.

PROC EnumerateWorkAreas()

Sets up the arrays of work areas for Pick, Place, and Other.

PROC PickPlaceSeq()

Specifies the sequence of the application, that is the logic of how the robot will pick and place from different queues.

This routine is called once every loop, which is counted as one pick in the pick rate statistics shown in the PickMaster production tab.

PROC Pick(num Index)

Executes one pick. The index defines which work area the item will be picked from.

PROC Place(num Index)

Executes one place. The given index defines which work area the item will be placed on.

PROC GoInterMidP(robtarget Pos)

Executes a motion to the intermediate position `Pos` (Used only for six axes robots).

TRAP SafeStopTrap

Trap routine to catch the stop I/O signal. This is executed if the stop I/O signal is set before `SafeStop` is called from the `PickPlace` routine.

TRAP PickTuneTrap

Trap routine to attach the tuned values from the PickMaster to the corresponding variables.

Hidden routines in ppaBase module

Overview

Following are the hidden routines in the `ppaBase` module.

PROC ResetEvent()

Resets some variables. This routine is only executed in the `RESET` system event shelf.

Continues on next page

PROC PowerOnEvent()

Resets some variables. This routine is executed only in the `POWER_ON` system event shelf.

PROC StopEvent()

Clears the robot path if the robot is in a coordinated motion when the stop occurs. This routine is only executed in the `STOP` system event shelf.

PROC RestartEvent()

This routine is only executed in the `RESTART` system event shelf. If the robot is currently in a coordinated motion, this routine will force the program to restart the program from the level that has an error handler for the raised error `PPA_RESTART`.

PROC NewSource()

Creates a new item source and initiates the `ItmSrcData` variable. PickMaster calls this routine for each work area when the project starts.

PROC ClearAll()

Resets all important variables and deletes all item sources. This routine is called when the project is stopped.

PROC PickRateInit()

Initiates the pick rate calculation.

PROC PickRateReset()

Resets the pick rate calculation.

PROC CheckAx4Rev ()

Checks if it is necessary to reset the fourth axis on the IRB340.

PROC ResetAx4 (VAR mecunit MechUnit)

Resets the fourth axis.

PROC NotifyClearAll ()

Tells PickMaster that `ClearAll` is executed.

PROC NotifySafeStop ()

Tells PickMaster that `SafeStop` is executed.

PROC NotifyRunning ()

Tells PickMaster that the process is running.

PROC NotifyWaitForExe ()

Tells PickMaster that the RAPID program is waiting for new order.

```
PROC WaitForExeOrder ()
```

PROC IncrPicks ()

Increments the pick calculation.

PROC ppaDropWobj(PERS wobjdata Wobj)

Encapsulates the `DropWobj` instruction. See *Application manual - Conveyor tracking* for more information

Continues on next page

PROC ppaWaitWobj(PERS wobjdata Wobj, \num RelDist \num MaxTime \bool TimeFlag)

Encapsulates the `WaitWobj` instruction. See *Application manual - Conveyor tracking* for more information

PROC ppaCnvGenInstr(VAR mecunit MechUnit, num cnvcmd, cnvgendata Data)

Encapsulates the `CnvGenInstr` instruction. See *Application manual - Conveyor tracking* for more information

PROC WalkTheData()

Traces the content of the array variables *ItmSrcData* and *NonCnvWOData*, which can be useful when trying to find an error. It prints the file TheData.log on the system directory on the controller.

TRAP PickRateTrap

Trap routine to calculate the correct pick rate for the robot.

5.4 Service routines

User defined service routines

It is possible to create new routines and variables. All procedure in the PPASERVICE module can be executed as service routines. All service variable names must be set in a ServiceVarX.

The service variables can only be of type num.

Maximum number of service variables is ten.

This page is intentionally left blank

6 Advanced programming examples

6.1 Example: Mixing one pick work area and two place work areas

Description of example

In this example we use one pick work area with two types of items. The items are put on two out work areas depending on type of item.

1 Pick item from pick work area
2 Define type of item
3 Place on out work area

Example code

```
PROC PickPlaceSeq()
  Pick PickWorkArea{1};
  IF PickTarget.Type = 1 THEN
    Place PlaceWorkArea{1};
  ELSEIF PickTarget.Type = 2 THEN
    Place PlaceWorkArea{2};
  ENDIF
ENDPROC
```

6.2 Example: Mixing two pick work areas and one place work area

Description of example

In this example, we use the place work area as master to decide which item is needed to fill a pattern, which in turn defines pick work area to pick from.

1 Check next item target type
2 Decide which work area to pick from
3 Pick item from pick work area
4 Place on out work area

Example code

```
PROC PickPlaceSeq()
  VAR num PlaceType:=0;

  NextItmTgtType
    ItmSrcData{PlaceWorkArea{1}}.ItemSource,
    PlaceType;
  IF PlaceType = 1 THEN
    Pick PickWorkArea{1};
  ELSEIF PlaceType = 2 THEN
    Pick PickWorkArea{2};
  ENDIF
  Place PlaceWorkArea{1};
ENDPROC
```

6.3 Example: Mixing with one pick and one place work area

Description of example

In this example we use the place work area as master to decide which item is needed to fill a pattern, which in turn defines which item to pick.

1 Check next item target type
2 Pick item from pick work area
3 Place on out work area

Example code

```
PROC Pick(num Index)
  VAR num PickType:=0;
  VAR num PlaceType:=0;

  WObjPick:=ItmSrcData{Index}.Wobj;
  NextItmTgtType
    ItmSrcData{PlaceWorkArea{1}}.ItemSource,PlaceType;
  TEST PlaceType
  CASE 4:
    PickType:=1;
  CASE 5:
    PickType:=2;
  CASE 6:
    PickType:=3;
  ENDTEST
  GetItmTgt ItmSrcData{Index}.ItemSource, PickTarget
       \ItemType:=PickType;
  TriggL \Conc, RelTool(PickTarget.RobTgt, 0, 0,
       -ItmSrcData{Index}.OffsZ), MaxSpeed,
       ItmSrcData{Index}.VacuumAct1, z20, PickAct1 \WObj:=WObjPick;
  MoveL \Conc, PickTarget.RobTgt, LowSpeed, z5 \Inpos:=
       ItmSrcData{Index}.TrackPoint, PickAct1 \WObj:=WObjPick;
  GripLoad ItemLoad;
  TriggL RelTool(PickTarget.RobTgt, 0, 0, -ItmSrcData{Index}.OffsZ),
       LowSpeed, ItmSrcData{Index}.Ack, z20, PickAct1
       \WObj:=WObjPick;
ENDPROC
```

6.4 Example: Double pick single place

Description of example

The robot shall pick up two items, one-by-one, on the infeeder conveyor, and then place both items on the outfeed conveyor. This operation requires a picking tool with two vacuum ejectors.

Implementation

As a starting point, create a simple working setup with one robot.

The RAPID program needs to be modified. To edit the RAPID program, go to the **Project view**, select a robot and display the drop down menu, select the **Rapid program** and select **Edit...**.

The `PickPlaceSeq` routine shall perform two Pick routine calls to handle the first and the second pick. It will then perform one Place routine call to handle the simultaneous placing of the picked up items. See the following example code.

```
!
! Procedure PickPlaceSeq
!
! The Pick and Place sequence.
! Edit this routine to specify how the robot shall execute the
    movements.
!

!***********************************************************
PROC PickPlaceSeq()
  Pick PickWorkArea{1}, 1;
  Pick PickWorkArea{1}, 2;
  Place PickWorkArea{1};
ENDPROC
```

For the Pick routine, see the following example code. Note the usage of `PickAct2` and `VacuumAct2` for the second pick.

```
!***********************************************************
!
! Procedure Pick
!
! Executes a pick movement.
! Edit this routine to modify how the robot shall
! execute the pick movements.
! Needs to be changed if more than one activator is used.
!
!***********************************************************
PROC Pick(num Index, num pickNo)
  IF Index > 0 THEN
    WObjPick:=ItmSrcData{Index}.Wobj;
    GetItmTgt ItmSrcData{Index}.ItemSource,PickTarget;
    IF pickNo = 1 THEN
      TriggL\Conc,RelTool(PickTarget.RobTgt,0,0,
      -ItmSrcData{Index}.OffsZ),
```

Continues on next page

```
        MaxSpeed,ItmSrcData{Index}.VacuumAct1,z20,
        PickAct1\WObj:=WObjPick;
        TriggL\Conc,PickTarget.RobTgt,LowSpeed,ItmSrcData{Index}.SimAttach1,
        z5\Inpos:=ItmSrcData{Index}.TrackPoint,
        PickAct1\WObj:=WObjPick;
        GripLoad ItemLoad;
        TriggL
        RelTool(PickTarget.RobTgt,0,0,-ItmSrcData{Index}.OffsZ),
        LowSpeed,ItmSrcData{Index}.Ack,z20,PickAct1\WObj:=WObjPick;
      ELSEIF pickNo = 2 THEN
        TriggL\Conc,RelTool(PickTarget.RobTgt,0,0,-ItmSrcData{Index}.OffsZ),
        MaxSpeed,ItmSrcData{Index}.VacuumAct2,
        z20,PickAct2\WObj:=WObjPick;
        TriggL\Conc,PickTarget.RobTgt,LowSpeed,ItmSrcData{Index}.SimAttach2,
        z5\Inpos:=ItmSrcData{Index}.TrackPoint,
        PickAct2\WObj:=WObjPick;
        GripLoad ItemLoad;
        TriggL
        RelTool(PickTarget.RobTgt,0,0,-ItmSrcData{Index}.OffsZ),
        LowSpeed,ItmSrcData{Index}.Ack,z20,
        PickAct2\WObj:=WObjPick;
      ENDIF
    ELSE
      ErrWrite "Missing item distribution", "Cannot pick because no
           item distribution contains current work area."
      \RL2:="Please check configuration";
      SafeStop;
    ENDIF
  ENDPROC
```

The tooldata `PickAct1` is used at the first pick. The tooldata `PickAct2` is used at the second pick. Update `PickAct1` and `PickAct2` (defined in module `ppaUser.sys`): Define the tool center point in the center of the controlled vacuum ejector. Update also the weight and the center of mass. Save the updates of the RAPID program, close the editor, and apply the updates.

For the Place routine see the following example. Note the usage of `VacuumOff1` and `VacuumOff2` for the simultaneous placing of both held items.

```
!*********************************************************
!
! Procedure Place
!
! Executes a place movement.
! Edit this routine to modify how the robot shall
! execute the place movements.
! Needs to be changed if more than one activator is used.
!
!*********************************************************
PROC Place(num Index)
  IF Index > 0 THEN
    WObjPlace:=ItmSrcData{Index}.Wobj;
    GetItmTgt ItmSrcData{Index}.ItemSource,PlaceTarget;
```

Continues on next page

```
      MoveL\Conc,RelTool(PlaceTarget.RobTgt,0,0,-ItmSrcData{Index}.OffsZ),
      MaxSpeed,z20,PlaceAll\WObj:=WObjPlace;
      TriggL\Conc,PlaceTarget.RobTgt,LowSpeed,ItmSrcData{Index}.VacuumRev1\T2:=
      ItmSrcData{Index}.VacuumOff1\T3:=ItmSrcData{Index}.VacuumOff2\T4:=
      ItmSrcData{Index}.VacuumRev2\T5:=ItmSrcData{Index}.SimDetach1\T6:=
      ItmSrcData{Index}.SimDetach2,z5\Inpos:=
      ItmSrcData{Index}.TrackPoint,PlaceAll\WObj:=WObjPlace;
      GripLoad load0;
      TriggL RelTool(PlaceTarget.RobTgt,0,0,-ItmSrcData{Index}.OffsZ),
      LowSpeed,ItmSrcData{Index}.Ack,z20,PlaceAll\WObj:=WObjPlace;
    ELSE
      ErrWrite "Missing item distribution", "Cannot place because no
           item distribution contains current work area."
      \RL2:="Please check configuration";
      SafeStop;
    ENDIF
  ENDPROC
```

The tooldata `PlaceAll` (defined in module `ppaUser.sys`) is used at place. Update `PlaceAll`: Define the tool center point in the center of the controlled vacuum ejectors. Update also the weight and the center of mass. Save the updates of the RAPID program, close the editor, and apply the updates.

Note

Use the same method to setup a tool with more than two activators. However, a few additional setup steps are required. For example, using a tool with 3-4 activators requires the following additional steps:

1. Select two I/O boards as controller option. Alternatively, create additional signals `goVacBlow3`, `goVacBlow4`, `doVacuum3`, `doVacuum4`, `doBlow3`, and `doBlow4`. The first bit of `goVacBlowX` shall overlap the signal `doVacuumX`. The second bit of `goVacBlowX` shall overlap the signal `doBlowX`.
2. Update the `SetTriggs` routine. Enable the `TriggEquip` events `VacuumAct3`, `VacuumOff3`, `VacuumAct4`, and `VacuumOff4` by removing the comments on these lines.

6.5 Example: Selecting item depending on clearance zone

Description of example

In this example, we select items on a conveyor belt depending on the clearance zone around the item, that is if there is any other item target within a specified area. This is useful when it is important that the gripper does not touch surrounding objects.

The selection algorithm selects the object that is closest to the exit limit in x-direction and has no locking objects in the selection shape.

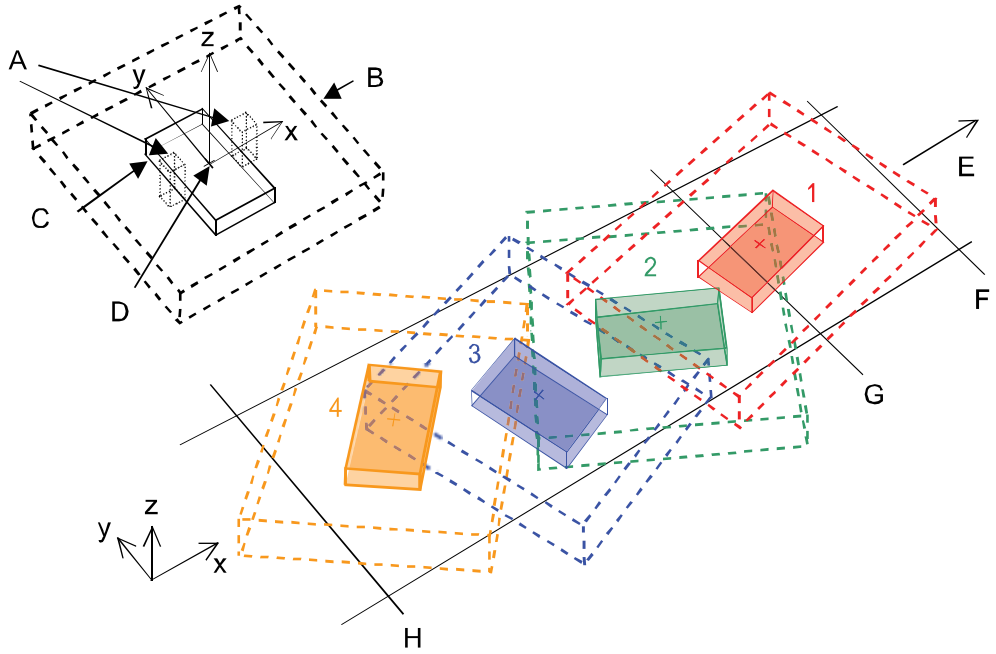
Use the check limit in x-direction as a parameter to the `GetItmTgt` instruction. This makes it possible to define the starting point from where the first object will be picked. The instruction will try to retrieve the first object between the check and enter limits. This will cause the selection algorithm to take all objects between the check limit and the exit limit into consideration when checking for the nearest objects. Therefore the distance between the check limit and the exit limit will be at least the diameter of the largest item.

The illustration below shows how the items are selected depending on the position and the orientation. The robot will first pick item 4 and then item 3. The other two will never be picked.

- Item 1 cannot be picked because it has passed the check limit, and item 2 is inside its selection shape.
- Item 2 cannot be picked because the positions of items 1 and 3 are inside its selection shape.
- Item 3 cannot be picked because item 4 is inside its selection area.
- Item 4 can be picked because no other item is its selection shape.
- Item 3 will be picked after item 4 is no longer present.

Continues on next page

6.5 Example: Selecting item depending on clearance zone
Continued



xx0800000323

A	Grippers
B	Selection shape
C	Item
D	Item target position
E	Product flow direction
F	Exit
G	Check limit
H	Enter

See *selectiondata - Selection data on page 154*.

Example code

```
PROC Pick(num Index)
  VAR selectiondata sel_data;
  VAR robtarget draw_target;
  VAR num check_limit;

  sel_data.ShapeType:=BOX;
  sel_data.ConsiderType:=BitOr(ITEMS_TO_USE,ITEMS_BYPASS);
  sel_data.GeometricData.x:=60;
  sel_data.GeometricData.y:=70;
  sel_data.GeometricData.z:=10;sel_data.GeometricData.radius:=0;
  sel_data.Offset.OffsetRelation:=ITEM_COORD_DIR;
  sel_data.Offset.OffsetPose.trans.x:=0;
  sel_data.Offset.OffsetPose.trans.y:=0;
  sel_data.Offset.OffsetPose.trans.z:=0;
  sel_data.Offset.OffsetPose.rot.q1:=1;
  sel_data.Offset.OffsetPose.rot.q2:=0;
```

Continues on next page

```
  sel_data.Offset.OffsetPose.rot.q3:=0;
  sel_data.Offset.OffsetPose.rot.q4:=0;
  check_limit:=150;

  WObjPick:=ItmSrcData{Index}.Wobj;
  GetItmTgt ItmSrcData{Index}.ItemSource,PickTarget
       \Limit:=check_limit\Selection:=sel_data;
  TriggL \Conc, RelTool(PickTarget.RobTgt, 0, 0,
       -ItmSrcData{Index}.OffsZ), MaxSpeed,
       ItmSrcData{Index}.VacuumAct1, z20, PickAct1\WObj:=WObjPick;
  MoveL \Conc, PickTarget.RobTgt, LowSpeed, z5 \Inpos:=
       ItmSrcData{Index}.TrackPoint, PickAct1\WObj:=WObjPick;
  GripLoad ItemLoad;
  TriggL RelTool(PickTarget.RobTgt, 0, 0, -ItmSrcData{Index}.OffsZ),
       LowSpeed, ItmSrcData{Index}.Ack, z20,
       PickAct1\WObj:=WObjPick;
ENDPROC
```

6.6 Example: Sorting in negative y-direction

Description of example

In this example, we shuffle items off a conveyor belt without touching surrounding objects. The shuffle movement is done perpendicular on the horizontal plane to the right side of the conveyor and the manipulator motion is coordinated with the conveyor motion.

The sorting algorithm selects the item closest to the exit limit in x-direction and has no locking objects in its selection shape.

The selection shape is defined as a long box. The shape's x-value is used to define the corridor width, the y-value must be more than half the width of the conveyor belt and the z-value must be greater than the largest difference in height among all items.

Set the y-value in the `OffsetData` to the negative y-value of the shape, the selection box will be moved out to the right.

As a result there must be a clear corridor to the right of every item before it is shuffled.

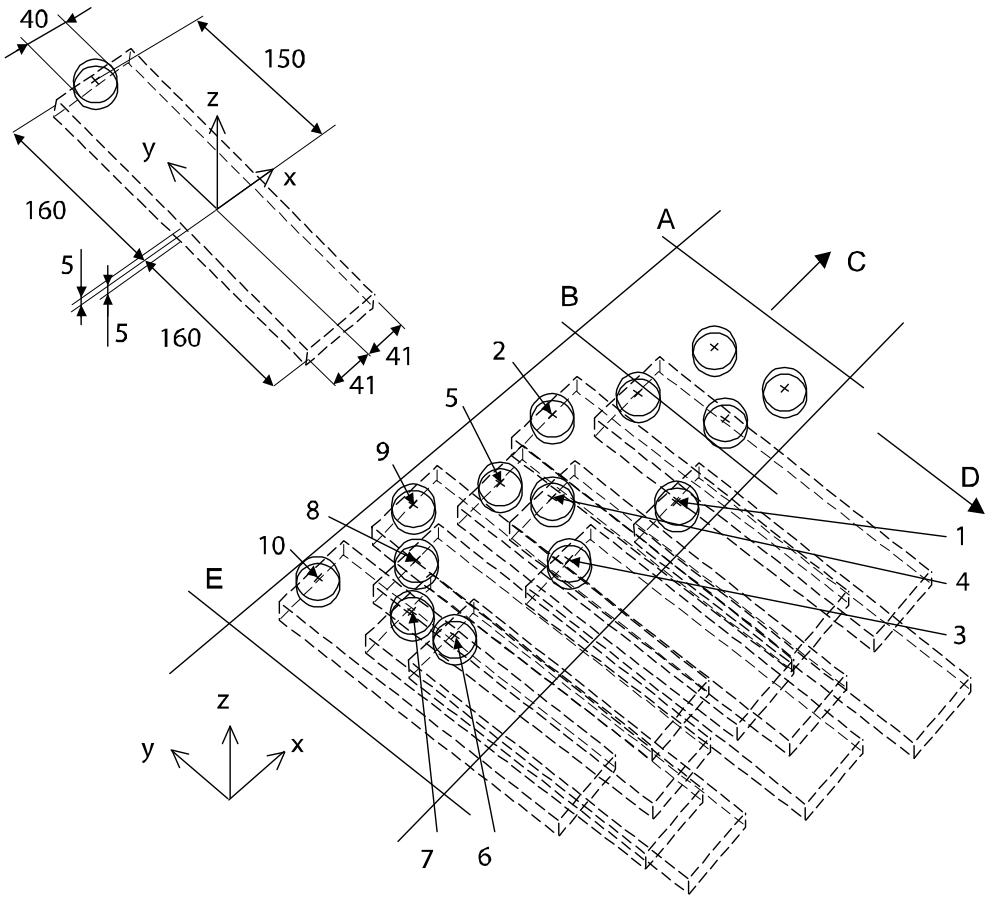
The algorithm will check both upwards and downwards the production flow for other items.

Use the check limit in the x-direction as a parameter to the `GetItmTgt` instruction, to define the starting point from where the first item will be shuffled. The instruction will try to shuffle the first item between the check and enter limits. This will also cause the selection algorithm to take all items between the check limit and the exit limit into consideration when checking for the nearest items. Therefore the distance between the check limit and the exit limit will be at least the diameter of the largest item.

In the illustration below, all items will be shuffled off to the right side of the conveyor belt. Because each item needs a clear zone, that is the shape of the *ShapeType*,

Continues on next page

the items will be shuffled off in the order 1 to 10 as numbered in the illustration below.



xx0800000324

A	Exit
B	Check limit
C	Product flow direction
D	Sort direction
E	Enter

Example code

```
PROC Pick(num Index)
  VAR selectiondata y_sort;
  VAR robtarget draw_target;
  VAR num check_limit;

  y_sort.ShapeType:=BOX;
  y_sort.ConsiderType:=BitOr(ITEMS_TO_USE,ITEMS_BYPASS);
  y_sort.GeometricData.x:=41;
  y_sort.GeometricData.y:=160;
  y_sort.GeometricData.z:=5;
  y_sort.GeometricData.radius:=0;
  y_sort.Offset.OffsetRelation:=FRAME_COORD_DIR;
```

Continues on next page

```
    y_sort.Offset.OffsetPose.trans.x:=0;
    y_sort.Offset.OffsetPose.trans.y:=-150;
    y_sort.Offset.OffsetPose.trans.z:=0;
    y_sort.Offset.OffsetPose.rot.q1:=1;
    y_sort.Offset.OffsetPose.rot.q2:=0;
    y_sort.Offset.OffsetPose.rot.q3:=0;
    y_sort.Offset.OffsetPose.rot.q4:=0;
    check_limit:=150;

    WObjPick:=ItmSrcData{Index}.Wobj;
    GetItmTgt ItmSrcData{Index}.ItemSource,PickTarget
         \Limit:=check_limit\Selection:= y_sort;
    TriggL\Conc, RelTool(PickTarget.RobTgt, 0, 0,
         -ItmSrcData{Index}.OffsZ), MaxSpeed,
         ItmSrcData{Index}.VacuumAct1, z20, Gripper\WObj:=WObjPick;
    MoveL\Conc, PickTarget.RobTgt, LowSpeed, z5
         \Inpos:=ItmSrcData{Index}.TrackPoint, Gripper
         \WObj:=WObjPick;
    GripLoad ItemLoad;
    draw_target:=PickTarget.RobTgt;
    draw_target.trans.y:=-200;
    draw_target.rot:=[0,1,0,0];
    TriggL draw_target, LowSpeed, ItmSrcData{Index}.Ack, z20,
         Gripper\WObj:=WObjPick;
ENDPROC
```

6.7 Controlling conveyor speed using an analog signal

Adding a new analog signal

Picking PowerPac supports control of conveyor speed using an analog output signal from the controller. To enable this feature, create an analog output signal (if not present already) and then configure a mapping between this signal and the conveyor speed property.

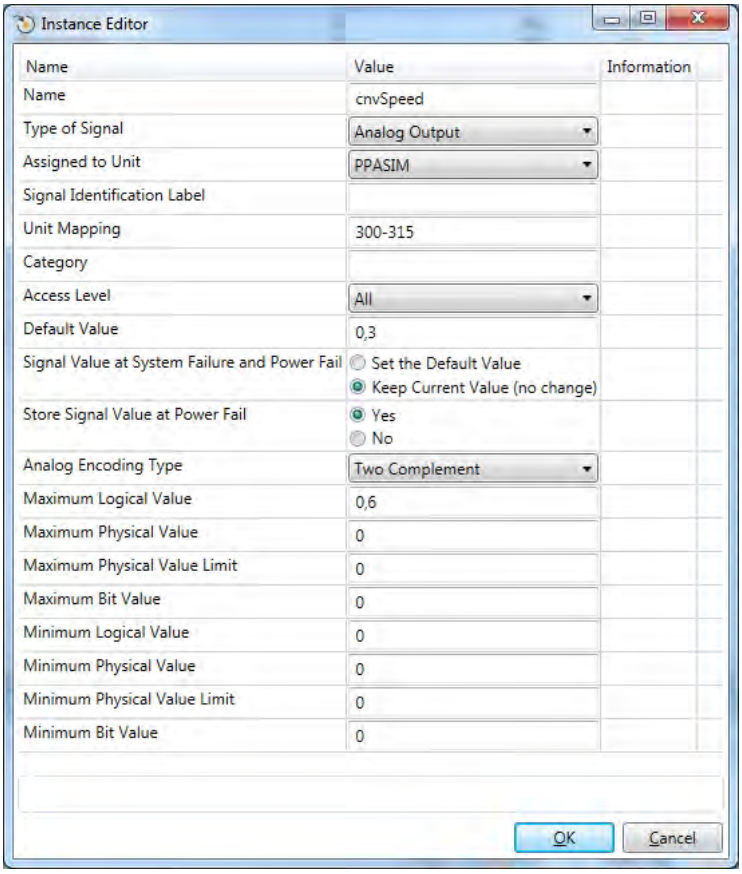
You can add a new analog output signal using the **Controller** tab of **RobotStudio**.

To add a new analog signal:

1. Click **Configuration Editor**, and select **I/O**.

   The I/O editor is displayed.
2. Right-click **Signal** and select **New Signal**.

   The **Instance Editor** window is displayed.



xx1300002279

3. Type a name for the signal in the **Name** field.
4. Select **Analog Output** from the **Type of signal** list.

   The corresponding parameters are loaded in the **Instance Editor**.
5. Select **PPASIM** in the **Assigned to Unit** list.
6. Type a unit mapping in the **Unit Mapping** field and ensure that it doesn't interfere with other signals mapping.

Continues on next page

7 Select **All** from the **Access Level** list. Access level must be set to **All** to set speed value at runtime.
8 Set minimum and maximum values for the other parameters.
9 Click **OK**.

   The analog signal is created. The controller needs to be restarted in order to use the signal.
10 Restart the controller.
11 In the Tree view browser **Line** tab right-click on the conveyor and select **Settings**.

   The **Conveyor Properties** window is displayed.
12 Click the **Edit Signals** button.

   The **Edit conveyor IO Signals** window is displayed.
13 Select the controller in which the analog signal was created from the **Controllers** list.
14 In the **I/O signal** column select the created I/O signal and assign it to the conveyor speed parameter.
15 Click **OK**.

> **Note**
>
> Once the I/O signal is assigned, the conveyor speed is controlled by the analog signal. Changes to the conveyor speed elsewhere in the application is not applicable. The unit of speed is in m/s.

7 RAPID reference information

7.1 Instructions

7.1.1 AckItmTgt - Acknowledge an item target

Usage

`AckItmTgt` is used to acknowledge that an `itmtgt` received with `GetItmTgt` from an item source has been used (For example, handled by the robot, skipped or put back in the queue for later usage). Normally, acknowledge is setup as a `TriggL` event on the path (using the `Ack` or `Nack` triggdata from sourcedata) to make sure acknowledge does not occur before any movements related to the target has been finished. However, if the received `itmtgt` shall be skipped or put back in the queue for later usage, movements related to the target may not be needed. Then it is convenient to use this instruction instead. Only after the acknowledge has been made, a new `itmtgt` can be fetched from the item source.

Basic example

```
VAR itmtgt PlaceTarget;
GetItmTgt ItmSrcData{Index}.ItemSource, PlaceItem;
AckItmTgt ItmSrcData{Index}.ItemSource, PlaceItem, FALSE
     \Skip:=TRUE;
```

Arguments

```
AckItmTgt ItemSource ItemTarget Acknowledge [\Skip] [\Type]
```

`ItemSource`

Data type: `itmsrc`

The item source from where the item target has been received with `GetItmTgt`.

`ItemTarget`

Data type: `itmtgt`

The item target to acknowledge.

`Acknowledge`

Data type: `bool`

The status of acknowledge. TRUE if the `itmtgt` has been handled (picked or placed)by the robot and FALSE otherwise, in which case the `itmtgt` is put back into the queue.

`Skip`

Data type: `bool`

Indicates if the `itmtgt` shall be skipped. If set to TRUE it will not be possible to receive the `itmtgt` again with `GetItmTgt`. If combined with `Acknowledge` = FALSE the `itmtgt` will be passed on for possible handling by downstream robots. If combined with `Acknowledge` = TRUE, skip will have no effect. If Skip is set to FALSE the `itmtgt` will either be considered as handled by the robot (when

Continues on next page

7.1.1 AckItmTgt - Acknowledge an item target
Continued

combined with `Acknowledge` = TRUE), or put back in the queue for later usage (when combined with `Acknowledge` = FALSE).

`Type`

Data type: `num`

Modifies the type of the `itmtgt`. If combined with `Acknowledge` = FALSE and `Skip` = TRUE, the item will be passed on to downstream robots according to the configured distribution of the new item type.

If combined with `Acknowledge` = FALSE and `Skip` = FALSE, the item will be put back in the queue with the new item type and can still be received with `GetItmTgt`. The item type will only be changed locally; the item type and the distribution of the item will not change for downstream robots.

If combined with `Acknowledge` = TRUE, type change will have no effect.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable ERRNO will be set to:

Error code	Description
`ERR_ITMSRC_UNDEF`	`itmsrc` undefined.

Limitations

The `itmtgt` must be received with the instruction `GetItmTgt`.

Syntax

```
AckItmTgt
  [ItemSource ':=' ] <variable (VAR) of itmsrc>,
  [ItemTarget ':=' ] <var or pers (INOUT) of itmtgt>,
  [Acknowledge':=' ] <expression (IN) of bool>,
  [\Skip ':=' ] <expression (IN) of bool>,
  [\Type ':=' ] <expression (IN) of num>;
```

Related information

For information about	See
The data type `itmtgt`	*itmtgt - Item target data on page 151*.

7.1.2 FlushItmSrc - Flush an item source

Usage

`FlushItmSrc` is used to flush an item source. The instruction clears the item source buffers, sets the scene number to one and flushes the encoder board.

Basic example

```
FlushItmSrc PlaceSource;
```

Flushes the earlier created item source object *PlaceSource*.

Arguments

```
FlushItmSrc ItemSource
```

`ItemSource`

Data type: `itmsrc`

The created item source.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable `ERRNO` will be set to:

Error code	Description
`ERR_ITMSRC_UNDEF`	`itmsrc` undefined

Limitations

To avoid potential problems, this instruction should be executed only when the last item target definitely has been acknowledged.

Syntax

```
FlushItmSrc
  [ItemSource ':=' ] <variable (VAR) of itmsrc>;
```

7.1.3 GetItmTgt - Get the next item target

Usage

`GetItmTgt` is used to get the next available `itmtgt` in the item source queue between the enter and the exit limit of the work area. The RAPID program waits in this instruction until the next item is possible to reach or the timeout occurs.

Basic examples

Basic examples of the instruction `GetItmTgt` are illustrated below.

Example 1

```
GetItmTgt PlaceSource, PlaceItem;
```

Receives a place item from the PlaceSource when there is one that can be used.

Example 2

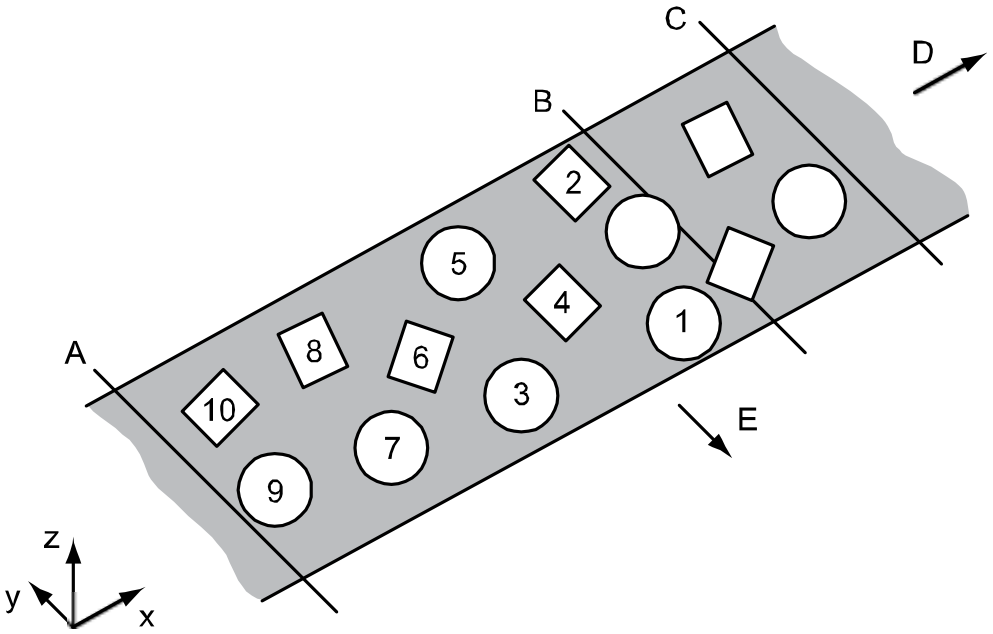
```
...
VAR selectiondata neg_y_sort;

neg_y_sort.ShapeType:=BOX;
neg_y_sort.ConsiderType:=BitOr(ITEMS_TO_USE,ITEMS_BYPASS);
neg_y_sort.GeometricData.x:=60;
neg_y_sort.GeometricData.y:=500;
neg_y_sort.GeometricData.z:=10;
neg_y_sort.GeometricData.radius:=0;
neg_y_sort.Offset.OffsetRelation:=FRAME_COORD_DIR;
neg_y_sort.Offset.OffsetPose.trans.x:=0;
neg_y_sort.Offset.OffsetPose.trans.y:=-500;
neg_y_sort.Offset.OffsetPose.trans.z:=0;
neg_y_sort.Offset.OffsetPose.rot.q1:=1;
neg_y_sort.Offset.OffsetPose.rot.q2:=0;
neg_y_sort.Offset.OffsetPose.rot.q3:=0;
neg_y_sort.Offset.OffsetPose.rot.q4:=0;
IF pick_type = 2 THEN pick_type := 1; ELSE
  pick_type := 2
ENDIF

GetItmTgt PickSource, PickItem \ItemType:=pick_type \Limit:=100
     \Selection:=neg_y_sort;
```

Retrieves a pick item from the *PickSource* with negative y-sorting and type request. The type is alternating between two types. The `Limit` argument tells from where to start the search.

In the example graphic below, the sorting is in positive x-direction, negative y-direction, and operating on two different object types. The two object types should

Continues on next page

be chosen in an alternating pattern starting with the circular. This will give the order as numbered 1-10 in the graphic.



xx0900000451

A	Enter
B	Check limit
C	Exit
D	Product flow direction
E	Sort direction
1-10	Sort order

Arguments

```
GetItmTgt ItemSource, ItemTarget [\MaxTime] [\TimeFlag] [\ItemType]
     [\Limit] [\SortData] [\Selection] [\Val1Min] [\Val1Max]
     [\Val2Min] [\Val2Max] [\Val3Min] [\Val3Max] [\Val4Min]
     [\Val4Max] [\Val5Min] [\Val5Max]
```

`ItemSource`

Data type: `itmsrc`

The item source from which the item target should be received.

`ItemTarget`

Data type: `itmtgt`

The received item target.

`[\MaxTime]`

Data type: `num`

The maximum waiting time permitted, expressed in seconds. If this time runs out before the item target is retrieved and no `TimeOut` flag is given, the error handler

Continues on next page

will be called with the error code ERR_PPA_TIMEOUT. If there is no error handler, the execution will be stopped.

[\TimeFlag]

Data type: bool

The output parameter that contains the value TRUE if the maximum permitted waiting time runs out before an item target is received. If this parameter is included in the instruction, it is not considered to be an error if the max time runs out. This argument is ignored if the MaxTime argument is not included in the instruction.

[\ItemType]

Data type: num

Specifies which item type number is requested. The instruction waits until an item target with the requested type number is available to be executed.

[\Limit]

Data type: num

Modifies the distance from where the item target is received. The instruction will return the next item target above this limit. If this argument is excluded, the instruction will return the next item target above the exit limit.

The distance is specified in millimeters from the center of the robot. The value is positive if the limit is beyond the center of the robot, in the moving direction of the feeder. This argument is only valid when a conveyor is used.

[\SortData]

Data type: sortdata

This data structure defines how the items shall be sorted.

[\Selection]

Data type: selectiondata

This data structure defines how the items are selected.

[\Val1Min]

Data type: num

Specifies minimum value for itmtgt parameter Val1. The instruction waits until an item target fulfilling this condition is available for execution.

[\Val1Max]

Data type: num

Specifies maximum value for itmtgt parameter Val1. The instruction waits until an item target fulfilling this condition is available for execution.

[\Val2Min]

Data type: num

Specifies minimum value for itmtgt parameter Val2. The instruction waits until an item target fulfilling this condition is available for execution.

[\Val2Max]

Data type: num

Continues on next page

Specifies maximum value for `itmtgt` parameter Val2. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val3Min]`

Data type: `num`

Specifies minimum value for `itmtgt` parameter Val3. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val3Max]`

Data type: `num`

Specifies maximum value for `itmtgt` parameter Val3. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val4Min]`

Data type: `num`

Specifies minimum value for `itmtgt` parameter Val4. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val4Max]`

Data type: `num`

Specifies maximum value for `itmtgt` parameter Val4. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val5Min]`

Data type: `num`

Specifies minimum value for `itmtgt` parameter Val5. The instruction waits until an item target fulfilling this condition is available for execution.

`[\Val5Max]`

Data type: `num`

Specifies maximum value for `itmtgt` parameter Val5. The instruction waits until an item target fulfilling this condition is available for execution.

Program execution

If there is no item target in buffer or any item targets available in the working area, the program execution waits in this instruction until an item is considered as inside the working area.

If the `MaxTime` argument is specified then the wait time is supervised. If the waiting time exceeds the value of `MaxTime` and the `TimeFlag` argument is used, then the program will continue. If `TimeFlag` is not used, then an error is raised. If `TimeFlag` is specified, it will be set to TRUE if the time is exceeded, otherwise it will be set to `FALSE`.

The `Limit` argument modifies the limit from where the item target shall be received.

If the `SortData` argument is specified the instruction will return the item target that is the closest to the exit limit in x-direction and depending of the absence of other objects in direction of the sorting, the first object in the sort direction will be selected. The `CheckBoundry` distance defines the required clearance distance

Continues on next page

around an object. The sorting will check both upwards and downwards the production flow for presence of other item targets. If this argument is combined with the `Limit` argument the sorting algorithm will also take all objects between the limit and the exit limit into consideration when checking the safety distance for the nearest objects. If more than one robot is used in a shared position source system, that is load balancing or `ATC`, we strongly recommend using the `Selection` argument instead with a proper selection data, as `SortData` does not take items that are bypassing in consideration when sorting.

If the `Selection` argument is specified, the instruction will return the item target that is the closest to the exit limit in x-direction, which has no other item targets inside the specified shape. If this argument is combined with the `Limit` argument the selection algorithm will also take all objects between the limit and the exit limit into consideration when checking the distance for the nearest objects. This is highly recommended to avoid collisions.

If values are specified for the optional arguments `Va1Xmin` or `Va1Xmax`, the instruction will return an item target that fulfills the required maximum and minimum values for `Va1X`.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable `ERRNO` will be set to:

Error code	Description
`ERR_ITMSRC_UNDEF`	`itmsrc` undefined.
`ERR_PPA_TIMEOUT`	Timeout without any error flag.

Syntax

```
GetItmTgt
  [ItemSource ':=' ] <variable (VAR) of itmsrc>,
  [ItemTarget ':=' ] <var or pers (INOUT) of itmtgt>
  [\MaxTime ':=' ] <expression (IN) of num>
  [\TimeFlag ':=' ] <var or pers (INOUT) of bool>
  [\ItemType ':=' ] <expression (IN) of num>
  [\Limit ':=' ] <expression (IN) of num>
  [\SortData ':=' ] <expression (IN) of sortdata>
  [\Selection ':=' ] <expression (IN) of selectiondata>
  [\Val1Min ':=' ] <expression (IN) of num>
  [\Val1Max ':=' ] <expression (IN) of num>
  [\Val2Min ':=' ] <expression (IN) of num>
  [\Val2Max ':=' ] <expression (IN) of num>
  [\Val3Min ':=' ] <expression (IN) of num>
  [\Val3Max ':=' ] <expression (IN) of num>
  [\Val4Min ':=' ] <expression (IN) of num>
  [\Val4Max ':=' ] <expression (IN) of num>
  [\Val5Min ':=' ] <expression (IN) of num>
  [\Val5Max ':=' ] <expression (IN) of num>;
```

Continues on next page

Related information

For information about	See
The data type `itmtgt`	*itmtgt - Item target data on page 151*.
The data type `selectiondata`	*selectiondata - Selection data on page 154*.
The data type `sortdata`	*sortdata - Sort data on page 157*.

7.1.4 NextItmTgtType - Get the type of the next item target

Usage

`NextItmTgtType` is used to get the type of the next item target (`itmtgt`) in the item source buffer. If the `Limit` distance parameter is given, the instruction will return the type of the next item target above the limit. The RAPID program waits in this instruction until there is an item in this queue.

Basic examples

```
NextItmTgtType PlaceSource, PlaceType
```

Retrieves the type of the next `itmtgt` in the *PlaceSource*.

Arguments

```
NextItmTgtType ItemSource ItemType [\Limit] [\MaxTime] [\TimeFlag]
```

`ItemSource`

Data type: `itmsrc`

The item source that the item target type should be retrieved from.

`ItemType`

Data type: `num`

The retrieved item target type.

`[\Limit]`

Data type: `num`

This is the limit from where the type is retrieved. The instruction will return the type of the next item target above this limit. If this argument is excluded, the instruction will return the type of the next item target above the exit limit.

The distance is calculated in millimeters from the center of the robot. The value is positive if the limit is beyond the center of the robot, in the moving direction of the conveyor.

This argument is only valid when a conveyor is used.

`[\MaxTime]`

Data type: `num`

The maximum waiting time permitted, expressed in seconds. If this time runs out before the item target is retrieved and no `TimeOut` flag is given, the error handler will be called with the error code `ERR_PPA_TIMEOUT`. If there is no error handler, the execution is stopped.

`[\TimeFlag]`

Data type: `bool`

The output parameter that contains the value `TRUE` if the maximum permitted waiting time runs out before an item target is retrieved. If this parameter is included in the instruction it is not considered to be an error if the max time runs out.

This argument is only used if the `MaxTime` argument is used.

Continues on next page

Program execution

If there is no item target in buffer or any item targets above the `Limit`, the program execution waits in this instruction until there is an item in the buffer.

If the `MaxTime` argument is specified then the wait time is supervised. If the waiting time exceeds the value of `MaxTime` and the `TimeFlag` argument is used, then the program will continue. If `TimeFlag` is not used, then an error is raised. If `TimeFlag` is specified, this will be set to TRUE if the time is exceeded, otherwise it will be set to FALSE.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable `ERRNO` will be set to:

Error code	Description
`ERR_ITMSRC_UNDEF`	`itmsrc` undefined.
`ERR_PPA_TIMEOUT`	Timeout without any error flag

Syntax

```
NextItmTgtType
  [ItemSource ':=' ] <variable (VAR) of itmsrc>,
  [ItemType ':=' ] <var or pers (INOUT) of num>
  [\Limit ':=' ] <expression (IN) of num>
  [\MaxTime ':=' ] <expression (IN) of num>
  [\TimeFlag ':=' ] <var or pers (INOUT) of bool>;
```

Related information

For information about	See
The data type `itmtgt`	*itmtgt - Item target data on page 151*.

7.1.5 QStartItmSrc - Start queue in item source

Usage

`QStartItmSrc` is used to start the queue in an item source. This instruction must be used when starting a new program or after flushing.

Basic example

```
QStartItmSrc PlaceSource;
```

The queue of objects in the item source *PlaceSource* is started.

Arguments

```
QStartItmSrc ItemSource
```

`ItemSource`

Data type: `itmsrc`

The started item source.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable `ERRNO` will be set to:

Error code	Description
`ERR_ITMSRC_UNDEF`	`itmsrc` undefined

Syntax

```
QStartItmSrc
  [ItemSource ':=' ] <variable (VAR) of itmsrc>;
```

Related information

For information about	See
The instruction `QStopItmSrc`	*QStopItmSrc - Stop queue in item source on page 145*.

7.1.6 QStopItmSrc - Stop queue in item source

Usage

`QStopItmSrc` is used to stop the queue in an item source.

Basic example

```
QStopItmSrc PlaceSource;
```

The queue of objects in the item source *PlaceSource* is stopped.

Arguments

```
QStopItmSrc ItemSource
```

`ItemSource`

Data type: `itmsrc`

The stopped item source.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable `ERRNO` will be set to:

Error code	Description
`ERR_ITMSRC_UNDEF`	`itmsrc` undefined

Syntax

```
QStopItmSrc
  [ItemSource ':=' ] <variable (VAR) of itmsrc>;
```

Related information

For information about	See
The instruction `QStartItmSrc`	*QStartItmSrc - Start queue in item source on page 144*.

7.1.7 ResetFlowCount - Reset flow counter

Usage

`ResetFlowCount` is used to reset the flow counter. The flow counter indicates the number of objects that has passed the exit limit of a conveyor work area since last reset. The value of the flow counter can be retreived with the function `GetFlowCount`

Basic example

```
ResetFlowCount PlaceSource;
```

Resets the flow counter for an item source.

Arguments

```
ResetFlowCount ItemSource
```

`ItemSource`

Data type: `itmsrc`

The item source.

Error handling

The following recoverable errors can be generated. The errors can be handled in an errorhandler. The system variable `ERRNO` will be set to:

Error code	Description
`ERR_ITMSRC_UNDEF`	`itmsrc` undefined

Syntax

```
ResetFlowCount[ItemSource ':=' ] <variable (VAR) of itmsrc>;
```

Related information

For information about	See
The function `GetFlowCount`	*GetFlowCount - Get number of passed items on page 150*.

7.2 Functions

7.2.1 GetQueueLevel - Get queue level

Usage

`GetQueueLevel` is used to get current number of item targets in an item source fulfilling certain conditions.

Basic example

```
reg1 := GetQueueLevel(PlaceSource);
```

`reg1` is assigned the current number of item targets in the item source `PlaceSource`.

Return value

Data type: `num`

The current number of item targets in the item source.

Arguments

```
GetQueueLevel (ItemSource [\ItmType] [\MinLimit] [\MaxLimit])
```

`ItemSource`

Data type: `itmsrc`

The item source that the current number of item targets should be retrieved from.

`\ItmType`

Data type: `num`

Only items of the specified type number will be counted.

`\MinLimit`

Data type: `num`

Defines the minimum distance to the robot center from where an item will be counted. A negative value indicates that the limit is upstreams from the robot center. A positive value indicates that the limit is downstreams. The parameters does not affect indexed work areas.

`\MaxLimit`

Data type: `num`

Defines the maximum distance to the robot center from where an item will be counted. A negative value indicates that the limit is upstreams from the robot center. A positive value indicates that the limit is downstreams. The parameter does not affect indexed work areas.

Continues on next page

7.2.1 GetQueueLevel - Get queue level
Continued

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable ERRNO will be set to:

Error code	Description
ERR_ITMSRC_UNDEF	itmsrc undefined

Syntax

```
GetQueueLevel '('
  [ItemSource ':=' ] <variable (VAR) of itmsrc> ')'
  [\ItmType ':=' ] <expression (IN) of num>
  [\MinLimit ':=' ] <expression (IN) of num>
  [\MaxLimit ':=' ] <expression (IN) of num>;
```

A function with a return value of the data type num.

7.2.2 GetQueueTopLevel - Get queue top level

Usage

`GetQueueTopLevel` is used to get the maximum number of item targets that simultaneously have been in the buffer of an item source.

Basic examples

```
reg1 := GetQueueTopLevel(PlaceSource);
```

reg1 is assigned the maximum number of item targets that simultaneously have been in the item source *PlaceSource*.

Return value

Data type: `num`

The maximum number of item targets that simultaneously have been in the item source.

Arguments

```
GetQueueTopLevel (ItemSource)
```

`ItemSource`

Data type: `itmsrc`

The item source that the current number of item targets should be retrieved from.

Error handling

The following recoverable errors can be generated. The errors can be handled in an error handler. The system variable `ERRNO` will be set to:

Error code	Description
`ERR_ITMSRC_UNDEF`	`itmsrc` undefined

Syntax

```
GetQueueTopLevel '('
  [ItemSource ':=' ] <variable (VAR) of itmsrc> ')';
```

A function with a return value of the data type `num`.

7.2.3 GetFlowCount - Get number of passed items

Usage

`GetFlowCount` is used to get the total number of items that has passed the exit limit of a conveyor work area since `ResetFlowCount` was executed. Items that the robot handles will not be counted (even if they pass the exit limit before picking/placing occurs).

Basic example

```
VAR num counter;
ResetFlowcount PlaceSource;
WaitTime 10;
counter := GetFlowCount(PlaceSource);
```

counter is assigned the number of items originating from `PlaceSource` that has passed the exit limit.

Return value

Data type: `num`

The number of items that has passed the exit limit since `ResetFlowCount` was executed.

Arguments

```
GetFlowCount (ItemSource)
```

`ItemSource`

Data type: `itmsrc`

The item souce.

Error handling

The following recoverable errors can be generated. The errors can be handled in an errorhandler. The system variable `ERRNO` will be set to:

Error code	Description
`ERR_ITMSRC_UNDEF`	`itmsrc` undefined

Syntax

```
GetFlowCount '('[ItemSource ':=' ] <variable (VAR) of itmsrc> ')';
```

A function returns value of the data type num.

Related information

For information about	See
The instruction `ResetFlowCount`	*ResetFlowCount - Reset flow counter on page 146*.

7.3 Data types

7.3.1 itmtgt - Item target data

Usage

`itmtgt` is used to describe one pick or place item.

Description

`Itmtgt` identifies an item to pick or place. It contains the position and some additional data.

Components

`tag`

Data type: `num`

Sequential number identifying the item. Can be modified by a user hook for free usage. Is restricted to integer values.

`type`

Data type: `num`

Type of item.

`scene`

Data type: `num`

Sequential number identifying the scene, corresponding for example to a picture taken by the vision system.

`robtgt`

Data type: `robtgt`

The pick or place position.

`val1`

Data type: `num`

Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

`val2`

Data type: `num`

Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

`val3`

Data type: `num`

Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

`val4`

Data type: `num`

Continues on next page

7.3.1 itmtgt - Item target data
Continued

Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

`val5`

Data type: `num`

Optional. Can be used to carry additional item specific information, for example, from a user hook. It is of data type float.

Examples

Example 1

```
CONST itmtgt pickpos :=
     [1,2,1,0,0,0,0,0,[[20,40,8],[1,0,0,0],[0,0,0,0],
     [9E+9,9E+9,9E+9,9E+9,0,0]]];
```

A pick position is defined. The external axis related to the used conveyors must be set to zero, that is not marked as unused (by stating 9E+9). Example: if you have two conveyors, set the two last external axis positions to zero.

Structure

```
<dataobject of itmtgt>
  <tag of num>
  <type of num>
  <scene of num>
  <val1 of num>
  <val2 of num>
  <val3 of num>
  <val4 of num>
  <val5 of num>
  <dataobject of robtarget>
    <trans of pos>
      <x of num>
      <y of num>
      <z of num>
    <rot of orient>
      <q1 of num>
      <q2 of num>
      <q3 of num>
      <q4 of num>
    <robconf of confdata>
      <cf1 of num>
      <cf4 of num>
      <cf6 of num>
      <cfx of num>
    <extax of extjoint>
      <eax_a of num>
      <eax_b of num>
      <eax_c of num>
      <eax_d of num>
      <eax_e of num>
      <eax_f of num>
```

Continues on next page

Related information

For information about	See
Positioning instructions.	*Technical reference manual - RAPID overview*.
Coordinate systems.	*Technical reference manual - RAPID overview*.
Handling configuration data	*Technical reference manual - RAPID overview*.
Configuration of external axes.	*Technical reference manual - System parameters*.
What is a quaternion?	*Technical reference manual - RAPID overview*.

7.3.2 selectiondata - Selection data

Usage

`selectiondata` is used to describe the selection criteria. It is also used to describe item sorting.

Description

`selectiondata` is used to set the criteria for sorting and clearance area when retrieving item targets from an item source.

Components

`ShapeType`

Data type: `shapetype`

Specifies the shape of the clearance area that should be used.

- `SHAPE_UNDEFINED` specifies that no selection is used.
- `BOX` specifies that there must be a clear box shape around the item target position where no other item targets are present.
- `CYLINDER` specifies there must be a clear cylinder shape around the item target position where no other item targets are present.
- `SPHERE` specifies that there must be a clear sphere shape around the item target position where no other item targets are present.

`ConsiderType`

Data type: `aconsidertype`

Specifies which items in the queue that should be taken in consideration when selecting.

- `ITEMS_TO_USE` specifies that only items marked for use by this queue are considered in the selection.
- `ITEMS_BYPASS` specifies that only items marked to pass by this queue are considered in the selection.
- `ITEMS_PICKED` specifies that only items marked as already picked, by this queue or by a former queue in the line, are considered in the selection.
- `ITEMS_PLACED` specifies that only items marked as already placed, by this queue or by a former queue in the line, are considered in the selection.

If items with different marks should be taken into consideration when selecting an item, then use a bit-or operation with the consideration types. (RAPID function `BitOr(<byte>,<byte>)`.)

`GeometricData`

Data type: `geodata`

The data that defines the geometric shape dimensions (x, y, z and radius).

- A `BOX` shape is defined by the x, y, and z-values.
- A `CYLINDER` shape is defined by the radius value and the height is defined by the z-value.

Continues on next page

- A SPHERE shape is defined by the radius value.

The orientation of the shape's coordinate system is defined by the offset data component. By default it is the coordinate system of the shape aligned to the workobject or conveyor frame. Note that all shapes origin are placed in the center of the shape and the values are the distance to every plane in both positive and negative direction. That is, if a box is defined as x: 10, y: 15 and z: 20 the box will have a size of 20 mm in x-direction, 30 mm in y-direction and 40 in z-direction. If no offset is used the check for other items in range will be done 10 mm before, 10 mm after, 15 mm left of, 15 mm right of, 20 mm above, and 20 mm underneath every item.

`Offset`

Data type: `offsetdata`

The offset consists of `OffsetRelation` (`offsetreltype`) and `OffsetPose` (`pose`).

The `OffsetRelation` can be of two different types.

- `FRAME_COORD_DIR` indicates that the rotation in the `OffsetPose` is relative to the workobject or conveyor frame coordinate system.
- `ITEM_COORD_DIR` indicates that the rotation in the `OffsetPose` is relative to the item coordinate system of the item to check.

The `OffsetPose` is used to move the center of the shape away from the item position, for example, if the grip position of the item is not at the center of real object to pick.

Examples

```
VAR selectiondata clear_rect:= [BOX,ITEMS_TO_USE,[22,15,5,0],
     [FRAME_COORD_DIR,[[0,7,0],[1,0,0,0]]]];
```

Limitations

The orientation must be normalized; that is the sum of the squares must equal 1.

$q1^2 + q2^2 + q3^2 + q4^2 = 1$

Structure

```
<dataobject of selectiondata>
  <ShapeType of shapetype>
  <ConsiderType of considertype>
  <GeometricData of geodata>
    <x of num>
    <y of num>
    <z of num>
    <radius of num>
  <Offset of offsetdata>
    <OffsetRelation of offsetreltype>
    <OffsetPose of pose>
      <trans of pos>
        <x of num>
        <y of num>
        <z of num>
```

Continues on next page

7.3.2 selectiondata - Selection data
Continued

```
<rot of orient>
  <q1 of num>
  <q2 of num>
  <q3 of num>
  <q4 of num>
```

Related information

For information about	See
The data type `pose`	*Technical reference manual - RAPID Instructions, Functions and Data types*.
The function `BitOr`	*Technical reference manual - RAPID Instructions, Functions and Data types*.
What is a quaternion?	*Technical reference manual - RAPID overview*.
Example using `selectiondata`	*Example: Selecting item depending on clearance zone on page 125*.

7.3.3 sortdata - Sort data

Usage

`sortdata` is used to describe the sorting criteria.

Description

`sortdata` is used to set the criteria for sorting item targets from an item source.

Components

`SortType`

Data type: `sorttype`

Type of sorting that is going to be used.

- `UNSORT_TYPE` tells that no sorting is used.
- `POS_Y_SORT_TYPE` tells that the sorting shall be done from the positive y-direction of the work area.
- `NEG_Y_SORT_TYPE` tells that the sorting shall be done from the negative y-direction of the work area.

`CheckBoundary`

Data type: `num`

The clearance distance for sorting, in millimeters. The distance is defined as the minimum distance to the next item in the sorting direction.

`SortDirOffset`

Data type: `num`

An offset distance beyond the item target in the sort direction. Is used to define the inner limit for the corridor in which no other item targets are allowed.

Examples

```
VAR sortdata y_sort:=[NEG_Y_SORT_TYPE ,78, 52];
```

Structure

```
<dataobject of sortdata>
  <SortType of sorttype>
  <CheckBoundary of num>
  <SortDirOffset of num>
```

This page is intentionally left blank

Index

A
AckItmTgt, instruction, 133
ATC, 13, 68

C
container, 12, 48
container pattern, 12, 54
conveyor start/stop signal, 36
conveyor work area, 12, 35

D
data types
  itmtgt, 151
  selectiondata, 154
  sortdata, 157
distribution, 12, 65
  settings, 65

F
flow, 12
flow handler, 12, 43
FlushItmSrc, instruction, 135
functions
  GetFlowCount ¶, 150
  GetQueueLevel, 147
  GetQueueTopLevel, 149

G
GetItmTgt, instruction, 136
GetQueueLevel, function, 147
GetQueueTopLevel, function, 149

H
hotspot, 64, 96

I
indexed work area, 12, 37
instructions
  AckItmTgt, 133
  FlushItmSrc, 135
  GetItmTgt, 136
  NextItmTgtType, 142
  QStartItmSrc, 144
  QStopItmSrc, 145
  ResetFlowCount ¶, 146
item, 12, 48
itmtgt, data type, 151

J
job, 12, 59

L
load balancing, 12, 67

N
NextItmTgtType, instruction, 142
noncnvwobjdata, 109

P
position available signal, 36
position generator signal, 45, 47

Q
QStartItmSrc, instruction, 144
QStopItmSrc, instruction, 145
queue idle signal, 36

S
safety, 9
selectiondata, data type, 154
sensor, 12, 39
signals
  conveyor start/stop, 36
  position available, 36
  position generator, 45, 47
  queue idle, 36
  strobe, 47
  trig, 45, 47
sortdata, data type, 157
sourcedata, 108
strobe signal, 47

T
trigger signal, 45, 47

W
work area, 12, 34

Contact us

ABB AB
Discrete Automation and Motion
Robotics
S-721 68 VÄSTERÅS, Sweden
Telephone +46 (0) 21 344 400

ABB AS, Robotics
Discrete Automation and Motion
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 51489000

ABB Engineering (Shanghai) Ltd.
No. 4528 Kangxin Highway
PuDong District
SHANGHAI 201319, China
Telephone: +86 21 6105 6666

www.abb.com/robotics

3HAC048042-001, Rev F, en