



ROBOTICS

Operating manual

Machining Software



Trace back information:
Workspace Main version a572
Checked in 2024-04-15
Skribenta version 5.5.019

Operating manual

Machining Software

1.4.0

Document ID: 3HAC080110-001

Revision: E

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, damage 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.

This manual and parts thereof must not be reproduced or copied without ABB's written permission.

Keep for future reference.

Additional copies of this manual may be obtained from ABB.

Original instructions.

© Copyright 2021-2024 ABB. All rights reserved.
Specifications subject to change without notice.

Table of contents

Overview of the manual	7
Network security	9
Safety	10
1 Overview	11
2 Getting started	15
3 Creating, loading and exporting projects	19
4 Creating and tuning programs	21
4.1 Overview	21
4.2 Defining path	23
4.3 Defining instructions	29
5 Auto-calibrating	33
5.1 Overview	33
5.2 Calibrating toolkits	37
5.3 Calibrating cutters	42
5.4 Calibrating external axes and work objects	47
6 RAPID programming	53
6.1 RAPID exporting and reloading	54
6.2 RAPID instructions	59
6.2.1 MachL - Moves linearly	59
6.2.2 MachJ - Moves using joint movement	61
6.2.3 MachC - Moves circularly	63
6.2.4 WaveLStartPoint - Moves to the wave start point	65
6.2.5 WaveL - Moves along the wave path	66
6.3 RAPID data types	68
6.3.1 MachiningPose - Coordinate transformations in machining programs	68
6.3.2 MachineProcess - Process definitions in machining programs	69
6.3.3 WaveDirection - Wave path processing directions	70
6.3.4 WaveDistribution - Distribution information of wave nodes	71
6.3.5 WaveNode - wave nodes defined for wave paths	72
6.3.6 WaveShape - wave patterns defined for wave paths	73
7 Reference information	75
7.1 Messages and errors	75
7.2 Troubleshooting	78
7.3 Drawings	79
Index	81

This page is intentionally left blank

Overview of the manual

About this manual

This manual describes how to use Machining Software to work with machining applications on the web user interface using PC or in the Machining application using FlexPendant.

Usage

This manual should be used during the work with Machining Software.

Who should read this manual?

This manual is intended for application engineers, commissioning engineers and operators.

Prerequisites

The reader should have basic knowledge of:

- Industrial robots and their terminology
- RAPID programming language
- RobotStudio
- Machining PowerPac add-in

References

Reference	Document ID
<i>Operating manual - IRC5 Integrator's guide</i>	3HAC050940-001
<i>Operating manual - Integrator's guide OmniCore</i>	3HAC065037-001
<i>Operating manual - Emergency safety information</i>	3HAC027098-001
<i>Safety manual for robot - Manipulator and IRC5 or OmniCore controllerⁱ</i>	3HAC031045-001
<i>Operating manual - Machining PowerPac - Machining Functionality</i>	3HAC054781-001
<i>Technical reference manual - RAPID overview</i>	3HAC050947-001
<i>Technical reference manual - RAPID Instructions, Functions and Data types</i>	3HAC050917-001
<i>Technical reference manual - System parameters</i>	3HAC050948-001
<i>Operating manual - RobotStudio</i>	3HAC032104-001
<i>Application manual - Force control with software and hardware</i>	3HAC070206-001

ⁱ This manual contains all safety instructions from the product manuals for the manipulators and the controllers.

Revision

Revision	Description
A	First edition

Continues on next page

Revision	Description
B	<p>Published with version 1.1. The following updates are done in this revision:</p> <ul style="list-style-type: none">• Renamed the product from Machining Shopfloor HMI to Machining Software.• Changed descriptions of option Shopfloor and Shopfloor Advanced to Machining Standard and Machining Premium, respectively.• Supported working with OmniCore controller.• Supported working with projects using the MSF application on FlexPendant.• Added calibration procedure for crossbeam-type toolkit.• Updated calibration procedure for cutters, separating cutters into sander type and machining tool type.• Added the wave path settings and related RAPID information.• Added the multiple instruction exporting function.• Added new error information.
C	<p>Published with version 1.2.0. The following updates are done in this revision:</p> <ul style="list-style-type: none">• Added calibration toolkit crossbeam and probe setup information.• Updated calibration procedure of probe-type toolkit.
D	<p>Published with version 1.3.0. The following updates are done in this revision:</p> <ul style="list-style-type: none">• Updated safety information.• Added information about how to create new programs in Machining Software.• Added the force control settings.• Added a note for working with Machining application on FlexPendant.• Removed information about exporting programs to local file.• Updated programming rules for modules CalibData_HMI and Main_test_ProgramID.
E	<p>Published with version 1.4.0. The following updates are done in this revision:</p> <ul style="list-style-type: none">• Added a new calibration method <i>Polyhedron</i> for work object calibration.• Added a note to the required RobotWare, RobotStudio and Machining PowerPac - Machining Functionality versions for using the latest Machining Software.• Added the drawing of polyhedron block as reference information.• Added a troubleshooting for the reported error 40661 during calibration using a probe.

Network security

Network security

This product is designed to be connected to and to communicate information and data via a network interface. It is your sole responsibility to provide, and continuously ensure, a secure connection between the product and to your network or any other network (as the case may be).

You shall establish and maintain any appropriate measures (such as, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of anti-virus programs, etc) to protect the product, the network, its system and the interface against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information. ABB Ltd and its entities are not liable for damage and/or loss related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

Safety

General

Machining Software is a product used for machining applications and shall be used for setting, teaching, programming and operation phases during the lifecycle.

Machining Software is not a safety related part in a robot system. Using Machining Software cannot reduce the safety risk level of the robot system. The integrator of the robot system is required to perform an assessment of the hazards and risks, and make sure using Machining Software will not increase the safety risk level of the robot system.

Machining Software shall be used with robot systems that are designed and installed in accordance with applicable standards.

Safety of personnel

When a robot is in automatic mode, the robot control system operates in accordance with the robot program. The robot program can have a pause, long stop in movement followed by a start of a movement. The robot program can also do changes in operation, path and speed, of the manipulator through external signal(s). The pattern of the behavior of a manipulator can therefore not be safely predicted without the knowledge of the robot program. A deviation from an unsafe prediction will be observed as an unexpected movement.

Therefore, a safety guard device (such as a safety fence) must be active in automatic mode; and the operator must work outside the safety fence.

If the robot is operating in Auto mode, the users shall be close to a control panel when using Machining Software from a browser so that the control panel can be used to stop the robot system timely when needed.

It is the integrator's responsible for the safety of the robot system, and for that the safety devices necessary to protect people working with the robot system are designed and installed correctly.

Related information

Before beginning work with the robot, make sure you are familiar with all the general safety aspects and specific safety information described in this manual as well as safety manuals for manipulators and controllers.

You can find safety descriptions in following but not limited to following manuals:

- *Operating manual - Emergency safety information* (3HAC027098-001)
- *Operating manual - General safety information* (3HAC031045-001)
- Safety chapter in product manuals for manipulators and controllers

1 Overview

Introduction

Machining Software (previous known as Machining Shopfloor HMI) is a RobotWare option that allows users to work with machining projects on site using PC or FlexPendant.

With Machining Software, users can define machining processes by creating programs using web browser on PC or in the dedicated Machining application on FlexPendant. It is also possible to synchronize or load programs created in Machining PowerPac - Machining Functionality (hereinafter referred as Machining PowerPac) add-in in RobotStudio, and then tuning the programs in Machining Software. Different from Machining PowerPac that provides offline programming functions, Machining Software can load the created or tuned programs to the connected controller (virtual or real) directly. This improves programming efficiency and reduce the onsite commissioning time.

Machining Software also provides the auto-calibration function that allows users to define calibration toolkits (eg. probe), cutters, external axis and work objects. The calibrated data can be copied to and reused in RAPID for other projects, which simplifies the calibration process.



Note

The Machining application on FlexPendant is only supported by OmniCore controllers operating in RobotWare 7.X.

Key features

Machining Software provides the following main features:

- Web-based access to machining projects (using PC)
- Dedicated Machining application for working with machining projects (using FlexPendant)
- Creating programs to define machining processes
- Tuning programs created by Machining PowerPac
- Auto-calibration on calibration toolkit, cutters, external axis and work objects

Versions

Machining Software provides two versions, Machining Standard (option 877-2 for IRC5 and 3418-1 for OmniCore) and Machining Premium (option 877-3 for IRC5 and 3418-2 for OmniCore), with different user access to functions. The following table lists the main functions to which the two options can access.

Function		Standard	Premium
File operations	File loading	X	X
	File export	X	X

Continues on next page

1 Overview

Continued

Function		Standard	Premium
Program operations	Program creating		X
	Program synchronization and loading		X
	Program tuning		X
Auto-calibration	Toolkit calibration	X	X
	Cutter calibration	X	X
	Work object calibration	X	X



Note

Only one Machining Software version can be installed on a controller at a time. For the web-based Machining Software, if the version is changed, clean the browser cache and restart the browser to make the new version take effect. Otherwise, a version incompatibility message will be displayed.

Prerequisites

Hardware and software

Requiring...	Note ⁱ
PC	Note Required only for web-based Machining Software using PC. <ul style="list-style-type: none">• OS: Windows 10, 64-bit• CPU: 4x2.0Hz or faster processors• RAM: 8GB at minimum• Resolution: 1024 x 768 Pixels• An account with administrator's privileges
Browser	Note Required only for web-based Machining Software using PC. Google Chrome and Microsoft Edge are recommended.
Controller	<ul style="list-style-type: none">• IRC5 controller in RobotWare 6.15.04 or later• OmniCore controller in RobotWare 7.14.2or later
FlexPendant ⁱⁱ	FlexPendant must be connected and available to use.
RobotStudio	RobotStudio 2022 or later
Machining PowerPac - Machining Functionality	Machining PowerPac 2022.1 or later Available in https://new.abb.com/products/robotics/robotstudio/downloads .

ⁱ Listed versions of RobotWare, RobotStudio and Machining PowerPac- Machining Functionality are the minimum versions required for the latest Machining Software version.

ⁱⁱ If the Machining application runs slowly when FlexPendant is used to work with machining projects, it is recommended to use the web-based Machining Software on PC or use a FlexPendant with a larger RAM.

Continues on next page

License

Machining Software requires a license to activate. Contact ABB to order the option Machining Standard or Machining Premium and obtain a license.

**Note**

If the option Machining Premium is selected, option Multitasking (option 623-1 for IRC5 and 3114-1 for OmniCore) will be selected together.

User grants

Machining Software requires a user account to log in, which must have been granted access permission to the connected controller and corresponding grants to specific functions. Refer to the following table for the required grants.

Functions	Required grant... ⁱ
Access to the main page of Machining Software	Read access to controller disks
New file, load file and save file	Write access to controller disks
Synchronize programs created by Machining Power-Pac to Machining Software	I/O write access
Click Apply button to apply changes	Load program
Request write access	Program debug
Access to calibration functions	Calibration
Access to program functions	Edit RAPID code
Click Calibration button to start calibration	<ul style="list-style-type: none"> • Load program • Program debug • Execute program
Selection of mechanical unit (Mechanism-type work object)	Required only for RobotWare 6.X <ul style="list-style-type: none"> • Remote restart • Modify configuration

ⁱ For RobotWare 6.X, controller restart is required after grants are modified. For RobotWare 7.X, grant modification takes effect in real time.

If the Default User is active, users can also use the user name **Default User** and its password **robotics** to log in as the default user.

For details about user account management, see information about User Authentication System (UAS) in the operation manual of the controller.

This page is intentionally left blank

2 Getting started

Installing

Generally, Machining Software is integrated in the controller system at delivery. It is also available as an add-in RobotStudio. To add it to an existing controller or do an update, use the following procedure to install:

- 1 Start RobotStudio and open the **Add-Ins** tab. The **Gallery** window is displayed.
- 2 In the displayed **Gallery** window, use the **Search** function or **Common tags** to find the Machining Software add-in.
- 3 Click the displayed add-in icon.
- 4 In the right pane, click **Add**.
The package is automatically installed and listed in the **Add-in** navigation tree in the left pane of the window.
- 5 In the **Controller** tab page, click the **Modify Installation** button in the **Configuration** group.
- 6 In the **Modify Installation** dialogue, connect to a real controller or select/create a virtual controller.
- 7 Follow the instructions in the installation wizard and add the Machining Software add-in to the controller.



Note

You have to choose the Machining Software version in the **Machining** list according to the license you obtained; otherwise, a warning message is displayed.

Only one Machining Software version is allowed to be installed on the controller at a time. To change the version, deselect the original version and select the new one.

The Machining Software add-in is displayed in the controller overview if it is successfully added to the controller.

Logging in

You can log in Machining Software in one of the following ways:

- Use web browser in PC
- Use the Machining application in FlexPendant

The modifications are automatically synchronized between the web-based Machining Software and Machining application in FlexPendant, which are opened for the same robot system. To view the latest modifications made in the other way, refresh the web page or reopen the setting window in the application.

Continues on next page

2 Getting started

Continued

Detailed logging in procedures are as follows.

Using PC



Note

Before working with web-based Machining Software, make sure the PC opening the Machining Software web page has connected to the required controller and works in the same local subnet of the controller.

- 1 Start the browser.

Google Chrome and **Microsoft Edge** are recommended.

- 2 Enter the IP address of the connected controller in the Address bar.

	Virtual controller	Real controller
RobotWare 6.X	http://127.0.0.1/docs/MSF.html	http://Controller IP address/docs/MSF.html
RobotWare 7.X	https://127.0.0.1:80/docs/MSF.html	https://Controller IP address/docs/MSF.html

- 3 In the displayed window, enter the user name and password.

Make sure the account has the access permission to the connected controller.
You can also enter as the Default User.

Using FlexPendant

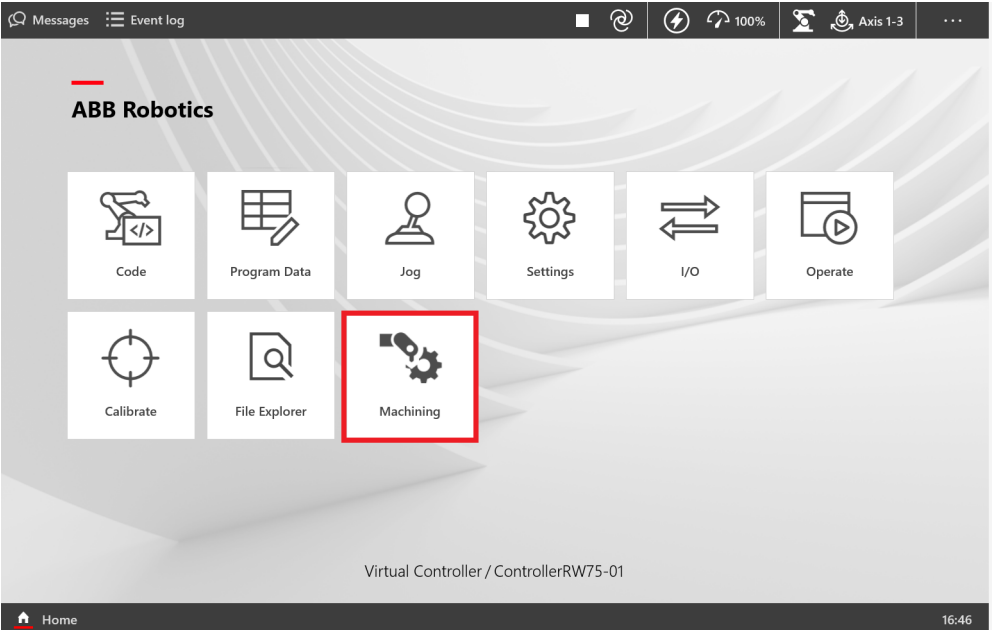


Note

The Machining application on FlexPendant is only supported by OmniCore controllers operating in RobotWare 7.X.

Continues on next page

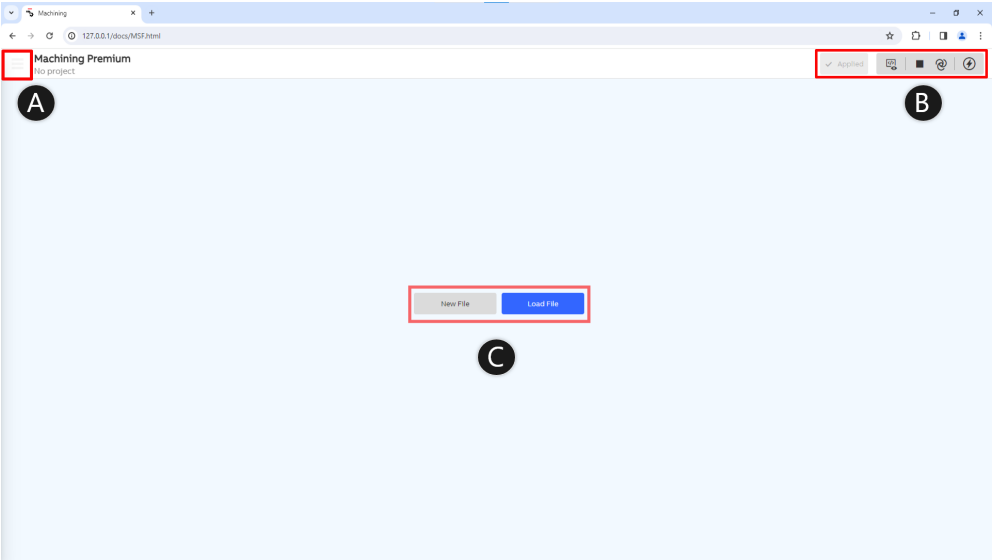
You can access the main window of Machining Software directly by tapping the following application icon on the FlexPendant touch screen.




xx2200000576

The user interface

Home page





xx2100002395

	Item	Description
A	Menu	<div>Displays the menu when clicking the hamburger button.</div> <div><div> Note</div><div>The menu is available to use only when a project has been created or loaded.</div></div>

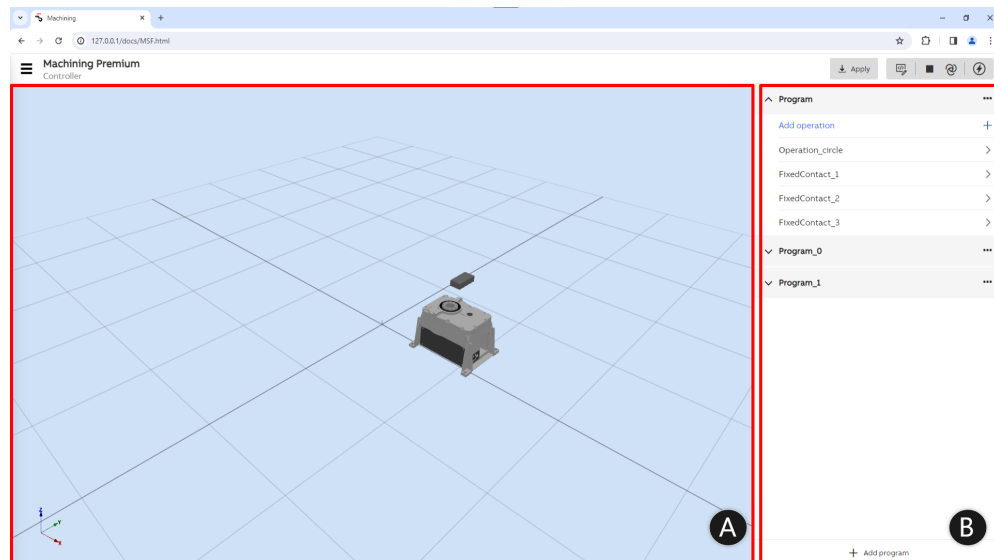
Continues on next page

2 Getting started

Continued

	Item	Description
B	Status bar	<p>Shows the controller status.</p> <p>You can click the icons to display the status details, including controller name, controller status, operating mode and motor status.</p> <p>In the status details page, you can click Request Write Access to get permission of program editing. Otherwise, the program settings can only be viewed. After the write access is granted, an Apply button is displayed next to the status bar.</p> <p> Note</p> <p>The status bar is displayed only for web-based Machining Software using PC. In the Machining application on FlexPendant, only the Apply button is available.</p>
C	Start buttons	<p>Starts working with Machining Software by clicking New File or Load File.</p> <ul style="list-style-type: none"> New File: a new project is created and auto-calibration function is available. Load File: an existing project can be loaded from the controller or a local folder. <p> Note</p> <p>If there is an unsaved project existing, a message window is displayed before you can click the start buttons, urging you to save or discard the previous editing.</p>

Main working page



xx2100002396

	Item	Description
A	3D graphics window	<p>Shows the 3D graphics of the station and provides an intuitive and realtime preview of the editings.</p> <p>You can also drag the mouse to control the graphics view.</p>
B	Setting window	Displays the detailed setting pages.

3 Creating, loading and exporting projects

About Machining Software project

Machining Software project document is a .mpstn file that contains data of a machining process. It stores information about operations, toolkits, cutters, work objects and cell layout. A project can contain all or part of the information.

Users can create a new project or work on an existing project in Machining Software. The generated or modified projects can be exported to local folder or the connected controller.

Creating a new project

The following table lists the ways to create a new project in Machining Software.

Create project from...	Description
Home page	In the home page, click New File .
Menu	<ol style="list-style-type: none"> 1 Click the hamburger button on the upper-left corner of the window to display the menu. 2 Choose File > New File. <p>A message is displayed prompting you to save any unsaved data.</p>

Loading an existing project


The following table lists the ways to load an existing project to Machining Software.

Load project by...	Description
Synchronizing from the Machining PowerPac add-in	<p>After programs are created in the Machining PowerPac add-in in RobotStudio:</p> <ol style="list-style-type: none"> 1 Click Sync in the Shopfloor Tool group. 2 Check the information shown in the displayed window and click OK. 3 Choose the controller to which Machining Software is connected and click OK. <p>The programs are automatically synchronized to Machining Software.</p> <ol style="list-style-type: none"> 4 Click Accept or Decline in the Machining Software.
Loading from a local file	<ol style="list-style-type: none"> 1 In the home page, click Load File and then choose Load file, or in the main working page, click the menu bar and then choose File > Load file. 2 In the displayed window, choose the required program file in .mpstn format from a local folder. 3 Click Open.
Loading from the controller	<ol style="list-style-type: none"> 1 In the home page, click Load File and then choose Load file from controller, or in the main working page, click the menu bar and then choose File > Load file from controller. 2 In the displayed window, choose the required program file in .mpstn format. <p>Only the files saved in the HOME folder of the controller system are displayed.</p> <ol style="list-style-type: none"> 3 Click Load.

Continues on next page

Exporting a project

The following table lists the ways to export a project.

Export project to...	Description
Local folder	<div> Note</div> <p>Exporting project to local is valid only for web-based Machining Software using PC.</p> <ol style="list-style-type: none">1 In the main working page, click the menu bar and then choose File > Save file.2 In the displayed window, choose the items to be downloaded. By default, all the items are selected. You can also select or remove a category in the left pane, or choose or remove part of items in a specified category.3 Click Save. A .mpstn project file is download to the specified local folder.
Controller	<ol style="list-style-type: none">1 In the main working page, click the menu bar and then choose File > Save file to controller.2 In the Contents area in the displayed window, choose the items to be downloaded. By default, all the items are selected and shown by their categories in the Contents area. You can click the drop-down arrow, and in the displayed window, select or remove a category in the left pane, or choose or remove part of items in a specified category.3 Click OK.4 Click Browse to select a folder in controller system for download. The HOME folder is selected by default.5 Enter a name in the File Name text box.6 Click Save.

4 Creating and tuning programs

4.1 Overview

Introduction

Machining processes are defined by programs and each program is a group of operations with detailed path and instruction settings. Programs can be created directly in Machining Software, synchronized from Machining PowerPac or loaded from a Machining Software project file.

All the available programs are listed in the program setting window. You can click the unfold button next to a program to expand or collapse the list of operations that are included in the program.



Note

Only the Machining Software in advanced version (option Machining Premium) supports the program creating, synchronization, loading and tuning.



Note

For web-based Machining Software, before working with the programs, make sure the writing access to the controller has been requested and granted. To request write access, click the icon on the status bar and then click **Request Write Access** in the status details page.

Starting with programs and operations

If a project has available programs, the programs are listed in the program setting window. You can click the required program and operation, and then start tuning it.

To create new programs and operations, perform following procedure:

- 1 Click the menu bar in the main working page and then click **Program**.
This step is required only when the program setting window is not displayed.
- 2 In the program setting window in the right pane, click **Add program**.
- 3 In the displayed dialog box, specify the program name and then click **OK**.
The program is listed in the setting window. You can also rename or remove the program by clicking the hamburger icon for the program and then choosing **Rename** or **Delete** respectively.
- 4 Click the unfold button next to the required program to expand the program.
- 5 Click **Add operation** to add a new operation for the selected program.
After adding, the operation setting window is directly displayed for the operation.
- 6 In the displayed window, further define paths and instructions.
Default operation name is OprX. You can rename the operation in the **Name** text box, or click **Delete** to remove the operation from the program.

Continues on next page

4 Creating and tuning programs

4.1 Overview

Continued

Detailed procedures to define paths and instructions are specified in following sections.

4.2 Defining path

Procedure

Use the following procedure to define path properties:

- 1 Expand the required program and click the required operation.
The operation setting window is displayed.
- 2 In the **Name** text box on the **Path** tab page, keep the operation name or rename the operation.
The name is allowed to be specified with a maximum of 10 characters.
- 3 Specify the path type, which can be set to **Machining**, **Move** and **Force Control**.
This is a global setting for all the instructions associated with the operation.
- 4 If path type is set to **Force Control**, click the unfold button next to **Force Control Settings** to display a new window and set force control parameters.
See details in [Setting force control parameters on page 23](#).
- 5 Select the cutter to be used for processing and the work object to be processed from the **Tool** and **Wobj** list, respectively.
- 6 For existing operations that already have target positions defined, perform path smoothing or wave path setting if required.
See details in [Smoothing path on page 24](#) and [Setting wave path on page 25](#).
- 7 Click **Apply** in the status bar to apply the modifications to the connected controller.

Setting force control parameters

If the path type is set to **Force Control** for an operation, you can click **Force Control Settings** to enter a new window to set the force control parameters. Following table lists the available parameters.

Parameter	Description
FCStart	
Force reference frame	Defines which coordinate system the force control coordinate system is related to. The parameter can be set to either the work object coordinate system, the tool coordinate system or the path coordinate system. The default value is the tool coordinate system.
Force X(N)	Defines the constant reference force in the x direction of the force control coordinate system. If this parameter is omitted there will be zero contact force in the x direction. Default value is 0.
Force Y(N)	Defines the constant reference force in the y direction of the force control coordinate system. If this parameter is omitted there will be zero contact force in the y direction. Default value is 0.
Force Z(N)	Defines the constant reference force in the z direction of the force control coordinate system. If this parameter is omitted there will be zero contact force in the z direction. Default value is -10.

Continues on next page

4 Creating and tuning programs

4.2 Defining path

Continued

Parameter	Description
Speed FFW	If this parameter is enabled then feed forward regulation is used. If this parameter is disabled then regulation in force direction is done only with force control without help from programmed path. Enable this parameter if path is complex and programmed path is close to actual path. This parameter is enabled by default.
Threshold(%)	Percentage of the contact force that should be reached before move toward the robtarget starts. TCP moves in force direction until this percentage is reached. When percentage of force is reached, the movement toward target starts. Default value is 50.
Force Change Rate(N/s)	Tuning parameter to ramp up force. This parameter overrides configured value. Default value is 50.
Damping(%)	The relation value between the measured force and the applied resulting force. By default the value are 100% (of system parameter values), but it can be between 50% and infinity. Smaller values than 100% means that the robot is more sensitive to external force.
Timeout(s)	If force hasn't build up before this time is reached then continue with next instruction. Default value is 5.
Supervision Distance	The robot will stop if it has moved more than the distance Distance away from the programmed path. Default value for Distance is 20mm. This parameter is disabled by default.
FCEnd	
Force Change Rate(N/s)	Tuning parameter to ramp up force. Default value is 50.
Zero Contact Force(N)	Specifies a force limit and if the force is less than the specified value, then force control is deactivated. Default value is 50.

The parameters in **FCStart** and **FCEnd** group are associated with the RAPID instructions **FCPress1LStart** and **FCPressEnd**, respectively. See details in *Application manual - Force control with software and hardware*.

Smoothing path

Use the following procedure to smooth target positions in a path or path segment:

- 1 Expand the required program and click the required operation.
On the **Path** tab page of the displayed window, information of the tool and work object associated to the selected operation can be viewed.
- 2 Click **Smooth**.
The **Smooth** window is displayed.
- 3 Select a method from the **Smooth Method** list.
 - **Position:** the path will be smoothed based on the X/Y/Z coordinates of the targets.
 - **Orientation:** the path will be smoothed based on the Rx/Ry/Rz coordinates of the targets.

Continues on next page

- **Process:** the path will be smoothed based on the axis settings of the tool.
- 4 Select targets from the **Segment** list for defining a path or path segment.
By default, all the targets in the whole path are selected. You can also select several targets to define a path segment. The number of selected targets can be viewed.
In the **Segment** window,
 - Click two targets as the first and last targets of the path/path segment. All the targets in between will be selected.
 - Click **Select All** to re-select all the targets. This button is available only when part of targets are selected.
 - Click **OK** to accept the selection or click **Cancel** to discard the selection.
 - 5 Click the **Local** or **Wobj** tab to choose the reference coordinate system of the targets, on which the path smoothing is based.
 - 6 Modify target positions to smooth path.
 - Only the positions of fixed targets can be modified. The first and last targets in a path/path segment are fixed targets by default. For other targets, turn the **Fixed Target** switch on to change the target characteristics.
 - You can drag the target to a required position in 3D graphics window or type a value in the corresponding coordinates text box for a specific target.
 - If **Position** is selected in the **Smooth Method** list, the **X(mm)**, **Y(mm)** and **Z(mm)** parameters referring to the variations on the X, Y and Z axis can be set.
 - If **Orientation** is selected in the **Smooth Method** list, the **Rx(deg)**, **Ry(deg)** and **Rz(deg)** parameters referring to the variations on the orientation rotating around the X, Y and Z axis can be set.
 - If **Process** is selected in the **Smooth Method** list, the **Indentation(mm)**, **Tilt Angle(deg)** and **Lead Angle(deg)** parameters referring to the tool position based on the TCP can be set.
 - 7 Click **Save** to accept the settings.
You can also click **Reset** to cancel the settings.
 - 8 Click **Apply** in the status bar to apply the modifications to the connected controller.

Setting wave path

What is wave path?

Wave path settings are used to generate new paths based on the original machining paths in operations generated by Machining PowerPac. Enabling wave path for a specified machining path will wiggle the tool from side to side in the defined pattern when the tool moves along the path. With wave path, better processing effects could be achieved because it could provide sufficient processing on contact areas. It especially benefits to deburring and grinding applications.

Continues on next page

4 Creating and tuning programs

4.2 Defining path

Continued

Wave path is available to use when:

- Target processing moves forward along X+ or Y+ directions of local coordinate system of targets.

The processing direction can be set in target configuration for an operation in Machining PowerPac.

- Distances between the targets in the original machining paths are even.

To obtain proper target distance, it is recommended to set interpolation method to **MaxLengthOnly** when setting process move parameters for operations in Machining PowerPac.

- Processing method is using tool tip.

In Machining PowerPac, the machining method for a tool shall be set to **Face Machining**.

Procedure

Use the following procedure to enable and set wave path for the path in a specified operation:

- 1 Expand the required program and click the required operation.
- 2 In the displayed new window, click **Wave Path**.
- 3 In the displayed **Wave Path** window, turn on the switch to enable wave path. Detailed settings are displayed.

- 4 Choose the machining direction, **Tool X+** or **Tool Y+**.

The tool will process along the selected direction based on the local coordinate system of the tool.

- 5 Choose a wave pattern or create a new pattern.

- To use a predefined pattern or an existing pattern, select the desired pattern from the list to apply the pattern. No further actions required.



Note

Three predefined patterns are available, square, circle and diamond. The patterns are allowed to be duplicated or edited by clicking **Edit** on the right corner of the window first and then clicking corresponding button.

For detailed procedure of editing a pattern, proceed to [6](#). The edited pattern will be applied to all the operations whose paths have enabled wave path with this pattern.

- To create a new pattern, click **New Pattern** and proceed to [6](#).

- 6 In the **Pattern Name** area of the **Edit Pattern** or **New Pattern** window, keep the automatically generated name or type another name as required.

- 7 Set the step for the pattern to loop.

The distance between two adjacent targets in the original machining path is considered as one step. The pattern loops in a distance based on the defined step. Default step is 3.

- 8 Set the basic source targets that could be used for setting the pattern.

Continues on next page

Source targets are the ones in the original path. They will be used as reference to generate wave nodes that will be the new targets in wave path for the tool to process. Default target number is 5.

- 9 Select the motion type that the tool moves from wave node to wave node.
 - **Linear:** the tool moves using MoveL, that is, in linear movement.
 - **Circular:** the tool moves along the path using MoveC, that is, in circular movement.
- 10 In the pattern preview window:
 - a Click a source target in blue to generate one or more wave nodes. Every wave node is shown with a number. The node numbers also represent the processing sequence, that is, the tool will move in number order.
 - b Click on the generated wave nodes and set the pattern with them.
 - The tool processes along the wave path in the wave node number order. The order could be changed by modifying the node number in the **Number** text box of the quickset dialog box. The pattern changes with the order change.
 - The wave node position can be modified by dragging it to the desired position or, in the quickset dialog box, setting the offsets from the targets where the wave node generated in both the machining direction and the direction perpendicular to the machining direction.
 - A single wave node can be deleted by clicking **Delete Node** in the quickset dialog box. It is also possible to clear all the wave nodes by clicking **Delete All Nodes** on the right upper corner of the preview window.

The defined pattern is displayed as a blue path in real time, and other looped patterns are displayed in grey.



Note

The final wave path depends on both the motion type and the way how the pattern is defined. If a pattern contains N wave nodes and,

- the motion type is **Linear**, positions of all the N wave nodes will be repeated in following looped patterns. That is, all the patterns contain N wave nodes.
- the motion type is **Circular**, the first wave node in defined pattern works as the wave start point but it is not repeated in the following looped patterns. That is, the following patterns will contain only $N-1$ wave nodes (wave node in number 1 is excluded).

Continues on next page

4 Creating and tuning programs

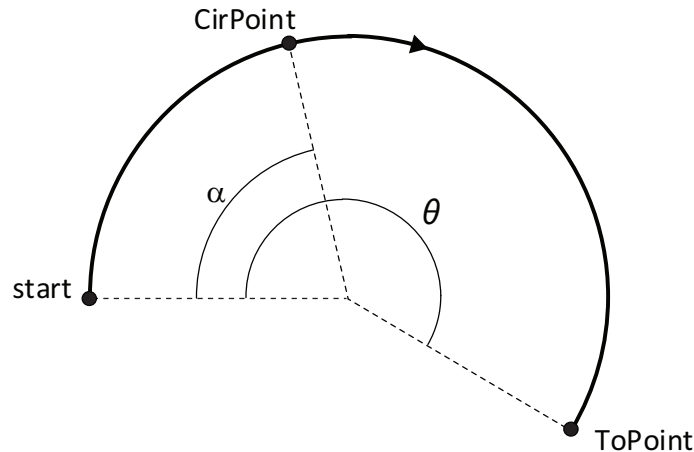
4.2 Defining path

Continued



Note

For a circular pattern, it requires a minimum of 3 wave nodes and the total node number should be $3+2N$ (N represents 0, 1, 2...). The wave node in number $3+2N$ and $3+2(N+1)$ function as the StartPoint and ToPoint in a circular movement, and the wave node between them will be the CirPoint, as shown in the following figure.



xx1700001575

There are some limitations in how the CirPoint and the ToPoint can be placed.

- Minimum distance between start and ToPoint is 0.1 mm
- Minimum distance between start and CirPoint is 0.1 mm
- Minimum distance between CirPoint and ToPoint is 0.1 mm
- If the system parameter *Restrict placing of circle points* is set to Yes, then the following additional limitations are active:
 - The angle of the circular path (θ in the picture above) may not be larger than 240° .
 - The circle point must be in the middle part of the circular path (α must be 25-75% of θ , according to the picture above).

For more details, refer to the description of MoveC in *Technical reference manual - RAPID Instructions, Functions and Data types*.

11 Click **Save** to return to the Wave Path window.



Note

Wave path settings could also be modified in RAPID programs. See [RAPID instructions on page 59](#) for more information.

4.3 Defining instructions

Procedure

An instruction includes information such as movement mode and movement speed on a specific target. Use the following procedure to define instructions:

- 1 Expand the required program and click the required operation.
- 2 In the displayed operation setting window, click the **Instruction** tab.

All the instructions associated with the selected operation are listed. If there is no available instruction, click **Add instruction** to add new instructions.

- 3 Click the unfold button next to the instructions to be edited.



Tip

You can click the **Filter** button to filter instructions by types, approach, process or depart.

- To edit a specific instruction, click it to enter the setting window.
- To edit a group of instructions as a whole, click **Select** in the right corner of the window to select more than one instructions and then click **Edit** at the bottom of the window to display the setting window.



Note

Instructions could also be selected and exported as arrays by clicking the **Export** icon first and then choosing to save to local or controller.

Note that exporting to local is only available for web-based Machining Software using PC.



Note

Available parameters for setting a specific instruction and setting a group of instructions are similar. But you can only set position, start event and end event for a specific instruction. Therefore, the following steps take the setting of a specific instruction as an example.

- 4 If the path type is set to **Force Control**, click the unfold button next to **Type**, and in the opened page, select a process type, **FCStart**, **FCVia**, **FCEnd** or **Move**.
- 5 Select the motion type, **Linear**, **Joint** or **Circular**.



Note

The selection of global path type, process type (only for *Force Control* path type) and motion type selected for the associated operation will define the actual instruction that will be applied in the RAPID programs and used in the process. See [Reference information on page 31](#).

Continues on next page

4 Creating and tuning programs

4.3 Defining instructions

Continued

- 6 Select a movement speed (speed data) from the **Speed** list.
- 7 Select the position termination mode (zone data) from the **Zone** list.
- 8 If the path type is set to **Force Control**, specify the force size in the direction defined in the coordinate system chosen in the **Force Control Settings** window.

The **Force** value is set to 10 by default.

- 9 Click the unfold button next to **TargetX**.
- 10 In the displayed **TargetX** window:
 - a Click the **Local**, **Wobj** or **Process** tab to choose the reference coordinate system on which the modification is based.
 - b Drag the coordinate axis to a required position in 3D graphics window or type a value in the corresponding coordinates text box.
 - In the **Local** and **Wobj** tab page,
 - the **X(mm)**, **Y(mm)** and **Z(mm)** parameters referring to the variations on the X, Y and Z axis can be set;
 - the **Rx(deg)**, **Ry(deg)** and **Rz(deg)** parameters referring to the variations on the orientation rotating around the X, Y and Z axis can be set.
 - In the **Process** tab page that is displayed for the **Machining** path type, the **Indentation(mm)**, **Tilt Angle(deg)** and **Lead Angle(deg)** parameters referring to the tool position based on the TCP can be set.
 - c Set external axis positions, if external axes exist.

If the value **9E+09** is displayed, it refers that no external axis is connected and the positions cannot be set.
 - d Click **Save** to accept the settings.

- 11 If there are events defined and bound to the specified instruction, click the unfold button next to **Start Event** or **End Event** to set the events.

Only bound events are displayed and available for setting. You can delete, add or change event order for the instruction.

- 12 Click the unfold button next to **Optional Arguments**, and in the displayed window, set values for the arguments.
 - **Offset**: used to add a displacement in the work object coordinate system to a robot position. The value is defined as [x,y,z] in unit of mm.
 - **RelEuler**: used to add a displacement and/or a rotation in the active tool object coordinate system to a robot position. The value is defined as [x,y,z,Rx,Ry,Rz] in unit of mm.
 - **ID**: specifies the synchronization id and is mandatory in the MultiMove system if the movement is synchronized or coordinated synchronized. By using the id number the movements are not mixed up at the runtime.
 - **T**: used to specify the total time in seconds during which the robot moves.

Continues on next page

The arguments will be used in the RAPID instruction. See [RAPID instructions on page 59](#) for more information.

- 13 Click **Apply** in the status bar to apply the modifications to the connected controller.

Reference information

For path type *Move* and *Machining*, the actual instructions that will be applied in RAPID program depend on the selection of motion type, that is MoveJ/MoveL/MoveC and MachJ/MachL/MachC.

For path type *Force Control*, the actual instructions that will be applied in RAPID program depend on the selection of both process type and motion type. The following table lists the actual instructions will be used in different combinations.

For details about the instruction descriptions, see *Technical reference manual - RAPID Instructions, Functions and Data types*, [RAPID instructions on page 59](#) and *Application manual - Force control with software and hardware*.

		Motion type		
		Joint	Linear	Circular
Process type	FCStart	-	FCPress1LStart	-
	FCVia	-	FCPressL	FCPressC
	FCEnd	-	FCPressEnd	-
	Move	MoveJ	MoveL	MoveC

It allows to change the path type for an operation. After changing, all the instructions associated with this operation will change to the corresponding ones automatically, depending on the selection of motion type.

If changing the path type from *Move* or *Machining* to *Force control* with process type *FCStart/FCVia/FCEnd*, the instructions will change to *FCPress1LStart*, *FCPressL* and *FCPressEnd*. If the process type is *FCVia*, you can further change the motion type from *Linear* to *Circular*, and then the instruction will change to *FCPressC*.

This page is intentionally left blank

5 Auto-calibrating

5.1 Overview

Introduction

If a Machining Software project file contains the information of toolkits, cutters and work objects, users can define and calibrate their data with the auto-calibration function. For a new project, it also allows users to define new toolkits and work objects.

To calibrate cutters and work objects, a toolkit must be available to use and calibrated first. The following table lists the required toolkit for each cutter and workobject type.

Type		Required toolkit
Cutter	Orbit sander	Probe
	Random orbital sander	Probe
	Machining tool	Crossbeam
Work object	External axis (Mechanism)	Probe
	Wobjdata (work object)	Probe

Every time the calibration is successfully performed, the calibration data is recorded. A maximum of 10 latest records will be listed in the calibration history and available for reuse. You can view the calibration history in the setting windows by clicking the **History** icon on the right upper corner of the windows.



Note

For web-based Machining Software, before performing calibration, make sure the writing access to the controller has been requested and granted. To request write access, click the icon on the status bar and then click **Request Write Access** in the status details page.

Toolkit setup

Crossbeam

Crossbeam is an optical sensor that radiates two perpendicular laser beams (X-direction and Y-direction) within its inner for TCP measurement. By moving the tool on a circular or square path within the crossbeam and interrupting the laser beams, the TCP could be observed and then referenced for tool calibration. It is also possible to determine the center point of the crossbeam where both laser beams cross. At this center point, it can be checked whether the calibration was correct since both laser beams are interrupted at the same time.

To calibrate cutters using a crossbeam in Machining Software, the crossbeam must be available to use and calibrated first. Detailed procedures about how to set up and configure a crossbeam, see the user manual from the vendor. Following

Continues on next page

5 Auto-calibrating

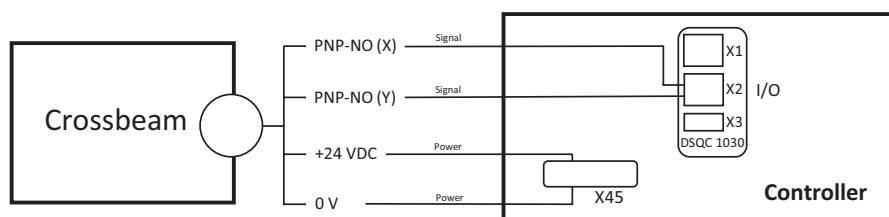
5.1 Overview

Continued

lists the necessary preparations of using the crossbeam with the Machining Software:

- The crossbeam can have a circular inner, which is also recommended, or a square inner. To calibrate the crossbeam with the Machining Software, the **Inner Radius** parameter must be properly set according to the crossbeam inner shape. For circular inner, it is set to the circle radius; for square inner, it is set to the short diameter.
- The crossbeam shall connect to a controller with an I/O device, which could provide 2 pins for power connection and 2 pins for signal connection.

The following figure illustrates a typical connection between the crossbeam and OmniCore with the digital base device DSQC1030 supporting the communication over Ethernet/IP.



xx2300001215



Note

If your controller is an IRC controller, the XS16 connector shall be used for power connection. If your controller is installed with a DSQC652, the X3 connector shall be used for signal connection.

For connections with other controllers or I/O boards, contact your local ABB Robotics Service representative, see <http://www.abb.com/>.

- Signals for identifying the laser beams in X direction and Y direction are configured using RobotStudio.
 - 1 In the **Controller** ribbon tab in the RobotStudio, click **Configuration** and then click **I/O system**.
 - 2 In the **Type** pane of the **Configuration - I/O system** window, right-click **Signal** and choose **New Signal**.
 - 3 In the displayed **Instance Editor** dialog box,
 - specify a name to the x-direction or y-direction laser beam
 - set **Type of Signal** to **Digital Input**
 - select the connected IO board in the **Assigned to Device** drop-down list
 - set **Default Value** to **0**
 - remain default settings for other parameters

Continues on next page

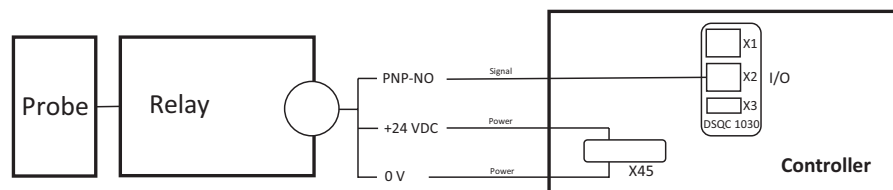
Probe

Probe is a displacement sensor that commonly consists of a sensor and a tooling ball (or known as needle). The tooling ball can "feel" the contact surfaces and, by measuring the displacement, feed back the position of the approached target.

To calibrate cutters or work objects using a probe in Machining Software, the probe must be available to use and calibrated first. Detailed procedures about how to set up and configure a probe, see the user manual from the vendor. Following lists the necessary preparations of using the probe with the Machining Software:

- The tooling ball locates at the end of the probe and can be in any size. To calibrate the probe with the Machining Software, the **Ruby Ball Radius** parameter must be properly set according to the actual tooling ball radius.
- The probe shall connect to a relay first and then to a controller with an I/O device, which could provide 2 pins for power connection and 1 pin for signal connection..

The following figure illustrates a typical connection between the probe and OmniCore with the digital base device DSQC1030 supporting the communication over Ethernet/IP.



xx2300001217

**Note**

If your controller is an IRC controller, the XS16 connector shall be used for power connection. If your controller is installed with a DSQC652, the X3 connector shall be used for signal connection.

For connections with other controllers or I/O boards, contact your local ABB Robotics Service representative, see <http://www.abb.com/>.

- Signal for identifying the tooling ball is configured using RobotStudio.
 - 1 In the **Controller** ribbon tab in the RobotStudio, click **Configuration** and then click **I/O system**.
 - 2 In the **Type** pane of the **Configuration - I/O system** window, right-click **Signal** and choose **New Signal**.
 - 3 In the displayed **Instance Editor** dialog box,
 - specify a name to the tooling ball of the probe
 - set **Type of Signal** to **Digital Input**
 - select the connected IO board in the **Assigned to Device** drop-down list
 - set **Default Value** to **0**

Continues on next page

5 Auto-calibrating

5.1 Overview

Continued

- remain default settings for other parameters

5.2 Calibrating toolkits

Preparation procedure for new toolkit

For a new toolkit, perform the following preparation actions:

- 1 Click the hamburger button on the upper-left corner of the window to display the menu.
- 2 Click **Calibration**.
- 3 In the displayed **Toolkit** setting window on the right, click **New** at the bottom of the window.
- 4 Choose the required tool type, **Probe** or **CrossBeam**.

Preparation procedure for existing toolkit

For an existing toolkit, perform the following preparation actions:

- 1 Click the hamburger button on the upper-left corner of the window to display the menu.
- 2 Click **Calibration**.
- 3 In the displayed **Toolkit** setting window on the right, click the **More options** icon next to the required toolkit.
- 4 Click an icon according to your requirement.
 - **Edit**: to edit basic information of the toolkit, such as name and trigger signal.
 - **Define**: to define the toolkit calibration settings and perform calibration.



Note

The **Define** item is available only when a trigger signal has been set for the toolkit.

- **Delete**: to delete the toolkit.

Calibration procedure - toolkit (probe type)

Use the following procedure to define and calibrate a probe:

- 1 In the **Probe Name** area of the **Probe** setting window, keep the automatically generated name or type another name as required.
- 2 Turn on/off the **Robot Hold** switch based on where the probe is installed.
 - If the probe is hold by the robot, turn on the switch, that is, set it to **True**.
 - If the probe is stationary or hold by another robot, turn off the switch, that is, set it to **False**.
- 3 In the **tooldata** area, check the current probe position and calibration status.
- 4 Choose a signal from the **Trigger Signal** drop-down list.
After a signal is chosen, the **Define** button at the bottom of the page is available.
- 5 Set **Trigger Value** to 0 or 1.

Continues on next page

5 Auto-calibrating

5.2 Calibrating toolkits

Continued

The trigger value represents the returned value when the probe contacts the calibrating object.

- 6 Set **Ruby Ball Radius (mm)** to the actual radius of the tooling ball at the end of the probe.
- 7 Click **Define** to enter the calibration process.
The **Define ProbeName** window is displayed.
- 8 Click **Ball** to enter the setting window for defining the calibrating ball that is the calibration target for the probe to touch.

In the displayed **Ball** window,

- Set **Ball Radius (mm)** to the actual value.
The ball is automatically named with its radius.
- Set **Accessible Offset (mm)**, which defines a safe area for the probe to touch the ball.

This parameter defines a height from the top point of the ball. The safe area refers to the ball surface covered by the projecting plane of the height. You can also drag the arrow in the 3D graphics window to set the safe area and the setting is displayed in real time.

The **Accessible Offset (mm)** value must be equal to or smaller than the **Ball Radius (mm)** value.

Click **Save** to return.

- 9 Click **Search Settings** to enter the setting window for defining how the probe goes to search.

In the displayed **Search Settings** window,

- Set **Search Mode** and **Accuracy Level**.
Five accuracy levels are predefined, with level 4 set by default.
The higher the level is selected, the more the touchpoints are defined and the higher the calibration accuracy is, but the slower the calibration process is. For example, in default level 4, nine touchpoints are defined and the calibration is relatively more accurate, but it will take estimate 240 s for the calibration.
- Click the **Advanced** arrow to expand the settings and set **Search Offset** and **Approach Distance**.



Tip

You can click the **Information** icon at the upper-right corner of the **Search Settings** window to view the parameter definitions.

Continues on next page

- 10 Jog the probe to touch the ball at the **Contact**, **Home** and other four **Pose** positions, and click **Update Position** to record corresponding values.

**Note**

The **Contact** position is the one where the probe touches the calibrating ball, and must be updated before the probe is jogged to the **Home** position. When jogging the probe to the four **Pose** positions, make sure the tooling ball of the probe only rotates around the **Home** position without any displacement.

- 11 Click **Calibrate** and then click **Play**.
- If the controller is in Auto mode, the calibration process runs automatically.
 - If the controller is in Manual mode, follow the instructions in the displayed messages to perform the calibration.
- 12 Click **Save** to save the data after the calibration done.

Calibration procedure - toolkit (crossbeam type)
**Note**

Before calibration, make sure a valid cutter has been defined. See [Calibration procedure - cutters \(machining tool type\) on page 45](#).

Use the following procedure to define and calibrate a crossbeam:

- 1 In the **Name** area of the **CrossBeam** setting window, keep the automatically generated name or type another name as required.
- 2 Turn on/off the **Robot Hold** switch based on where the crossbeam is installed.
 - If the crossbeam is hold by the robot, turn on the switch, that is, set it to **True**.
 - If the crossbeam is stationary or hold by another robot, turn off the switch, that is, set it to **False**.
- 3 In the **Wobjdata** area, check the crossbeam position and calibration status. By default, the wobjdata **wobj0** is displayed.
 - If the crossbeam is installed with X and Y lasers whose directions close to the X and Y directions of the coordination system for wobj0, remain the wobjdata.
 - If the crossbeam is defined with other wobjdata, click **Copy wobjdata from controller** and select valid wobjdata for the crossbeam.
 - If X and Y lasers of the crossbeam are not in the same X and Y directions of the coordination system for wobj0 and no wobjdata is valid for the crossbeam, define the crossbeam using the **User defined with 3 points** method in FlexPendant.
- 4 Choose signals for both X and Y lasers from the **Trigger Signal** drop-down lists.

Continues on next page

5 Auto-calibrating

5.2 Calibrating toolkits

Continued

The **Define** button at the bottom of the page is available only after the signals are selected.



Note

For crossbeam, the trigger value is always be 1, which represents the returned value when the corresponding laser is interrupted by the calibrating objects.

- 5 Set **Inner Radius (mm)** to the actual inner radius of the crossbeam.
- 6 Click **Define** to enter the calibration process.
The **Define CrossbeamName** window is displayed.
- 7 Select a cutter from the **Cutter** drop-down list.
Only cutters in machining tool type and in opposite robot-hold position to the crossbeam are available to choose.
- 8 Click **Search Settings** to enter the setting window for defining how to search with the crossbeam.

In the displayed **Search Settings** window,

- Set **Search Direction** and **Search Speed**.
- Click the **Advanced** arrow to expand the settings and set **Search Depth (mm)** and **Search Radius (mm)**.



Note

The larger the search radius is, the higher the search precision is and the less the possibility of reporting search error messages is. However, a larger search radius may also cause collisions. Always set the search radius properly.



Tip

You can click the **Information** icon at the upper-right corner of the **Search Settings** window to view the parameter definitions.

- 9 Jog the cutter to approach the inner circle of the crossbeam at the **Home**, **ContactXY** and other four **Pose** positions, and click **Update Position** to record corresponding values.



Note

The positions are related to both the workobject coordinate system of the crossbeam and the tool coordinate system of the selected cutter.

Continues on next page



Note

The **ContactXY** position is the one where the cutter reaches the crosspoint of the X and Y lasers, that is, the center point of the crossbeam inner circle. When jogging the cutter to the four **Pose** positions, make sure the cutter only rotates around the center point of the crossbeam inner circle.

10 Click **Calibrate** and then click **Play**.

- If the controller is in Auto mode, the calibration process runs automatically.
- If the controller is in Manual mode, follow the instructions in the displayed messages to perform the calibration.

11 Click **Save** to save the data after the calibration done.

5.3 Calibrating cutters

Preparation procedure for new cutters

For a new cutter, perform the following preparation actions:

- 1 Click the hamburger button on the upper-left corner of the window to display the menu.
- 2 Click **Calibration**.
- 3 In the displayed **Cutter** setting window on the right, click **New** at the bottom of the window.

Preparation procedure for existing cutters

For an existing cutter, perform the following preparation actions:

- 1 Click the hamburger button on the upper-left corner of the window to display the menu.
- 2 Click **Calibration**.
- 3 In the displayed **Cutter** setting window on the right, click the **More options** icon next to the required cutter.
- 4 Click an icon according to your requirement.
 - **Edit**: to edit basic information of the cutter, such as name and tool type.
 - **Define**: to define the cutter calibration settings and perform calibration.
 - **Delete**: to delete the cutter.

Calibration procedure - cutters (sander type)

Setting basic cutter information

Use the following procedure to set basic information of the cutter:

- 1 In the **Name** area of the **Cutter** setting window, keep the automatically generated name or type another name as required.
- 2 Turn on/off the **Robot Hold** switch based on where the cutter is installed.
 - If the cutter is hold by the robot, turn on the switch, that is, set it to **True**.
 - If the cutter is stationary or hold by another robot, turn off the switch, that is, set it to **False**.
- 3 In the **tooldata** area, check the current cutter position and calibration status. By default, the tooldata **tool0** is displayed.
 - If the TCP of the cutter is at position of tool0, remain the tooldata.
 - If the cutter is defined with other tooldata, click **Copy tooldata from controller** and select valid tooldata for the cutter.
 - If the TCP of the cutter is not at position of tool0 and no tooldata is valid for the cutter, create tooldata for the cutter in FlexPendant.
- 4 Choose the sander-type cutter from the **Tool Type** list.
 - **Orbital Sander**: choose this type if the sanding paper is square.
 - **Random Orbital Sander**: choose this type if the sanding paper is circle.

Continues on next page

- 5 Click **Define** to enter the calibration process.

The **Define CutterName** window is displayed.

Defining and calibrating the cutter

In the **Define CutterName** window, use the following procedure to define and calibrate the sander:

- 1 Click **Calibration Method** to select the required method.
 - **DOF3**: calibration will be performed on the surface of sanding paper on the sander.
 - **DOF6**: calibration will be performed on the sander.

Click **Save** to return.

- 2 Click **Shape** to enter the page for setting the shape of the sanding paper or sander based on the previous selection of the sander type and calibration method.

Cutter type	Calibration method	Shape type	Parameter description
Orbital sander	DOF3	Default Pad	<ul style="list-style-type: none"> • Length/Width/Height Defines the shape of square sanding paper. • Rotation Defines a degree based on which the XY plane of the sanding paper or sander rotates around the Z direction.
		CAD	<ul style="list-style-type: none"> • Load Sander Model Click to upload a 3D model from local PC. Make sure the Z direction of the model is the same as the polishing direction. • Rotation Defines a degree based on which the XY plane of the sanding paper or sander rotates around the Z direction.
	DOF6	CAD	<ul style="list-style-type: none"> • Load Sander Model Click to upload a 3D model from local PC. Make sure the Z direction of the model is the same as the polishing direction. • Rotation Defines a degree based on which the XY plane of the sanding paper or sander rotates around the Z direction.

Continues on next page

5 Auto-calibrating

5.3 Calibrating cutters

Continued

Cutter type	Calibration method	Shape type	Parameter description
Random or-orbital sander	DOF3	Default Pad	<ul style="list-style-type: none">• Radius Defines the shape of circle sanding paper.• Rotation Defines a degree based on which the XY plane of the sanding paper or sander rotates around the Z direction.
	DOF6	CAD	<ul style="list-style-type: none">• Load Sander Model Click to upload a 3D model from local PC. Make sure the Z direction of the model is the same as the polishing direction.• Rotation Defines a degree based on which the XY plane of the sanding paper or sander rotates around the Z direction.



Note

To ensure the calibration validity, the defined shape shall not be larger than the actual shape of the sanding paper or sander.

Click **Save** to return.

- 3 Click **Touchpoints** to add the touchpoints used for calibration.
 - If the calibration method is **DOF3**, add touchpoints directly by clicking on the surface of the model shown in the 3D view.
 - If the calibration method is **DOF6**, choose a plane first and then add touchpoints on the selected plane.



Note

If there are steps on a plane, adding touchpoints in different steps may affect the final calibration result slightly.

Effective Touchpoints in Total shows the quantity of effective touchpoints. You could also remove the selection by clicking **Clear Points**.

After selection, click **Save** to return.

- 4 Choose a calibrated probe from the **Probe** list.
- 5 Click **Search Settings** to enter the setting window for defining how the probe goes to search.

In the displayed **Search Settings** window,

- Set **Search Mode** and **Search Speed**.
- Click the **Advanced** arrow to expand the settings and set **Search Offset** and **Approach Distance**.

Continues on next page

**Tip**

You can click the **Information** icon at the upper-right corner of the **Search Settings** window to view the parameter definitions.

- 6 Jog the probe to touch the sanding paper surface or sander at the required position and click **Update Position** to record the value.

- If the calibration method is DOF3, the **Home** position is required.
- If the calibration method is DOF6, the **Home**, **Home_X+**, **Home_Y+**, **Home_X-** and **Home_Y-** positions are required.

**Note**

The positions are related to both the tool coordinate system of the selected probe and the tool coordinate system of the sander itself. At the same time, the sander tooldata will be referred as wobjdata.

- 7 Click **Calibrate** and then click **Play**.

- If the controller is in Auto mode, the calibration process runs automatically.
- If the controller is in Manual mode, follow the instructions in the displayed messages to perform the calibration.

- 8 Click **Save** to save the data after the calibration done.

Calibration procedure - cutters (machining tool type)
Setting basic cutter information

Use the following procedure to set basic information of the cutter:

- 1 In the **Name** area of the **Cutter** setting window, keep the automatically generated name or type another name as required.
- 2 Turn on/off the **Robot Hold** switch based on where the cutter is installed.
 - If the cutter is hold by the robot, turn on the switch, that is, set it to **True**.
 - If the cutter is stationary or hold by another robot, turn off the switch, that is, set it to **False**.
- 3 In the **tooldata** area, check the current cutter position and calibration status. By default, the tooldata **tool0** is displayed.
 - If the TCP of the cutter is at position of tool0, remain the tooldata.
 - If the cutter is defined with other tooldata, click **Copy tooldata from controller** and select valid tooldata for the cutter.
 - If the TCP of the cutter is not at position of tool0 and no tooldata is valid for the cutter, use the tooldata tool0.
- 4 Choose the machining tool-type cutter from the **Tool Type** list.
- 5 Click **Define** to enter the calibration process.
The **Define CutterName** window is displayed.

Continues on next page

5 Auto-calibrating

5.3 Calibrating cutters

Continued

Defining and calibrating the cutter

In the **Define CutterName** window, use the following procedure to define and calibrate the cutter:

- 1 Choose a calibrated crossbeam from the **CrossBeam** list.
- 2 Click **Search Settings** to enter the setting window for defining how the probe goes to search.

In the displayed **Search Settings** window,

- Set **Search Mode** and **Search Speed**.
- Click the **Advanced** arrow to expand the settings and set **Search Offset** and **Approach Distance**.



Tip

You can click the **Information** icon at the upper-right corner of the **Search Settings** window to view the parameter definitions.

- 3 Jog the cutter to approach the inner circle of the crossbeam at the **Home** and **ContactXY** positions, and click **Update Position** to record corresponding values.



Note

The positions are related to both the workobject coordinate system of the selected crossbeam and the tool coordinate system of the cutter itself.



Note

The **ContactXY** position is the one where the cutter reaches the crosspoint of the X and Y lasers, that is, the center point of the crossbeam inner circle.

- 4 Click **Calibrate** and then click **Play**.
 - If the controller is in Auto mode, the calibration process runs automatically.
 - If the controller is in Manual mode, follow the instructions in the displayed messages to perform the calibration.
- 5 Click **Save** to save the data after the calibration done.

5.4 Calibrating external axes and work objects

Preparation procedure for new external axis/work object

For a new external axis/work object, perform the following preparation actions:

- 1 Click the hamburger button on the upper-left corner of the window to display the menu.
- 2 Click **Calibration**.
- 3 In the displayed **Wobj** setting window on the right, click **New** at the bottom of the window.
- 4 Click the **Mechanism** or **Wobj** icon as required.

Preparation procedure for existing external axis/work object

For an existing toolkit, perform the following preparation actions:

- 1 Click the hamburger button on the upper-left corner of the window to display the menu.
- 2 Click **Calibration**.
- 3 In the displayed **Wobj** setting window on the right, click the **More options** icon next to the required external axis/work object.
- 4 Click an icon according to your requirement.
 - **Edit**: to edit basic information of the external axis/work object, such as name.
 - **Define**: to define the external axis/work object calibration settings and perform calibration.
 - **Delete**: to delete the external axis/work object

Calibration procedure - external axis

Use the following procedure to define and calibrate an external axis:

- 1 In the **Name** area of the **Mechanism** setting window, keep the automatically generated name or type another name as required.
- 2 Select an external axis from the **Mechanism Unit** list.

A message will be displayed if the base frame of the selected external axis is different in the Machining Software and in the controller, prompting you to load data from controller or overwrite the data in the controller.

 - If loading from the controller, the axis information and base frame of the external axis are displayed automatically.
 - If overwriting, you will be promoted to restart the controller. The data takes effect only after the controller is restarted. After the controller restarts, you can view the updated axis information in the Machining Software by reselecting the external axis.
- 3 Click **Define** to enter the calibration process.

The **Define ExternalAixsName** window is displayed.
- 4 Choose a calibrated probe from the **Probe** list.

Continues on next page

5 Auto-calibrating

5.4 Calibrating external axes and work objects

Continued

- 5 Click **Ball** to enter the setting window for defining the calibrating ball that is placed on the external axis and the calibration target for the probe to touch.

In the displayed **Ball** window,

- Set **Ball Radius (mm)** to the actual value.
The ball is automatically named with its radius.
- Set **Accessible Offset (mm)**, which defines a safe area for the probe to touch the ball.
This parameter defines a height from the top point of the ball. The safe area refers to the ball surface covered by the projecting plane of the height.
The **Accessible Offset (mm)** value must be equal to or smaller than the **Ball Radius (mm)** value.
- Move the **Number of touchpoints** slider to set the quantity of points on the ball for the probe to touch.

Click **OK** to return.

- 6 Click **Search Direction** to select a direction for the probe to touch the calibrating ball.

The direction could be based on the probe coordinate system (**Tool** tab) or work object coordinate system (**Wobj** tab).

- 7 Click **Search Settings** to enter the setting window for defining how the probe goes to search.

In the displayed **Search Settings** window,

- Set **Search Mode** and **Search Speed**.
- Click the **Advanced** arrow to expand the settings and set **Search Offset** and **Approach Distance**.



Tip

You can click the **Information** icon at the upper-right corner of the **Search Settings** window to view the parameter definitions.

- 8 Jog the probe to touch the ball at **AngleX+**, **Angle2**, **Angle3** and **Angle4** positions, and click **Update Position** to record corresponding values.

More positions can be added by clicking the **+** button.



Note

The positions are related to both the workobject coordinate system of the wobj0 (robot base frame) and the tool coordinate system of the selected probe.



Note

Before jogging the probe to the **AngleX+** position, make sure that the external axis is in 0° position.

Continues on next page

- 9 Click **Calibrate** and then click **Play**.
- If the controller is in Auto mode, the calibration process runs automatically.
 - If the controller is in Manual mode, follow the instructions in the displayed messages to perform the calibration.
- 10 Click **Save** to save the data after the calibration done.
- A message displays, promoting you to restart the controller. The calibration data takes effect only after the controller restarts.

Calibration procedure - work object

Use the following procedure to define and calibrate a work object:

- 1 In the **Wobj Name** area of the **Wobj** setting window, keep the automatically generated name or type another name as required.
- 2 Turn on/off the **Robot Hold** switch based on where the work object is installed.
 - If the work object is hold by the robot, turn on the switch, that is, set it to **True**.
 - If the work object is stationary, or hold by another robot or external axis, turn off the switch, that is, set it to **False**.
- 3 Click **External Axis** to enable or disable the external axis. If enabled, select the valid external axis from the list.

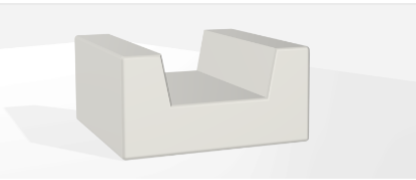



Note

External Axis is valid for setting only when **Robot Hold** is set to **False**.

- 4 In the **Wobjdata** area, check the current work object position and calibration status.
- 5 Click **Define** to enter the calibration process.
The **Define WobjName** window is displayed.
- 6 Choose the calibration method, **Polyhedron** or **Cube**, based on the work object shape from the **Calibration Method** list.

Following figures illustrate the shapes of the polyhedron and cube.

Polyhedron	Cube
 <p>Polyhedron [High Accuracy] [Time-Consuming]</p> <p>xx2400000697</p>	 <p>Cube [Moderate Accuracy] [Moderate Time-Consuming]</p> <p>xx2400000698</p>

Continues on next page

5 Auto-calibrating

5.4 Calibrating external axes and work objects

Continued



Note

Calibration method *Polyhedron* is available from Machining Software version 1.4.0. Using the Polyhedron calibration mode will obtain a higher accuracy calibration result but it consumes more time.

- 7 Choose a calibrated probe from the **Probe** list.



Tip

Only probes whose robot-hold status is opposite to the to-be-calibrated work object are available to choose.

- 8 Click **Polyhedron** or **Cube** to enter the setting window for defining the calibration target for the probe to touch.

Method	Action
Polyhedron	<p>a Set Length(mm) and Width(mm) to the actual values. You can prepare a standard polyhedron block by referring to the drawing with predefined dimensions. The standard block is in 120 mm length and 100 mm width. See Drawing for polyhedron block on page 79.</p> <p>You can also customize a block as required. When designing the block, make sure that the difference between the length value and width value is larger than 10 mm.</p> <p>b View other parameters. Top Height(mm), Mid Height(mm) and Concave Depth(mm) are in fixed value; Mid Width(mm) is half of the defined Width(mm) value.</p> <p>c Click Save to return.</p>
Cube	<p>a Set Length (mm), Width (mm) and Height (mm) to the actual values.</p> <p>b Click the Touchpoint Layout arrow to display the panel for selecting point matrix. At least a 2x2 matrix, that is, 4 points, must be selected.</p> <p>c Click Save to return.</p>

- 9 Click **Search Settings** to enter the setting window for defining how the probe goes to search.

In the displayed **Search Settings** window,

- Set **Search Mode**.
- If it is *Cube*, set **Search Speed**.
- If it is *Polyhedron*, set **Accuracy**.

Five accuracy levels are available for choose. Number of touchpoints on each block surface for each level can be previewed in 3D graphics window in real time, and the estimated calibration time is also presented for reference. The higher the accuracy level is, the more calibration time is taken.

- Click the **Advanced** arrow to expand the settings and set **Search Offset** and **Approach Distance**.

Continues on next page

**Tip**

You can click the **Information** icon at the upper-right corner of the **Search Settings** window to view the parameter definitions.

- 10 Select a corner on the work object as the origin point in the 3D graphics window.

Only after the origin point is selected, the position information can be displayed and target updating can be performed. You can click **Change Origin** to change an origin point.

**Note**

The coordination system is based on whether the work object is hold by the robot. If yes, the tool coordinate system is used; otherwise, the work object coordinate system is used.

- 11 Jog the probe to touch the work object at the **Origin**, **X** and **Y** positions, and click **Update Position** to record corresponding values.

**Note**

If the external axis is enabled, before jogging the probe to the **Origin** position, make sure that the external axis is in 0°.

**Note**

The **Origin**, **X** and **Y** positions are related to both wobj0 (robot base frame) and the tool coordinate system of the selected probe.

The **Home**, **XOYHome**, **XOZHome** and **YOZHome** positions are related to the workobject coordinate system of the work object, which has been calculated based on the **Origin**, **X** and **Y** positions.

**Note**

The **Origin** position is the one where the probe touches the origin point. The **X** position is the corner point where the probe touches after moving along the **X** axis direction of the origin point. The **Y** position is a point on the positive **Y**-axis side.

The **Home**, **XOYHome**, **XOZHome** and **YOZHome** positions are valid for updating only after the **Origin**, **X** and **Y** positions are updated.

After the **Home** position is updated, the **XOYHome**, **XOZHome** and **YOZHome** positions update automatically. You can further update the three positions separately.

Continues on next page

5 Auto-calibrating

5.4 Calibrating external axes and work objects

Continued

12 Click **Calibrate** and then click **Play**.

- If the controller is in Auto mode, the calibration process runs automatically.
- If the controller is in Manual mode, follow the instructions in the displayed messages to perform the calibration.

13 Click **Save** to save the data after the calibration done.

6 RAPID programming

Overview

RAPID programming is supported only by Machining Software in advanced version (option Machining Premium).

Continues on next page

6 RAPID programming

6.1 RAPID exporting and reloading

6.1 RAPID exporting and reloading

RAPID exporting

Machining Software allows users to apply the created or tuned programs to the connected controller. When the program is ready, click the **Apply** button to apply the programs.




Note

The **Apply** button is valid only when the writing access has been requested and granted. If the controller is in Auto mode, the writing access will be automatically granted to Machining Software; if the controller is in Manual mode, the writing access have to be granted in FlexPendant.

If the writing access is held by another client, release the access first and request from Machining Software again.

A main module and several related modules will be generated.

Module	Description
CalibData_HMI	Includes the tool data, work object data, speed data, zone data and calibration result data.
EventDefGroup_ <i>ProgramID</i>	If events are specified for instructions, the event data will be loaded in the EventDefGroup_ <i>ProgramID</i> module. <i>ProgramID</i> specifies the program to which the events belong.
Group_ <i>ProgramID</i> _OperationID_000	Includes the path data. <i>ProgramID</i> and <i>OperationID</i> specify the program and operation to which the instruction belongs.
Main_test_ <i>ProgramID</i>	The main module of the Machining Software program.
ProcessDefGroup_ <i>ProgramID</i> _OperationID_000	Includes the machining process data. <i>ProgramID</i> and <i>OperationID</i> specify the program and operation to which the machining process belongs.
TargetDefGroup_ <i>ProgramID</i> _OperationID_000	Includes the target point data. <i>ProgramID</i> and <i>OperationID</i> specify the program and operation to which the target point belong.
WaveGroup_ <i>ProgramID</i> _OperationID_000	Includes the wave path data, including wave pattern, speed data, zone data and work object data. <i>ProgramID</i> and <i>OperationID</i> specify the program and operation that enables wave path. <div> Note The speed data and zone data are not defined for wave paths in Machining Software. They are set to v50 and z0 respectively by default and allow users to modify in RAPID editor.</div>
InstructionName_T	Includes the point array data for wave paths. <i>InstructionName</i> specifies the name of the instruction that enables wave path.

Continues on next page

Module	Description
<i>PatternName_NodeMod</i>	Includes the wave path pattern data. <i>PatternName</i> starts with "C" or "L" that specifies the motion type, circular or linear and follows with the user-defined pattern name.

Exporting RAPID to controller

The modules are loaded into the `T_ROB1` task of the controller after clicking **Apply**.

RAPID reloading

The applied Machining Software modules must be edited based on programming rules to ensure successful reloading to Machining Software.



Note

Wave path modules could be edited in RAPID editor without programming limitations and applied to the controller directly, but they could not be reloaded from the controller to Machining Software.

In module,	Programming rule
<i>CalibData_HMI</i>	<ul style="list-style-type: none"> Values of the tooldata and wobjdata can be modified. Do not modify the names of the tooldata or wobjdata. <p>Values of the tooldata or wobjdata whose name is modified cannot be reloaded to Machining Software.</p>
<i>EventDefGroup_ProgramID</i>	<ul style="list-style-type: none"> Events included in a procedure (between PROC and ENDPROC) can be modified, either the complete event routine or the parameters defined in the event routine. Do not insert new event routines in an event routine. Do not modify the procedure name. <p>Event routines in the procedure cannot be reloaded to Machining Software even the events are modified.</p> <ul style="list-style-type: none"> Do not delete event routines. <p>Deleted events will not be deleted from Machining Software and will be exported again when program download/export is executed in Machining Software.</p>
<i>Group_ProgramID_OperationID_000</i>	<ul style="list-style-type: none"> Values of the speeddata and zonedata in the instructions can be modified, but only predefined values are allowed. Do not modify the names of the robtarget, RCS, tooldata, wobjdata or process in the instructions. <p>Values of the robtarget, RCS, tooldata, wobjdata or process whose name is modified cannot be reloaded to Machining Software.</p> <ul style="list-style-type: none"> Do not delete instructions. <p>Deleted instructions will not be deleted from Machining Software and will be exported again when program download/export is executed in Machining Software.</p>
<i>Main_test_ProgramID</i>	Not allowed to be edited.

Continues on next page

6 RAPID programming

6.1 RAPID exporting and reloading

Continued

In module,	Programming rule
ProcessDefGroup_ProgramID_OperationID_000	<ul style="list-style-type: none">• Values of the EngageDistance, TiltAngle and LeadAngle components in MachineProcess data type can be modified.• Do not modify the names of the PreRoutine or PostRoutine components in MachineProcess data type. Values of the components whose name is modified cannot be reloaded to Machining Software.• Do not modify the names of processes in MachineProcess data type. Values of the EngageDistance, TiltAngle and LeadAngle components cannot be reloaded to Machining Software even they are modified.• Do not add new processes in MachineProcess data type. Values of the new processes cannot be reloaded to Machining Software.• Do not delete processes in MachineProcess data type. Deleted processes will not be deleted from Machining Software and will be exported again when program download/export is executed in Machining Software.
TargetDefGroup_ProgramID_OperationID_000	<ul style="list-style-type: none">• Value of the robtarget can be modified.• Do not modify RCS values. Modified RCS values cannot be reloaded to Machining Software.• Do not modify the target names. Value of the target whose name is modified cannot be reloaded to Machining Software.• Do not add new robtarget or RCS. Values of the new robtarget and RCS cannot be reloaded to Machining Software.• Do not delete robtarget or RCS. Deleted robtarget and RCS will not be deleted from Machining Software and will be exported again when program download/export is executed in Machining Software.

All the Machining Software modules (wave path related modules excluded) are also declared with "`! Generated by ABB Machining Software HMI - Machining Functionality for ABB Robot xxx`", which cannot be removed. Before reloading the modules to the Machining Software, verify that the declaration remains and the modules are programmed following the rules.

Reloading from controller

When reloading the modules from the controller to the Machining Software,

- if there is no modification made in the Machining Software, the programs will be reloaded to Machining Software directly.
- if there is modifications made in the Machining Software, a message will be displayed, prompting you to synchronize from the controller or keep the modification in Machining Software.

Reloading from local file

The modules saved to the local folder must be loaded to the `T_ROB1` task of the controller first, and then be programmed and reloaded to Machining Software.

Continues on next page

Multiple instruction exporting

Introduction

If the targets in a program is more than 2000, it is recommended to enable the multiple instruction exporting function. With the function enabled, every 200 instructions will be grouped and included in a separate module in the T_Large task. The modules in the T_Large task will be further called by the main routine in the T_ROB1 task. This facilitates program programming and reading.

The multiple instruction exporting function is supported only by Machining Software in advanced version (option Machining Premium). To enable the function, make sure the Multitasking option is selected and then select option Machining Premium > More than 2000 instructions > Add T_large task in the **Modify Installation** dialogue.

Working with multiple instruction exporting function

- 1 Open RobotStudio.
- 2 In the Add-Ins tab, click **Machining 202X.X**.
- 3 In the **Machining** tab, choose **Export Settings** from the **Export** list in the **Path Tools** group.
- 4 In the displayed **Export Settings** window, choose **Large_Export_Rules_Library.yml** from the **Active Template** drop-down list in the **Export template library** area, and then click **OK**.
- 5 In the **Program** tab page on the left pane of the window, right-click the required program and choose **Export RAPID** from the shortcut menu.
 - Information of every 200 targets is included in an LN file. If less than 200 targets are left for the last file, default data will be added to make sure 200 target items are included in the file.
 - The LN files are named with suffix "_LN_{index}", in which "index" refers to the file number.
 - The LN files are generated and exported to directory
\$HOME/LN{ProgramName}/Opr_{ProgramID}_{Operation_ID}.
- 6 Click the **Controller** tab and, in the **RAPID** category in the **Controller** navigation tree in the displayed window:
 - Right-click the T_Large task, choose **Load Module** from the shortcut menu and load modules `BackgroundCalibData.mod` and `BackgroundMain.mod`.
 - Right-click the T_ROB1 task, choose **Load Module** from the shortcut menu and load modules `MotionCalibData.mod`, `MotionMain.mod` and, if any, `MotionEventDefGroup`.
- 7 Click the **RAPID** tab and remove the selection of the T_ROB1 task from the **Selected tasks** list in the **Test and Debug** group.
- 8 Set the program pointer to the `GenerateModules_Opr` routine in the T_Large task.
- 9 Run the program to start RAPID compiling.
Corresponding binary files are generated for the LN files in the same directory.

Continues on next page

6 RAPID programming

6.1 RAPID exporting and reloading

Continued

- 10 Click the **RAPID** tab and select the T_ROB1 task from the **Selected tasks** list in the **Test and Debug** group.
- 11 Set the program pointer to the main routine in the T_ROB1 task.
- 12 Run the program.

6.2 RAPID instructions

6.2.1 MachL - Moves linearly

Usage

In machining programs, `MachL` is used to move the tool center point (TCP) linearly to a given destination.

This instruction can only be used in the main task `T_ROB1` or, if in a MultiMove System, in Motion tasks.

Arguments

`offset`

Data type: `pose`

`offset` is used to add an offset to the robot position in the object coordinate system.

`RelEuler`

Data type: `MachiningPose` (See [MachiningPose - Coordinate transformations in machining programs on page 68](#))

`RelEuler` is used to add a displacement and/or a rotation, expressed in the active tool coordinate system, to a robot position.

`ToPoint`

Data type: `robtarget`

`ToPoint` defines the destination point of the robot and external axes.

`\RCS`

Data type: `pose`

`RCS` defines the coordinate system for machining the contact point. The Z-axis direction is the same as the normal vector. The value of the this argument is generated automatically in Machining PowerPac and not allowed to be changed.

`\ID`

Data type: `identno`

The argument `\ID` is mandatory in the MultiMove systems, if the movement is synchronized or coordinated synchronized. This argument is not allowed in any other case. The specified id number must be the same in all the cooperating program tasks. By using the id number the movements are not mixed up at the runtime.

`Speed`

Data type: `speeddata`

The speed data that applies to movements. Speed data defines the velocity for the tool center point, the tool reorientation, and external axes.

`Zone`

Data type: `zonedata`

Continues on next page

6 RAPID programming

6.2.1 MachL - Moves linearly

Continued

Zone data for the movement. Zone data describes the size of the generated corner path.

Tool

Data type: tooldata

The tool in use when the robot moves. The tool center point is the point moved to the specified destination position.

\Wobj

Data type: wobjdata

The work object (coordinate system) to which the robot position in the instruction is related.

This argument can be omitted and if so then the position is related to the world coordinate system. If, on the other hand, a stationary tool or coordinated external axes are used then this argument must be specified to perform a linear movement relative to the work object.

\Corr

Data type: switch

Correction data written to a corrections entry by the instruction CorrWrite will be added to the path and destination position if this argument is present.

The RobotWare option Path Offset is required when using this argument.

\TLoad

Data type: loaddata

The \TLoad argument describes the load attached to the tool flange of the robot.

process

Data type: MachineProcess (See [MachineProcess - Process definitions in machining programs on page 69](#))

The process argument describes the required process parameters and other necessary information for machining applications.

6.2.2 MachJ - Moves using joint movement

Usage

In machining programs, `MachJ` is used to move quickly from one point to another when that movement does not have to be in a straight line.

The robot and external axes move to the destination position along a non-linear path. All axes reach the destination position at the same time.

This instruction can only be used in the main task `T_ROB1` or, if in a MultiMove system, in Motion tasks.

Arguments

`offset`

Data type: `pose`

`offset` is used to add an offset to the robot position in the object coordinate system.

`RelEuler`

Data type: `MachiningPose` (See [MachiningPose - Coordinate transformations in machining programs on page 68](#))

`RelEuler` is used to add a displacement and/or a rotation, expressed in the active tool coordinate system, to a robot position.

`ToPoint`

Data type: `robtarget`

`ToPoint` defines the destination point of the robot and external axes.

`\RCS`

Data type: `pose`

`RCS` defines the coordinate system for machining the contact point. The Z-axis direction is the same as the normal vector. The value of the this argument is generated automatically in Machining PowerPac and not allowed to be changed.

`\ID`

Data type: `identno`

The argument `\ID` is mandatory in the MultiMove systems, if the movement is synchronized or coordinated synchronized. This argument is not allowed in any other case. The specified id number must be the same in all the cooperating program tasks. By using the id number the movements are not mixed up at the runtime.

`Speed`

Data type: `speeddata`

The speed data that applies to movements. Speed data defines the velocity for the tool center point, the tool reorientation, and external axes.

`Zone`

Data type: `zonedata`

Continues on next page

6 RAPID programming

6.2.2 MachJ - Moves using joint movement

Continued

Zone data for the movement. Zone data describes the size of the generated corner path.

Tool

Data type: tooldata

The tool in use when the robot moves. The tool center point is the point moved to the specified destination position.

\Wobj

Data type: wobjdata

The work object (coordinate system) to which the robot position in the instruction is related.

This argument can be omitted and if so then the position is related to the world coordinate system. If, on the other hand, a stationary tool or coordinated external axes are used then this argument must be specified to perform a linear movement relative to the work object.

\TLoad

Data type: loaddata

The \TLoad argument describes the load attached to the tool flange of the robot.

process

Data type: MachineProcess (See [MachineProcess - Process definitions in machining programs on page 69](#))

The process argument describes the required process parameters and other necessary information for machining applications.

6.2.3 MachC - Moves circularly

Usage

In machining programs, `MachC` is used to move the tool center point (TCP) circularly to a given destination. During the movement the orientation normally remains unchanged relative to the circle.

This instruction can only be used in the main task `T_ROB1` or, if in a MultiMove system, in Motion tasks.

Arguments

`offset`

Data type: `pose`

`offset` is used to add an offset to the robot position in the object coordinate system.

`RelEuler`

Data type: `MachiningPose` (See [MachiningPose - Coordinate transformations in machining programs on page 68](#))

`RelEuler` is used to add a displacement and/or a rotation, expressed in the active tool coordinate system, to a robot position.

`CirPoint`

Data type: `robtarget`

The circle point of the robot. The circle point is a position on the circle between the start point and the destination point.

`\RCS_Cir`

Data type: `pose`

`RCS` defines the coordinate system for circle point.

`ToPoint`

Data type: `robtarget`

`ToPoint` defines the destination point of the robot and external axes.

`\RCS_To`

Data type: `pose`

`RCS` defines the coordinate system for destination point.

`\ID`

Data type: `identno`

The argument `\ID` is mandatory in the MultiMove systems, if the movement is synchronized or coordinated synchronized. This argument is not allowed in any other case. The specified id number must be the same in all the cooperating program tasks. By using the id number the movements are not mixed up at the runtime.

Continues on next page

6 RAPID programming

6.2.3 MachC - Moves circularly

Continued

Speed

Data type: speeddata

The speed data that applies to movements. Speed data defines the velocity for the tool center point, the tool reorientation, and external axes.

Zone

Data type: zonedata

Zone data for the movement. Zone data describes the size of the generated corner path.

Tool

Data type: tooldata

The tool in use when the robot moves. The tool center point is the point moved to the specified destination position.

\Wobj

Data type: wobjdata

The work object (coordinate system) to which the robot position in the instruction is related.

This argument can be omitted and if so then the position is related to the world coordinate system. If, on the other hand, a stationary tool or coordinated external axes are used then this argument must be specified to perform a linear movement relative to the work object.

\TLoad

Data type: loaddata

The \TLoad argument describes the load attached to the tool flange of the robot.

process

Data type: MachineProcess (See [MachineProcess - Process definitions in machining programs on page 69](#))

The process argument describes the required process parameters and other necessary information for machining applications.

6.2.4 WaveLStartPoint - Moves to the wave start point

Usage

In machining programs, `WaveLStartPoint` is used to specify the position of the wave start point. If the motion type is circular, this instruction is operated and the robot moves to the specified wave start point first to process. If the motion type is linear, this instruction is skipped and the robot directly moves to the next wave node to process.

This instruction can only be used in the main task `T_ROB1` or, if in a MultiMove System, in Motion tasks.

Arguments

`WShape`

Data type: `WaveShape` (See [WaveShape - wave patterns defined for wave paths on page 73](#))

`WShape` specifies the wave pattern, circular or linear.

`ToTarget`

Data type: `robtarget`

`ToTarget` specifies the sourcing target generated by Machining PowerPac, which is used as reference for generating the wave start node.

`Speed`

Data type: `speeddata`

The speed data that applies to movements. Speed data defines the velocity for the tool center point, the tool reorientation, and external axes.

`Zone`

Data type: `zonedata`

Zone data for the movement. Zone data describes the size of the generated corner path.

`Tool`

Data type: `tooldata`

The tool in use when the robot moves. The tool center point is the point moved to the specified destination position.

`\Wobj`

Data type: `wobjdata`

The work object (coordinate system) to which the robot position in the instruction is related.

This argument can be omitted and if so then the position is related to the world coordinate system. If, on the other hand, a stationary tool or coordinated external axes are used then this argument must be specified to perform a linear movement relative to the work object.

6 RAPID programming

6.2.5 WaveL - Moves along the wave path

6.2.5 WaveL - Moves along the wave path

Usage

In machining programs, WaveL is used to move the tool center point (TCP) along the wave path.

This instruction can only be used in the main task T_ROB1 or, if in a MultiMove System, in Motion tasks.

Arguments

WShape

Data type: WaveShape (See [WaveShape - wave patterns defined for wave paths on page 73](#))

WShape specifies the wave pattern, circular or linear.

TargetArray

Data type: robtarget

TargetArray specifies the array of sourcing targets generated by Machining PowerPac, which are used as reference for generating wave nodes.

Start

Data type: num

Start is the index for the start point, where the wave path starts.

End

Data type: num

End is the index for the end point, where the wave path ends.

Speed

Data type: speeddata

The speed data that applies to movements. Speed data defines the velocity for the tool center point, the tool reorientation, and external axes.

Zone

Data type: zonedata

Zone data for the movement. Zone data describes the size of the generated corner path.

Tool

Data type: tooldata

The tool in use when the robot moves. The tool center point is the point moved to the specified destination position.

\Wobj

Data type: wobjdata

The work object (coordinate system) to which the robot position in the instruction is related.

Continues on next page

This argument can be omitted and if so then the position is related to the world coordinate system. If, on the other hand, a stationary tool or coordinated external axes are used then this argument must be specified to perform a linear movement relative to the work object.

6 RAPID programming

6.3.1 MachiningPose - Coordinate transformations in machining programs

6.3 RAPID data types

6.3.1 MachiningPose - Coordinate transformations in machining programs

Usage

Data of the type `MachiningPose` describes how a coordinate system is displaced and rotated.

Components

X

Data type: `num`

The displacement in X-axis position of the coordinate system.

Y

Data type: `num`

The displacement in Y-axis position of the coordinate system.

Z

Data type: `num`

The displacement in Z-axis position of the coordinate system.

Rx

Data type: `num`

The orientation in X-axis position of the coordinate system.

Ry

Data type: `num`

The orientation in Y-axis position of the coordinate system.

Rz

Data type: `num`

The orientation in Z-axis position of the coordinate system.

6.3.2 MachineProcess - Process definitions in machining programs

Usage

Data of the type `MachiningProcess` describes parameters specially used during machining process.

Components

`EngageDistance`

Data type: `num`

The pressure amount with which the tool will move a further distance towards to the work object.

`TiltAngle`

Data type: `num`

The angle (inclined right or left) that the tool will rotate along the forward direction of the machining path.

`LeadAngle`

Data type: `num`

The angle (inclined forward or backward) that the tool will rotate along the tangent direction to the forward direction of the machining path.

`PreRoutine`

Data type: `string`

The component `PreRoutine` defines the event that is executed before a motion instruction.

`PostRoutine`

Data type: `string`

The component `PostRoutine` defines the event that is executed after a motion instruction.

6 RAPID programming

6.3.3 WaveDirection - Wave path processing directions

6.3.3 WaveDirection - Wave path processing directions

Usage

Data of the type `WaveDirection` describes the coordinate system of the wave nodes and the processing forward direction of the wave paths.

Components

`CoordinateSystem`

Data type: `num`

Value of `TOOLSYSTEM` is 0, indicating the tool coordinate system is used.

`Direction`

Data type: `string`

Processing forward direction of the wave path.

6.3.4 WaveDistribution - Distribution information of wave nodes

Usage

Data of the type `WaveDistribution` describes the node array, that is, how the nodes distribute.

Components

Nodes

Data type: `string`

Name of the node array.

NodeCount

Data type: `num`

Quantity of nodes in the node array.

StartPoint

Data type: `pos`

Wave start point of the wave path. This parameter works only for the wave path in circular pattern.

6 RAPID programming

6.3.5 WaveNode - wave nodes defined for wave paths

6.3.5 WaveNode - wave nodes defined for wave paths

Usage

Data of the type `WaveNode` describes how the wave nodes are defined based on the sourcing targets generated by Machining PowerPac.

Components

`Offset`

Data type: `pos`

Displacement offset of wave nodes from the sourcing targets based on which the nodes are generated.

`Index`

Data type: `num`

Index indicating the sourcing target from which the wave node is generated.

`MotionType`

Data type: `string`

The robot motion type, linear or circular.

6.3.6 WaveShape - wave patterns defined for wave paths

Usage

Data of the type `WaveShape` describes the wave pattern of the wave path.

Components

`WType`

Data type: `string`

Name of the wave path.

`WaveDistribution`

Data type: `WaveDistribution`

Distribution information of nodes in the wave path.

`WaveDirection`

Data type: `WaveDirection`

Coordinate system of nodes and processing forward direction of the wave path.

`nStep`

Data type: `num`

Step for the wave pattern to repeat.

`WMotion`

Data type: `string`

Wave motion function, whose default value is `WaveMotionCustomDefault`. It allows users to customize motion functions as requires. The function is available in the system module `WaveMotionModule`.

This page is intentionally left blank

7 Reference information

7.1 Messages and errors

Message list

ID	Message
118001	Calibration Process Information
118002	TouchMotion Information
118003	Parameter error

Error list

118004 Array Length is too short

Description	The array length is too short.
Consequence	The program is stopped.
Causes	None
Actions	None

118005 Distance is zero

Description	The distance between the home point and the contact point is 0.
Consequence	The program is stopped.
Causes	The home point is too close to the contact point.
Actions	None

118006 Tool and Wobj are wrong

Description	Active tool and active work object cannot have the RobHold parameter set to the same value, no matter both True or both False .
Consequence	The program is stopped.
Causes	The setting of the robhold component in the tooldata and wobjdata instructions is conflict.
Actions	Change the setting of the RobHold parameter for the active tool and active work object to different value.

118007 External Axis is not at Zero position

Description	The axis is not at zero position.
Consequence	The program is stopped.
Causes	The external axis is found not at the zero position.
Actions	Jog the external axis to the zero position.

118008 Array length mismatch

Description	The size of the array does not match the size used.
Consequence	The program is stopped.
Causes	The length of the array is less than the number of elements used.

Continues on next page

7 Reference information

7.1 Messages and errors

Continued

Actions	Modify the number of elements used.
---------	-------------------------------------

118009 Precision is not enough

Description	Data or position is not defined accurately.
Consequence	The program is stopped.
Causes	No enough data or the positions do not have the required relations or not specified with enough accuracy.
Actions	Increase the number of data and improve the data correlation.

118010 Collision occurred during probe movement

Description	Collision occurs during probe movement.
Consequence	The program is stopped.
Causes	The probe has collided with an obstacle in the cell during its movement.
Actions	<ol style="list-style-type: none">1 Check the probe retraction status, succeeded or failed.2 If failed, go to Manual mode.3 Manually run the robot to move the probe away from the object.4 Resume operation by restarting the program.

118011 Probe retraction succeeded

Description	The manipulator has attempted to move the probe back away from the obstacle, into which it collided, and succeeded.
Consequence	The system is ready to go back to normal operation.
Causes	None
Actions	None

118012 Probe retraction failed

Description	The manipulator has attempted to move the probe back away from the obstacle, into which it collided, and failed
Consequence	The system is NOT ready to go back to normal operation.
Causes	This may be caused by the probe being stuck to the object into which it collided.
Actions	<ol style="list-style-type: none">1 Go to Manual mode.2 Manually run the robot to move the probe away from the object.3 Resume operation by restarting the program.

118100 Robot-hold status confliction for crossbeam and its related tool

Description	Robot-hold status of the tool and crossbeam (or the wobjdata defined for the crossbeam) conflict.
Consequence	The program is stopped.
Causes	The tool selected for crossbeam calibration and the crossbeam (or the wobjdata defined for the crossbeam) are both set as in fixed position or hold by the robot.
Actions	<ol style="list-style-type: none">1 Check the robot-hold status of each item.2 Modify the required one.

Continues on next page

118101 Robot-hold status confliction for original and calibrated tooldata

Description	Robot-hold status included in the original tooldata and calibrated tooldata of the tool conflict.
Consequence	The program is stopped.
Causes	The robot-hold status in the calibrated tooldata of the tool is different from that in the original tooldata used for the tool.
Actions	<ul style="list-style-type: none">• Check the robot-hold status in the original and calibrated tooldata.• Modify the required one.

118610 Rapid module is missing

Description	The specified RAPID module cannot be found.
Consequence	The program is stopped.
Causes	The RAPID module is missing.
Actions	<ul style="list-style-type: none">• Check the file path and file name.• Check whether the file exist.

118611 LN file is missing

Description	The specified LN file cannot be found.
Consequence	The program is stopped.
Causes	The LN file is missing.
Actions	<ul style="list-style-type: none">• Check the file path and file name.• Check whether the file exist.

7.2 Troubleshooting

Robot stops and reports 40661, Search Error during probe searching

Description

During the calibration using a probe, the robot stops and event 40661 Search Error is reported when the probe moves to search towards the calibration target.

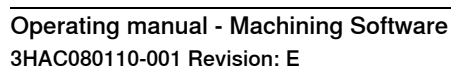
Possible causes

When the probe starts search towards a surface of the calibration target, the robot will turn its position to the same as the home position defined for this surface. With this position, if the searching target is around the singularity or the boundary of robot working range, the error is reported.

Action

In the defining window, jog the robot to another position and update the **Home** position to this new position. Then, start the calibration process again.

Drawing for polyhedron block



This page is intentionally left blank

Index

A

authentication
 user grant, 13

C

calibration
 cutter, 42
 external axis, 47
 toolkit, 37
 work object, 47
cutter, 42
 machining tool type, 45
 sander type, 42

E

external axis, 47

L

license, 13

N

network security, 9

P

path
 smooth path, 24
 wave path, 25

S

safety, 10
smooth path, 24

T

toolkit, 37
 crossbeam type, 39
 probe type, 37

V

version, 11
 license, 13

W

wave path, 25
work object
 cube, 49
 polyhedron, 49

**ABB AB****Robotics & Discrete Automation**

S-721 68 VÄSTERÅS, Sweden

Telephone +46 10-732 50 00

ABB AS**Robotics & Discrete Automation**

Nordlysvegen 7, N-4340 BRYNE, Norway

Box 265, N-4349 BRYNE, Norway

Telephone: +47 22 87 2000

ABB Engineering (Shanghai) Ltd.

Robotics & Discrete Automation

No. 4528 Kangxin Highway

PuDong New District

SHANGHAI 201319, China

Telephone: +86 21 6105 6666

ABB Inc.**Robotics & Discrete Automation**

1250 Brown Road

Auburn Hills, MI 48326

USA

Telephone: +1 248 391 9000

abb.com/robotics