ROBOTICS

Operating manual

Integrator's guide OmniCore

Trace back information:
Workspace 24A version a11
Checked in 2024-03-05
Skribenta version 5.5.019

Operating manual

OmniCore

RobotWare 7.14

Document ID: 3HAC065037-001

Revision: R

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.

Table of contents

This page is intentionally left blank

Overview of this manual

About this manual

This manual contains advanced instructions for OmniCore based robot systems using a FlexPendant.

The daily operations are described in *Operating manual - OmniCore*. This manual describes aspects for commissioning, as well as advanced instructions that are not used by the operator when the robot is running in production.

Note

It is the responsibility of the integrator to conduct a risk assessment of the final application.

It is the responsibility of the integrator to provide safety and user guides for the robot system.

Note

Screenshots in this manual are generally intended to show a language version corresponding to the language of the manual. In some cases, a translated manual still uses English screenshots if the localized user interface was not available at the time of publishing the manual.

Usage

This manual should be used during commissioning and when making changes to the robot that are outside the scope of everyday operations. This manual should be used together with *Operating manual - OmniCore* that describes more common operations, and the product manual for the robot controller.

Note

Before any work on or with the robot is performed, the safety information in the product manual for the controller and manipulator must be read.

Who should read this manual?

This manual is intended for:

- integrators
- product technicians
- service technicians
- robot programmers

Prerequisites

The reader should:

- Have read and understood the safety instructions in the product manuals for the robot.
- Be familiar with the concepts described in *Operating manual - OmniCore*.

Continues on next page

Continued

- Be trained in robot operation.

References

Safety manual for robot - Manipulator and IRC5 or OmniCore controller	*3HAC031045-001*
Product manual - OmniCore E10	*3HAC079399-001*
Product manual - OmniCore C30	*3HAC060860-001*
Product manual - OmniCore C30 Type A	*3HAC089064-001*
Product manual - OmniCore C90XT	*3HAC073706-001*
Product manual - OmniCore V250XT Type A	*3HAC084692-001*
Product manual - OmniCore V250XT Type B	*3HAC087112-001*
Product manual - OmniCore V400XT	*3HAC081697-001*
Operating manual - OmniCore	*3HAC065036-001*
Operating manual - RobotStudio	*3HAC032104-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC065038-001*
Technical reference manual - RAPID Overview	*3HAC065040-001*
Technical reference manual - System parameters	*3HAC065041-001*
Application manual - Additional axes	*3HAC082287-001*
Application manual - CC-Link IE Field Basic	*3HAC082295-001*
Application manual - Controller software OmniCore	*3HAC066554-001*
Application manual - DeviceNet Master/Slave	*3HAC066562-001*
Application manual - EtherNet/IP Scanner/Adapter	*3HAC066565-001*
Application manual - PROFINET Controller/Device	*3HAC066558-001*
Application manual - RobotWare add-ins	*3HAC070207-001*
Application manual - Functional safety and SafeMove	*3HAC066559-001*

Tip

All documents can be found via myABB Business Portal, *www.abb.com/myABB*.

Revisions

Revision	Description
A	Released with RobotWare 7.0.
B	Released with RobotWare 7.0.1. • The safety information is moved to the product manuals for the controller and the manipulator. • References to the *Hold-to run-button* is replaced with *thumb button* in the manual. • SFTP and FTP are added in the list of application protocols in section *OmniCore application protocols on page 268*.
C	Released with RobotWare 7.0.2. • Updated information about add-ins in the installation procedure. • Updated information about queueing backups.

Continues on next page

Revision	Description
	• Added information about SafeMove.
D	Released with RobotWare 7.0.4. • Calibration information updated. • Updated NOTE regarding IP addresses in section "Set up the network connection". • Information adjusted in the following sections to include the controller C90XT: References, *The OmniCore controller on page 20*, *Connected Services configuration on page 54*, *Start RobotWare Installation Utilities on page 151*, "Network connections on the OmniCore".
E	Released with RobotWare 7.1. • Updated the section *Connected Services configuration on page 54*. • Information about Connected Services is added in the section *User Authentication System on page 260*. • Updated the section *Editing expressions on page 132*. • Section *Service Information System*: Added information about *Fleet Assessment* and caution about resetting counters. • New sections: - *Performing a controller disk cleanup on page 193* - *Managing system snapshots on page 188* - *Defining controller date and time on page 198* - *Setting the controller name on page 197* • Updated the section *Updating an existing RobotWare system via RobotStudio on page 171*. • *Creating a new installation package on page 181*: Updated with information about Distribution tab. • FlexPendant changes updated in entire manual. • Information about FTP removed in section "Set up the network connection". • Section "Network connections on the OmniCore" has been updated with NOTE regarding correct usage of the MGMT port.
F	Released with RobotWare 7.2. • Updated the section *Connected Services configuration on page 54*. • Added sections *Single point of control on page 74* and *Robots in collaborative applications on page 75*. • Added information about the new expression editor search bar in the section *Editing expressions on page 132*. • Added the section *Directory structure on OmniCore on page 34*. • A NOTE is added in the section *Restart, reset and recovery procedures on page 147* that the TEMP folder is emptied at controller restart if RAPID and system parameters are reset. • Minor corrections in section "Connections on the main computer". • EtherCAT is added in the list of protocols in section *OmniCore application protocols on page 268*.
G	Released with RobotWare 7.3. • Updated the section *Connected Services configuration on page 54*. • Minor corrections in section *Service Information System on page 76*, *RobotStudio on page 27* and *Directory structure on OmniCore on page 34*. • Section "Network connections on OmniCore" moved to chapter *Get started on page 37*. • Minor updates in section *RobotWare installation procedures on page 169*.

Continues on next page

Continued

Revision	Description
H	Released with RobotWare 7.4. • Information about OmniCore E line added in section *The OmniCore controller on page 20*. • Added the section *Creating user-defined data types on page 110*. • Sections "Network connections on OmniCore" and "Set up the network connection" replaced by *Ethernet networks on OmniCore on page 35*.
J	Released with RobotWare 7.5. • Updated the section *What is saved on backup? on page 158*. • Minor changes to section *Creating virtual controllers on page 180*. • Information about V line controller added in *References on page 10*, *The OmniCore controller on page 20* and *Ethernet networks on OmniCore on page 35*.
K	Released with RobotWare 7.6. • Added information about UDPUC in the section *OmniCore application protocols on page 268*. • Updated information about Integrated Vision in the section *OmniCore application protocols on page 268*. • Updated the NOTE in the section *Gateway types on page 56*. • Minor updates in section *RobotWare installation procedures on page 169*. • Sections **Reset RAPID program**, **Reset RAPID program and system parameters** and **Reset safety settings** merged into new section: *Reset user data on page 153* • Reference to AM CC Link IE added. • Information about I/O Network added in section *Configuring firewall settings on page 52*.
L	Released with RobotWare 7.7. • Information added in section *I/O signals on page 137* that two industrial network masters can be run in parallel on the OmniCore controller. • New section: *Add-in installation via FlexPendant on page 203*.
M	Released with RobotWare 7.8. • Added information about OPC UA protocol in the section *OmniCore application protocols on page 268*. • Information about application grants added in *User Authentication System on page 260*.
N	Released with RobotWare 7.10. • Information about OmniCore V250XT Type A added in *References on page 10* and *The OmniCore controller on page 20*. • Updated various sections to reflect the change of transferring the RAPID data editing functionality from **Calibrate** to **Program Data**. • Minor corrections in *RobotWare installation procedures on page 169*. • Information about port forwarding added in *Port forwarding on page 53*. • Information about communication and application protocols updated in section *Network architecture and communication on page 253* and *OmniCore application protocols on page 268*. • Information about certificate replacement updated in *Certificate handling on page 271*. • Added information about TuneMaster in section *Network architecture and communication on page 253*. • Minor corrections in section *Ethernet networks on OmniCore on page 35*.
P	Released with RobotWare 7.12. • Updated the section *RobotWare installation procedures on page 169*.

Continues on next page

Revision	Description
	• Added the new section *Manage configuration files on page 166*. • New section *Collecting diagnostics on page 199*. • Information about file formats updated in *Handling of modules on page 95*.
Q	Released with RobotWare 7.13. • ABB Connected Services is the new name for the functionality previously known as ABB Ability. During a period of time, both names will appear in and on our products. • Information about OmniCore V250XT Type B added in *References on page 10* and *The OmniCore controller on page 20*. • Information about OmniCore V400XT added in *References on page 10* and *The OmniCore controller on page 20*. • Updated the section *Ethernet networks on OmniCore on page 35*. • Drive system table in *Working with option selections on page 176* updated. • Added information that the restart type **Reset System** does not reset the topic *Communication* in the system parameter configuration. • Minor corrections in *OmniCore application protocols on page 268*. • Restructured information in the following sections: *Get started on page 37* and *Configuration on page 45*. • Information about "Apply and reset" and "Update history" added in *Updating an existing RobotWare system via RobotStudio on page 171*. • "Package Installer" renamed to "Apply Update". • Updated the section *Manage configuration files on page 166*. • Updated the section *Get started on page 37*.
R	Released with RobotWare 7.14. • Updated the section *Connected Services configuration on page 54*. • *Drive system types on page 178* updated. • Information about mass software update added in *RobotWare installation procedures on page 169*.

Product documentation

Categories for user documentation from ABB Robotics

The user documentation from ABB Robotics is divided into a number of categories. This listing is based on the type of information in the documents, regardless of whether the products are standard or optional.

Tip

All documents can be found via myABB Business Portal, *www.abb.com/myABB*.

Product manuals

Manipulators, controllers, DressPack, and most other hardware is delivered with a **Product manual** that generally contains:

- Safety information.
- Installation and commissioning (descriptions of mechanical installation or electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals and expected life time of parts).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Calibration.
- Troubleshooting.
- Decommissioning.
- Reference information (safety standards, unit conversions, screw joints, lists of tools).
- Spare parts list with corresponding figures (or references to separate spare parts lists).
- References to circuit diagrams.

Technical reference manuals

The technical reference manuals describe reference information for robotics products, for example lubrication, the RAPID language, and system parameters.

Application manuals

Specific applications (for example software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful).
- What is included (for example cables, I/O boards, RAPID instructions, system parameters, software).
- How to install included or required hardware.
- How to use the application.

Continues on next page

- Examples of how to use the application.

Operating manuals

The operating manuals describe hands-on handling of the products. The manuals are aimed at those having first-hand operational contact with the product, that is production cell operators, programmers, and troubleshooters.

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.

Open source and 3rd party components

Open source and 3rd party components

ABB products use software provided by third parties, including open source software. The following copyright statements and licenses apply to various components that are distributed inside the ABB software. Each ABB product does not necessarily use all of the listed third party software components. Licensee must fully agree and comply with these license terms or the user is not entitled to use the product. Start using the ABB software means accepting also referred license terms. The third party license terms apply only to the respective software to which the license pertains, and the third party license terms do not apply to ABB products. With regard to programs provided under the GNU general public license and the GNU lesser general public license licensor will provide licensee on demand, a machine-readable copy of the corresponding source code. This offer is valid for a period of three years after delivery of the product.

ABB software is licensed under the ABB end user license agreement, which is provided separately.

RobotWare

For RobotWare, there is license information in the folder **licenses** in the RobotWare distribution package.

OpenSSL

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

CTM

For OIeOS, the Linux based operating system used on the conveyor tracking module (CTM), a list of copyright statements and licenses is available in the file /**etc**/**licenses.txt** located on the CTM board and accessible via the console port or by downloading the file over SFTP.

For the CTM application, a list of copyright statements and licenses is available in the file /**opt**/**ABB.com**/**ctm**/**licenses.txt** located on the CTM board and accessible via the console port or by downloading the file over SFTP.

This page is intentionally left blank

1 Welcome to OmniCore

1.1 About this section

Overview

This section presents an overview of the FlexPendant, the OmniCore controller, and RobotStudio.

A robot consists of a robot controller, the FlexPendant, RobotStudio, and one or several manipulators or other mechanical units.

This manual describes a robot without options, not a robot system. However, in a few places, the manual gives an overview of how options are used or applied. Most options are described in detail in their respective application manual.

1.2 The OmniCore controller

Overview of OmniCore

The OmniCore controller contains all the functions needed to move and control the manipulator, and delivers flexibility, connectivity, and performance. The OmniCore controller gives ABB robots the ability to perform their tasks in a highly efficient manner and also increases the flexibility to incorporate the latest digital technologies. The controller comes with ABB's powerful operating system, RobotWare 7.

The controller can be equipped with additional offerings, such as industrial network protocols, vision solutions, and force control.

OmniCore E line

OmniCore E line is an ultra slim controller for confined spaces and high density lines within the OmniCore family, offering only the essential functions together with full motion performance and precision. The E line controller is designed for stand-alone and slave installations and has a very compact design with integrated hardware and RobotWare functions.

OmniCore E10

The OmniCore E10 controller has only one base configuration without electronic hardware options. For more information about the OmniCore E10 controller, see *Product manual - OmniCore E10*.

OmniCore C line

OmniCore C line is the compact line of controllers within the OmniCore family, offering significant size reduction and flexible integration possibilities without any compromise on performance or precision.

OmniCore C30

The OmniCore C30 controller offers a compact solution suitable for applications where there is less need for additional equipment inside. For more information about the OmniCore C30 controller, see *Product manual - OmniCore C30*.

OmniCore C90XT

The OmniCore C90XT controller offers a compact solution suitable for most applications with room for some additional equipment inside. For more information about the OmniCore C90XT controller, see *Product manual - OmniCore C90XT*.

Continues on next page

OmniCore V line

OmniCore V line is a versatile and powerful controller with high degree of flexibility covering a wide range robot and applications. V line supports external axis and provides flexible configuration opportunities.

OmniCore V250XT Type A

The OmniCore V250XT Type A controller offers a compact, yet flexible, solution for advanced applications and robots sizes up to IRB 6700. For more information about the OmniCore V250XT Type A controller, see *Product manual - OmniCore V250XT Type A*.

OmniCore V250XT Type B

The OmniCore V250XT Type B controller offers a compact, yet flexible, solution for advanced applications and robots sizes up to IRB 6700. The controller supports up to three additional drive units and has 15 liter optional space inside.

For more information about the OmniCore V250XT Type B controller, see *Product manual - OmniCore V250XT Type B*.

OmniCore V400XT

The OmniCore V400XT controller offers a compact, yet flexible, solution for advanced applications and robots sizes up to IRB 7600. The controller supports up to six additional drive units and has 50 liter optional space inside.

For more information about the OmniCore V400XT controller, see *Product manual - OmniCore V400XT*.

1.3 The FlexPendant

Introduction to the FlexPendant

The FlexPendant is a hand held operator unit that is used for many of the tasks when operating a robot: running programs, jogging the manipulator, modifying programs, and so on.

The FlexPendant is designed for continuous operation in harsh industrial environment. Its touchscreen is easy to clean and resistant to water, oil, and accidental welding splashes.

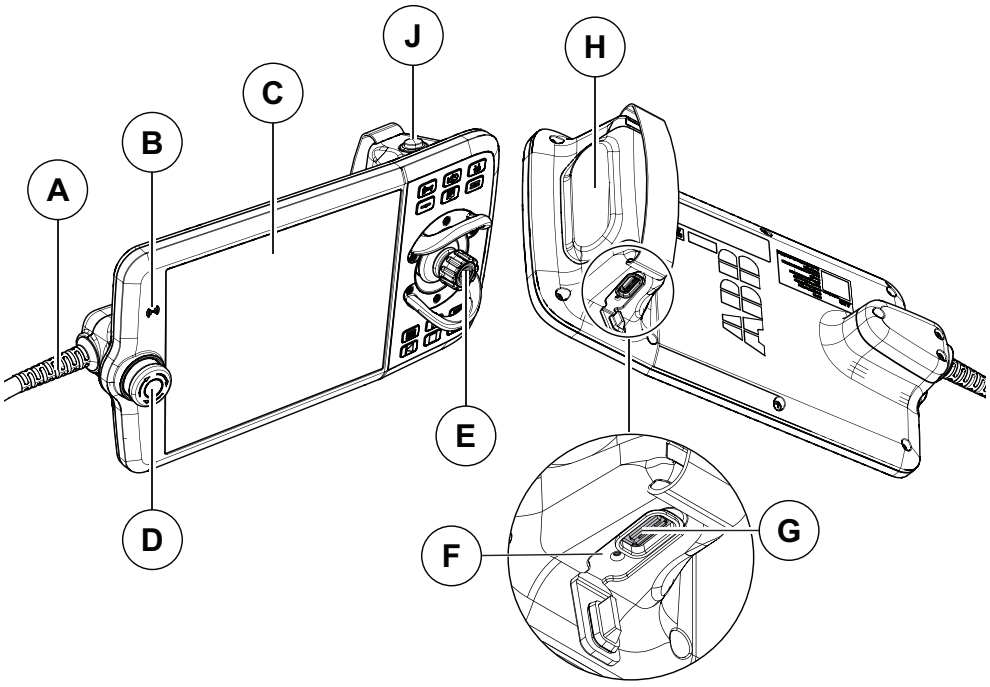
The FlexPendant consists of both hardware and software and is a complete computer in itself. It is connected to the robot controller by an integrated cable and connector.

Note

If protective gloves are used, these must be compatible with touchscreens when using the FlexPendant.

Main parts

These are the main parts of the FlexPendant.



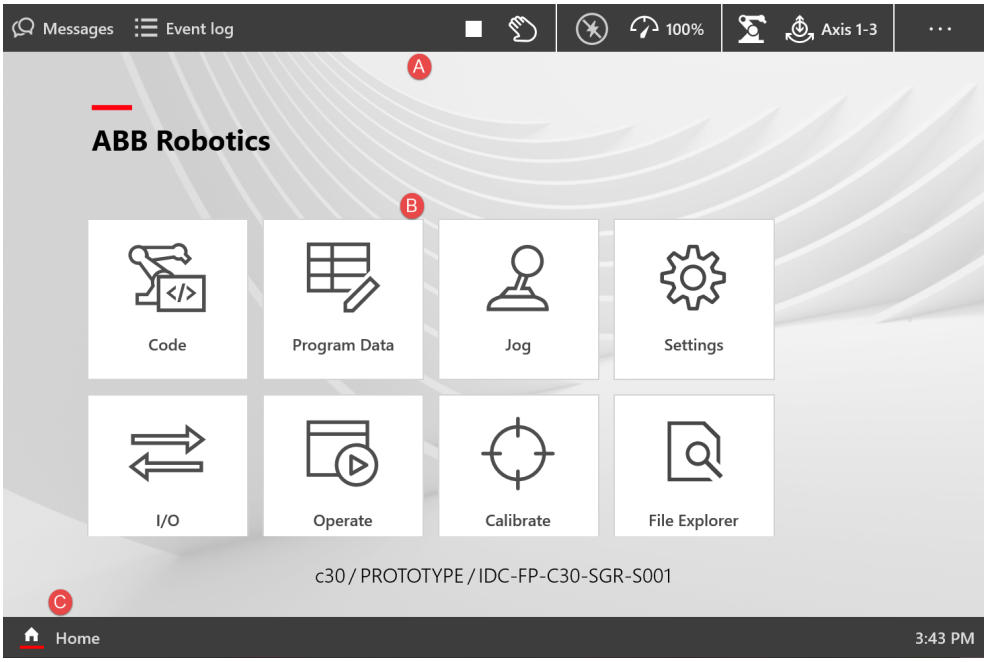
xx1700001891

A	Connector
B	RFID reader (functionality not yet implemented)
C	Touchscreen
D	Emergency stop device
E	Joystick

Continues on next page

F	Reset button
G	USB port
H	Three-position enabling device. For details, see *Three-position enabling device on page 25*
J	Thumb button. For details, see *Thumb button on page 26*.

Touchscreen



xx1800001181

A	Status bar buttons	Allows you to navigate to operator messages, event logs, and QuickSet window.
B	Applications	The applications that are required for operating a robot system are available in the Home Screen. By default, the Home screen displays all the applications available to you.
C	Home button	From any window tap the Home button to navigate to the Home screen of FlexPendant. The Home screen view is also the default view of the FlexPendant during startup.

Emergency stop device

On delivery, the emergency stop device on the FlexPendant is able to initiate the emergency stop function affecting the manipulator(s) and additional axis only.

Joystick

Use the joystick to move the manipulator. This is called jogging the robot. There are several settings for how the joystick will move the manipulator.

Reset button

If the FlexPendant freezes during operation, press the reset button to restart the FlexPendant.

The reset button resets the FlexPendant, not the system on the controller.

Continues on next page

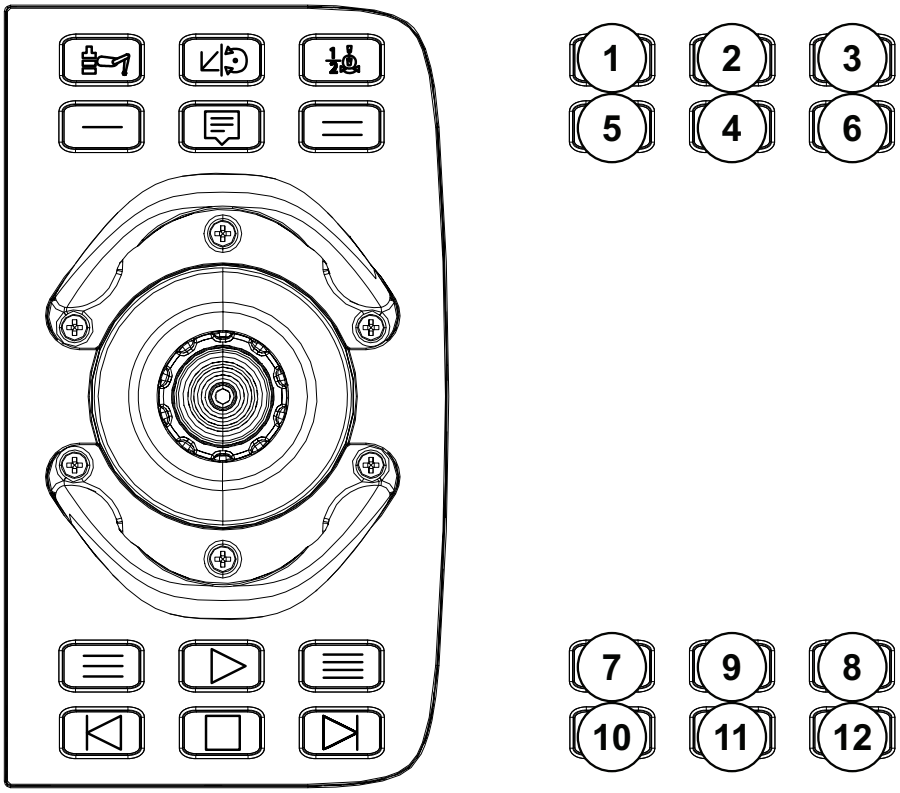
USB port

Connect a USB memory to the USB port to read or save files. For example, to load and save programs and modules, save and restore backups, and so on. The USB memory name and drive letter *(X:)* is displayed in dialogs.

> **Note**
>
> Close the protective cap on the USB port when not used.

Hard buttons

The following hard buttons are available on the FlexPendant.



xx1700001892

Label	Description
1	Mechanical unit button. Allows you to select a mechanical unit.
2	Motion mode button 1. Allows you to toggle the motion mode between reorient and linear.
3	Motion mode button 2. Allows you to toggle the motion mode between axis 1-3 and axis 4-6.
4	Messages button. Allows you to open the QuickSet window. **Note** Press the Messages button for a longer duration to capture a screenshot of the current screen. For more details, see *Operating manual - OmniCore*.

Continues on next page

Label	Description
5, 6, 7, 8	Programmable keys, 1 to 4. Programmable keys are hardware buttons on the FlexPendant that can be used for dedicated, specific functions set by the user.
9	**START** button. Starts the program execution.
10	**Step BACKWARD** button. Executes one instruction backward.
11	**STOP** button. Stops the program execution.
12	**Step FORWARD** button. Executes one instruction forward.

Note

The user interface of the panel in Virtual FlexPendant is slightly different. For more details, see *Operating manual - OmniCore*.

Three-position enabling device

CAUTION

The person using the three-position enabling device is responsible to observe the safeguarded space for hazards due to robot motion and any other hazards related to the robot.

The three-position enabling device is located on the FlexPendant. When continuously held in center-enabled position, the three-position enabling device will permit robot motion and any hazards controlled by the robot. Release of or compression past the center-enabled position will stop the robot motion.

CAUTION

For safe use of the three-position enabling device, the following must be implemented:

- The three-position enabling device must never be rendered inoperational in any way.
- If there is a need to enter safeguarded space, always bring the FlexPendant. This is to enforce single point of control.

CAUTION

On the IRB 14050, the three-position enabling device is not active unless a valid SafeMove configuration is active in the controller.

Note

To enforce single-point of control from the FlexPendant, press and release the three-position enabling device twice.

Continues on next page

Note

YuMi robots with SafeMove requires using the enabling device.

On YuMi robots without SafeMove the enabling device is disabled, hence, not used.

Thumb button

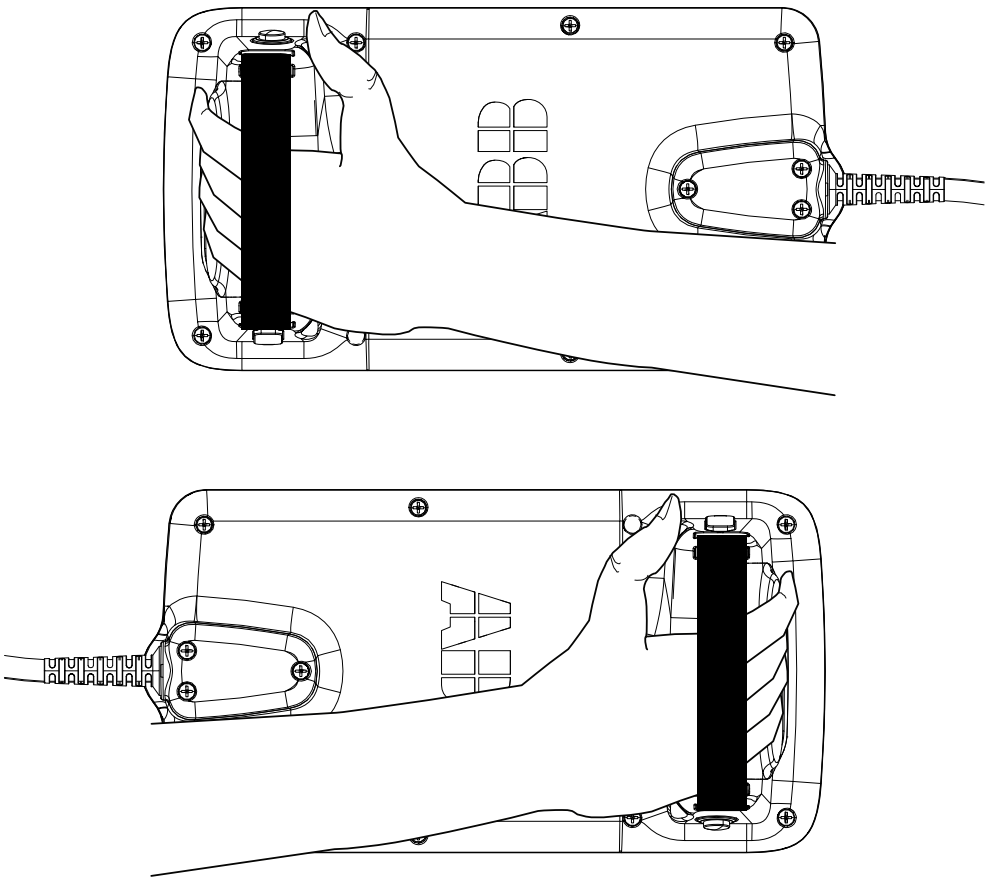
For robots used in collaborative application, the thumb button is used to enable the lead-through functionality.

For robots supporting the mode manual full speed, the button is used as hold-to-run.

How to hold the FlexPendant

FlexPendant is typically operated while being held in the hand. The right-handed users use their left-hand to support the FlexPendant while their right-hand performs the operations on the touch screen. However, the left-handed users can easily adapt FlexPendant for their use.

For more details, see *Operating manual - OmniCore*.



xx1800000045

1.4 RobotStudio

Overview of RobotStudio

RobotStudio is an engineering tool for the configuration and programming of ABB robots, both real robots on the shop floor and virtual robots in a PC. To achieve true offline programming, RobotStudio utilizes ABB VirtualRobot™ Technology.

RobotStudio has adopted the Microsoft Office Fluent User Interface. The Office Fluent UI is also used in Microsoft Office. As in Office, the features of RobotStudio are designed in a workflow-oriented way.

With add-ins, RobotStudio can be extended and customized to suit the specific needs. Add-ins are developed using the RobotStudio SDK. With the SDK, it is also possible to develop custom SmartComponents which exceed the functionality provided by RobotStudio's base components.

For more information, see *Operating manual - RobotStudio*.

RobotStudio for real controllers

RobotStudio allows, for example, the following operations when connected to a real controller:

- Controller.Software.isRobotWare6Installing and modifying RobotWare systems on controllers, using the **Modify Installation** function.
- Text-based programing and editing, using the **RAPID Editor**.
- File manager for the controller.
- Administrating the User Authorization System.
- Configuring system parameters.

1.5 FlexPendant applications

The FlexPendant applications

The FlexPendant contains applications for controlling the robot. There are different application packages depending on the options selected for the robot. The *Limited App Package* is always included, unless another app package is selected.

There are more applications available than those listed below. These can be specific for the selected products and options, for example, application software, or applications for controlling grippers and tools.

Code

The **Code** application is used to create new programs, modify existing programs, and so on.

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
Create new programs, edit existing programs			✓
View and edit RAPID modules and RAPID routines			✓
Debug Options PP to main, cursor to program pointer, goto position, call routine, cancel routine, check program, view system data, next move instruction			✓
Teach position (ModPos)			✓
Check for syntactic and semantic error			✓

If the option *Program Package* is not selected then programs must be created and edited using RobotStudio.

Program Data

The **Program Data** application is used to view and edit RAPID data.

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
View and edit RAPID data (program data)			✓
Manage payload data	✓	✓	
Manage tool data	✓	✓	
Manage work object data	✓	✓	

Jog

The **Jog** application is used to jog the ABB industrial robot using an intuitive touch based user interface or using a joystick.

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
Joystick jog	✓	✓	

Continues on next page

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
Touch jog		✓	
Align tool		✓	
Lead-through	✓[1]	✓[1]	
Jog supervision	✓	✓	
GoTo (jog to target)		✓	
3D visualization	✓	✓	

Settings

The **Settings** application is used to configure the general settings of OmniCore controller and FlexPendant. Controller configuration includes Network, ABB Connected Services, Time and Language, Backup, Restore, System diagnostics and so on. FlexPendant configuration includes background settings and programmable keys.

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
System About, hardware devices, software resources	✓	✓	
Network Status, WAN settings, DNS Client	✓	✓	
ABB Connected Services Status, Connected Services status, configure 3G/WiFi/wired Status, Connected Services status, configure 4G/3G/Wi-Fi/wired	✓	✓	
Configure Connected Services	✓	✓	
Backup and Recovery Backup, restore, system diagnostics, restart, reset user data, RobotWare Installation Utilities	✓	✓	
Date & time	✓	✓	
Region & language	✓	✓	
Programmable keys	✓	✓	

I/O

The **I/O** application is used to manage the I/O signals. Signals are configured with system parameters.

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
Show industrial networks	✓	✓	
View all I/O signals	✓	✓	

1 Only applicable for compatible manipulators, currently IRB 14050 and CRB 15000.

Continues on next page

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
Display I/O signals with respect to category	✓	✓	
Filter signals	✓	✓	
Sort signals	✓	✓	
Set signals	✓	✓	
Bit values	✓	✓	
Navigate to device specific signals	✓	✓	
Identify device	✓	✓	
Scan EDS	✓	✓	
Activate and deactivate devices	✓	✓	
Start	✓	✓	
Scan	✓	✓	
Firmware upgrade	✓	✓	

Operate

The **Operate** application is used to view the program code while the program is running. Controller data can be configured for viewing the data in the form of dashboards. Updates during production are shown here.

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
View dashboards		✓	
Configure dashboards		✓	
Load and execute RAPID programs	✓	✓	
View loaded RAPID programs	✓	✓	
Teach position (ModPos) of robtargets in loaded RAPID programs	✓	✓	
Reset program pointer to Main	✓	✓	
Show program pointer position	✓	✓	
Show motion pointer position	✓	✓	
Execute service routines	✓	✓	

Calibrate

The **Calibrate** application is used for calibration and definition of frames for ABB robots.

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
Mechanical unit calibration	✓	✓	
Update revolution counters	✓	✓	
Edit motor offset values	✓	✓	
Load motor offset values	✓	✓	

Continues on next page

Feature	*Limited App Package* [3120-1]	*Essential App Package* [3120-2]	*Program Package* [3151-1]
Fine calibration	✓	✓	
Robot memory	✓	✓	
Base frame calibration	✓	✓	
Execute calibration specific service routines	✓	✓	

File Explorer

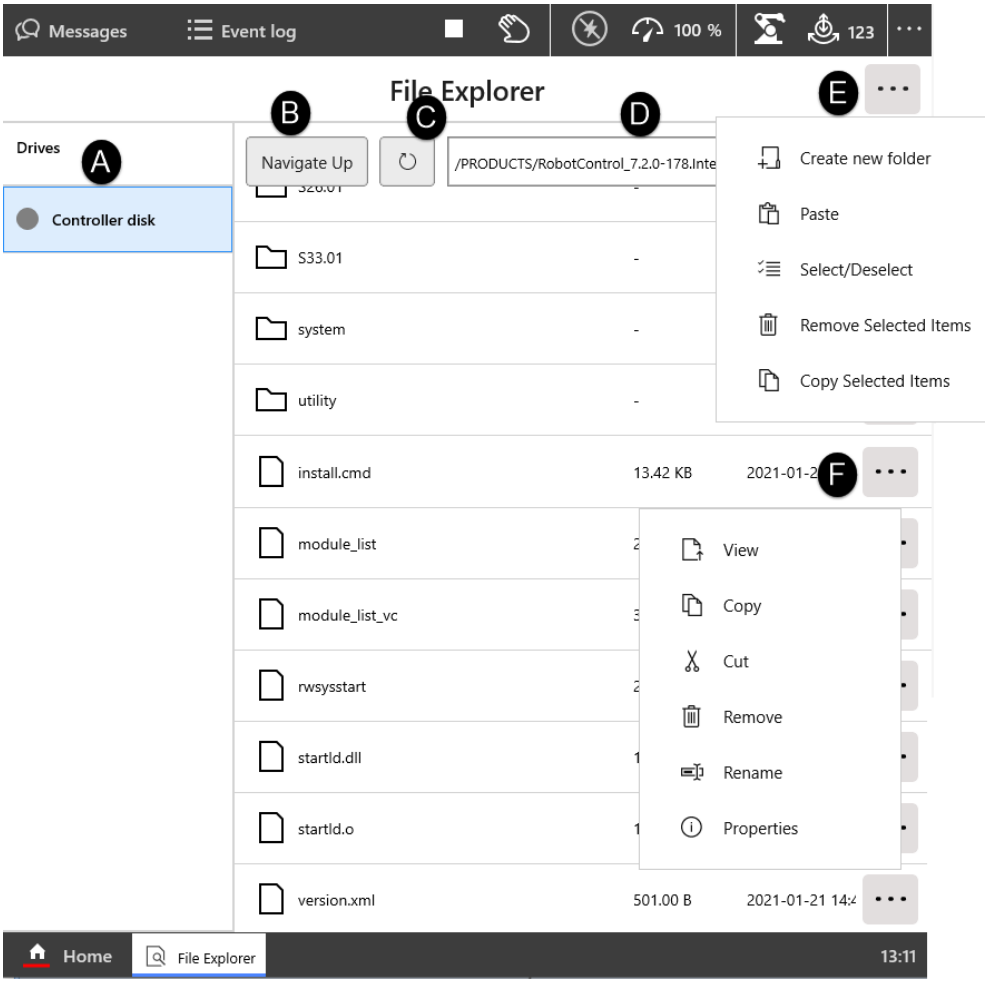
The File Explorer is a file manager, similar to Windows Explorer, with which you can view, rename, delete, or move files and folders on the controller or on a connected external USB drive.

Note

The file explorer supports operations on the following file formats: TXT, CFG, PNG, XML, ZIP, JPG, MOD, PGF, LOG, and MODX.

Continues on next page

To manage files and folders, from the Home screen, open **File Explorer**. The file explorer window is displayed. The following image and table provides information regarding the functions available in the file explorer window.



xx2100000050

Label	Description
A	Displays the available drives. If a USB drive is connected to the FlexPendant that is also displayed here.
B	Navigates to the folder up by one level.
C	Refreshes the files and folders.
D	Displays the path of the selected folder.
E	Displays the options available for a selected folder. • **Create new folder**: Creates a new folder in the selected folder. • **Paste**: Pastes the copied files or folders in the selected folder. • **Select/Deselect**: Selects or clear the selection for a set of files and folders. • **Remove Selected Items**: Removes the selected items. • **Copy Selected Items**: Copies the selected items.

Continues on next page

Label	Description
F	Displays the options available for a selected item. • **View**: Allows you to view the selected text or picture files. • **Copy**: Copies the selected item. • **Cut**: Cuts the selected item. • **Remove**: Deletes the selected item. • **Rename**: Changes the name of the selected item. • **Properties**: Displays the properties of the selected item.

Note

The following grants are required for full access to controller disk:

- **Read access to controller disks**
- **Write access to controller disks**

Without the **Read and Write access to controller disks** grant you may get access to some folders in controller disk like `/TEMP` but not all of them.

While moving the file and folders following are the possible scenarios:

- Moving files and folders within the controller disk.
- Moving files and folders from controller to USB drive and vice versa.

Note

It is not possible to move or copy files and folders within a USB drive.

SafeMove

The application **SafeMove** is used to configure some parts of SafeMove. See *Application manual - Functional safety and SafeMove*. For full SafeMove configuration, see *Visual SafeMove* in RobotStudio.

1.6 Directory structure on OmniCore

Default base directory structure

The default base public directory structure on the OmniCore controller consists of the following directories:

Directory	Description
HOME	Intended for user files and for use by RAPID programs. Data stored in the *HOME* directory is included in the Backup and restore function. **Note** Add-in data should not be stored in this directory. The *ADDINDATA* directory is used for this purpose instead.
DATA	Data stored in the *DATA* directory is not included in the Backup and restore function.
ADDINDATA	The *ADDINDATA* directory contains a number of sub-directories used for the RobotWare add-ins. See *Application manual - RobotWare add-ins* for detailed information about the add-in directories.
TEMP	The *TEMP* directory is for the storage of temporary files. **CAUTION** The *TEMP* directory is cleaned on system reset and during system updates.
BACKUP	The *BACKUP* directory is used for for saving backups.
RAMDISK	The *RAMDISK* directory, located in the RAM disk (nonpersistent), is used for high-performance logging. **CAUTION** Content is lost on each restart of the controller.

To make sure that all RAPID programs and URL-s in the robot web-service API work properly, make sure that you only use the locations specified above (using the predefined environment variables).

Note

RobotWare add-ins can add additional directories and files to the default base directory structure of your controller, so the actual default structure on your controller may have additional files and directories.

1.7 Ethernet networks on OmniCore

Network segment overview

The Ethernet networks used by OmniCore are distributed into the following segments:

Network segment	C30	C90XT V250XT Type A	V250XT Type B V400XT	E10	Usage
Private Network	I/O (Scalable I/O) ETHERNET SWITCH	I/O (Scalable I/O) ETHERNET SWITCH	DEV	DEVICE	Process equipment local to this specific robot.
	MGMT (Management)	MGMT (Management)	MGMT (Management)	MGMT (Management)	ABB service personnel.
	HMI (FlexPendant)	HMI (FlexPendant)	HMI (FlexPendant)	HMI (FlexPendant)	FlexPendant connection.
ABB Connect Network	ABB Connect	ABB Connect	ABB Connect	WAN 2	ABB Connected Services connection.
Public Network	WAN	WAN	WAN 1	WAN 1	Public/factory network.
			WAN 2		
I/O Network	LAN	LAN3	LAN	-	Secondary public/factory network. Isolated from WAN.

Note

For information regarding location of the Ethernet port connectors, see the Product manual for the respective OmniCore controller.

This page is intentionally left blank

2 Get started

Main steps for installing and configuring OmniCore

The following table shows an example of the installation and configuration steps that can be taken if you want to get started with OmniCore and have access to the Public Network.

	Action	Information
1	Install the robot equipment.	Mechanical installation and electrical connections between manipulator and controller is described in the *Product manual* of the robot and controller respectively.
2	Make sure the safety circuits of the system are properly connected to the robot cell or have jumper connections installed (if required).	How to connect the safety circuits is detailed in the product manual for the robot controller.
3	Connect the FlexPendant to the controller.	The FlexPendant and its major parts and functions are detailed in section *The FlexPendant on page 22* How to connect the FlexPendant to the controller is detailed in section *Detaching and attaching a FlexPendant on page 41*.
4	Switch the power on.	Use the main switch on the controller.
5	Install RobotStudio on a PC.	Proceed as detailed in *Operating manual - RobotStudio*. RobotStudio is used to create a system to run on the controller, but at this point (prior to the first start) a system is already installed by the manufacturer.
6	Connect the controller to a PC (through the management port).	Proceed as detailed in section *Ethernet networks on OmniCore on page 35*.
7	Start RobotStudio on the PC.	Proceed as detailed in *Operating manual - RobotStudio*.
8	Install RobotWare system (if required).	Proceed as detailed in section *RobotWare installation procedures on page 169*.
9	Configure the firewall settings in the Firewall Manager.	Proceed as detailed in section *Configuring firewall settings on page 51*. **Note** The first time you configure the firewall settings, you must be connected through the MGMT port (Private network). **Note** All default services and application protocols, except DHCP client, are disabled by default on the Public Network. For information about what protocols are necessary in different scenarios, see *Configuring firewall settings on page 51*.
10	Define IP adress for the Public Network.	Proceed as detailed in section *Public Network on page 47*.

Continues on next page

Continued

	Action	Information
11	Allow connection to controller from RobotStudio on Public Network through WAN port.	Must be enabled on the FlexPendant, Public Network settings, see *Public Network on page 47*.
12	Restart the controller.	
13	The robot system is now ready for configuration and programming.	

Continues on next page

2.1 System start OmniCore

Prerequisites before start

This procedure details the main steps required to start the system when the power has been switched off.

All information is based on the assumption that *working system software has already been installed* on the robot controller, as the case would be at first start directly after delivery.

Note that there may be more information available than the one referred to in the procedure.

System start

This procedure details all required steps to start the system for the first time. For everyday start, step 4 is normally the only required step.

	Action	Information
1	Install the robot equipment.	Mechanical installation and electrical connections between manipulator and controller is described in the *Product manual* of the robot and controller respectively.
2	Make sure the safety circuits of the system are properly connected to the robot cell or have jumper connections installed (if required).	How to connect the safety circuits is detailed in the product manual for the robot controller.
3	Connect the FlexPendant to the controller.	The FlexPendant and its major parts and functions are detailed in section *The FlexPendant on page 22* How to connect the FlexPendant to the controller is detailed in section *Detaching and attaching a FlexPendant on page 41*.
4	Switch the power on.	Use the main switch on the controller.
5	If the controller or manipulator have been replaced with spare parts, make sure the calibration values, revolution counters and serial numbers are updated correctly.	Normally, only the revolution counters require updating, which is to be performed as detailed in the product manual for the robot. If required, transfer the calibration data from the serial measurement board as detailed in *Robot memory on page 226* for systems *without* the Absolute Accuracy option. If required, enter the calibration data as detailed in *Update calibration data using the FlexPendant on page 222* for systems *with* the Absolute Accuracy option.
6	Install RobotStudio on a PC.	Proceed as detailed in *Operating manual - RobotStudio*. RobotStudio is used to create a system to run on the controller, but at this point (prior to the first start) a system is already installed by the manufacturer.
7	Connect the controller to a PC (through the management port) or to the network (if used).	Proceed as detailed in section *Ethernet networks on OmniCore on page 35*.
8	Start RobotStudio on the PC.	Proceed as detailed in *Operating manual - RobotStudio*.

Continues on next page

	Action	Information
9	Restart the controller.	
10	The robot system is now ready for configuration and programming.	

2.2 Detaching and attaching a FlexPendant

Introduction

With the option *Hot swappable FlexPendant [3018-1]* it is possible to detach and attach the FlexPendant from an OmniCore controller in automatic mode, without interrupting the ongoing process.

Detaching the FlexPendant in manual mode will always result in an emergency stop.

Note

Detaching the FlexPendant is possible only if the logged in user has the **Detach the FlexPendant** grant.

CAUTION

Before detaching the FlexPendant, another emergency stop shall be available.

How to configure emergency stops is described in the product manual for the controller, see *References on page 10*.

CAUTION

With a detached FlexPendant, there is no visual identification of the operating mode.

CAUTION

A FlexPendant that is not connected to the robot must be stored out of sight so that it cannot be mistaken for being in use.

CAUTION

The FlexPendant connector shall only be used to connect the FlexPendant.

For details, see the product manual for the respective robot controller.

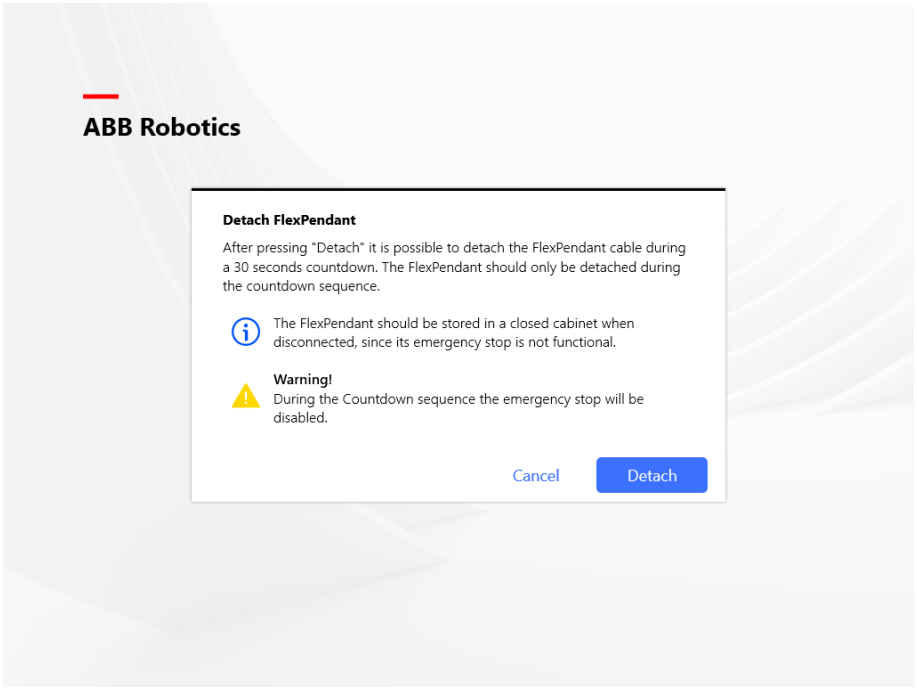
Detaching the FlexPendant in automatic mode

Use the following procedure to detach the FlexPendant in automatic mode:

1. On the status bar, tap the **QuickSet** button.
2. Tap the **Logout/Restart** tab.
3. In the **FlexPendant** section, tap **Detach FlexPendant**.

Continues on next page

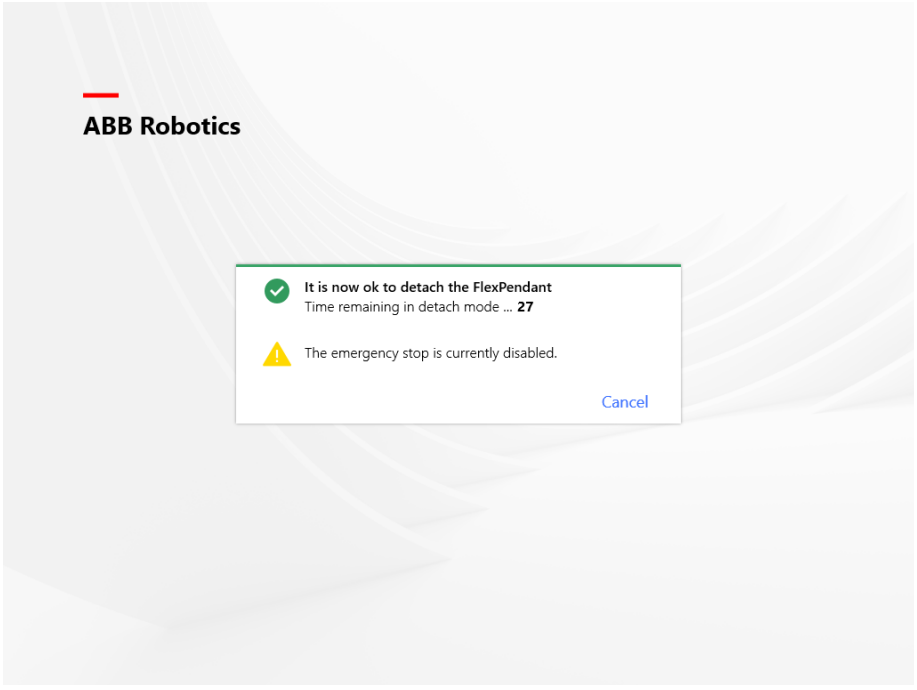
The **Detach FlexPendant** window is displayed.



xx1900000403

4 Tap **Detach**.

A popup window with 30 seconds countdown timer is displayed.



xx1900000404

5 When the countdown is progressing, detach the FlexPendant.

Continues on next page

When detached, the FlexPendant will shut down.

Note

If the FlexPendant is not detached within 30 seconds, the process for detach of the FlexPendant is aborted.

WARNING

If the FlexPendant is detached after the 30 seconds countdown has passed, the controller will enter emergency stop state.

Attaching the FlexPendant

CAUTION

Always inspect the connector for dirt or damage before attaching. Clean or replace any damaged parts.

Attach the connector to the controller and tighten the locking ring or screws.

CAUTION

Make sure that the emergency stop device is not pressed in before attaching the FlexPendant.

This page is intentionally left blank

3 Configuration

3.1 Configuring networks

Connection of industrial networks

Factory wide I/O network

A factory wide I/O network should be connected to the WAN/WAN1 port on the controller, or to the LAN/LAN3 port if the I/O network needs to be isolated from the network already connected to WAN/WAN1.

> **Note**
>
> For OmniCore E10:
>
> A factory wide I/O network should be connected to the WAN1 port on the controller.

Connect multiple ports

> **Note**
>
> It is not supported to connect multiple ports of the OmniCore controller to the same external switch unless static VLAN isolation is applied on the external switch.

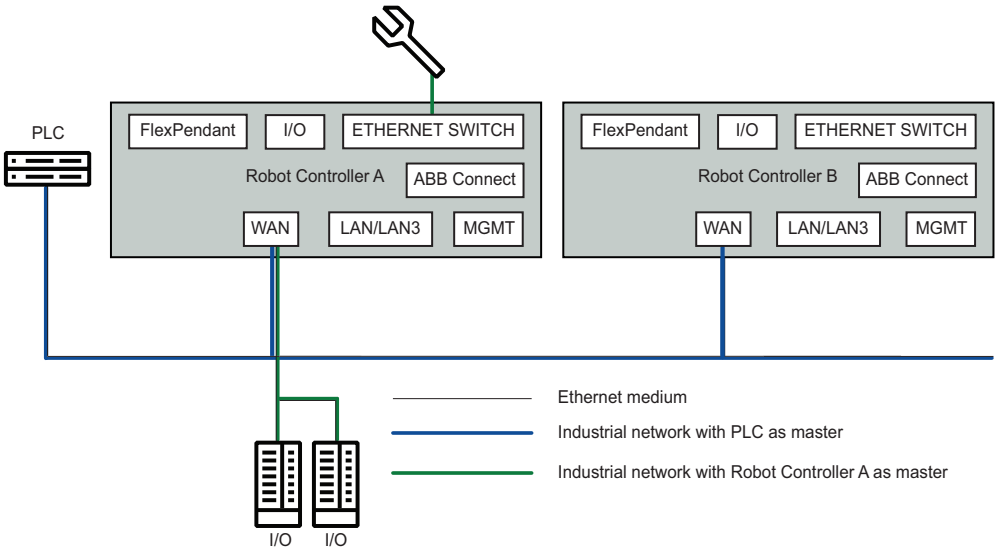
Private Network segments of multiple controllers

> **Note**
>
> Private Network segments of multiple controllers cannot be connected to each other.

Continues on next page

Combined industrial networks

There are many possible solutions of combined industrial networks. For example, the robot controller can be both master and slave on the Public Network as well as master on the Private Network.



xx2100002085

This can be implemented by setting up an EtherNet/IP adapter on the Public Network and an EtherNet/IP scanner targeting both the Public and Private Networks at the same time. The traffic for both industrial networks on the Public Network can share the ethernet medium with each other and with other non-industrial network traffic.

Reserved IP addresses

Before configuring IP addresses manually, it is important to be aware that some ranges are reserved by the robot controller. Configuring units or networks that collide with these reserved addresses will most likely cause network problems.

The following IP address ranges are allocated for internal functions on the controller:

- 192.168.125.0 - 255
- 192.168.126.0 - 255
- 192.168.127.0 - 255

No other robot controller connected network can be on a subnet that overlaps with any of the above reserved IP addresses. If a subnet mask in the class B range has to be used, then a private address of class B must be used to avoid any overlapping. Contact your local network administrator regarding network overlapping. See section "Communication" in *Technical reference manual - System parameters*.

Note

It is not recommended using leading zeros in dot-decimal notation of IP addresses. The numbers may wrongly be interpreted as octal numbers. Different behaviors on virtual and real controllers may be experienced.

Continues on next page

Private Network

The Private Network has a static configuration with IP address 192.168.125.1/24 and hosts a DHCP server. The purpose of the Private Network is to connect the computers within the robot controller as well as I/O networks and process equipment local to the robot. Many IP addresses are reserved on this network, so it is recommended that new units get their IP address from the DHCP server.

Note

Never connect another DHCP server to any of the ports connected to the Private Network. There cannot be two DHCP servers on the same network. It might cause an erroneous behavior of both internal and external units.

ABB Connect Network

Configuration of the IP settings for the ABB Connect Network shall be done manually when the controller is equipped with a wired Connected Services Gateway (DSQC1041) that is connected to an Internet gateway, using an IP address provided by the network administrator.

Note

On OmniCore E line controllers, the WAN2 port is equivalent to a wired Connected Services Gateway (DSQC1041).

The IP address configuration is done either in RobotStudio or on the FlexPendant.

See *Connected Services configuration on page 54* for instructions.

As an exception to the reserved ranges mentioned in section *Reserved IP addresses on page 46*, the ABB Connect Network is allowed to configure IP addresses within the reserved range 192.168.126.0 - 255, but not within 192.168.125.0 - 255 or 192.168.127.0 - 255.

For security reasons, only outbound access on port 53 DNS and 443 HTTPS are allowed. The inbound access is blocked by an internal firewall and cannot be unblocked.

Note

For more information about ABB Connect Network and internet connection, see *Application manual - Controller software OmniCore*.

Public Network

The Public Network interface is typically connected to the factory network with a public IP address provided by the network administrator. The Public Network segment can be used for:

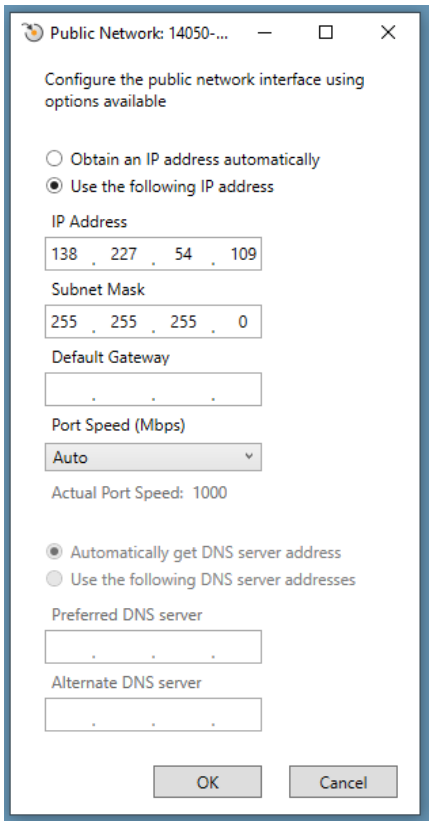
- Connecting a PC running RobotStudio. For more information see *Operating manual - RobotStudio*.
- Mounting FTP or NFS disks from the controller.
- Running Industrial Ethernet protocols.

Continues on next page

Most protocols are disabled by default in the controller firewall. See *Configuring firewall settings on page 52* for information about how to enable these protocols.

The Public Network can be configured with a fixed IP address, or as a DHCP client, either in RobotStudio or from the FlexPendant:

- **Defining network settings in RobotStudio:**
  1. In the **Configuration** browser, right-click the controller and select **Properties\Network Settings** and then **Public Network**.
  2. In the **Public Network** window, configure the network interface and click **OK**.

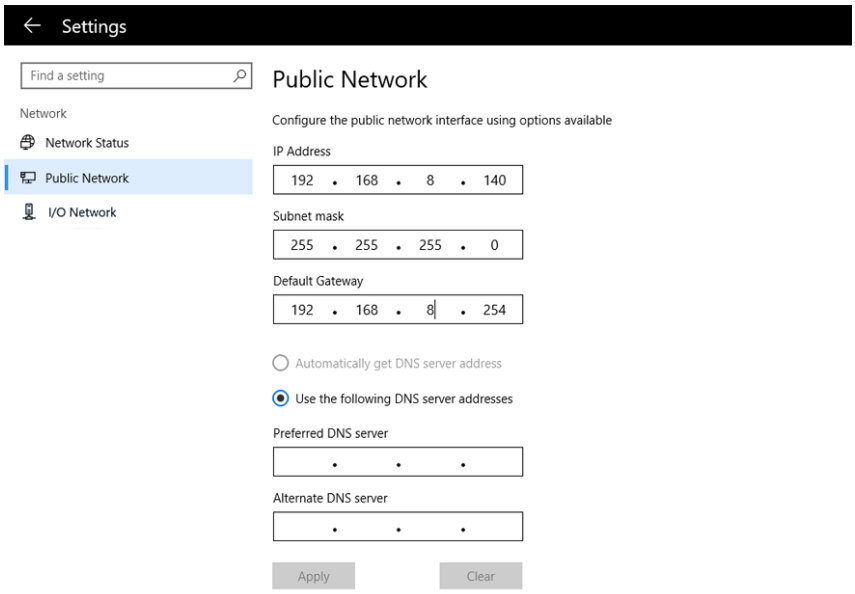


xx2000000489

- **Defining network settings on the FlexPendant:**
  1. On the start screen, tap **Settings**, and then select **Network** from the menu.

Continues on next page

2 Select **Public Network** and configure the network interface. Tap **Apply**.



xx2100002086

Parameters available in **Public Network**

The following table provides information about the parameters available in **Public Network**:

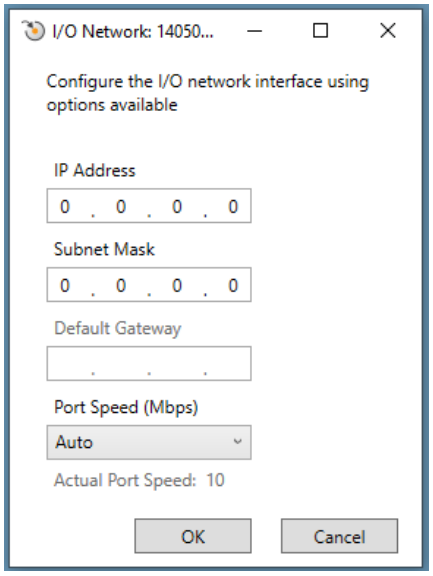
Parameters	Description
Obtain an IP address automatically	The controller will receive an IP address automatically from the DHCP server.
Use the following IP address	Enables manual update of **IP Address**, **Subnet Mask**, and **Default Gateway**.
IP Address	Enter the IP address.
Subnet Mask	Enter the subnet mask.
Default Gateway	Enter the default gateway.
Automatically get DNS server address	The controller will receive a DNS IP address automatically from the DHCP server.
Use the following DNS server address	Enables manual update of **Preferred DNS Server** and **Alternate DNS Server**.
Preferred DNS Server	Type the IP address of the preferred DNS server on the Public Network.
Alternate DNS Server	Type the IP address of the alternative DNS server on the Public Network.
Allow connection to Controller from RobotStudio on public network through WAN port	Enables or disables the connection to controller from RobotStudio on a public network.

Continues on next page

I/O Network

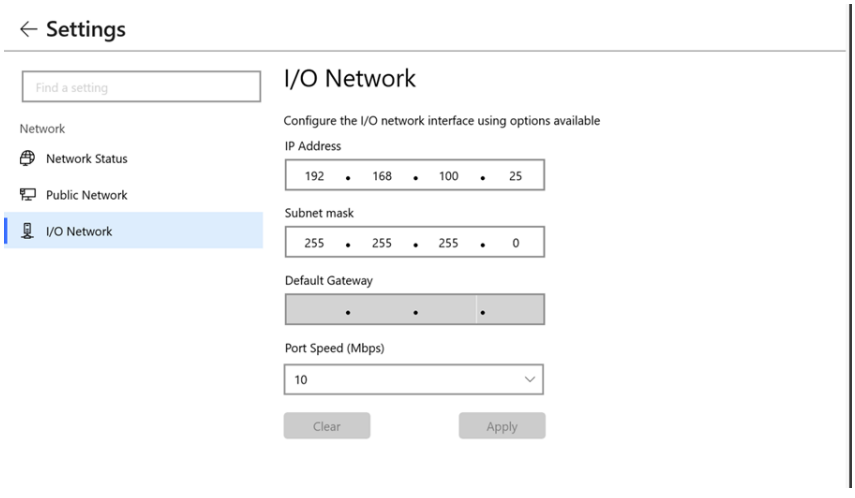
The I/O Network is needed when an Industrial Ethernet network must be isolated from the Public Network. It can be configured either in RobotStudio or from the FlexPendant:

- **Defining network settings in RobotStudio:**
  1. In the **Configuration** browser, right-click the controller and select **Properties\Network Settings** and then **I/O Network**.
  2. In the **I/O Network** window, configure the network interface and click **OK**.



xx2200000389

- **Defining network settings on the FlexPendant:**
  1. On the start screen, tap **Settings**, and then select **Network** from the menu.
  2. Select **I/O Network** and configure the network interface. Tap **Apply**.



xx2100002097

3.2 Configuring firewall settings

Overview

The Firewall Manager is used to configure pre-registered network services on the controller by enabling or disabling them.

Note

All default services and application protocols, except DHCP client, are disabled by default on the Public Network. All communication via the Public Network must be manually enabled.

Note

Only users with UAS grant **Modify network security properties** can modify the firewall settings.

Note

Only pre-registered Network Services can be configured in the Firewall Manager settings. The user cannot add new Network Services in the Firewall Manager and can only change the parameters **Enable on Public Network**, **Enable on Private Network** and **Enable on I/O Network** for pre-registered Network Services.

Default configuration

The following table contains default values for pre-registered network services. These services will always be shown. The list might contain more services depending on which options are installed in the system.

Network Service (pre-registered)	Enable on Public Network	Enable on Private Network	Enable on I/O Network
Bonjour	No	No	No
ConnectedServices	No	Yes	No
DHCP_Client	Yes	No	No
EtherNetIP	No	Yes	No
Netscan	No	Yes[i]	No
RapidSockets	No	No	No
RobAPI	No	Yes[i]	No
RobICI	No	Yes	Yes
RobotWebServices	No	Yes[i]	No
syslog	No	Yes	No
OpcUaServer	No	Yes	No
UDPUC	No	Yes	No

[i] This value cannot be changed. It must be enabled on the private network for connection of RobotStudio and FlexPendant to the controller.

Continues on next page

Note

All default services and application protocols, except DHCP client, are disabled by default on the Public Network. All communication via the Public Network must be manually enabled.

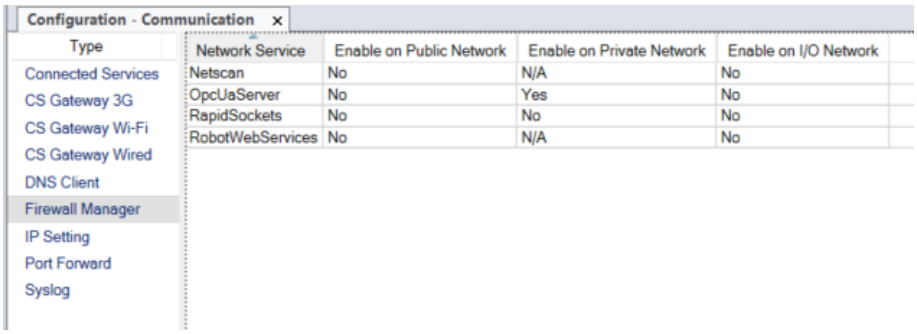
Tip

In order to be able to connect to the Public network and perform basic RAPID tasks, the following must be enabled in the firewall:

- Bonjour
- Netscan
- RobAPI
- RobotWebServices

Configuring firewall settings

1. In RobotStudio, select **Add Controller < Connect to Controller**.
2. Select **Request Write Access**.
3. Select **Configuration < Communication**.
4. Select **Firewall Manager**. The following view is displayed:



xx1900000352

5. Select **YES** or **NO** for each network service, indicating if they should be enabled or disabled on the respective network.

3.3 Port forwarding configuration

Port forwarding

The port forwarding configuration can be used to reach a server/device located on a different network in the controller. One example is having a built-in webserver on a device on the I/O network that needs to be reached from a Private or Public network.

This functionality uses Network Address Translation (NAT) and will automatically opens the associated port in the controller firewall. For the communication to work in both ways, the server/device must have a default gateway set that points back to the controller. For instance, a server/device on the private network needs to have a default gateway set to 192.168.125.1 which is the controller address on that network.

The following forwarding is possible:

- Public Network -> Private Network
- Public Network -> I/O Network
- Private Network -> I/O Network

Note

Since network traffic will flow through the controller, high bandwidth applications could negatively affect the controller performance.

The port forwarding is configured through system parameters, Communication topic. See *Technical reference manual - System parameters* for detailed information.

3.4 Connected Services configuration

3.4.1 Introduction

ABB Connect

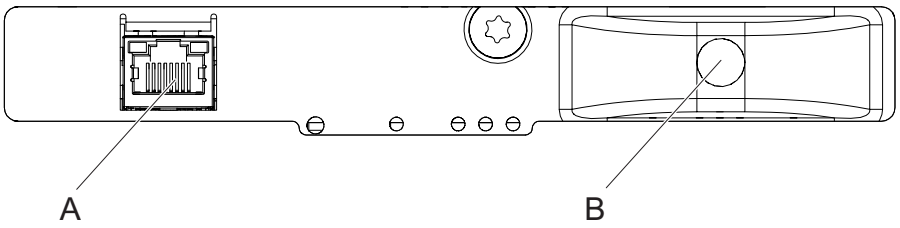
> **Note**
>
> ABB Connected Services is the new name for the functionality previously known as ABB Ability. During a period of time, both names will appear in and on our products.

Overview

The functionality Connected Services enables the OmniCore controller to send data to ABB Connect cloud solution through a Connected Services Gateway module or directly through the public port.

Ports on 3G gateway

The following illustration shows the ports in the Connected Services 3G gateway module.



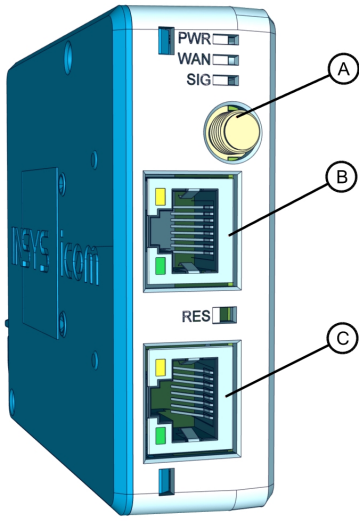
xx1800002955

A	Ethernet port - Available only for wired module.
B	Antenna connection point - Available only for 3G/Wi-Fi module.

Continues on next page

Ports on 4G gateway

The following illustration shows the ports in the Connected Services 4G gateway module.



xx2300000614

A	Cellular antenna (SMA socket)
B	ETH1 - Ethernet port 1
C	ETH2 - Ethernet port 2

Note

It is important to install the antenna when you use Connected Services gateway Wi-Fi/3G.

Note

If the Connected Services Gateway is not used, the public port (WAN/WAN1) is used.

Note

There is no Connected Services gateway on the E10 controller. Hence, the ABB Connect port or the public port (WAN1) is used with an external gateway for Connected Services.

Note

The Connected Services Gateway Wired is not required for C90XT, V250XT, or V400XT. The connection can be done directly on the ABB Connect port of the main computer.

Continues on next page

3.4.1 Introduction
Continued

For information regarding the location of Connected Services gateway module in the OmniCore controller, see the Product manual for the respective OmniCore controller.

Gateway types

The following Connected Services Gateway types are available:

- Connected Services Gateway 3G - DSQC1039
- Connected Services Gateway Wi-Fi - DSQC1040
- Connected Services Gateway Wired - DSQC1041
- Connected Services Gateway 4G - EU - DSQC1093
- Connected Services Gateway 4G - US - DSQC1093A

Note

The robot controller will automatically detect 3G or Wi-Fi Connected Services gateway type.

An IP address needs to be defined in the Connected Services Gateway Wired configuration, for the Connected Services Gateway Wired to be detected. In this case, the detection will be set to Wired, if the 3G or Wi-Fi are not automatically detected.

For more details, see *Configuring Connected Services Gateway Wired - DSQC1041 on page 60*.

Note

The robot controller will not automatically detect the 4G Connected Services Gateway. This module will be managed as an external wired connection. It is then required to configure it as a Connected Services Gateway Wired.

Note

If the 4G gateway option is pre-installed, the configuration is already implemented.

3.4.2 Configuring Connected Services Gateway using FlexPendant

Overview

The modules must be configured based on the module type (3G/4G/Wi-Fi/Wired).

> **Note**
>
> The configuration page is visible in FlexPendant only if the Connected Services Gateway has been detected. If the configuration page is not visible, use RoboStudio to configure Connected Services Gateway. For more details, see *Configuring Connected Services Gateway using Robotstudio on page 64*.

Configuring Connected Services Gateway 3G - DSQC1039

If the connected module is 3G, it is enabled by default.

> **Note**
>
> By default the module is installed with ABB SIM card using the default values and it will connect to the network. If the module is not using ABB SIM card then you have to configure the settings.

Use the following procedure to configure Connected Services Gateway 3G:

1. On the start screen, tap **Settings**.
2. Tap **ABB Connected Services** > **3G Connection**.

   The **3G Connection** parameters with default values are displayed.
3. Configure the parameters according to the mobile network settings.

   For details about the parameters available in **3G Connection**, see *Parameters available in 3G Connection on page 58*.
4. Tap **Apply**.

   The Connected Services Gateway 3G is configured and a confirmation window is displayed.
5. Tap **Yes**.

   The controller is restarted and changes are applied.
6. Verify the connection information of Connected Services Gateway 3G on **Settings** > **ABB Connected Services** > **Connectivity Status** > **Module connection**.

   If the module is connected to the internet the status will show as connected. And it receives an IP address, Gateway, and DNS from the network provider.

   For more details on module information, refer *Connected Services Gateway 3G pages on page 68*.

Continues on next page

3.4.2 Configuring Connected Services Gateway using FlexPendant
Continued

Parameters available in **3G Connection**

The following table provides information about the parameters available in **3G Connection**:

	Parameters	Description
Basic **Note** This section gives information about basic configuration parameters.	**Automatic Configuration of Cellular 3G connection**	**Yes**: Loads the default configuration values. **No**: Enables the field to edit the configuration values.
	Enable 3G connection	Enables or disables the Connected Services Gateway 3G module. The available values are **Yes** or **No**.
	Roaming	Enables or disables the roaming. By default, roaming is enabled in ABB SIM. The available values are **Yes** or **No**.
	Access Point/APN	Name of the Access point to connect to the mobile network. The default Access point is *abbrobotics.com* for ABB SIM cards.
	User/Password	Username/password of Access point name. Leave this field blank if there is no username/password.
Advanced **Note** This section gives information about advanced configuration parameters.	**Pin**	If the SIM is secured with a PIN, type the PIN number in this field. By default, ABB SIM card has no PIN.
	Scan for Operators	Enabling the **Scan for Operators** toggle button displays the **Scan** button next to the **Operator** field.
	Operator	Type the operator ID to force connection to a specific operator. Leave this field blank for the auto-detection of the operator. If the **Scan for Operators** toggle button is enabled, tap on the **Scan** button to search on the network and display the available operators in the Operator list. You can then select an operator from the Operator list. The operator ID consists of the MCC and MNC. Concatenated to identify the operator, for example: operator ID 46001 (460 China, 01 first operator in China). • MCC: Mobile Country Code • MNC: Mobile Network Code
	Band	Select a specific network band. The available values are **Automatic**, **GSM**, and **UMTS**.
	Authentication	Select the authentication method. The available values are **Automatic**, **CHAP**, and **PAP**.
	Idle	Type the Idle time (in seconds) to specify the idle time required before hanging up the connection. **Note** This feature is not yet available.
	Delay	Type the duration (in seconds) between the time-out and retry. **Note** This feature is not yet available.

Continues on next page

Configuring Connected Services Gateway Wi-Fi - DSQC1040

Connected Services Gateway Wi-Fi is enabled by default but must be configured manually.

Use the following procedure to configure Connected Services Gateway Wi-Fi:

1. On the start screen, tap **Settings**.
2. Tap **ABB Connected Services** > **Wi-Fi Connection**.

   The **Wi-Fi Connection** parameters are displayed.
3. Enable and configure the parameters according to the Wi-Fi network settings.

   For details about the parameters available in **Wi-Fi Connection**, see *Parameters available in Wi-Fi Connection on page 59*.
4. Tap **Apply**.

   The Connected Services Gateway Wi-Fi is configured and a confirmation window is displayed.
5. Tap **Yes**.

   The controller is restarted and changes are applied.
6. Verify the connection information of Connected Services Gateway Wi-Fi on **Settings** > **ABB Connected Services** > **Connectivity Status** > **Module connection**.

   If the module is connected to the internet the status will show as connected. And it receives an IP address, Gateway, and DNS from the network provider.

   For more details on module information, refer *Connected Services Gateway Wi-Fi pages on page 70*.

Parameters available in **Wi-Fi Connection**

The following table provides information about the parameters available in **Wi-Fi Connection**:

Parameters	Description
Enable Wi-Fi Connection	Enables or disables the CS Gateway Wi-Fi module. The available values are **Yes** or **No**.
SSID	Type the SSID of the wireless network to which the module need to be connected.
Key	Type the security key of the SSID.
Security	Select the type of security required. The available values are: • Automatic • IEEE802.1X (not yet implemented) • None • WEP • WPA2-PSK

Continues on next page

Configuring Connected Services Gateway Wired - DSQC1041

The Connected Services Gateway Wired is enabled by default but must be configured manually.

> **Note**
>
> The Connected Services Gateway Wired is an extension of the ABB Connect port and not a real gateway.
>
> The configuration will define the ABB Connect network port IP, the mask, the external gateway and the DNS which must be used.

Use the following procedure to configure Connected Services Gateway Wired:

1. On the start screen, tap **Settings**.
2. Tap **ABB Connected Services** > **Wired Connection**.

   The **Wired Connection** parameters are displayed.
3. Enable and configure the parameters according to the Wired network settings.

   For details about the parameters available in **Wired Connection**, see *Parameters available in Wired Connection for Connected Services 4G Gateway on page 62*.
4. Tap **Apply**.

   The Connected Services Gateway Wired is configured and a confirmation window is displayed.
5. Tap **Yes**.

   The controller is restarted and the changes are applied.

> **Note**
>
> It is not possible to verify the internet connectivity directly in the wired connection.

Parameters available in **Wired Connection**

The following table provides information about the parameters available in **Wired Connection**:

Parameters	Description
State	Enables or disables the CS Gateway Wired module.
IP Address	Type the IP address of the ABB Connect port on the wired network. **Note** Before assigning the IP address for the wired module make sure the IP address has been assigned to your module by the network administrator.
Subnet Mask	Type the subnet mask of the ABB Connect port.
Default Gateway	Type the IP address of the external gateway.
Preferred DNS Server	Type the IP address of the external Primary DNS Server.

Continues on next page

Parameters	Description
Alternate DNS Server	Type the IP address of the external Secondary DNS Server, if available.

Note

The DNS can be set to blank if:

- the proxy is defined.
- the DNS resolution is done with the proxy.
- the proxy is provided as an IP address.

Configuring Connected Services Gateway 4G - DSQC1093 and DSQC1093A

The Connected Services Gateway 4G is enabled by default but must be configured manually like Connected Services Gateway Wired.

Note

The Connected Services Gateway 4G is connected to the ABB Connect port and act as an external gateway.

The configuration will define the ABB Connect network port IP, the mask, the external gateway and the DNS which must be used.

Note

If the 4G gateway option is pre-installed, the configuration is already implemented.

Use the following procedure to configure Connected Services Gateway 4G:

1. On the start screen, tap **Settings**.
2. Tap **ABB Connected Services** > **Wired Connection**.

   The **Wired Connection** parameters are displayed.
3. Enable and configure the parameters according to the Wired network settings.

   For details about the parameters available in **Wired Connection**, see *Parameters available in Wired Connection on page 60*.
4. Tap **Apply**.

   The Connected Services Gateway Wired is configured and a confirmation window is displayed.
5. Tap **Yes**.

   The controller is restarted and the changes are applied.

Note

It is not possible to verify the internet connectivity directly in 4G Gateway. For support, see the *Troubleshooting 4G Gateway* chapter in Controller Product Manual.

Continues on next page

3.4.2 Configuring Connected Services Gateway using FlexPendant
Continued

Parameters available in **Wired Connection** for Connected Services 4G Gateway

The following table provides information about the parameters available in **Wired Connection**:

Parameters	Description
State	Enables or disables the CS Gateway Wired module.
IP Address	Type the IP address of the ABB Connect port on the wired network. **Note** The IP address is 192.168.126.2
Subnet Mask	Type the subnet mask of the ABB Connect port. The subnet is 255.255.255.0
Default Gateway	Type the IP address of the external gateway. The IP address is 192.168.126.1
Preferred DNS Server	Type the IP address of the external Primary DNS Server. The DNS server is 192.168.126.1
Alternate DNS Server	Leave blank, not used yet.

Configuring without Connected Services Gateway

Connected Services can be used through the Public Port of the controller, if there is no Connected Services Gateway installed.

WARNING

The customer must ensure that a firewall is installed on its network to prevent unexpected external accesses if the Public port is connected to Internet.

Use the following procedure to configure the Public port:

1. On the start screen, tap **Settings**.
2. Tap **Network** > **Public Network**.
   The **Public Network** parameters are displayed.
3. Enable and configure the parameters according to the Public network settings.
   For details about the parameters available in **Public Network**, see *Parameters available in Public Network on page 63*.
4. Tap **Apply**.
   The Connected Services Gateway public is configured and a confirmation window is displayed.
5. Tap **Yes**.
   The controller is restarted and the changes are applied.

Continues on next page

Parameters available in **Public Network**

The following table provides information about the parameters available in **Public Network**:

Parameters	Description
Automatically get an IP address	The controller will receive an IP address automatically from DHCP server.
Use the following IP address	The parameters (IP Address, Subnet Mask, and Default Gateway) fields are enabled to update manually.
IP Address	Type the IP address of the ABB Connect port on the public network.
Subnet Mask	Type the Subnet Mask of the ABB Connect port network.
Default Gateway	Type the Default Mask of the ABB Connect port network.
Automatically get DNS server address	The controller will receive a DNS IP address automatically from DHCP server.
Use the following DNS server address	The parameters (Preferred DNS Server and Alternate DNS Server) fields are enabled to updated manually.
Preferred DNS Server	Type the IP address of the preferred DNS Server on the public network.
Alternate DNS Server	Type the IP address of the alternative DNS Server on the public network.
Allow connection to Controller from RobotStudio on public network	Enables or disables the connection to Controller from RobotStudio on a public network.

3.4.3 Configuring Connected Services Gateway using Robotstudio

Overview

The modules must be configured based on the module type (3G/Wi-Fi/Wired).

Configuring Connected Services Gateway 3G - DSQC1039

If the connected module is 3G, it is enabled by default.

> **Note**
>
> By default the module is installed with ABB SIM card using the default values and it will connect to the network. If the module is not using ABB SIM card then you have to configure the settings.

Use the following procedure to configure Connected Services Gateway 3G:

1. In the **Controller** tab add controller.

   The controller is added.
2. Click **Request Write Access**, to get write access to update the configuration of the controller.
3. Click **Configuration** > **Communication** > **CS Gateway 3G**.

   The **CS Gateway 3G** parameters are displayed.
4. Right-click on any parameter and select **Edit CS Gateway 3G**.

   The **Instance Editor** is displayed with default values.
5. Configure the parameters according to the mobile network settings.

   For details about the parameters available in **3G Connection**, see *Parameters available in 3G Connection on page 58*.
6. Click **OK**.

   The Connected Services Gateway 3G is configured.
7. Restart the controller to apply the changes.
8. Verify the connection information of Connected Services Gateway 3G on **Device Browser** > **Hardware devices** > **Controller** > **Connected Services Gateway** > **Module connection**.

   If the module is connected to the internet the status will show as connected. And it receives an IP address, Gateway, and DNS from the network provider.

   For more details on module information, refer *Connected Services Gateway 3G pages on page 68*.

Configuring Connected Services Gateway Wi-Fi - DSQC1040

Connected Services Gateway Wi-Fi is enabled by default and must be configured manually.

Use the following procedure to configure Connected Services Gateway Wi-Fi:

1. In the **Controller** tab add controller.

   The controller is added.

Continues on next page

2 Click **Request Write Access**, to get write access to update the configuration of the controller.
3 Click **Configuration** > **Communication** > **CS Gateway WiFi**.

   The **CS Gateway WiFi** parameters are displayed.
4 Right-click on any parameter and select **Edit CS Gateway WiFi**.

   The **Instance Editor** is displayed.
5 Enable and configure the parameters according to the Wi-Fi network settings.

   For details about the parameters available in **Wi-Fi Connection**, see *Parameters available in Wi-Fi Connection on page 59*.
6 Once the parameters are configured click **OK**.

   The Connected Services Gateway Wi-Fi is configured.
7 Restart the controller to apply the changes.
8 Verify the connection information of Connected Services Gateway Wi-Fi on **Device Browser** > **Hardware devices** > **Controller** > **Connected Services Gateway** > **Module connection**.

   If the module is connected to the internet the status will show as connected, and it will receive an IP address, Gateway, and DNS from the network provider.

   For more details on module information, refer *Connected Services Gateway Wi-Fi pages on page 70*.

Configuring Connected Services Gateway Wired - DSQC1041

The Connected Services Gateway Wired is enabled by default and must be configured manually.

> **Note**
>
> The Connected Services Gateway Wired is an extension of the ABB Connect port and not a real gateway.
>
> The configuration will define the ABB Connect IP and mask and the extended gateway and DNS which must be used.

Use the following procedure to configure Connected Services Gateway Wired:

1 In the **Controller** tab add controller.

   The controller is added.
2 Click **Request Write Access**, to get write access to update the configuration of the controller.
3 Click **Configuration** > **Communication** > **CS Gateway Wired**.

   The **CS Gateway Wired** parameters are displayed.
4 Right-click on any parameter and select **Edit CS Gateway Wired**.

   The **Instance Editor** is displayed.
5 Configure the parameters according to the wired network settings.

   For details about the parameters available in **Wired Connection**, see *Parameters available in Wired Connection on page 60*.

Continues on next page

6 Click **OK**.

The Connected Services Gateway wired is configured.

7 Restart the controller to apply the changes.

> **Note**
>
> It is not possible to verify the internet connectivity directly in wired connection.

Configuring Connected Services Gateway 4G - DSQC1093 and DSQC1093A

The Connected Services Gateway 4G is enabled by default and must be configured manually.

> **Note**
>
> The Connected Services Gateway 4G is connected to the ABB Connect port and act as an external gateway.
>
> The configuration will define the ABB Connect IP and mask and the extended gateway and DNS which must be used.

> **Note**
>
> If the 4G gateway option is pre-installed, the configuration is already implemented.

Use the following procedure to configure Connected Services Gateway 4G:

1 In the **Controller** tab add controller.

The controller is added.

2 Click **Request Write Access**, to get write access to update the configuration of the controller.

3 Click **Configuration** > **Communication** > **CS Gateway Wired**.

The **CS Gateway Wired** parameters are displayed.

4 Right-click on any parameter and select **Edit CS Gateway Wired**.

The **Instance Editor** is displayed.

5 Configure the parameters according to the wired network settings.

For details about the parameters available in **Wired Connection**, see *Parameters available in Wired Connection for Connected Services 4G Gateway on page 62*.

6 Click **OK**.

The Connected Services Gateway wired is configured.

7 Restart the controller to apply the changes.

> **Note**
>
> It is not possible to verify the internet connectivity directly in 4G gateway. For support, see the *Troubleshooting 4G Gateway* chapter in Controller Product Manual.

Continues on next page

Configuring without Connected Services Gateway

Connected Services can be used through the Public Port of the controller, if there is no Connected Services Gateway installed.

> **Note**
>
> Connected Services on Public port is reserved for ABB Factory connectivity and not yet implemented for customers.

> **WARNING**
>
> The customer must ensure that a firewall is installed on its network to prevent unexpected external accesses if the Public port is connected to Internet.

Use the following procedure to configure DNS IP:

1. In the **Controller** tab add controller.

   The controller is added.
2. Click **Request Write Access**, to get write access to update the configuration of the controller.
3. Click **Properties** > **Network settings**.

   The **Network settings** window is displayed.
   - Obtain an IP address automatically: The controller will receive an IP address automatically from DHCP server.
   - Use the following Ip address: The parameters (IP Address, Subnet Mask, and Default Gateway) fields are enabled to update manually.
4. Click **OK**.

   The Public IP is configured.
5. To configure the DNS IP, go to **Configuration**, select and right click on **communication** and select **Configuration Editor** option.

   The **Configuration Editor** window is displayed.
6. Under **Type** select the **DNS Client** option.

   The **DNS Client** options are displayed
7. Right click on DNS client option and select **Edit DNS Client**.

   The **Instance Editor** window is displayed.
8. Update the values in **Instance Editor**:
   - Enable: Select **Yes** to enable the DNS IP.
   - 1st Name Server: Provide the first name server.
   - 2nd Name Server: Provide the second name server, if available.
9. Click **OK**.

   The DNS IP is configured.
10. Restart the controller to apply the changes.

3.4.4 Connected Services Gateway information

Connected Services gateway pages

The Connected Services gateway information pages display the details based on the detected module.

> **Note**
>
> Eth1 is the port connected to the robot controller.
>
> Eth2 is the port connected to the internet.

Connected Services Gateway 3G pages

Connected Services Gateway 3G pages from FlexPendant

The following Connection details are available in **Settings** > **ABB Connected Services** > **Connectivity Status**:

- Module connection
- Mobile connectivity
- Mobile configuration

The following module gateway information are available under **Connected Services Gateway** section in **Settings** > **System** > **Hardware devices**.

- Module information
- Firware 1

> **Note**
>
> In Robotstudio the gateway information of the Connected Services Gateway 3G pages are available in **Controller** > **Properties** > **Device Browser** > **Hardware devices** > **Connected Services Gateway**:

Module information

The following table provides information about the fields available in 3G Module information page:

Field	Description
Hardware version	Version number of the module hardware. For example, P2.
Installed module	Name of the installed module. For example, DSQC 1039 3G.
Serial number	Serial number of the hardware module. For example, 965971#1625000001.
Type	Type of the module. For example, DSQC1039.
Eth1 MAC ID	MAC address of the LAN port of the module.
Eth2 MAC ID	MAC address of the WAN port of the module.
Modem manufacturer	Name of the modem manufacturer. For example, QUALCOMM INCORPORATED.
Modem model	Model number of the modem.
Modem revision	Revision number of the modem.

Continues on next page

Field	Description
Modem S/N	Serial number of the modem available in the module.
IMEI	IMEI (International Mobile Equipment Identity) of the modem device.
ICCID	ICCID (Integrated Circuit Card Identifier) of the installed SIM card on the module.
IMSI	IMSI (International Mobile Subscriber Identity) of the installed SIM card on the module.

Firmware 1

The following table provides information about the fields available in Firmware page:

Field	Description
Software Version	Version number of the firmware software. For example, SW v1.15.

Module connection

The following table provides information about the fields available in 3G Module connection page:

Field	Description
Gateway Status	The status showing whether the module is connected to the network or not. The available status values are **disabled**, **idle**, **connected**, and **ready**.
Gateway IP	The public IP obtained from the network operator.
Eth2 IP	The public IP obtained from the network operator.
Eth2 Mask	Subnet mask of the connected network.
Eth2 Gateway	Default gateway of the connected network
Eth2 DNS1 and DNS2	Primary and the secondary DNS IP obtained from the network operator.
Eth2 Routes 1 - 6	Network Routes created by the module.
Connection Status	Connectivity status to the internet.
Eth1 IP	The IP address of LAN port of the module.
Eth1 Mask	The subnet mask of LAN port of the module.

Mobile connectivity

The following table provides information about the fields available in Mobile connectivity page:

Field	Description
Operator connection type	Type of the mobile connection used. The available values are **GSM** and **UMTS**.
Operator name	Name of the mobile operator.
Operator MCC (Country Code)	Mobile country code of the connected operator.
Operator MNC (Network Code)	Mobile network code for the connected operator.

Continues on next page

3.4.4 Connected Services Gateway information
Continued

Field	Description
Operator status	Connection registration status with the mobile operator.
Connection state	Connection status with the mobile operator.
Signal strength (dBm)	Strength of the signal. Signal strength of - 60 dBm is nearly perfect.
Signal level	Strength of the mobile signal network. 0-3 as displayed in module LEDs.
Cellular ID	Cellular ID of the mobile operator.
Location area code	Unique number of the current location where the SIM is connected.
Extended information	General information.

Mobile configuration

The following table provides information about the fields available in Mobile configuration page:

Field	Description
State	Current state of the module. For example, Enabled or Disabled.
Access point name	Access point name to connect to the network.
APN user	Username of APN.
Operator selection	The ID of the operator. Displays the name of the selected operator or the value is Automatic.
Band selection	Selected network band.
Authentication	Selected authentication method.
Idle time before hang up	Configured idle time (in seconds) before hanging up the connection.
Retry delay	Configured duration (in seconds) between the time-out and retry.
Roaming	Status of roaming. Possible values are enable and disable.

Connected Services Gateway Wi-Fi pages

Connected Services Gateway Wi-Fi pages from FlexPendant

The following Connection details are available in **Settings** > **ABB Connected Services** > **Connectivity Status**:

- Module connections
- Wi-Fi Connectivity
- Wi-Fi Configuration

The following module gateway information are available under **Connected Services Gateway** section in **Settings** > **System** > **Hardware devices**.

- Module information
- Firmware 1

Continues on next page

Note

In Robotstudio the gateway information of the Connected Services Gateway 3G pages are available in **Controller** > **Properties** > **Device Browser** > **Hardware devices** > **Connected Services Gateway**:

Module information

The following table provides information about the fields available in Wi-Fi Module information page:

Field	Description
Hardware version	Version number of the module hardware.
Installed module	Name of the installed module. For example, DSQC 1040 Wi-Fi.
Serial number	Serial number of the hardware module.
Type	Type of the module. For example, DSQC1040.
Eth1 MAC ID	MAC address of the LAN port of the module.
Eth2 MAC ID	MAC address of the WAN port of the module.

Firmware 1

The following table provides information about the fields available in Wi-Fi Firmware page:

Field	Description
Software Version	Version number of the firmware software. For example, `SW v1.15`.

Module connection

The following table provides information about the fields available in Wi-Fi Module connection page:

Field	Description
Gateway Status	The status showing whether the module is connected to the network or not. The available status values are **disabled**, **idle**, **connected**, and **ready**.
Gateway IP	The IP address obtained on the WAN port.
Eth2 IP	The public IP obtained from the network operator.
Eth2 Mask	Subnet mask of the connected network.
Eth2 Gateway	Default gateway of the connected network.
Eth2 DNS	Primary and the secondary DNS IP obtained from the network operator.
Eth2 Routes	Network Routes created by the module.
Connection Status	Connectivity status to the internet.
Eth1 IP	IP address of LAN port of the module.
Eth1 Mask	Subnet mask of LAN port of the module.

Continues on next page

3.4.4 Connected Services Gateway information
Continued

Wi-Fi connectivity

The following table provides information about the fields available in Wi-Fi connectivity page:

Field	Description
State	The connection status of the Connected Services Gateway Wi-Fi module.
SSID	The SSID of the wireless network to which the module is connected.
Security Type	Security method used for connecting to the network.
Signal Strength (dBm)	Strength of the signal. Signal strength of - 60 dBm is nearly perfect.
Signal Level	Strength of the network. 0-3 as displayed in module LEDs.

Wi-Fi configuration

The following table provides information about the fields available in Wi-Fi configuration page:

Field	Description
State	The connection state of the Connected Services Gateway Wi-Fi module.
SSID	The configured SSID used to connect to the wireless network.
Security Type	The security type selected during the configuration.

Connected Services Gateway Wired pages

Connected Services Gateway Wired pages from FlexPendant

The following module gateway information are available in **Settings** > **ABB Connected Services** > **System** > **Hardware devices**.

- Module information

Note

In Robotstudio the gateway information of the Connected Services Gateway 3G pages are available in **Controller** > **Properties** > **Device Browser** > **Hardware devices** > **Connected Services Gateway**:

Module information

The following table provides information about the fields available in Wired Module information page:

Field	Description
Installed Module	Type of the module. For example, DSQC 1041 Wired.

Note

The Connected Services Gateway 4G is seen as Connected Services Wired.

Continues on next page

No Connected Services Gateway pages

If Public port is configured, then the network status is displayed under Public section in **Settings** > **Network** > **Network Status**.

> **Note**
>
> In Robotstudio the public port information are available under **Controller** > **Properties** > **Controller Properties** > **Network Connection** > **WAN**.

The following table provides information about the fields available in Public page:

Field	Description
Port Name	The name of the public port.
Physical Address	The MAC address of the Public port.
DHCP Enabled	The state of the DHCP server is enabled/disabled.
IP Address	The IP address obtained from DHCP or IP address configured manually.
Subnet mask	The Subnet mask of the public port network.
Default Gateway	The Default Gateway of the public port network.
Preferred DNS server	The DNS IP address obtained from DHCP or DNS IP address configured manually.
Media state	The state of the network port where the network cable is connected or not.
Port Enabled	The state of the public port enabled/disabled.
Speed	The network speed of the public port.

3.5 Single point of control

Recommendations for configuration of single point of control

Single point of control is the ability to operate the robot such that initiation of robot motion is only possible from one source of control and cannot be overridden from another initiation source.

In manual mode, the FlexPendant always has highest priority and can be used to start and stop program execution, jog, and configure the system. Other clients can connect to the robot, for example RobotStudio.

In automatic mode, there is no difference in priority between clients connected to the robot. The FlexPendant can always be used to start or stop program execution. Any remote client must have the user grant UAS_REMOTE_START_STOP_IN_AUTO to be able to start or stop program execution in automatic mode. Any user with this grant should be located within eyesight of the robot, unless there are presence sensing devices installed that can prevent potentially hazardous situations.

Local presence and local client

As a rule of thumb, having local presence near the robot is recommended when changing operating mode, starting or stopping execution, or jogging. This is to ensure that no one else is near the robot before doing anything that can cause a potentially hazardous situation.

A local client is a client connected directly to the robot controller, not over the network. The FlexPendant is always local client.

To become logged in as local client you must have local presence. By design, only one client can be local at any given time.

With the FlexPendant, a user can verify local presence with the three-position enabling device. For robots without a connected FlexPendant, system input signals can be used to verify local presence.

CAUTION

It is the responsibility of the integrator to implement that local presence is set up in a correct way.

It is the responsibility of the integrator to implement that single point of control is set up in a correct way.

3.6 Robots in collaborative applications

Introduction

The collaborative robots from ABB can have specific functions that are not available on the standard industrial robots. The collaborative robots can also have differences in behavior compared to the standard industrial robots. These specific functions and differences in behavior are described in the product manual for the respective manipulator.

Indication of status for collaborative robots

The collaborative robots from ABB can have a lamp or light right that indicate status. If the light is integrated on the manipulator, then the description on how to configure this is described in the product manual for the manipulator. It is also possible to configure an external lamp or similar, using I/O signals. This is described in the product manual for the controller (section *Installation and commissioning*, *I/O system*), together with the manuals describing I/O configuration (applicable options are listed in the same section in the product manual for the controller).

3.7 Service Information System

3.7.1 Introduction to Service Information System (SIS)

Introduction

Service Information System (SIS) is a software function in the robot controller, that simplifies maintenance of the robot system. It supervises the operating time and mode of the robot, and alerts the operator when a maintenance activity is scheduled.

Maintenance is scheduled by setting the system parameters of the type *SIS Parameters*, see *Technical reference manual - System parameters*.

Service Information System also supervises the motor status on large robots during high load operations, see *Safety shutdown messages on page 85*.

Supervised functions

The following counters are available:

Calendar time

Calendar time is used for robot service intervals, based on calendar time.

Operation time

Operation time is used for robot service intervals, based on operational time.

Gearbox

Gearbox is used for estimating the service interval (remaining lifetime) of the gearbox. Each supervised gearbox has a time counter, based on the wear of the axis and the work load on the robot. The number of supervised gearboxes on the robot are different for each model.

SIS event logs reported as warnings instead of errors

By default SIS event logs are reported as errors, but it is also possible to have the SIS event logs reported as warnings instead of errors. The main difference is that warnings do not take focus on the FlexPendant, and that they have a different icon. This will affect calendar time events, production time events, and gearbox events.

For more information, see the parameter *Events as Warnings* in *Technical reference manual - System parameters*.

3.7.2 SIS counters

3.7.2.1 Calendar time

Description

Calendar time is used for robot service intervals, based on calendar time. This timer can, for example, be used to schedule when it is time for inspection of dampers or motor seals. This information is available in the maintenance schedule for the robot.

When the calendar time limit for maintenance is reached, a message is stored in the event log.

Components

The following information is available about calendar time in the service routine *ServiceInfo*.

Service information	Description
Prev. Service	Shows the date when the counter was reset the last time, that is, the date of the last service.
Elapsed Time	Shows the number of days that have elapsed since the counter was reset the last time.
Next Service	Shows the date of the next scheduled service.
Remaining Time	Shows the number of days remaining until the next scheduled service.

3.7.2.2 Operation time

Description

Operation time is used for robot service intervals, based on operational time. *Operation time* uses the time that the MOTORS ON signal is active and the brakes are released. This timer can, for example, be used to schedule when it is time to change oil. This information is available in the maintenance schedule for the robot.

When the operation time limit for maintenance is reached, a message is stored in the event log.

Components

The following information is available about operation time in the service routine *ServiceInfo*.

Service information	Description
Service Interval	Shows the specified service interval, in hours.
Elapsed Time	Shows the time, in hours and minutes, that has elapsed since the service interval was set the last time.
Remaining Time	Shows the operation time, in hours and minutes, remaining until the service interval has expired.

3.7.2.3 Gearbox

Description

Gearbox is used for estimating the service interval (remaining lifetime) of the gearbox. The estimate is based on used torque and speed (rpm). This information can be used as a guidance when planning gearbox replacement.

When the estimated gearbox life is reached, a message is stored on the event log.

Note

The maximum estimated service interval is 40,000 hours.

Components

The following information is available about the gearbox status in the service routine *ServiceInfo*.

Service information	Description
Axis x OK	The automatically calculated time parameter for the axis in question has not been exceeded.
Axis x NOK	The service interval for the axis in question has been reached.
Axis x N/A	No service time parameter calculation is available for the axis in question. **Note** This information is displayed when there is no supervision on gearbox x.

The following information is available for the gearbox time in the service routine *ServiceInfo*.

Service information	Description
Consumed time	Shows the consumed time as a percentage of the total amount of time.
Elapsed time	Shows the operation time, in hours, that has elapsed for axis x since measurement began.
Remaining time	Shows the time, in hours, remaining for axis x until the gearbox is planned to be replaced.

CAUTION

Incorrectly defined or reset counters can result in wrong information being displayed.

3.7.3 Using the SIS system

Introduction

This is a brief description of how to use the Service Information System (SIS).

Using the SIS system

Use this procedure to run the SIS system.

	Action	Reference
1	Determine which SIS functions to use.	See *Calendar time on page 77*, *Operation time on page 78* and *Gearbox on page 79*.
2	Define what values are adequate and suitable for your application in your production environment.	Recommendations for expected component life and service intervals are described in the *Product manual*, section *Maintenance*.
3	Enter these parameters in the system parameter configuration.	See *Setting the SIS parameters on page 81*.
4	Run the robot in normal operation.	
5	Reset the counter if a repair has been made, or if a counter for any other reason has been restarted.  **CAUTION** Resetting counters cannot be undone.	See *Resetting values on page 82*.
6	When a time limit is exceeded, a message is stored in the event log.	See *Reading the SIS logs on page 83*.
7	If the log containing the message is to be available from an external PC, or if the SIS parameters are to be entered from an external PC, a set of software tools are available to build such an application.	
8	Some robots can be programmed to give warnings if the motors are overheated and need to be cooled down.	See *Safety shutdown messages on page 85*.

3.7.4 Setting the SIS parameters

Introduction

The Service Information System is set up using system parameters. The values can be be based on the maintenance schedule for the robot, but must be adapted over time by the users as knowledge of the robot's working conditions are accumulated.

Since the warnings are to be used for purposes defined by the user, ABB cannot give any recommendations regarding their definitions.

Use RobotStudio to configure the system parameters of the types *SIS Parameters* and *SIS Single Parameters*, in the topic *Motion*.

Examples

These examples show how the system parameters can be set.

Operational Limit (h)

If the parameter is set to `2000`, SIS will alert the operator after 2000 hours in operation mode.

Operational Warning (%)

If the parameter is set to `90`, SIS will warn the operator after 1800 hours in operation mode. The total amount of hours is calculated from the percentage of *Operational Limit (h)*.

```
2000*0.9=1800
```

Calendar Limit (years)

If the parameter is set to `2`, SIS will alert the operator after 2 years.

Calendar Warning (%)

If the parameter is set to `50`, SIS will warn the operator after 1 year. The total amount of years is calculated from a percentage of *Calendar Limit (years)*.

```
2*0.5=1
```

Gearbox warning (%)

If the parameter is set to `90`, SIS will alert the operator after 90% of the expected service interval of each gearbox.

The robot system automatically detects and stores all required variables to calculate the expected service interval of each gearbox. This is done by extrapolating data from earlier operation into a function of time, using a formula including:

- Input and output torque.
- Gearbox spindle speed.
- Ambient temperature.
- Other variables.

Robot temperature (C) / Single temperature (C)

This parameter defines the ambient temperature used to estimate the remaining gearbox lifetime.

3.7.5 Resetting values

Resetting values

Counters can be reset at any time by running the service routine *ServiceInfo* from the FlexPendant, see *Operating manual - OmniCore*.

CAUTION

Resetting counters cannot be undone.

	Action
1	Start the **Program Editor**.
2	Tap **Debug** and then tap **Call Routine**.
3	Tap **ServiceInfo** to start the service routine.
4	If there is more than one robot, tap ROB_x to select robot.
5	In the service routine main window, select the counter to be reset (1= Calendar time, 2=Operation time and 3= Gearbox).
6	In the selected counter window, tap **Reset**.

3.7.6 Reading the SIS logs

Introduction

When a set counter value has been reached (for example the maximum allowed operation time before service), a message is shown in the event log.

Access to logs

In RobotStudio, use the function *Fleet Assessment* to extract SIS data that ABB can use to further investigate, if needed.

Duty Time

The total duty time can be seen on the FlexPendant.

On the ABB menu, tap **System Info** and then tap to expand **Hardware devices**, and continue through **Mechanical units** and **ROB_1** to **General SIS data**.

Service interval exceeded

If the service interval has exceeded the defined value, an error message (Service interval exceeded!) is displayed.

No data available

If no data is available for the defined value, a message (No data available!) is displayed when trying to show the data.

Available messages

The following messages can be shown:

Counter	SIS log message	Description
Calendar time	Service message: X calendar days to next service.	X number of calendar days remain until the manually set calendar time limit expires. How to set the limit is detailed in section *Setting the SIS parameters on page 81*.
Calendar time	Service message: Service is due! X calendar days since last service.	The manually set calendar time limit has expired. How to set the limit is detailed in section *Setting the SIS parameters on page 81*. Proceed with the required service as detailed in the *Product manual*.
Operation time	Service message: X production hours to next service.	X number of operation hours remain until the manually set operation time limit expires.How to set the limit is detailed in section *Setting the SIS parameters on page 81*.
Operation time	Service message: Service is due! X production hours since last service.	The manually set operation time limit has expired. How to set the limit is detailed in section *Setting the SIS parameters on page 81*. Proceed with the required service as detailed in the *Product manual*.

Continues on next page

3.7.6 Reading the SIS logs

Continued

Counter	SIS log message	Description
Gearbox time	Service message: X% of the service interval has expired for gearbox x!	X% of the gearbox hours remain until the estimated gearbox lifetime limit has expired. How to set the limit is detailed in *Setting the SIS parameters on page 81*.
Gearbox time	Service message: Gearbox x requires service!	The estimated gearbox lifetime limit has expired. Proceed with the required service as detailed in the *Product manual*.

3.7.7 Safety shutdown messages

Duty factor warning

The safety shutdown is a warning and an error message used to protect large robots from damaging the motors or gearboxes during high load operations. The warning/error is titled **50263 Duty factor warning**.

Usage

When robots work in high speed under heavy load for long periods of time, the motors and gearboxes will become hot. Letting the motors and gearboxes cool down occasionally will prevent damaging them.

The limit for how hard the motors and gearboxes can run depends on both torque, revolution, and also the ambient temperature. This can be simulated for an installation with the RobotStudio add-in *Gearbox Heat Prediction Tool*. The tool does not require a premium license.

If the motors or gearboxes risk being overheated the system will warn that the robot needs to cool down. A warning message is sent to the log and after 30 minutes an error message is also sent to the log. An error handler can be used to take adequate measures, for example turning off external equipment and moving the robot out of the working area.

If the warning (**50263 Duty factor warning**) is displayed often but it is verified that the motors and gearboxes are not overheated, then changing the system parameter that defines the ambient temperature (*Robot temperature (C)* or *Single temperature (C)*) can reduce the occurance.

CAUTION

If the error is ignored without letting the motor or gearbox cool down, then the lifetime of the motor or gearbox can be reduced.

Examples

These are examples of error handlers can be used. `IError` is used to order and enable an interrupt when an error occurs. Inside the TRAP function you program what actions to be done before the system is shut down.

Warning

```
TRAP trap_name_warn
  ! Your actions
END TRAP
..
..
PROC main()
  CONNECT errorint WITH trap_name_warn;
  IError MOTION_ERR\Error Id := 263, TYPE_WARN, errorint;
  ..
  ..
END main
```

Continues on next page

3.7.7 Safety shutdown messages
Continued

Error

```
TRAP trap_name_error
  ! Your actions
END TRAP
..
..
PROC main()
  CONNECT errorint WITH trap_name_error;
  IError MOTION_ERR\Error Id := 263, TYPE_ERR, errorint;
  ..
  ..
END main
```

Related information

For information about	See
IError	*Technical reference manual - RAPID Instructions, Functions and Data types*
TRAP	*Technical reference manual - RAPID Instructions, Functions and Data types*
StopMove	*Technical reference manual - RAPID Instructions, Functions and Data types*
StorePath	*Technical reference manual - RAPID Instructions, Functions and Data types*
Robot temperature (C) and *Single temperature (C)*	*Technical reference manual - System parameters*
Gearbox Heat Predition Tool	*Operating manual - RobotStudio*

4 Programming

4.1 Before you start programming

Programming tools

You can use both the FlexPendant and RobotStudio for programming. The FlexPendant is best suited for modifying programs, such as positions and paths, while RobotStudio is preferred for more complex programming.

How to program using RobotStudio is described in *Operating manual - RobotStudio*.

Define tools, payloads, and work objects

Define tools, payloads and work objects before you start programming. You can always go back and define more objects later, but you should define your basic objects in advance.

See *Creating a tool on page 111*.

WARNING

It is important to always define the actual tool load and, when used, the payload of the robot (for example, a gripped part). Incorrect definitions of load data can result in overloading of the robot mechanical structure. There is also a risk that the speed in manual reduced speed mode can be exceeded.

When incorrect load data is specified, it can often lead to the following consequences:

- The robot may not use its maximum capacity.
- Impaired path accuracy including a risk of overshooting.
- Risk of overloading the mechanical structure.

The controller continuously monitors the load and writes an event log if the load is higher than expected. This event log is saved and logged in the controller memory.

Define coordinate systems

Make sure the base and world coordinate systems have been set up properly during the installation of your robot system. Also make sure that additional axes have been set up.

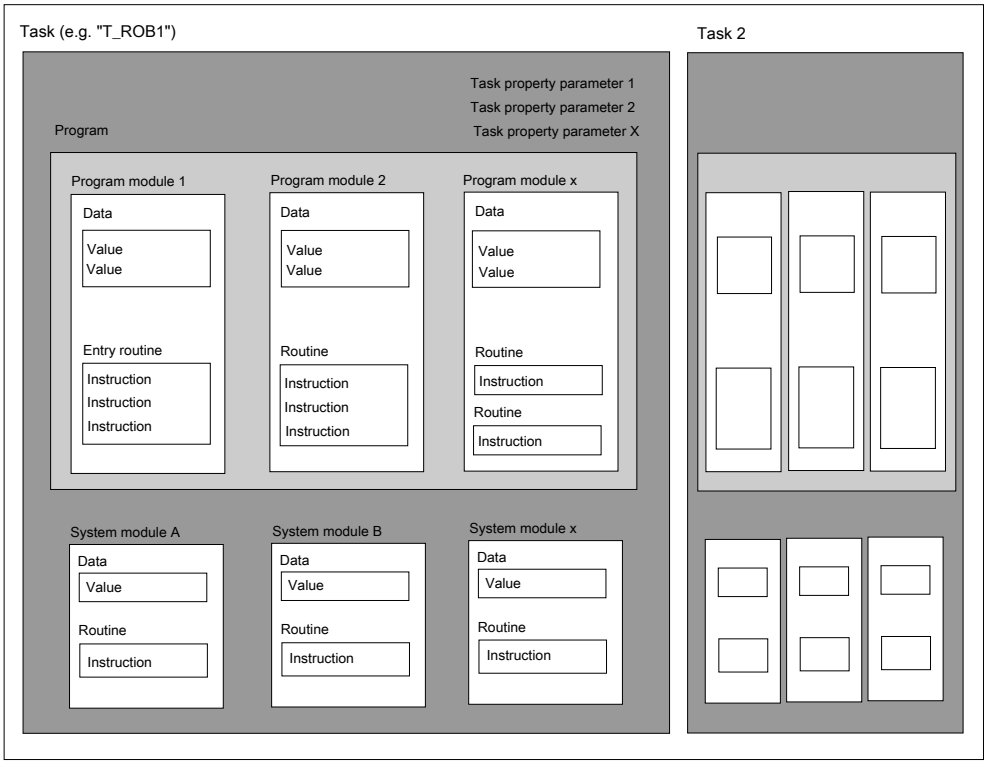
Define tool and work object coordinate systems before you start programming. As you add more objects later you also need to define the corresponding coordinate systems.

Tip

For more details about the RAPID language and structure, see *Technical reference manual - RAPID Overview* and *Technical reference manual - RAPID Instructions, Functions and Data types*.

4.2 The structure of a RAPID application

Illustration of a RAPID application



en0300000576

Parts

Part	Function
Task	Each task usually contains a RAPID program and system modules aimed at performing a certain function, e.g. spot welding or manipulator movements. A RAPID application may contain one task. If you have the *Multitasking* option installed, then there can be more than one task. Read more about *Multitasking* in *Application manual - Controller software OmniCore*.
Task property parameter	The task property parameters set certain properties for all task contents. Any program stored in a certain task, assumes the properties set for that task. The task property parameters are specified in *Technical reference manual - RAPID Overview*.
Program	Each program usually contains program modules with RAPID code for different purposes. Any program must have an entry routine defined to be executable.

Continues on next page

Part	Function
Program module	Each program module contains data and routines for a certain purpose. The program is divided into modules mainly to enhance overview and facilitate handling the program. Each module typically represents one particular robot action or similar. All program modules will be removed when deleting a program from the controller program memory. Program modules are usually written by the user.
Data	Data are values and definitions set in program or system modules. The data are referenced by the instructions in the same module or in a number of modules (availability depending on data type). Data type definitions are specified in the *Technical reference manual - RAPID Instructions, Functions and Data types*.
Routine	A routine contains sets of instructions, i.e. defines what the robot system actually does. A routine may also contain data required for the instructions.
Entry routine	A special type of routine, in English sometimes referred to as "main", defined as the program execution starting point. **Note** Each program **must** have an entry routine called "main", or it will not be executable. How to appoint a routine as entry routine is specified in *Technical reference manual - RAPID Overview*. The default name for main can be changed by the system parameter configurations, type *Task*. See *Technical reference manual - System parameters*.
Instruction	Each instruction is a request for a certain event to take place, e.g. "Run the manipulator TCP to a certain position" or "Set a specific digital output". The instructions, their syntax and function is thoroughly described in the *Technical reference manual - RAPID Instructions, Functions and Data types*.
System module	Each system module contains data and routines to perform a certain function. The program is divided into modules mainly to enhance overview and facilitate handling the program. Each module typically represents one particular robot action or similar. All system modules will be retained when "Delete program" is ordered. System modules are usually written by the robot manufacturer or line builder.

4.3 Programming

4.3.1 Handling of programs

Overview

This section provides information about the handling of robot programs. It describes how to:

- create a new program
- load an existing program
- save a program
- rename a program
- delete a program

> **Note**
>
> Each task must contain *one* program. The procedures in this section describes a single task system.

How to create a new program *when no program is available* is detailed in section *Creating a new program on page 91*.

Program Editor

In FlexPendant the RAPID programs are created and edited using **Program Editor** in **Code**.

If you toggle between the **Program Editor** and another view and back again, the **Program Editor** will show the same part of the code as long as the program pointer has not been moved. If the program pointer is moved, the **Program Editor** shows the code at the position of the program pointer.

The same behavior applies to **Advanced View** in **Operate**.

> **Note**
>
> When you open **Code**, the name of the module where the program pointer is in is displayed in the **Program Pointer location** section.

About program files

The program is saved as a folder, named as the program, containing the actual program file, of type pgf.

When loading a program you open the program folder and select the pgf file.

When renaming a program you rename the program folder and the program file.

When saving a loaded program which is already saved to the hard disk, you must not open the existing program folder. Instead, you should save the program folder again and overwrite the old version, or rename the program.

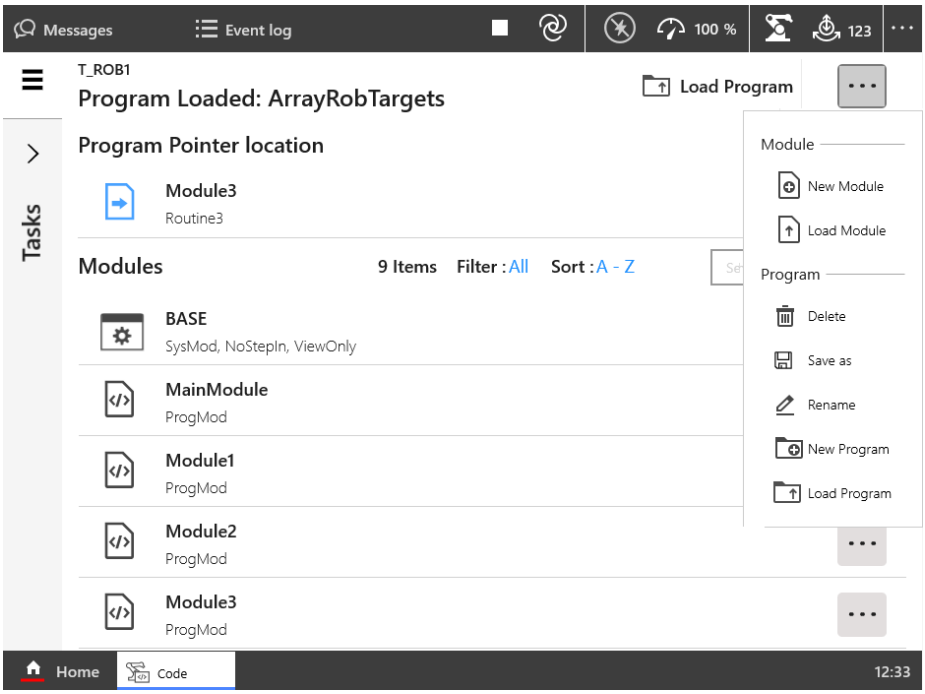
Continues on next page

Creating a new program

This section describes how to create a new program.

1 On the start screen, tap **Code**, and then select **Modules** from the menu.

2 On the **Context menu**, tap **New Program**.



xx1900000214

If there was already a program loaded, a warning dialog appears:

- Tap **Save** to save the loaded program.
- Tap **Don't Save** to close loaded program without saving it, i.e. delete from program memory.
- Tap **Cancel** to leave the program loaded.

3 Tap **Main Module**.

4 Add instructions to the program.

For details regarding adding instructions, see the section **Adding instructions** in *Operating manual - OmniCore*.

5 Tap **Check Program**.

6 Tap **Modules**.

7 On the **Context menu**, tap **Save as**.

8 Tap a name for the program in the **File Name** field.

9 Select the location for saving the new program file.

10 Tap **Save**.

The program is saved.

Once a program is created you can run the program. For details regarding running a program, see the section **Starting programs** in *Operating manual - OmniCore*.

Continues on next page

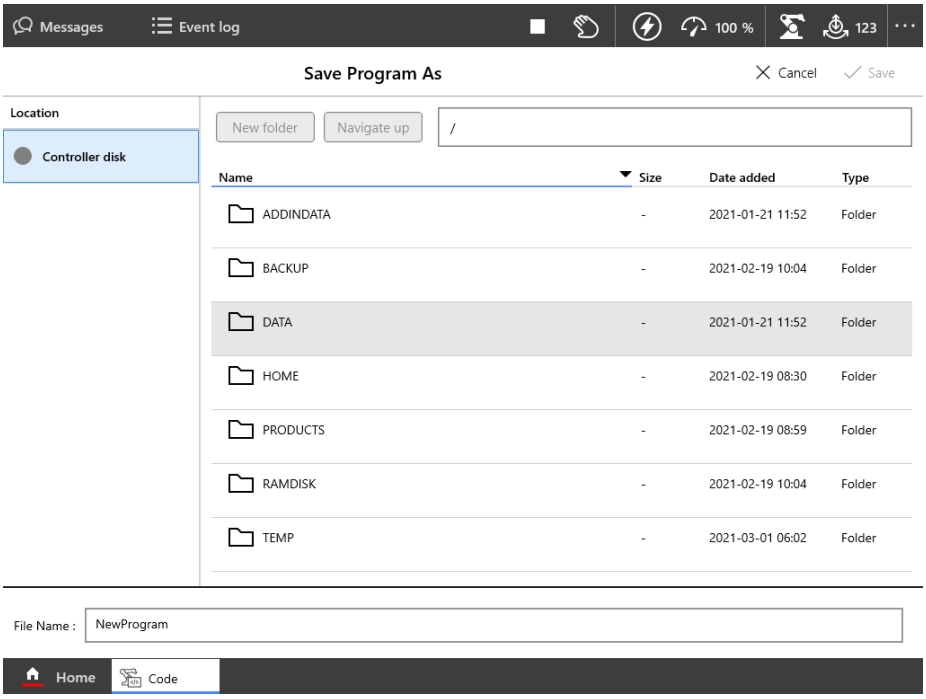
4.3.1 Handling of programs
Continued

Saving a program

This section describes how to save a loaded program to the controller hard disk.

A loaded program is automatically saved in the program memory, but saving to the controller hard disk is an extra precaution.

1. On the start screen, tap **Code**, and then select **Modules** from the menu.
2. On the **Context menu**, tap **Save as**.
3. Use the suggested program name or type a new name in the **File Name** field.



xx1900000223

4. Tap **Save**.

   The program is saved.

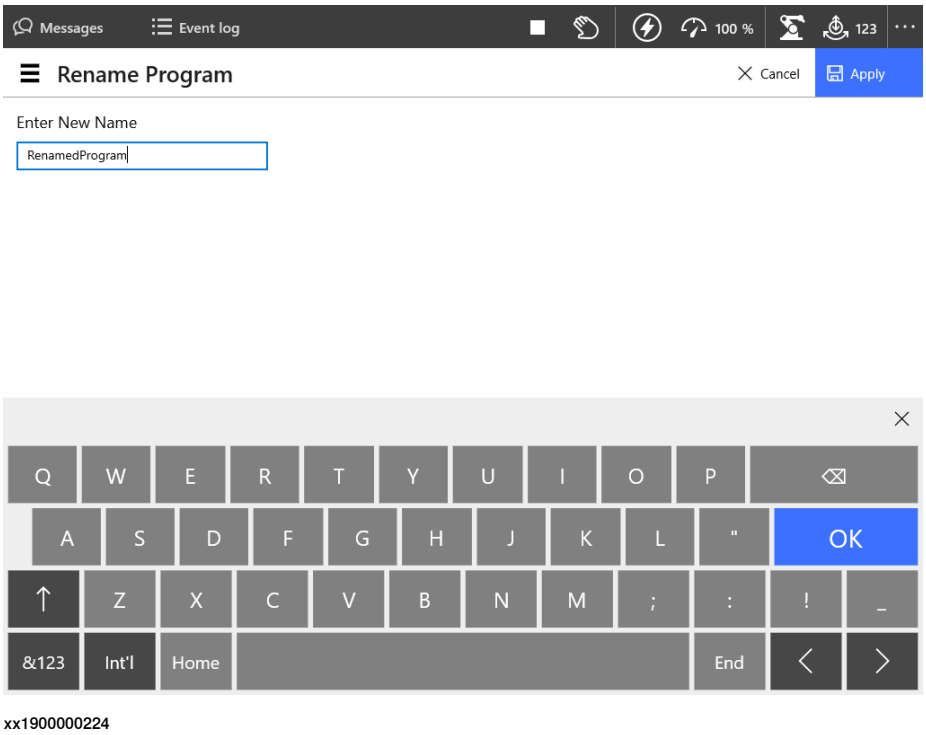
Renaming a loaded program

This section describes how to rename a loaded program.

1. On the start screen, tap **Code**, and then select **Modules** from the menu.
2. On the **Context menu**, tap **Rename**.

Continues on next page

3 In the **Enter New Name** field, type a new name for the selected program.



xx1900000224

4 Tap **Apply**.
The program is renamed.

Deleting a program

This section describes how to delete a program.

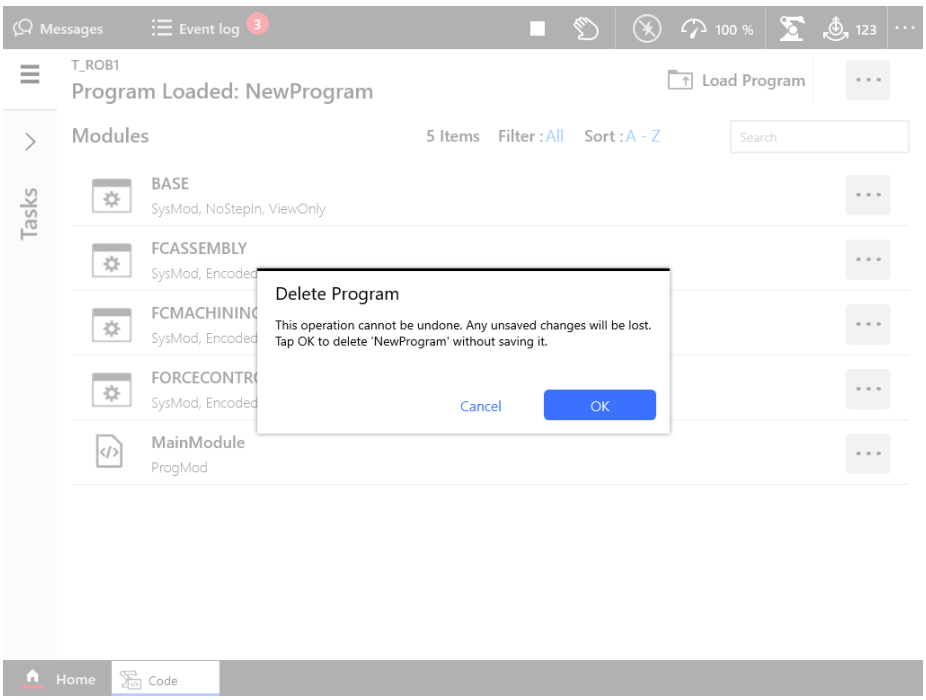
> **Note**
>
> You can only delete a loaded program.

1 On the start screen, tap **Code**, and then select **Modules** from the menu.
2 On the **Context menu**, tap **Delete**.

Continues on next page

4.3.1 Handling of programs
Continued

3 In the **Delete Program** confirmation window, tap **OK** to delete, or **Cancel** to keep the program intact.



xx1900000225

4.3.2 Handling of modules

Overview

This section details how to handle program modules. i.e.:

- create a new module
- load an existing module
- save a module
- rename a module
- delete a module

Creating a new module

This section describes how to create a new module.

1. On the start screen, tap **Code**, and then select **Modules** from the menu.
2. On the **Context menu**, tap **New Module**.
3. In the **Create New Module** window, enter a **Module Name**, and select if **Module Type** should be **Program** or **System**.

   Note

   How to later switch between these types is detailed in section *Changing type of module on page 98*.

4. Tap **Apply**.

   The module is created and displayed in the **Modules** section.

File format for modules

Note

In RobotWare 7.0 and earlier, the formats were `.mod` and `.sys`. When loading these in a RobotWare 7.1 controller or later using RobotStudio, they are automatically converted when saved. When saved, the new file extensions are `.sysx` and `.modx`. Note that the files must be converted, not just renamed.

To convert a file manually, the file must be saved as UTF-8 without BOM (Byte Order Mark).

Loading an existing module

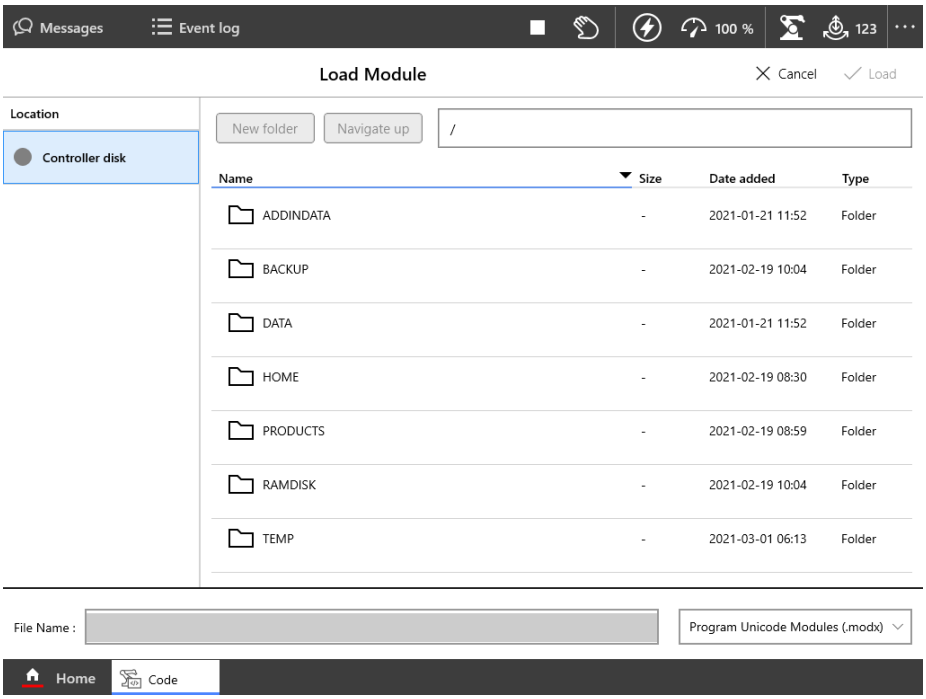
Use the following procedure to load an existing module.

1. On the start screen, tap **Code**, and then select **Modules** from the menu.
2. On the **Context menu**, tap **Load Module**.

Continues on next page

4.3.2 Handling of modules
Continued

3 Navigate and select the module from the location where it is saved.



xx1900000227

4 Tap **Load**.

The selected module is loaded.

Saving a module

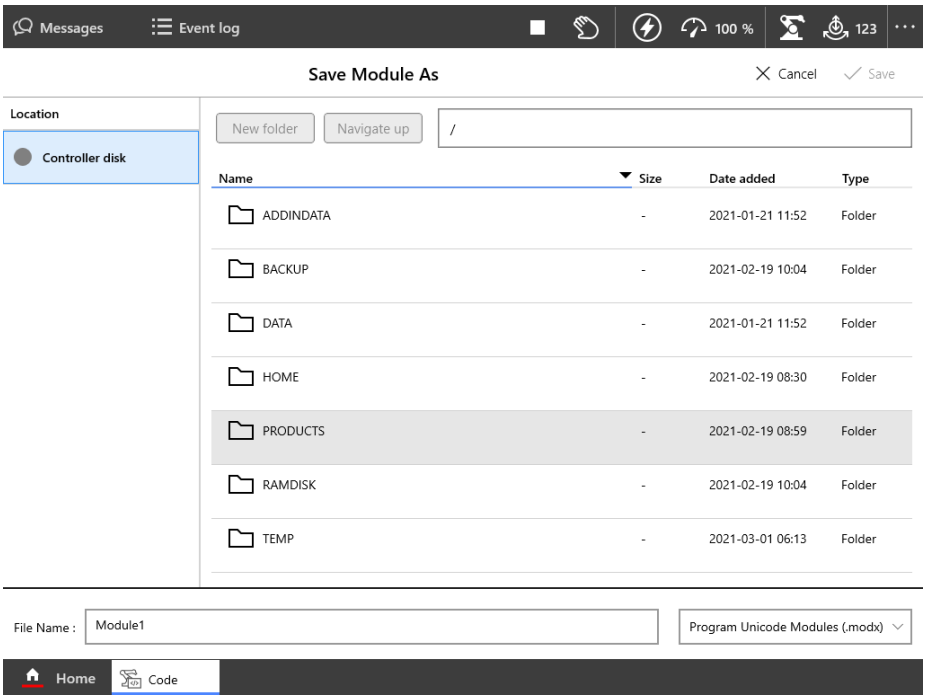
This section describes how to save a module.

1 On the start screen, tap **Code**, and then select **Modules** from the menu.

2 Tap **Save as** on the **Context menu** for the module.

Continues on next page

3 The **Save Module as** window is displayed:



xx1900000229

Tap and select a location for saving the module.

Use the suggested module name or enter a **File Name**.

4 Tap **Save**.

The module is saved in the selected location.

Renaming a module

This section describes how to rename a module.

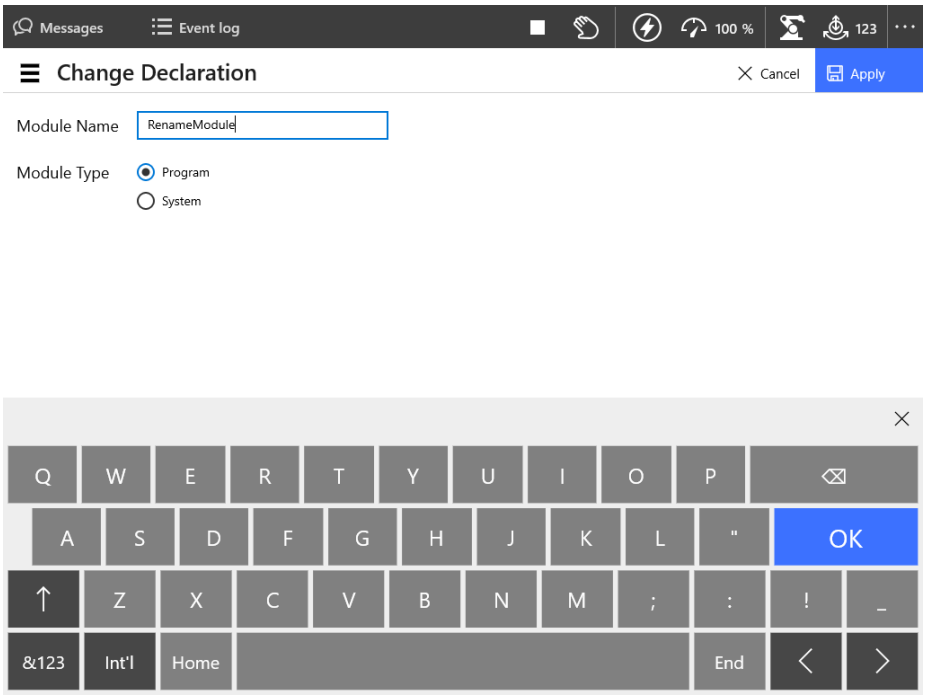
1 On the start screen, tap **Code**, and then select **Modules** from the menu.

2 Tap **Change Declaration** on the **Context menu** for the module.

Continues on next page

4.3.2 Handling of modules
Continued

The **Change Declaration** window is displayed.



xx1900000230

3 In the **Module Name** field type a new name for the module and then tap **Apply**.
The module is renamed.

Changing type of module

This section describes how to change the type of module.

1 On the start screen, tap **Code**, and then select **Modules** from the menu.
2 Tap **Change Declaration** on the **Context menu** for the module.
The **Change Declaration** window is displayed.
3 In the **Module Type** list select a type and then tap **Apply**.
The module type is changed.

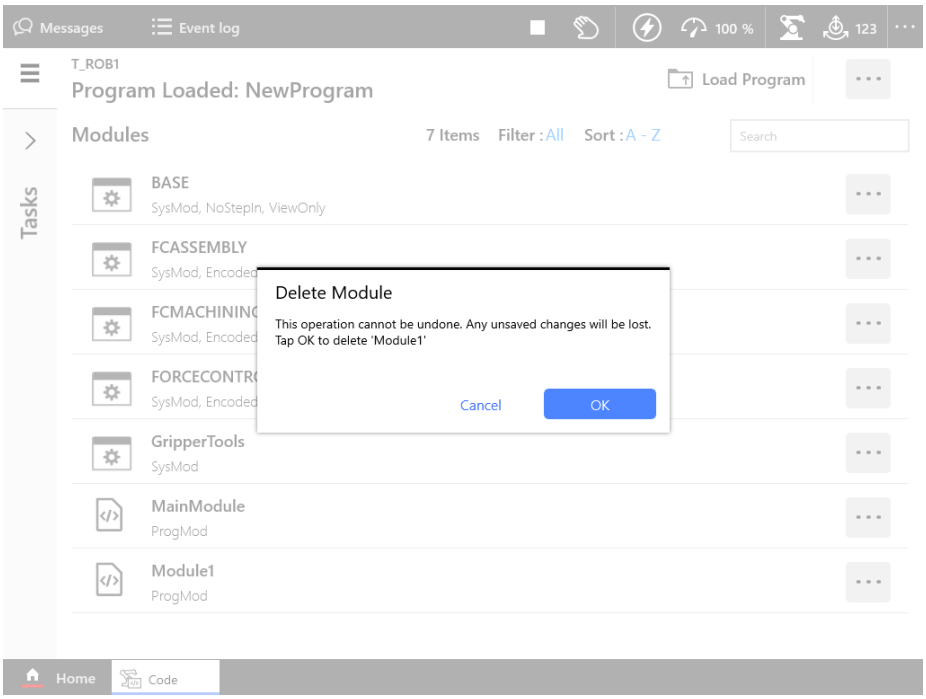
Deleting a module

This section describes how to delete a module from memory. If the module has been saved to disk, it will not be erased from the disk.

1 On the start screen, tap **Code**, and then select **Modules** from the menu.
2 Tap **Delete** on the **Context menu** for the module.

Continues on next page

3 The **Delete Module** confirmation window is displayed:



xx1900000231

4 Tap **OK**.

The selected module is deleted and removed from the module list.

4.3.3 Handling of routines

Overview

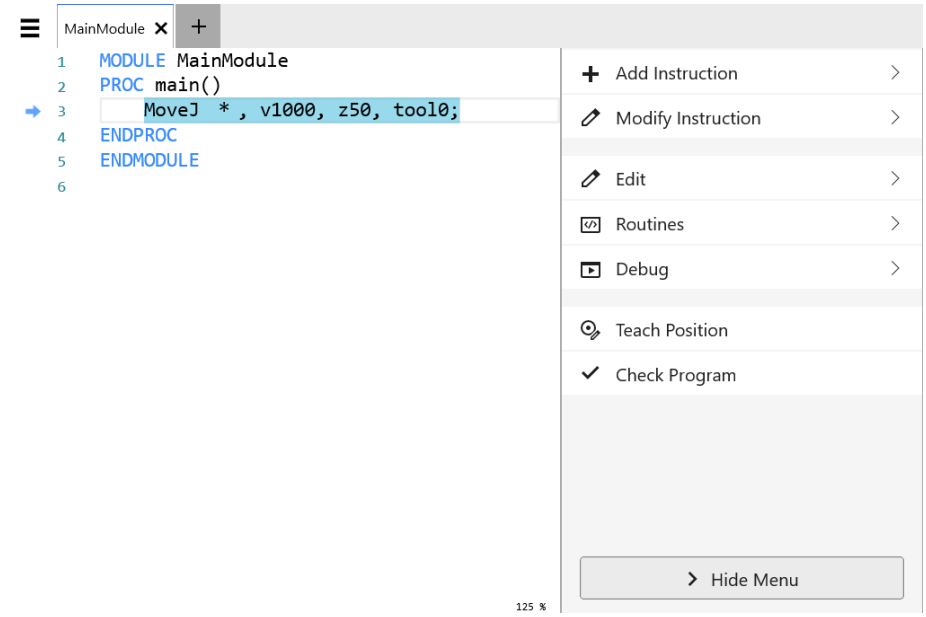
This section details how to handle program routines. i.e.:

- create a new routine
- create a copy of a routine
- change the declaration of a routine
- delete a routine

Creating a new routine

This section details how to create a new routine, set the declaration, and add it to a module.

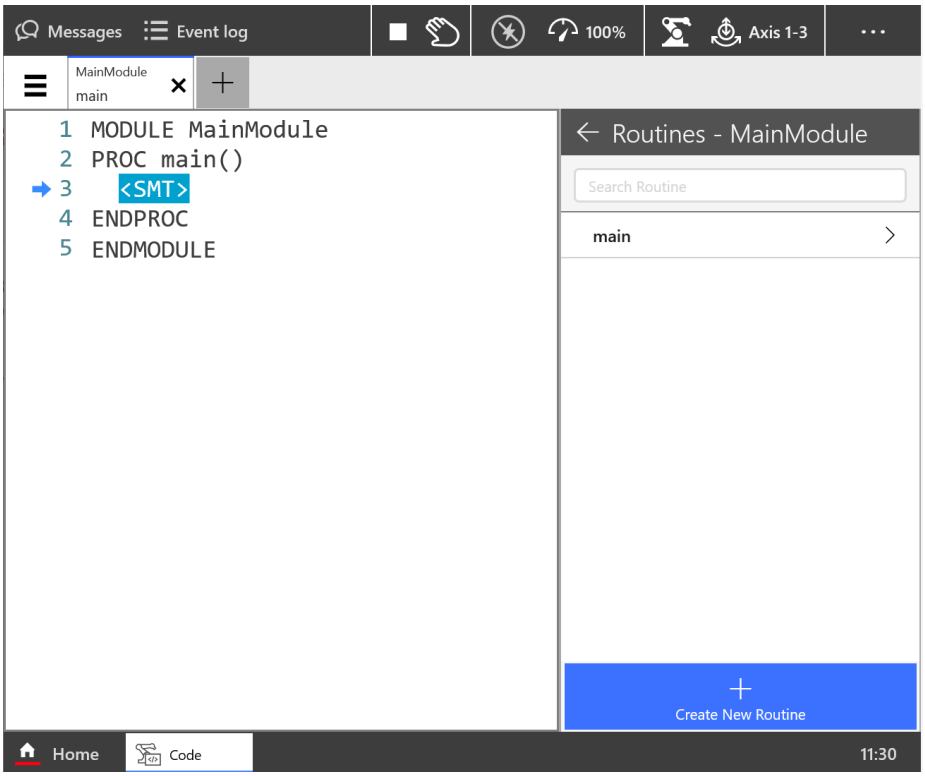
1. On the start screen, tap **Code**, and then select **Program editor** from the menu.
2. Tap **Routines** in the menu to the right.



xx1900000232

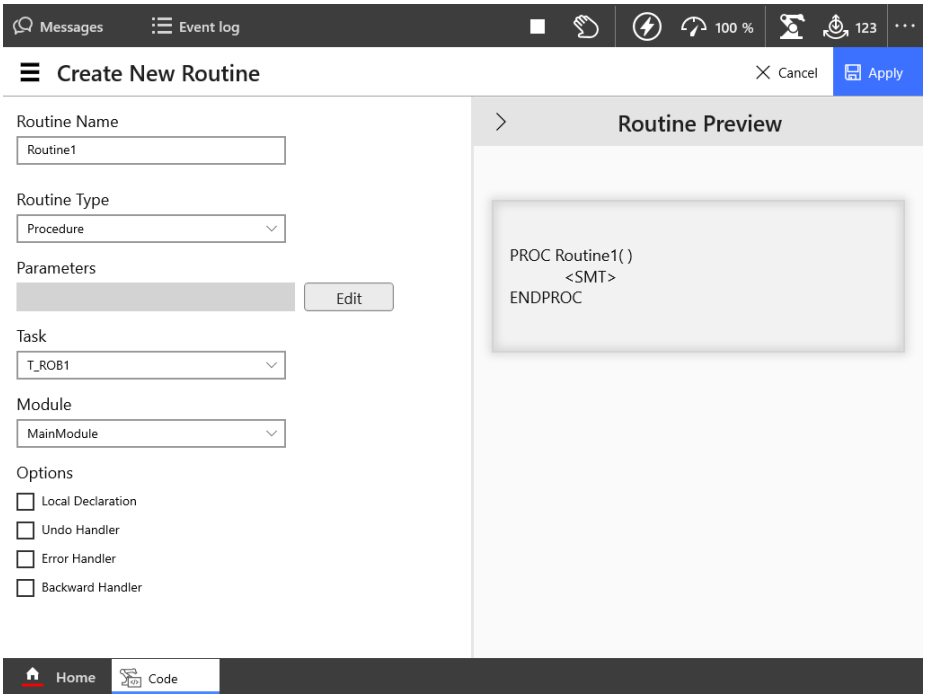
Continues on next page

3 Tap **Create New Routine**.



xx1900000251

4 The **Create New Routine** page is displayed:



xx1900000254

Complete the routine declaration by entering the following information:

- **Routine Name**

Continues on next page

- **Routine Type**
  - Procedure: used for a normal routine without return value
  - Function: used for a normal routine with return value
  - Trap: used for an interrupt routine
- **Parameters**

  Tap **Edit** to add parameters to the routine. See section *Defining parameters in routine on page 102* for more information.
- **Task**
- **Module**
- **Options**
  - Local Declaration

    Tap the checkbox to select **Local declaration** if the routine should be local.

    A local routine can only be used in the selected module.
  - Undo Handler
  - Error Handler
  - Backward Handler

Note

Use the **Preview** button to preview the values selected for the new routine.

5 Tap **Apply**

The new routine is created and displayed in the **Routines** list for the selected module.

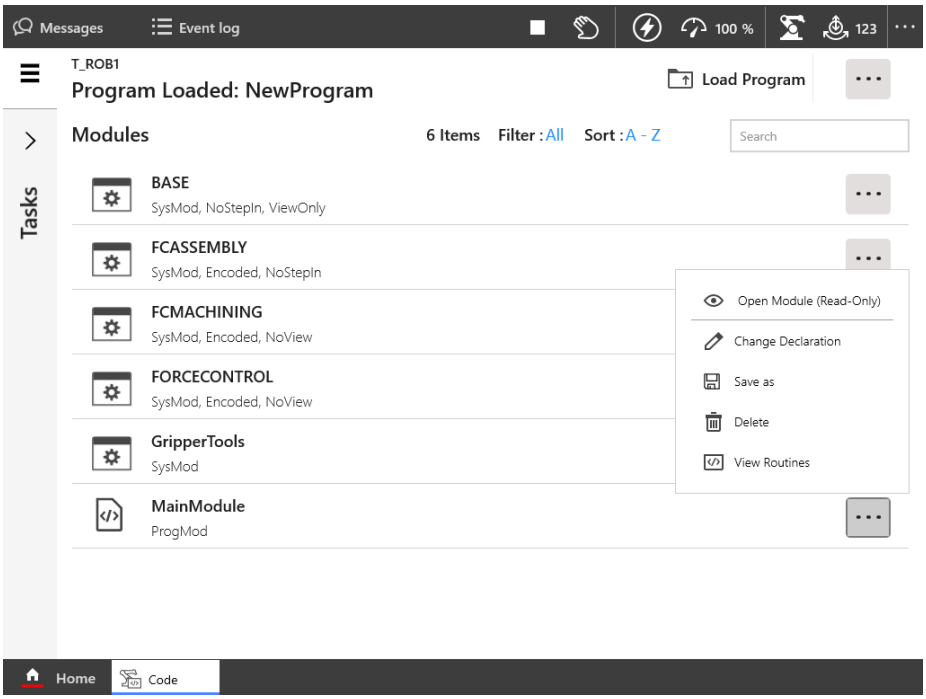
Defining parameters in routine

This section describes how to define parameters in a routine.

1 On the start screen, tap **Code**, and then select **Modules** from the menu.

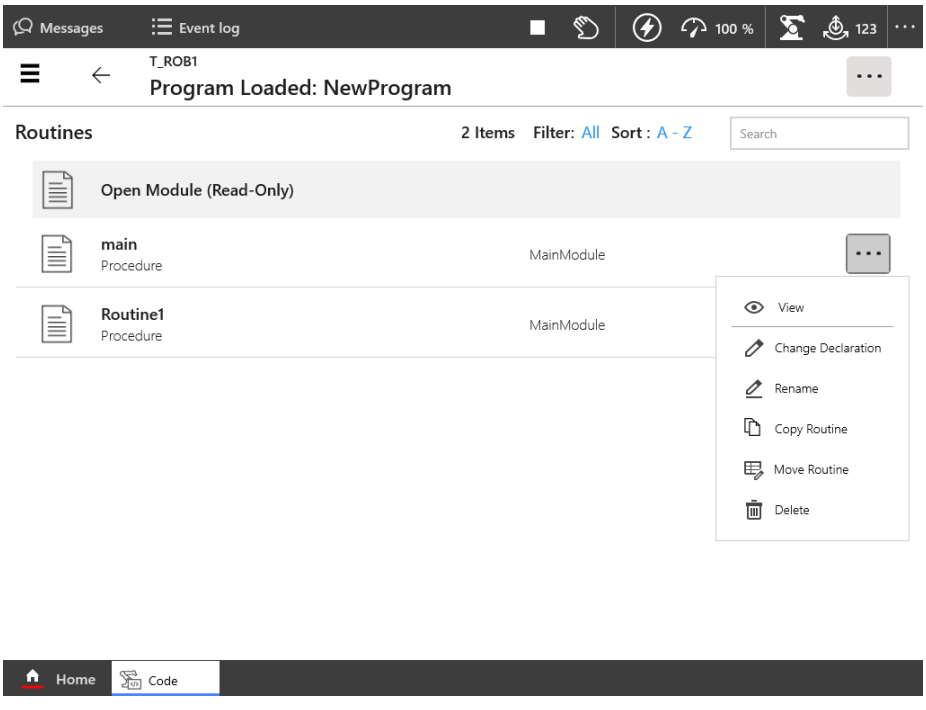
Continues on next page

2 Tap **View Routines** on the **Context menu** for the module.



xx1900000228

3 Tap **Change declaration** on the **Context menu** for the routine.



xx1900000271

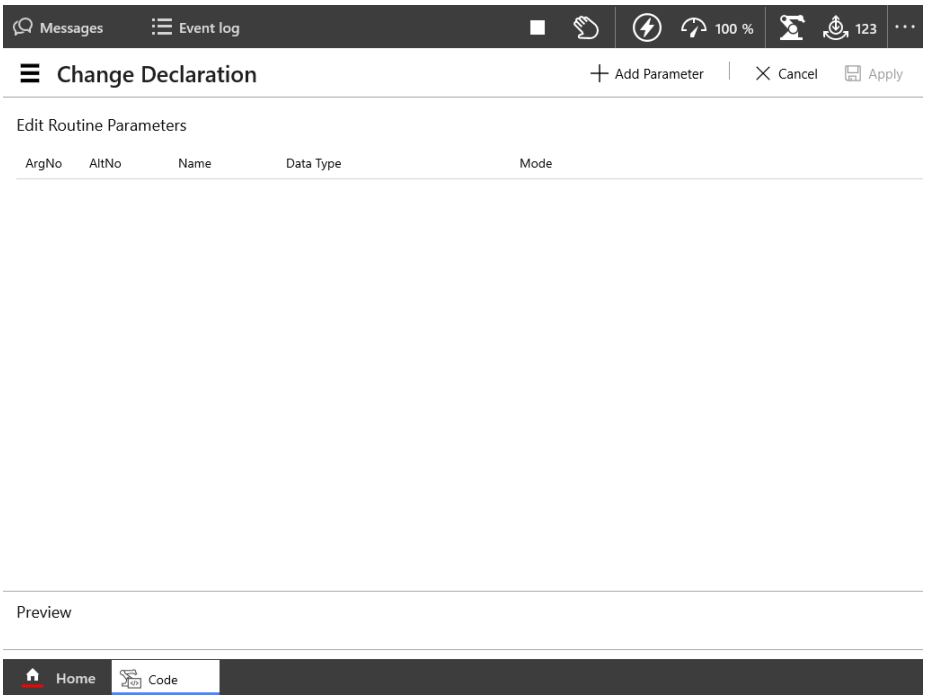
The **Change Declaration** page is displayed.

4 If no parameters are shown, tap the **Edit** button next to the **Parameters** field.

Continues on next page

4.3.3 Handling of routines
Continued

The **Edit Routine Parameters** page is displayed.



xx1900000272

5 Tap **Add Parameter** and select **Mandatory Parameter** or **Optional Parameter** according to your requirement.

The selected parameter is added to the **Edit Routine Parameters** list.

> **Note**
>
> Select an optional parameter and tap **Add Parameter** > **Optional Mutual Parameter** to add a parameter that is mutually optional with another parameter.

Read more about routine parameters in the RAPID reference manuals.

6 Type a **Name** for the parameter and tap **Apply**.

7 The new parameter is displayed in the list. Tap to select a parameter. To edit values, tap the value.

8 Tap **Apply**.

The selected parameters are added to the **Parameters** field in the routine declaration window.

Creating the copy of a routine

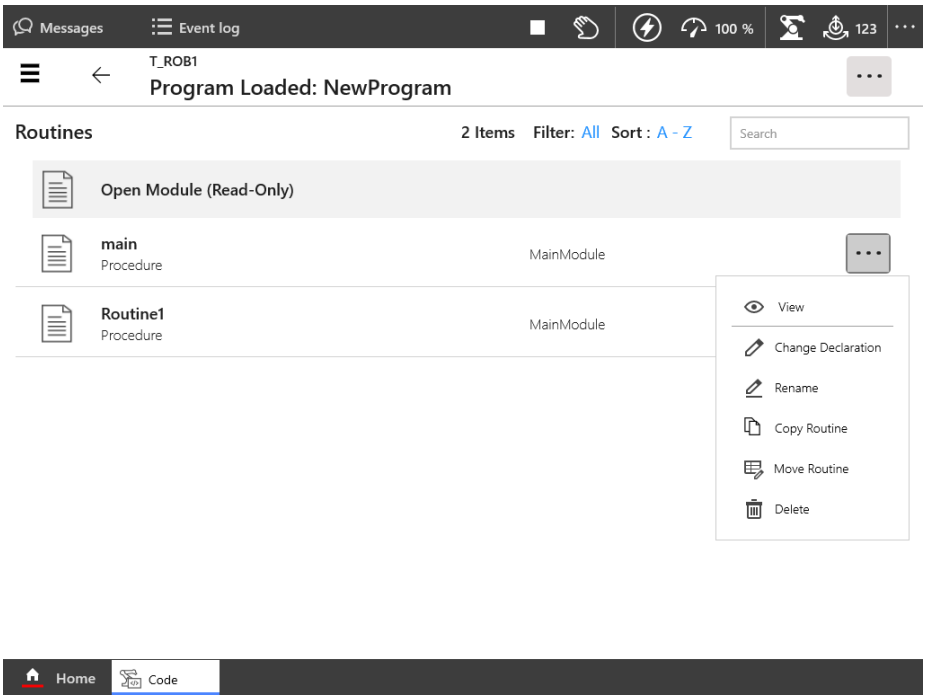
This section describes how to create a copy of a routine.

1 On the start screen, tap **Code**, and then select **Program Editor** from the menu.

2 Tap **View Routines** on the **Context menu** for the module.

Continues on next page

The **Routines** window is displayed.



xx1900000271

3 Tap **Copy Routine** on the **Context menu** for the routine.

The **Copy Routine** dialog is displayed.

4 Edit the name or other parameters according to your requirement.

5 Tap **Apply**.

A copy of the selected routine is created.

How to make all declarations is detailed in section *Creating a new routine on page 100*.

Changing the declaration of a routine

This section describes how to change the declaration of a routine.

1 On the start screen, tap **Code**, and then select **Program Editor** from the menu.

2 Tap **View Routines** on the **Context menu** for the module.

3 Tap **Change Declaration** on the **Context menu** for the routine.

The **Change Declaration** dialog is displayed.

4 Edit the values according to your requirement.

5 Tap **Apply**.

The changes to the routine are saved.

How to make all declarations is detailed in section *Creating a new routine on page 100*.

Continues on next page

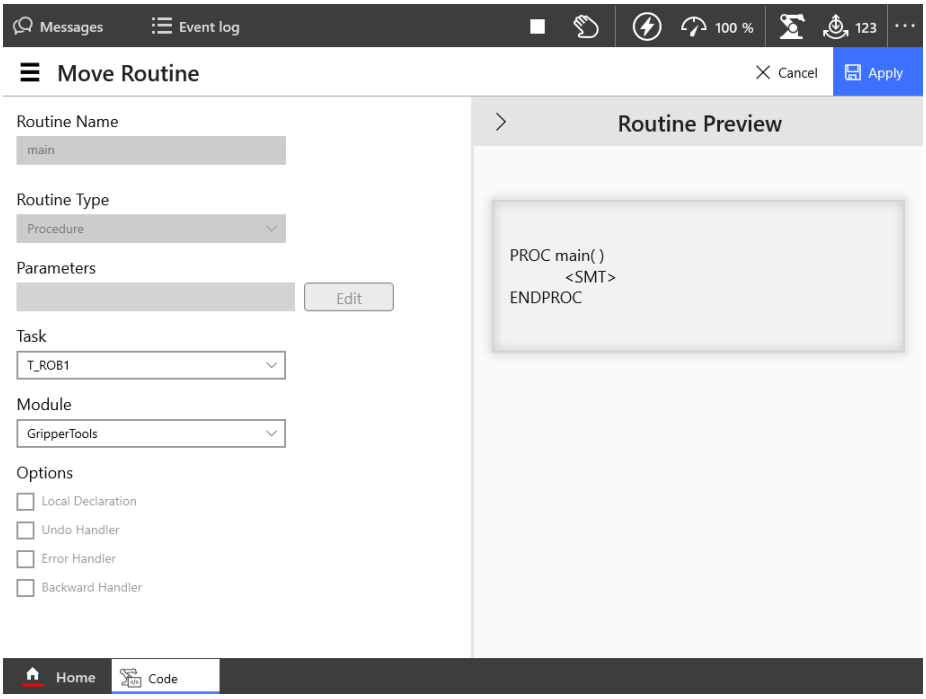
4.3.3 Handling of routines
Continued

Moving a routine

This section describes how to move a routine to another module.

1. On the start screen, tap **Code**, and then select **Program Editor** from the menu.
2. Tap **View Routines** on the **Context menu** for the module.
3. Tap **Move Routine** on the **Context menu** for the routine.

   The **Move Routine** dialog is displayed.



xx1900000310

4. Select the **Task** and **Module** to which the routine should be moved. Then tap **Apply**.

Renaming a routine

This section describes how to rename a routine.

1. On the start screen, tap **Code**, and then select **Program Editor** from the menu.
2. Tap **View Routines** on the **Context menu** for the module.
3. Tap **Rename** on the **Context menu** for the routine.
4. Type a new name for the routine in the **Enter New Name** field.
5. Tap **Apply**.

   The selected routine is renamed.

Deleting a routine

This section describes how to delete a routine from memory.

1. On the start screen, tap **Code**, and then select **Program Editor** from the menu.

Continues on next page

2 Tap **View Routines** on the **Context menu** for the module.
3 Tap **Delete** on the **Context menu** for the routine.
The **Delete Routine** conformation window is displayed.
4 Tap **OK**.
The selected routine is deleted.

4.4 Data types

4.4.1 Creating new data instance

Creating new data instance

This section details how to create new data instances of data types.

1 On the start screen, tap **Program Data**.

The **Data Types** dialog is displayed.

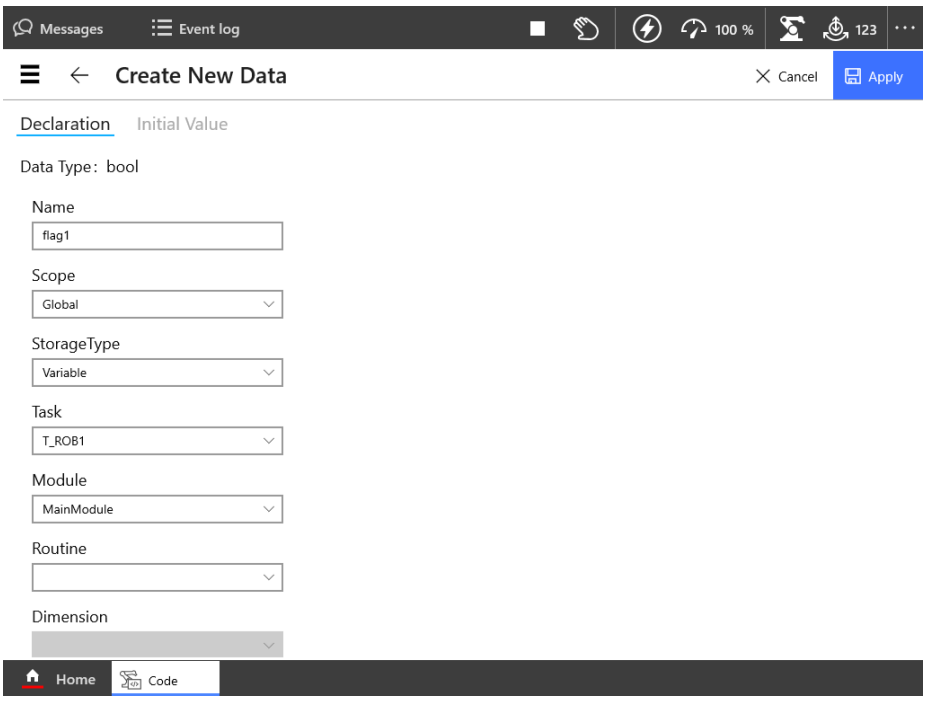
> **Tip**
>
> Define what data to be displayed by selecting either **Only used types** or **All data types**.

2 Tap the data instance type to be created, for example, **bool**.

3 Tap the context menu, and select **Create New Data**.

The **Create New Data** dialog is displayed.

4 In the **Declaration** tab, complete the following fields for the new data type:



xx1900000351

- **Name**: Type a new name.
- **Scope**: Set the accessibility for the data instance from the following options:
  - **Global** - reachable by all the tasks.
  - **Local** - reachable within the module.
  - **Task** - reachable within the task.

Continues on next page

- **Storage type**: Set the type of memory used for the data instance from the following options:
  - **Persistent** to retain the data between sessions.
  - **Variable** if the data instance is variable.
  - **Constant** if the data instance is constant.
- Tap the **Module** menu to select module.
- Tap the **Routine** menu to select routine.

5 In the **Initial Value** tab, select the values according to the selected data type.

6 Tap **Apply**.

The new data instance is created.

4.4.2 Creating user-defined data types

Introduction

A user-defined data type allows you to customize data according to your needs. The user-defined data types are created from RobotStudio.

Procedure

Use the following procedure to create a user-defined data type in RobotStudio and to use it in FlexPendant:

1. In the RobotStudio Controller tab ribbon, click **Request Write Access** in the **Access** group.

   The **Request for write access** window is displayed in FlexPendant.
2. Tap **Grant**.

   The write access is enabled in RobotStudio.
3. In the RobotStudio RAPID tab, under Controller browser, open the Module in which the data type needs to be created.

   The selected module is opened in the program editor.
4. Place the cursor just below the MODULE declaration and type the RECORD declaration. A sample RECORD declaration is shown in the following example.

   ```
   RECORD myRecordData1
     num myRecordData1_num;
     string myRecordData1_str;
     bool myRecordData1_bool;
   ENDRECORD
   ```

   Note

   If the RECORD declaration is not placed at the correct position, a syntax error is displayed for the module.

   Note

   The name of each data type must be unique.
5. In the RAPID tab ribbon, under **Controller** group, click **Apply**, and select **Apply All**.

   The changes are saved to the controller.
6. In the RobotStudio Controller tab ribbon, click **Release Write Access** in the **Access** group.
7. In FlexPendant open **Code** > **Program Data** > **All data types**.

   The newly created data type is displayed along with the other data types. Once a user-defined data type is created, you can create and edit data instance for this data type.

4.5 Tools

4.5.1 Creating a tool

What happens when you create a tool?

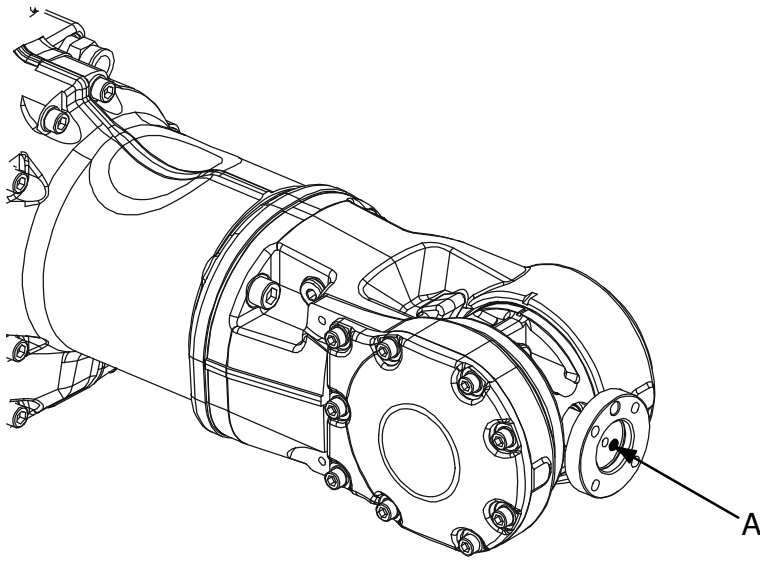
When you create a new tool a variable of the data type `tooldata` is created. The variable name will be the name of the tool. For more information on data types, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

The new tool has initial default values for mass, frame, orientation etc., which must be defined before the tool can be used.

How to create a tool

The tool center point of the default tool (tool0) is in the center of the robot's mounting flange and shares the orientation of the robot base.

By creating a new tool you define another tool center point. For more information about tool center points, see *What is the tool center point? on page 235*.



en0400000779

A	Tool center point, TCP, for tool0

1. On the start screen, tap **Program Data**.
2. Select **tooldata** from the **Data Types** list.
3. Tap **Create New Data** in the menu to the right.

   The **Create New Data** window is displayed.

Continues on next page

4.5.1 Creating a tool
Continued

4 Complete the tool information by typing or selecting information in the fields available under the **Declaration** and **Initial Value** tabs (see *Tool declaration settings on page 112*).

> **Note**
>
> In the **Mass** field under **Initial Value** tab, make sure to provide the mass of the attached tool in kg units.

5 Tap **Apply**.

The tool is created.

Tool declaration settings

If you want to change...	then...	Recommendation
the name of the tool	tap on the **Name** field and change name using the soft keyboard that appears.	Tools are automatically named `tool` followed by a running number, for example `tool10` or `tool21`. You are recommended to change this to something more descriptive such as gun, gripper or welder. **Note** If you change the name of a tool after it is referenced in any program you must also change all occurrences of that tool.
the scope	select the preferred scope from the **Scope** drop-down list.	Tools should be global, if it should be available to all the modules in the program.
the storage type	select the value from the **Storage Type** drop-down list.	Tool variables must always be persistent.
the task	select the value from the **Task** drop-down list.	
the module	select the module in which this tool should be declared from the **Module** drop-down list.	
the routine	select the value from the **Routine** drop-down list.	
the size of the data array's axes	select the value from the **Dimension** drop-down list.	

> **Note**
>
> The created tool is not useful until you have defined the tool data (TCP coordinates, orientation, weight etc.). See *Editing the tool data on page 117* and the section **Load identification service routine** in *Operating manual - OmniCore*.

4.5.2 Defining the tool frame

Preparations

To define the tool frame, you first need a reference point in the world coordinate system. If you need to set the tool center point orientation, you also need to affix elongators to the tool.

You also need to decide which method to use for the tool frame definition.

Available methods

There are three different methods which can be used when defining the tool frame. All three require that you define the cartesian coordinates of the tool center point. What differs is how the orientation is defined.

If you want to...	...then select
set the orientation the same as the orientation of the robot's mounting plate	**TCP (default orient.)**
set the orientation in Z axis	**TCP&Z**
set the orientation in X and Z axes	**TCP&Z,X**

How to define position and orientation

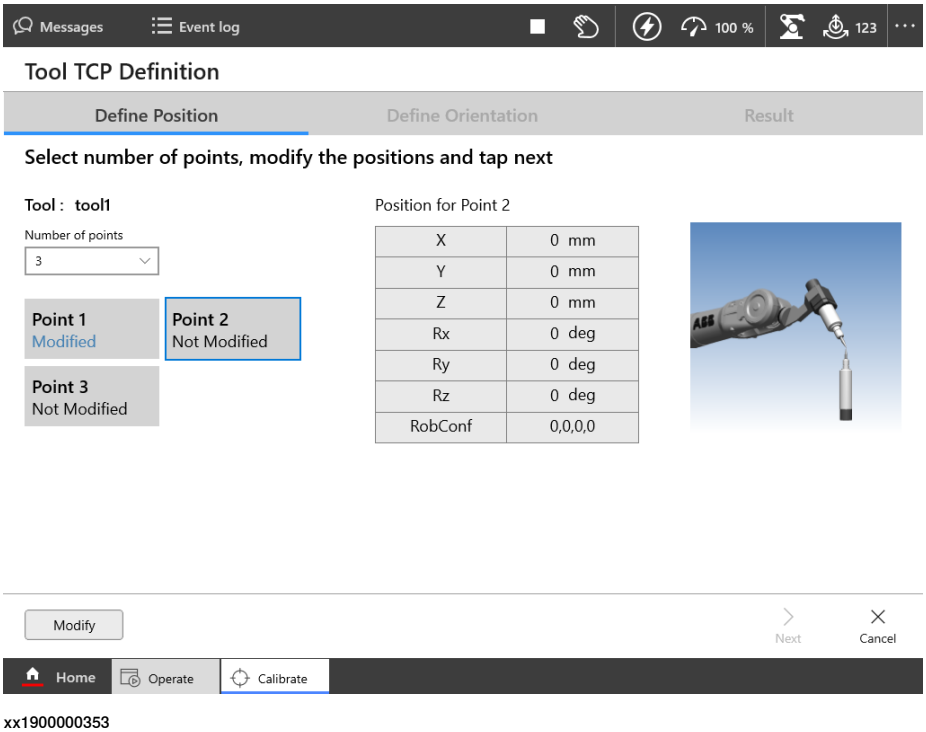
The following procedure describes how to select the method to be used when defining the tool frame:

1. On the start screen, tap **Program Data**, and then select **tooldata**.

   The data of type 'tooldata' is displayed.

2. Tap on the context menu of the tool that you want to edit, and select **Define**.

   The **Tool TCP Definition** window for the selected tool is displayed.



xx1900000353

Continues on next page

4.5.2 Defining the tool frame

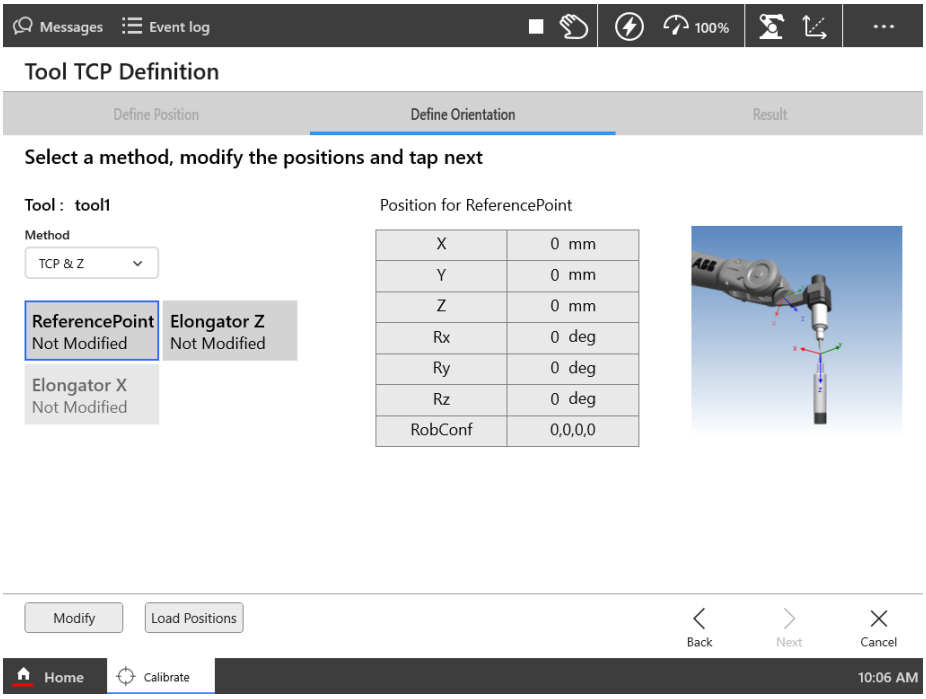
Continued

3 Select the number of approach points from the **Number of points** field. Usually 4 points are enough. If you choose more points to get a more accurate result, you should be equally careful when defining all of them.

4 Select a point, jog the robot to a required position, and then tap **Modify** to define the selected points. Repeat this step for all the points. See *How to proceed with tool frame definition on page 115*.

5 Tap **Next**.

The **Tool TCP Definition, Define Orientation** window is displayed.



xx1900000354

6 Select the **Method** to be used.

7 Select a point and tap **Modify** to modify the positions.

8 Tap **Next**.

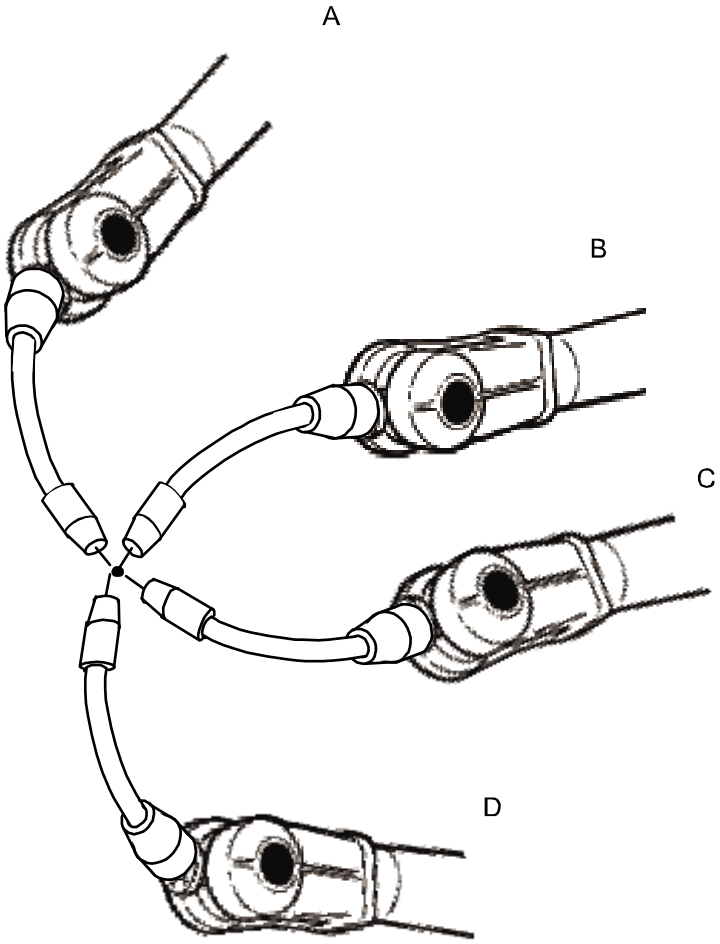
The **Tool TCP Definition, Calibration Result** window is displayed.

9 Tap **Finish** to save the calibration.

Continues on next page

How to proceed with tool frame definition

This procedure describes how to define the tool center point in Cartesian coordinates.



en0400000906

1 Jog the robot to an appropriate position, A, for the first approach point.

   Use small increments to accurately position the tool tip as close to the reference point as possible.

2 Tap **Modify** to define the point.

3 Repeat step 1 and 2 for each approach point to be defined, positions B, C, and D.

   Jog away from the fixed world point to achieve the best result. Just changing the tool orientation will not give as good a result.

4 If the method you are using is TCP&Z or TCP&Z,X orientation must be defined as well.

   Follow the instructions in *How to define elongator points on page 116*.

5 If, for some reason, you want to redo the calibration procedure described in step 1-4, tap **Cancel**.

Continues on next page

4.5.2 Defining the tool frame
Continued

6 Tap **Next**. The **Calculation Result** dialog box will now be displayed, asking you to cancel or to confirm the result before it is written to the controller.

For further information see *Is the calculated result good enough? on page 116*

How to define elongator points

This procedure describes how to define the orientation of the tool frame by specifying the direction of the z and/or x axis. You need to do this only if you the tool orientation should differ from that of the robot base. The tool coordinate system by default resembles the coordinate system of tool0, as illustrated in *Measuring the tool center point on page 118*.

1 Without changing the orientation of the tool, jog the robot so that the reference world point becomes a point on the desired positive axis of the rotated tool coordinate system.
2 Tap **Modify** to define the point.
3 Repeat step 1 and 2 for the second axis if it should be defined.

Is the calculated result good enough?

The **Calculation Result** dialog box displays the calculated result of the tool frame definition. You have to confirm that you accept the result before it can take effect in the controller. The alternative is to redo the frame definition in order to achieve a better result. The result **Mean Error** is the average distance of the approach points from the calculated TCP (tool center point). **Max Error** is the maximum error among all approach points.

It is hard to tell exactly what result is acceptable. It depends on the tool, robot type etc. you are using. Usually a mean error of a few tenths of a millimeter is a good result. If the positioning has been undertaken with reasonable accuracy the result will be okay.

As the robot is used as a measuring machine, the result is also dependent on where in the robot's working area the positioning has been done. Variation of the actual TCP up to a couple of millimeters (for large robots) can be found between definitions in different parts of the working area.The repeatability of any following TCP calibrations will thus increase if these are done close to the preceding ones. Note that the result is the optimal TCP for the robot in that working area, taking into account any discrepancies of the robot in the configuration at hand.

Tip

A common way to check that the tool frame has been correctly defined is to perform a reorientation test when the definition is ready. Select the reorient motion mode and the tool coordinate system and jog the robot. Verify that the tool tip stays very close to the selected reference point as the robot moves.

4.5.3 Editing the tool data

Tool data

Use the value settings to set the tool center point position and physical properties of the tool such as weight and center of gravity.

This can also be done automatically with the service routine LoadIdentify. See *Operating manual - OmniCore*.

> **CAUTION**
>
> If the tooldata is incorrectly defined there is a risk that the speed is higher than expected. This is particularly important in manual mode.

Editing the tool data

This section details how to edit the tool data.

1. On the start screen, tap **Program Data**,
2. Select **tooldata** from the list of data types.
3. Tap on the menu button next to the tool that you want to edit.

   The context menu is displayed.
4. Tap **Edit**.

   The **Edit Data** page is displayed.
5. In the **Declaration**, **Initial Value**, and **Current Value** tabs edit the parameters according to your requirement.
6. Tap **Apply**.

   The changes are saved.

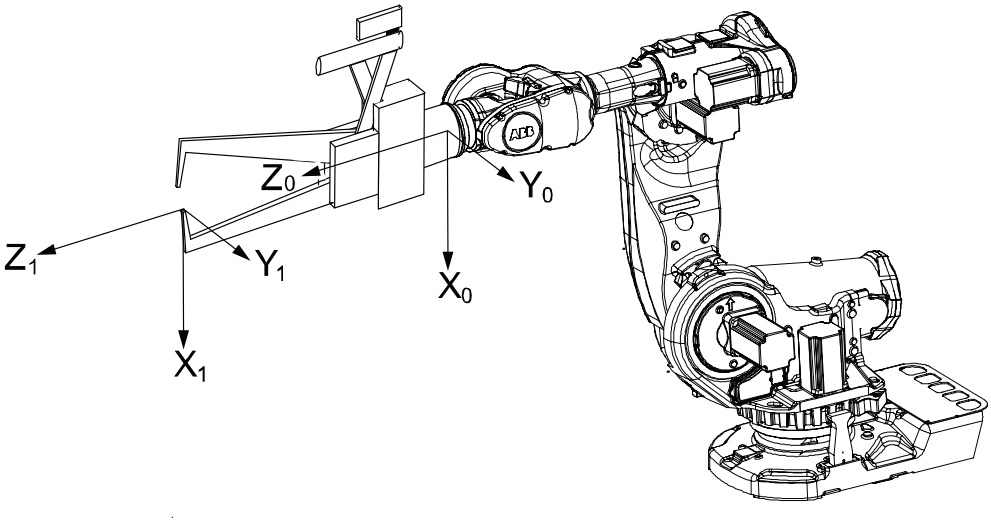
Continues on next page

4.5.3 Editing the tool data
Continued

Measuring the tool center point

The easiest way to define the tool center point, TCP, is usually to use the predefined method described in *Defining the tool frame on page 113*. If you use this method, you do not have to write any values for the frame as these are supplied by the method.

If you already have the measurements of the tool, or for some reason want to measure them manually, the values can be entered in the tool data.



en0400000881

X_0	X axis for tool0
Y_0	Y axis for tool0
Z_0	Z axis for tool0
X_1	X axis for the tool you want to define
Y_1	Y axis for the tool you want to define
Z_1	Z axis for the tool you want to define

1. Measure the distance from the center of the robot's mounting flange to the tool's center point along the X axis of tool0.
2. Measure the distance from the center of the robot's mounting flange to the tool's center point along the Y axis of tool0.
3. Measure the distance from the center of the robot's mounting flange to the tool's center point along the Z axis of tool0.

Editing the tool definition

	Action	Instance	Unit

1	Enter the cartesian coordinates of the tool center point's position.	`tframe.trans.x`
		`tframe.trans.y`
		`tframe.trans.z`
			[mm]

Continues on next page

	Action	Instance	Unit

2	If necessary, enter the tool frame orientation.	`tframe.rot.q1`
		`tframe.rot.q2`
		`tframe.rot.q3`
		`tframe.rot.q4`
			None

3	Enter the weight of the tool.	`tload.mass`	[kg]

4	If necessary, enter the tool's center of gravity.	`tload.cog.x`
		`tload.cog.y`
		`tload.cog.z`
			[mm]

5	If necessary, enter the orientation of the axis of moment	`tload.aom.q1`
		`tload.aom.q2`
		`tload.aom.q3`
		`tload.aom.q4`
			None

6	If necessary, enter the tool's moment of inertia.	`tload.ix`
		`tload.iy`
		`tload.iz`
			[kgm^2]

7	Tap **Save** to use the new values, **Cancel** to leave the definition unchanged.		

4.5.4 Setup for stationary tools

Stationary tools

Stationary tools are used, for instance, in applications that involve large machines such as cutters, presses and punch cutters. You may use stationary tools to perform any operation that would be difficult or inconvenient to perform with the tool on the robot.

With stationary tools, the robot holds the work object.

Make a tool stationary

This section describes how to make a tool stationery.

1. On the start screen, tap **Program Data**,
2. Select **tooldata** from the list of data types.
3. Tap on the menu button next to the tool that you want to edit.

   The context menu is displayed.
4. Tap **Edit**.

   The **Edit Data** page is displayed.
5. Tap the **Current Value** tab.

   The data that defines the selected tool is displayed.
6. In the **robhold** field set the value to **FALSE**.
7. Tap **Apply**.

   The selected tool is made stationery.

Make a work object robot held

This section describes how to make a work object robot held.

1. On the start screen, tap **Program Data**,
2. Select **wobjdata** from the list of data types.
3. Tap on the menu button next to the workobject that you want to edit.

   The context menu is displayed.
4. Tap **Edit**.

   The **Edit Data** page is displayed.
5. Tap the **Current Value** tab.

   The data that defines the selected tool is displayed.
6. In the **robhold** field set the value to **TRUE**.
7. Tap **Apply**.

   The changes are saved.

Set up the tool coordinate system

You use the same measurement methods to set up a stationary tool coordinate system as with tools mounted on the robot.

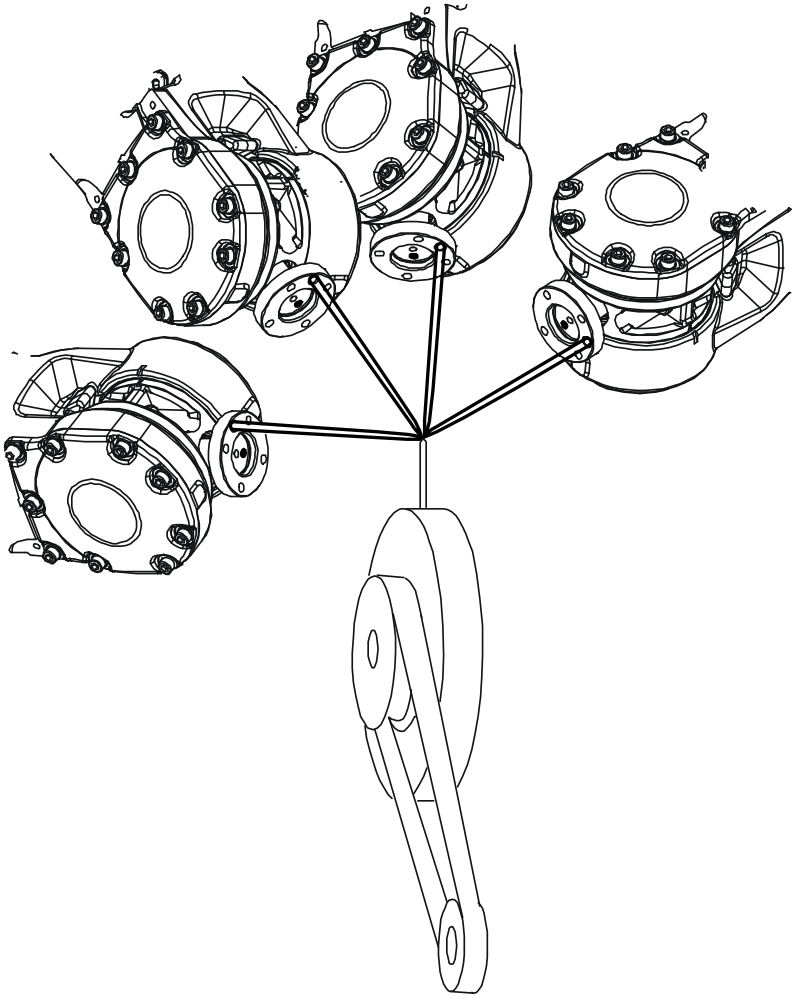
The world reference tip must, in this case, be attached to the robot. Define and use a tool with the reference tip’s measurements when you create approach points.

Continues on next page

You also need to attach elongators to the stationary tool if you need to set up the orientation.

You should enter the reference tip's tool definition manually to minimize errors when calculating the stationary tool's coordinate system.

You may enter the stationary tool's definition manually.



en0400000990

4.6 Work objects

4.6.1 Creating a work object

What happens when I create a work object?

A variable of the type `wobjdata` is created. The variable's name will be the name of the work object. For more information on data types, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

This is detailed in section *What is a work object? on page 237*.

Creating a work object

The work object's coordinate system is now identical with the world coordinate system.

To define the position and orientation of the work object's coordinate system, see *Work object declaration settings on page 122*.

1. On the start screen, tap **Program Data**.
2. Select **wobjdata** from the menu.

   The list of data of type wobjdata is displayed.
3. Tap **Create New Data** in the menu to the right.
4. In the **Declaration**, **Initial Value** tabs select the parameters according to your requirement.
5. Tap **Apply**

   The work object is created.

Work object declaration settings

If you want to change...	then...	Recommendation
the work object's name	enter a name in **Name**	Work objects are automatically named `wobj` followed by a running number, for example `wobj10`, `wobj27`. You should change this to something more descriptive. If you change the name of a work object after it is referenced in any program you must also change all occurrences of that work object.
the scope	select the scope of choice from the menu	Work objects should always be global to be available to all modules in the program.
the storage type	-	Work object variables must always be persistent.
the task	select the preferred task from the menu	
the module	select the module in which this work object should be declared from the menu	

4.6.2 Defining the work object coordinate system

Overview

Defining a work object means that the robot is used to point out the location of it. This is done by defining three positions, two on the x-axis and one on the y-axis.

When defining a work object you can use either the user frame or the object frame or both. The user select frame and the object frame usually coincides. If not, the object frame is displaced from the user frame.

How to define the work object coordinate system

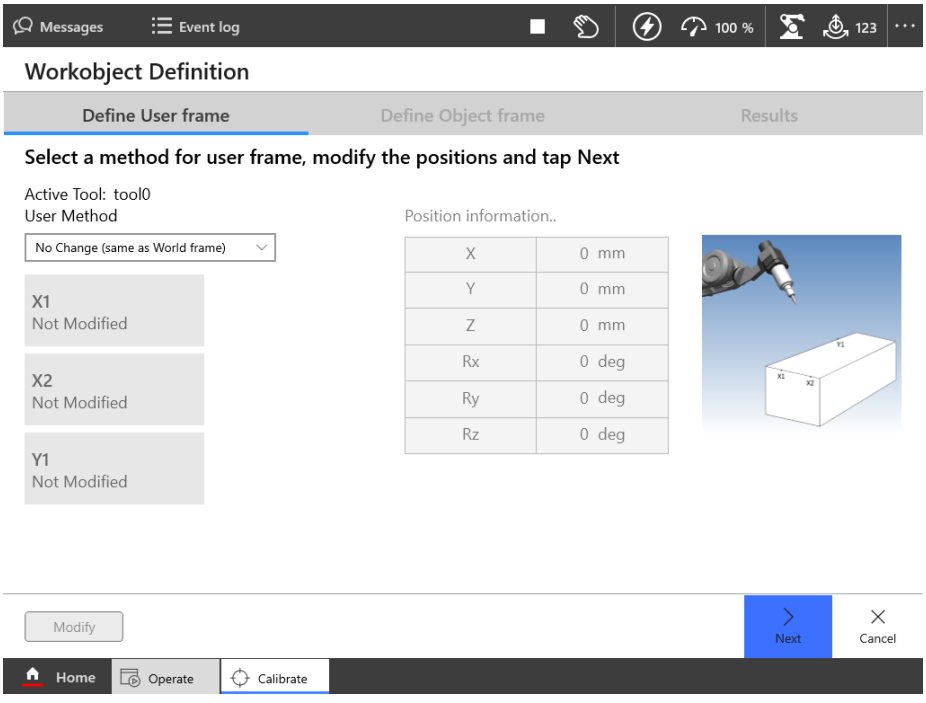
This procedure describes how to define the work object coordinate system. Note that this only works for a user created work object, not the default work object, wobj0.

1 On the start screen, tap **Program Data**, and then select **Workobject** from the list of data types.

2 Tap on the context menu next to the workobject that you want to define and select **Define**.

The **Workobject Definition** page is displayed. The **Define User frame** tab is displayed by default.

3 Select method from the **User method** drop down menu.



xx1900000451

4 Tap **Modify** to define the points. See *How to define the user frame on page 124*.

5 Tap **Next**.

The **Workobject Definition, Define Object frame** window is displayed.

6 Select the **Object Methods** to be used.

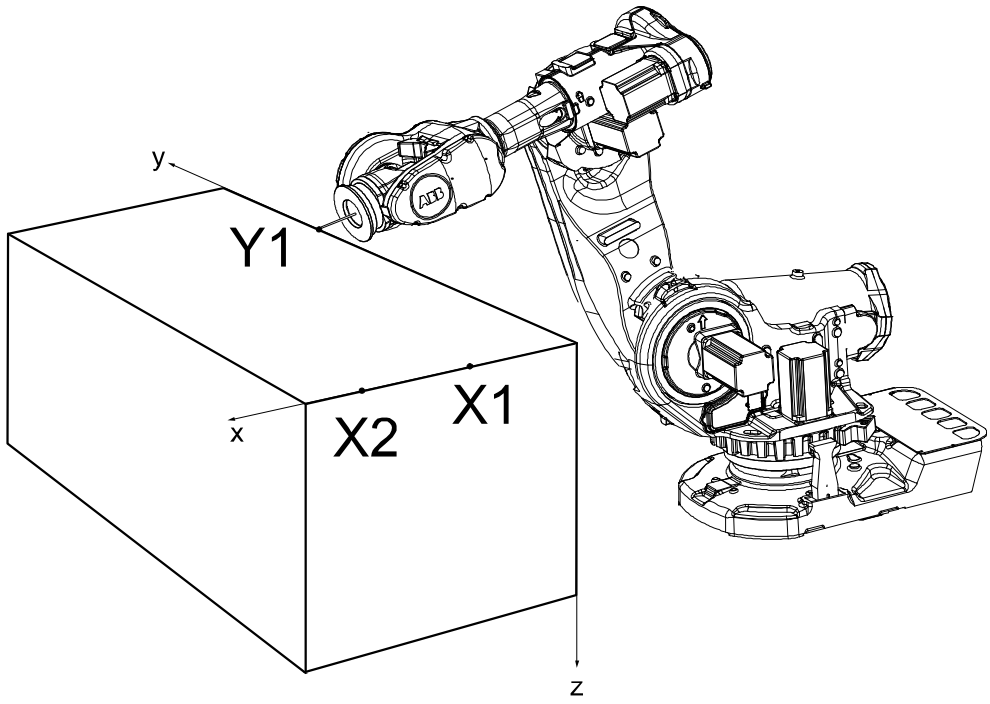
Continues on next page

4.6.2 Defining the work object coordinate system
Continued

7 Tap **Modify** to modify the positions. See *How to define the object frame on page 125*.
8 Tap **Next**.
The **Workobject Definition, Calculation Result** window is displayed.
9 Tap **Finish** to save the calibration.

How to define the user frame

This section details how to define the user frame.



en0400000887

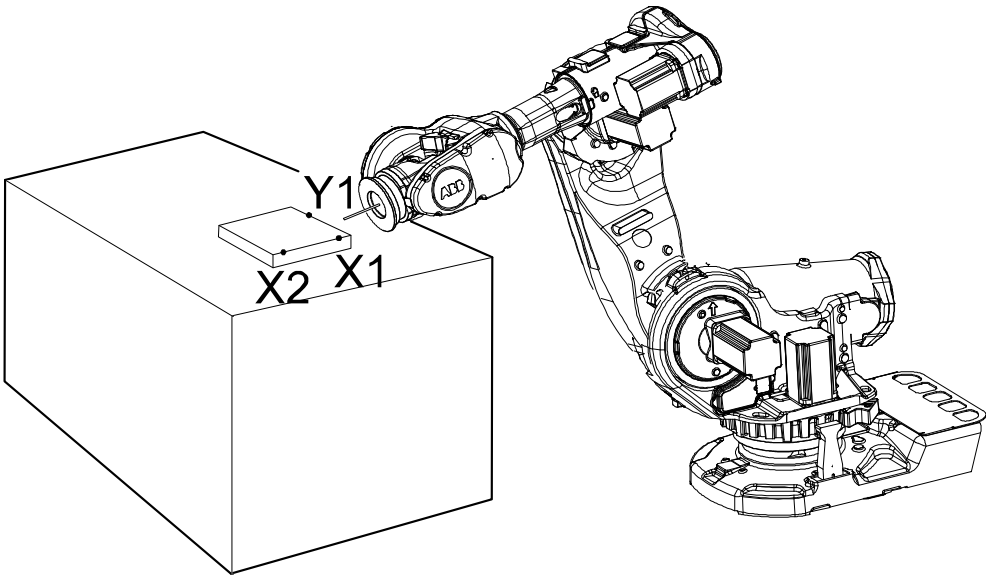
The x axis will go through points X1-X2, and the y axis through Y1.

1 In the **User method** drop down menu, select **User defined with 3 points**.
2 Press the three-position enabling device and jog the robot to the first (X1, X2 or Y1) point that you want to define.
Large distance between X1 and X2 is preferable for a more precise definition.
3 Select the point in the list.
4 Tap **Modify** to define the point.
5 Repeat steps 2 to 4 for the remaining points.

Continues on next page

How to define the object frame

This section describes how to define the object frame if you want to displace it from the user frame.



en0400000899

The x axis will go through points X1-X2, and the y axis through Y1.

1 In the **Object Methods** drop down menu, select **User defined with 3 points**.
2 See steps 2 to 4 in the description of *How to define the user frame on page 124*.

4.6.3 Editing the work object data

Overview

Use the work object data definition to set the position and rotation of the user and object frames.

How to display the work object data

1. On the start screen, tap **Program Data**, and then select **Workobject** from the list of data types.
2. Tap on the context menu next to the workobject that you want to edit and select **Edit**.

   The **Edit Data** page is displayed.
3. Tap the **Declaration, Initial Value, and Current Value** tabs.
4. View the values according to your requirement.

How to set user and object frame values manually

The easiest way to set the work object and user coordinate systems position is to use the method described in *Defining the work object coordinate system on page 123*. You can however edit the values manually using the following guide.

Values	Instance	Unit

The cartesian coordinates of the position of the object frame	`oframe.trans.x`
	`oframe.trans.y`
	`oframe.trans.z`
		mm

The object frame orientation	`oframe.rot.q1`
	`oframe.rot.q2`
	`oframe.rot.q3`
	`oframe.rot.q4`
		-

The cartesian coordinates of the position of the user frame	`uframe.trans.x`
	`uframe.trans.y`
	`uframe.trans.z`
		mm

The user frame orientation	`uframe.rot.q1`
	`uframe.rot.q2`
	`uframe.rot.q3`
	`uframe.rot.q4`
		-

Note

Editing work object data can also be done from the **Code** window.

4.6.4 Editing the work object declaration

Overview

Use the declaration to change how the work object variable can be used in the program's modules.

Displaying the work object declaration

1. On the start screen, tap **Calibrate**, and then select **Workobject** from the menu.
2. Tap the work object you want to edit.

   The **Edit Workobject** dialog is displayed.
3. Tap **Declaration**.

   The work object's declaration appears.
4. Proceed with changing the data as described in *Work object declaration settings on page 122* and then tap **Save**.

> **Note**
>
> If you change the name of a work object after it is referenced in any program you must also change all occurrences of that work object.

4.7 Payload

4.7.1 Creating a payload

What happens when I create a payload?

When you create a payload, a variable of the type `loaddata` is created. The variables name will be the name of the payload. For more information on data types, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

Adding a new payload and setting data declaration

The payloads coordinate system will be set to the position, including orientation, of the world coordinate system.

1. On the start screen, tap **Program Data**, and then select **loaddata**.

   The loaddata items are displayed.
2. Tap **Create New Data** in the menu to the right.

   The **Create New Data** window is displayed.
3. Complete the payload information (see *Payload declaration settings on page 129*).
4. Tap **Apply**.

   The payload is created.

WARNING

It is important to always define the actual tool load and, when used, the payload of the robot (for example, a gripped part). Incorrect definitions of load data can result in overloading of the robot mechanical structure. There is also a risk that the speed in manual reduced speed mode can be exceeded.

When incorrect load data is specified, it can often lead to the following consequences:

- The robot may not use its maximum capacity.
- Impaired path accuracy including a risk of overshooting.
- Risk of overloading the mechanical structure.

The controller continuously monitors the load and writes an event log if the load is higher than expected. This event log is saved and logged in the controller memory.

Continues on next page

Payload declaration settings

If you want to change...	...then...	Recommendation
the payload's name	enter a name in **Name**	Payloads are automatically named `load` followed by a running number, for example `load10`, `load31`. You should change this to something more descriptive. If you change the name of a payload after it is referenced in any program you must also change all occurrences of that payload's name.
the scope	select the scope of choice from the menu	Payloads should always be global to be available to all modules in the program.
the storage type	-	Payload variables must always be persistent.
the task	select the preferred task from the menu	
the module	select the module in which this payload should be declared from the menu	-

Setting the value for ModalPayLoadMode

The **ModalPayLoadMode** is defined in RobotStudio. See *Operating manual - RobotStudio*.

4.7.2 Editing the payload data

Overview

Use the payload data to set physical properties of the payload such as weight and center of gravity.

This can also be done automatically with the service routine LoadIdentify. See *Operating manual - OmniCore*.

Displaying the payload definition

1. On the start screen, tap **Program Data**, and then select **loaddata** from the list of data types.

   The **Data of type 'loaddata'** page displays the list of available loaddata.
2. Tap on the context menu next to the loaddata that you want to edit and select **Edit**.
3. Tap the payload for which you want to view the values.

   The **Edit Data** window is displayed.
4. Tap the **Current Value** tab and view the values.

Changing the payload data

This procedure describes how to manually enter the payload data. This can also be done automatically by running the service routine LoadIdentify.

	Action	Instance	Unit
1	Enter the weight of the payload.	`load.mass`	[kg]

2	Enter the payload's center of gravity.	`load.cog.x`
		`load.cog.y`
		`load.cog.z`
			[mm]

3	Enter the orientation of the axis of moment.	`load.aom.q1`
		`load.aom.q2`
		`load.aom.q3`
		`load.aom.q3`	

4	Enter the payload's moment of inertia.	`ix`
		`iy`
		`iz`
			[kgm^2]

5	Tap **Apply** to use the new values, **Cancel** to leave the data unchanged.	-	-

Using the `PayLoadsInWristCoords` parameter

By using the `PayLoadsInWristCoords` parameter, the loaddata for payloads can be specified relative to the wrist instead of the active TCP or work object. This can be useful if several tool or TCP or work objects (when tool is stationary) are used for one payload. In this case only one load identification is needed instead of one for each tool or TCP or work object. Thus it is possible to use the same payload loaddata for any robhold or stationary tool being active. This saves the time (for example, during commissioning).

Continues on next page

For more information about `PayLoadsInWristCoords`, see *Technical reference manual - System parameters* and *Technical reference manual - RAPID Instructions, Functions and Data types*.

4.8 Advanced programming

4.8.1 Editing instruction expressions and declarations

Expressions

An expression specifies the evaluation of a value. It can be used, for example:

- as a condition in an IF instruction
- as an argument in an instruction
- as an argument in a function call

Read more in *Technical reference manual - RAPID Overview* and *Technical reference manual - RAPID Instructions, Functions and Data types*.

Editing expressions

Use the following procedure to edit expressions in instructions.

1. On the start screen, tap **Code**, and then select **Code Editor** from the menu.
2. Tap and select the instruction that you want to change.
3. Tap **Modify Instruction**.
4. Tap **Expression Editor**.
5. Select an expression for editing. Following are the available options:
   - **+**: Adds an expression next to the current selected expression. Tap the new expression to define it.
   - **-**: Deletes an expression.
   - **()**: Inserts a parenthesis set around the selected expression.
   - **(x)**: Deletes a parenthesis set.
   - **Expression**:Allows you to edit the array expression elements.
   - **Change DataType**: Allows you to change the data type. This is detailed in section *Changing data type on page 132*.
   - **ABC**: Displays the soft keyboard.
   - Search bar: Allows you to filter the robtargets and quickly add it to the expression editor.
   - **New RAPID Data**: Creates a new data declaration, that is, adding a data declaration not previously used.
6. Tap **Apply**.

   The changes are saved.

Declarations and data types

When editing an expression, new data can be declared with the button **New**. More information about data declarations and how to edit them can be found in *Operating manual - OmniCore*.

Changing data type

Use the following procedure to change the data type:

1. On the start screen, tap **Code**, and then select **Code Editor** from the menu.

Continues on next page

2 Tap and select the instruction that you want to change.
3 Tap **Modify Instruction**.
4 Tap **Expression Editor**.
5 Select the expression for which you want to change the data type.
6 Tap **Change DataType**.
   The Modify Instruction window is displayed.
7 Select the required data type from the list.
8 Tap **OK**.
9 Tap **Apply**.
   The changes are saved.
10 Tap **Apply**.
   The changes are saved.

4.8.2 Deleting programs from hard disk

Overview

Programs can be deleted using **File Transfer** in RobotStudio. When deleting programs from the controller hard disk, the currently loaded program in the program memory is not affected.

The different memories are described in section *What is the memory? on page 145*.

4.8.3 Activating mechanical units

Overview

A mechanical unit can be active or deactive. Only active units are run when executing a program. Deactivated units will not run. This may be useful when programming or testing a program.

A robot **cannot** be deactivated.

The Activate function does not affect jogging. To select mechanical unit for jogging, use the **Mechanical unit** property in the **Jogging** menu.

Activating mechanical units

This procedure describes how to activate a mechanical unit.

1. On the start screen, tap **Jog**.
2. Make sure that the right mechanical unit is selected, then tap **Activate**.

   To deactivate an active mechanical unit, tap **Deactivate**.

> **Note**
>
> A robot cannot be deactivated.

Related information

Mechanical units can be active or deactive at start depending on the system setup, see *Technical reference manual - System parameters*, topic *Motion*.

This page is intentionally left blank

5 I/O signals

5.1 Basic procedures

5.1.1 Configuring I/O

Creating and editing industrial networks, devices and signals

The configuration of I/O is slightly different for different industrial networks. How to create and edit networks, devices and signals are described in the respective industrial network manuals.

Note

Two industrial network masters can be run in parallel on the OmniCore controller. It is the responsibility of the integrator to verify the behavior when two masters are used in one OmniCore.

5.1.2 Activating or deactivating I/O devices

Overview

Deactivating an I/O device makes the controller ignore the device. This can be useful during commissioning, for avoiding errors if the I/O device is not connected to the controller yet. The signals configured on the device are still be visible when it is deactivated, but the signal values are not available. The controller will not attempt to send or receive any signals on a deactivated device.

Activating the unit again will take it back to normal operation.

Procedure

Use the following procedure to activate or deactivate I/O devices:

1. On the start screen, tap **I/O**, and then select **I/O Devices** from the menu.
2. Use the search box to navigate to a particular device.
3. Select a device, and select **Activate** or **Deactivate** on the **Context menu**.

   The selected device is activated or deactivated.

Note

All the signals on the I/O device must have an access level that allows the local clients (for example, the FlexPendant) to have write access. If not, the device cannot be activated or deactivated from local clients. The access level is set with system parameters for each signal, see the types *Signal* and *Access Level* in the topic *I/O*.

Note

The device cannot be deactivated if the system parameter *Unit Trustlevel* is set to *0 (Required)*. *Unit Trustlevel* belongs to the type *Unit* in the topic *I/O*.

5.1.3 Alias I/O signals

Introduction

Alias I/O is used to define a signal of any type with an alias name. After the Alias I/O instruction is executed in the RAPID program, the Alias I/O signal can be viewed from the Alias I/O menu in the same way as the other signals from the Signal page of the I/O application.

Creating new signal data

Use the following procedure to create new alias I/O signal:

1. On the start screen, tap **Code**, and then select **RAPID data** from the menu.
2. In the **Data Types** section filter options, select **All data types**.
3. In the search box type `signaldi`.
4. On the **Context menu**, tap **Create New Data**.
5. In the **Name** field define the name of data instance. For example, **alias_di1**.
6. Tap the **Scope** list and select **Global**.
7. Tap **Apply**.

   The alias signal is created.
8. Repeat the preceding steps to create a **signaldo** data instance. For example, **alias_do1**.

Note

Consider the following example,

```
VAR signaldo alias_do1;
AliasIO do_1, alias_do1;
```

VAR declaration must be done global in the module.

After declaring `signaldi` and `signaldo` and executing the instruction `AliasIO do_1, alias_do1` the `alias_do1` signal is displayed in the Alias I/O menu in the same way the ordinary signals are displayed in the **I/O** application.

The `alias_do1` signal is active as long as the RAPID program is active and is displayed after the `AliasIO` instruction is executed.

Adding Alias I/O

Use the following procedure to add Alias I/O instructions and to view them on **Alias I/O** menu:

Note

Before adding Alias I/O instructions you must declare the RAPID variables of data type **signaldi** and **signaldo**, for example,**alias_di1** and **alias_do1** respectively, as described in *Creating new signal data on page 139*.

1. On the start screen, tap **Code**, and then open a program.
2. Tap **Add Instruction**.

Continues on next page

A large number of instructions, divided into several categories are available.

3 Tap the **Groups** tab and then tap **I/O**.
4 Select **AliasIO**.
5 Tap on **Exp. Edit**.
   The expression editor window is displayed.
6 For the **FromSignal** argument, tap **<EXP>** and select **Change Data Type**.
7 Select the **signaldi** data type and tap **OK**.
8 Select the signal to which the alias signal need to be associated.
   The selected signal is added to the expression editor window.
9 For the **ToSignal** argument, tap **<EXP>** and select **Change Data Type**.
10 Select **signaldi** from the list and click **OK**.
11 Select the argument value (for example, **alias_di1**) for the **To Signal** argument.
12 Tap **Apply**.
13 Tap **Check Program**.
14 Tap **Debug** and select **PP to Main**.
15 Press the **Start** button on the FlexPendant and run the program.
16 On the start screen, tap **I/O**, and then select **Signals** from the menu.
17 In the **Filter** list select the **Alias Signals** category.
   The new data declaration created from the RAPID program is available.
18 Repeat the preceding steps to add the **signaldo** data type and view from **Alias Signals** category.

Note

Currently only digital signals `signaldi` and `signaldo` are supported.

5.2 Safety signals

5.2.1 Safety I/O signals

General

In the OmniCore controller's basic and standard form, certain I/O signals are dedicated to specific safety functions. These are listed below with a brief description of each.

All signals can be viewed in the I/O menu on the FlexPendant.

Safety I/O signals

Signal name	Description	Bit value condition	From - To
scExternalEmergencyStop	External Emergency stop	1 = Closed 0 = Button is pressed	Safety controller - main computer
scAutoStop	Auto stop	1 = Closed	Safety controller - main computer
scEnableSwitch	Enabling device	1 = Enabled	Safety controller - main computer
scDeviceEnable1	Motors on/off order from main computer	1 = Ok from main computer to go to motors on state 0 = Go to motors off state	Safety controller - main computer
scDriveEnableAllowed		1 = Ok from safety controller to go to motors on state	Safety controller - main computer
scLocalEmergencyStop	Local Emergency stop	1 = Closed 0 = Button is pressed	Safety controller - main computer
scDriveEnableFeedback	Feedback signal from motor contactors	1 = Motor contactors are activated	Safety controller - main computer
scDriveEnable	Activation signal for motor contactors	1 = Activation of motor contactors	Safety controller - main computer
scEmergencyStop	Status for both emergency stops	1 = Closed 0 = Any emergency stop button is pressed	Safety controller - main computer
MotOnPB	Motors on push button	1 = Pressed	External push button - main computer
ManRSReqTPU	Manual reduced speed mode request from TPU	1 = Request	Main computer - internal status
ManFSReqTPU	Manual high speed mode request from TPU	1 = Request	Main computer - internal status

Continues on next page

5.2.1 Safety I/O signals
Continued

Signal name	Description	Bit value condition	From - To
AutoReqTPU	Auto mode request from TPU	1 = Request	Main computer - internal status
MotLmpPB	Motors on lamp	0 = Standby 1 = Motors on Blink = Guard stop	Main computer - external lamp
ManRS	Manual reduced speed mode	1 = Selected	Main computer - internal status
ManFS	Manual high speed mode	1 = Selected	Main computer - internal status
Auto	Auto mode	1 = Selected	Main computer - internal status

6 RobotWare system

6.1 What is a RobotWare system?

The RobotWare system

A RobotWare system is the software that runs on a controller. It consists of the specific RobotWare products for the robots connected to the controller, configuration files, and RAPID programs.

The RobotWare license

What products and options (supported robot models, options, etc.) that can be included in the RobotWare system is determined by the RobotWare license.

When running a RobotWare system on a real controller it has to be built with the license that was delivered with the robot or purchased from ABB Robotics.

For running a RobotWare system on a virtual controller (e.g. for simulations in RobotStudio) either a license from a real controller or a virtual license can be used. Using a license from a real controller is a quick way to ensure that the virtual system matches the RobotWare system on a real controller. Using a virtual license provides possibility to simulate and evaluate any robot model with any configuration. A RobotWare system built with a virtual license can however never be run on a real controller.

Default configuration of RobotWare system

A new RobotWare system that only contains the RobotWare products and the default configurations is called an empty system. When robot or process specific configurations are made, I/O signals are defined or RAPID programs are created, the system is no longer considered empty.

Resetting the RobotWare system is a procedure which returns it into the default empty state. This is different from deleting the system, the procedure which completely removes the installed RobotWare system and requires new installation (see *RobotWare installation procedures on page 169*).

Empty controller

Empty Controller is a Robot Controller on which no RobotWare System is installed. The only software present on the controller in that case is the RobotWare Installation Utilities, which cannot be deleted by the user. For more informations see *RobotWare installation procedures on page 169*.

6.2 RobotWare add-ins

About RobotWare add-ins

RobotWare add-ins are software packages that can be developed by either ABB or third-party developers. The goal of add-ins is to make it easier to add new functionality to the controller or to extend or configure existing functionality. An add-in can contain any files that are recognized and supported by RobotWare, such as RAPID modules, configuration files and FlexPendant WebApps.

The add-in contents are usually created using tools such as RobotStudio. The content is then used in the Add-In Packaging tool (available for download from *ABB Library Download Center*) to create a RobotWare add-in. For more information about add-ins and the creation process, see *Application manual - RobotWare add-ins*.

6.3 Memory and file handling

6.3.1 What is the memory?

Overview

When using the term *memory*, different things may be implied:

- The controller mass memory unit (hard disk, flash disk, or other drive)
- The hard disk of some other unit connected to the same LAN as the robot controller, serving as a storage for software.

Controller mass memory unit

This is the main mass storage unit of the controller, i.e. the controller mass memory. Depending on controller version, it may be a flash disk, hard drive, or other type such as solid-state drive, and it may vary in size. It contains all necessary software for operating the robot, and is the unit on which RobotWare is installed.

LAN unit

This may be used as extra mass storage device if the one in the controller is not sufficient. It is not normally considered a part of the robot system.

6.3.2 File handling

File handling and storing

Backups, programs, and configurations etc. are saved as files in the robot system. These files are handled either in a specific FlexPendant application, such as the **Program Editor**, or in an editor in RobotStudio.

Files can be stored on a number of different drives, or memory devices, such as:

- Controller mass memory unit
- Network drive
- USB drive

These drives are all used in the same way and available when saving or opening files using an application on the FlexPendant.

Limitations

The maximum length for a file name is 99 characters and the maximum length for a file path including the file name is 247 characters.

> **Note**
>
> Some additional options may have other restrictions on the length of file names and file paths. For more information see *Application manual - Controller software OmniCore*.

Related information

Technical reference manual - Event logs for RobotWare 7.

What is the memory? on page 145

6.4 Restart, reset and recovery procedures

6.4.1 Restart overview

When do I need to restart a running controller?

ABB robot systems are designed to operate unattended for long times. There is no need to periodically restart functioning systems.

The controller has built-in uninterruptible power supply. Switching off the power of the controller will shut down the system in a controlled manner.

Restart the robot system:

- to apply configuration changes that require a restart.
- to clear a system failure (SYSFAIL) if it has occurred.

Restart types

A number of restart types are available:

Situation:	Restart type:	Detailed in section:
You want to restart the RobotWare system. The current programs and configurations will not be removed.	**Restart**	*Restart on page 149*.
You want to deactivate the system and enter the RobotWare Installation Utilities.	**Start RobotWare Installation Utilities**	*Start RobotWare Installation Utilities on page 151*.
You want to delete all user loaded RAPID programs. **Note** This cannot be undone.	**Reset RAPID program**	*Resetting RAPID program on page 153*.
You want to return to the default system settings. **Note** This will remove all user loaded RAPID programs and all user defined configurations from memory and start RobotWare System in default empty configuration, except system parameters in the topic *Communication* (SIO). The content in the **TEMP** folder in the controller will also be emptied.	**Reset RAPID program and System Parameters**	*Resetting RAPID program and system parameters on page 153*.
You want to reset the safety configuration. **Note** This will remove all user defined safety settings. The controller will start up with the basic safety settings.	**Reset safety configuration**	*Resetting safety settings on page 153*.

Continues on next page

6.4.1 Restart overview

Continued

Situation:	Restart type:	Detailed in section:
The controller failed to shut down correctly. You want to restart the controller with a recovery backup. **Note** Only users with elevated used grants can perform an auto recovery. **Note** Changes to the controller after the auto recovery backup was created will not be restored.	**Auto Recovery**	*Perform auto recovery on page 155*

Related information

Technical reference manual - Event logs for RobotWare 7.

6.4.2 Restart

What happens with my RobotWare system?

The RobotWare system will be stopped.

All system parameters and programs will be saved to the controller mass memory.

During the restart process the system state will be resumed. Static and semistatic tasks will be started. Programs can be started from the point they were stopped.

Restarting this way will activate any configuration changes entered using RobotStudio.

Restart the controller

This section describes how to restart the controller and preserve the current system/parameters and RAPID programs.

1 On the start screen, tap **Settings**.

2 At the bottom of the **Settings** screen tap **Restart Controller**.

The controller is restarted.

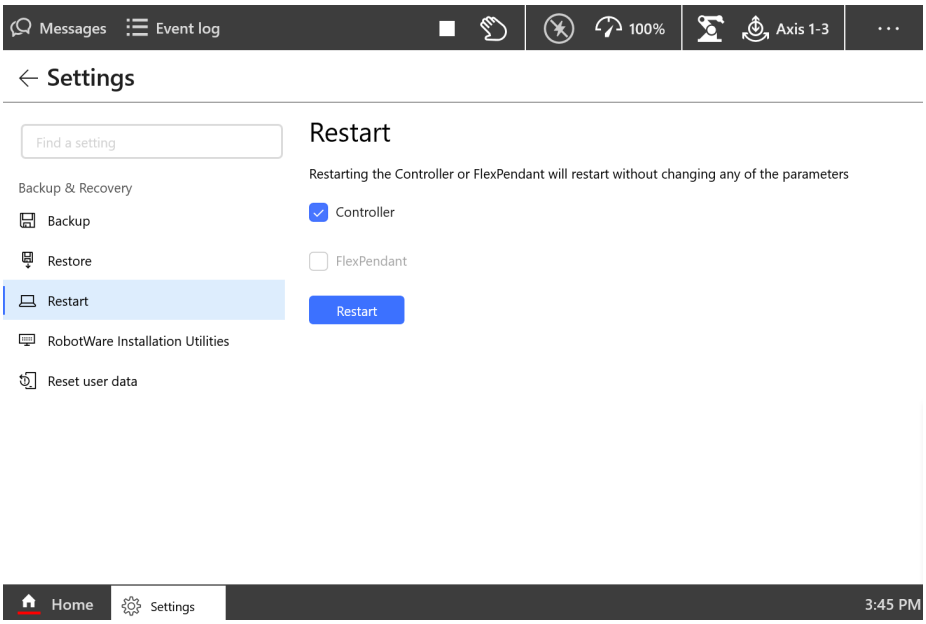
Restart the controller and/or FlexPendant

This section describes how to restart the Controller or Flexpendant and preserve the current system/parameters and RAPID programs.

1 On the start screen, tap **Settings**, and then tap **Backup & Recovery**.

2 On the **Backup & Recovery** menu, tap **Restart**.

The restart dialog is displayed.



xx1900000123

Select **Controller** or **FlexPendant**.

3 Tap **Restart**.

Continues on next page

The Controller and/or Flexpendant is restarted.

> **Note**
>
> When the Controller is restarted the Flexpendant is also automatically restarted.

Restart Controller through QuickSet menu

Use the following procedure to restart the controller using the QuickSet menu:

1. Tap on the **QuickSet** button and select the **Logout/Restart** tab.
   The Logout/Restart page is displayed.
2. In the **Controller** section tap **Restart**.
   The **Restart** window is displayed.
3. Tap **OK**.
   The controller is restarted.

Restart FlexPendant through QuickSet menu

There are certain cases under which only the FlexPendant needs to be restarted. For example, restart the FlexPendant while troubleshooting.

> **Note**
>
> If the FlexPendant freezes during operation, press the reset button to restart the FlexPendant. For more details, see *Reset button on page 23*.

Use the following procedure to restart the FlexPendant using the QuickSet menu:

1. Tap on the **QuickSet** button and select **Logout/Restart**.
   The **Logout/Restart** page is displayed.
2. In the **FlexPendant** section tap **Restart FlexPendant**.
   The **Restart** window is displayed.
3. Tap **OK**.
   The FlexPendant is restarted.

6.4.3 Start RobotWare Installation Utilities

What happens with my RobotWare System?

The RobotWare System will be saved and deactivated and then controller will enter into RobotWare Installation Utilities.

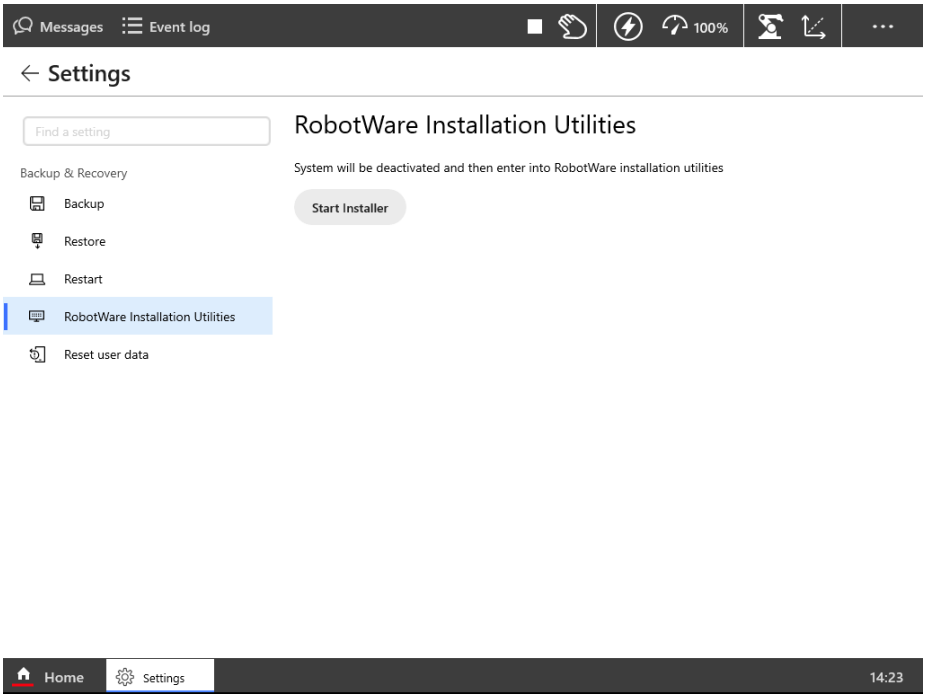
RobotWare Installation Utilities is for advanced maintenance only and the robot cannot run while in this mode.

Start RobotWare Installer

1 On the start screen, tap **Settings**, and then **Backup & Recovery**.

2 On the **Backup & Recovery** menu, tap **RobotWare Installation Utilities**.

The **RobotWare Installation Utilities** dialog is displayed.



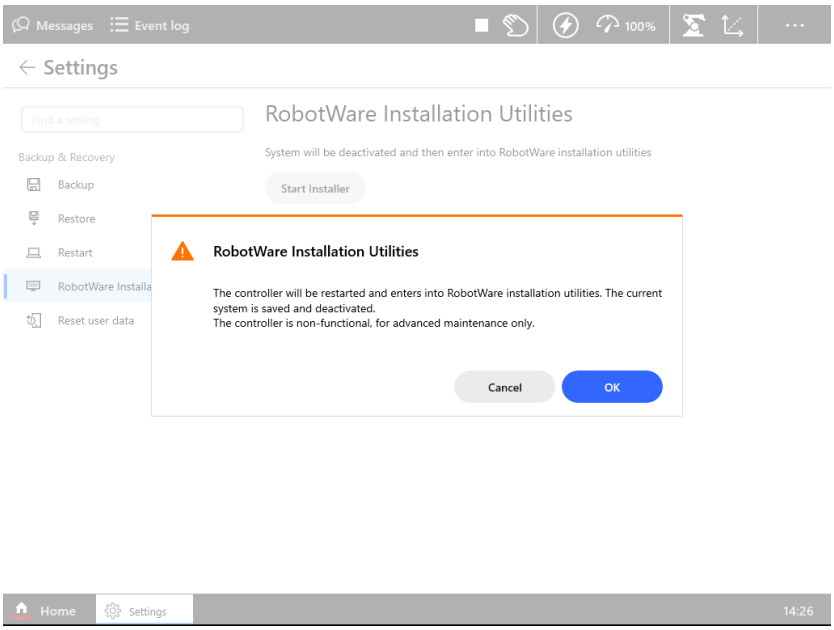
xx1900000109

3 Tap **Start Installer**.

Continues on next page

6.4.3 Start RobotWare Installation Utilities
Continued

The **RobotWare Installation Utilities** window is displayed.



xx1900001255

4 Tap **OK**.

The controller is restarted and the RobotWare Installation Utilities is started.

RobotWare startup error

If nothing seems to be starting and the PC STAT LED is solid red, possible fault scenarios are main computer hardware, firmware or that software not signed by ABB Robotics have been installed. Try to restart the system up to three times and if that doesn't help, the main computer should be returned to ABB for investigation.

For more information about troubleshooting, see the product manual for the controller, listed in *References on page 10*.

6.4.4 Reset user data

Overview

It is recommended to perform a user data reset:

- when you want to reset the RAPID program in the system.
- when you want to restore the system to its original state by resetting RAPID program and system parameters (except the topic *SIO/Communication*).
- when you want to remove the user configured safety settings and load the default safety settings.

Resetting RAPID program

After restart, the system state will be resumed except for manually loaded programs and modules. Static and semistatic tasks are started from the beginning, not from the state they had when the system was stopped.

Modules will be installed and loaded in accordance with the current configuration. System parameters will not be affected.

Resetting RAPID program and system parameters

After restart, the system returns to the default empty state and any changes done to system parameters and RAPID programs will be lost. Instead, system parameters and other settings are read from the originally installed system on delivery (except the topic *SIO/Communication*).

CAUTION

When the controller is restarted, the content in the **TEMP** folder in the controller is also emptied. To avoid problems, move any important content before resetting these parameters and restarting the controller.

Resetting safety settings

The user defined safety settings will be replaced by an empty default configuration and all safety information stored in the system will be erased. After the restart, a new safety configuration will be needed, synchronization between the safety controller and the robot controller needs to be performed and the safety configuration needs to be locked.

Procedure

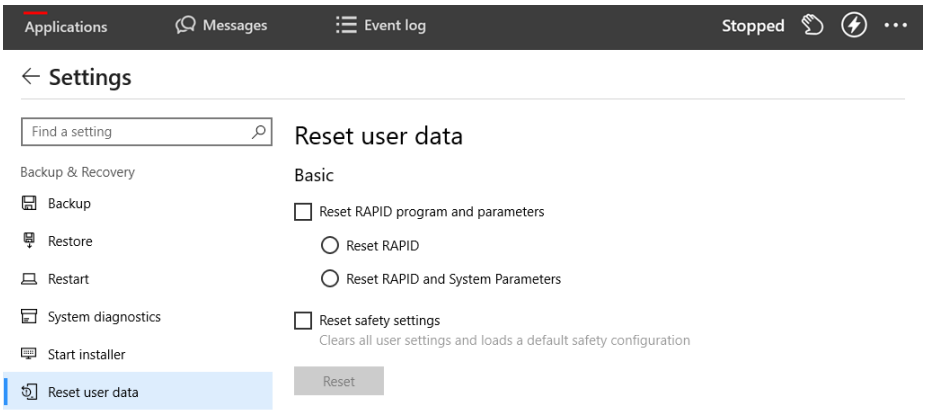
Use the following procedure to reset the user data:

1 On the start screen, tap **Settings**, and then **Backup & Recovery**.
2 On the **Backup & Recovery** menu, tap **Reset user data**.

Continues on next page

6.4.4 Reset user data
Continued

The **Reset user data** dialog is displayed.



xx1900000124

3 Select one of the following options:
   - **Reset RAPID**: To reset the RAPID data of the loaded system.
   - **Reset RAPID and System Parameters**: To restore the system to its original state by resetting the RAPID and system parameters (except the topic *SIO/Communication*).

     Note

     If this option is selected, the content in the **TEMP** folder will be emptied when the controller is restarted.

   - **Reset safety setting**: To remove the user configured safety settings and load the default safety settings.
4 Tap **Reset**.

   A confirmation message is displayed.
5 Tap **OK**.

   The controller is restarted and the system is updated according to the selected reset data settings.

6.4.5 Perform auto recovery

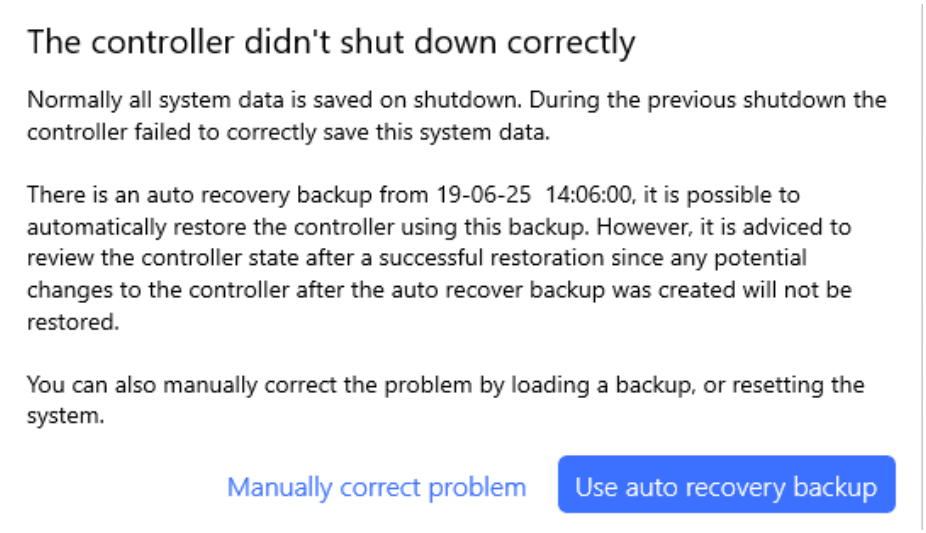
What will happen with my system?

If the controller failed to shut down correctly, you will be prompted at the next startup of the FlexPendant with information about the shutdown problems. You can choose either to use a recovery backup or to manually correct the problem. If auto recovery is selected, the controller will be restarted with the latest available recovery backup.

Perform auto recovery

This section describes how to perform an auto recovery.

1 If the controller failed to shut down correctly, the following information is displayed at the startup of the FlexPendant:



xx1900001252

Continues on next page

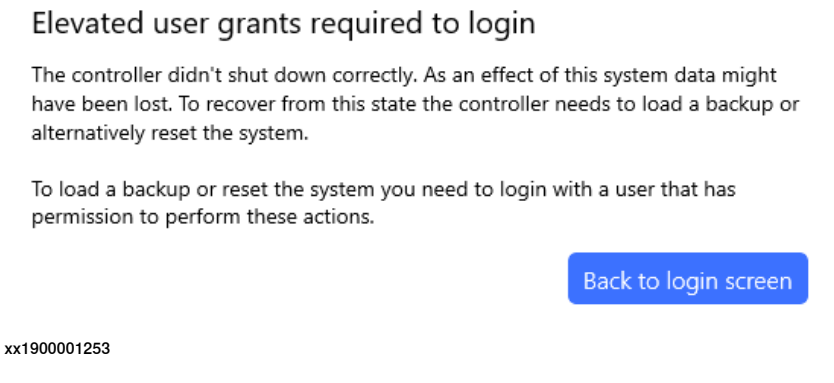
6.4.5 Perform auto recovery
Continued

You have two options:

- Tap **Use auto recovery backup**. If this option is selected, the controller is restarted with the recovery backup.

Note

Only users with elevated user grants can perform an auto recovery:



- Tap **Manually correct problem**. If this options is selected, you can inspect the system and view e.g. the event log and RAPID programs. You can then select to do the recovery from the **Settings app (Backup & Recovery)** or to manually select a backup to restore.

6.4.6 Reflashing firmware

Overview of reflashing

After replacing hardware units, such as axis computer, buses, etc., or installing newer versions of RobotWare, the system will automatically attempt reflashing the unit in order to maintain hardware/software compatibility if that is needed.

Reflashing is the process of loading appropriate firmware (hardware specific software) onto a specific unit running this software during operation.

The units currently using the reflash function are:

- Power units
- Drive units
- Axis computer (not available on all variants)
- Robot signal exchange proxy
- Main computer FPGA
- Safety module
- Connected Services Gateway module 3G or Wifi

Reflashing process

The automatic reflashing process, described below, must not be disturbed by switching off the controller while running:

	Event	Information
1	When the system is restarted, the system checks the hardware and firmwares used.	The result can be: • Hardware OK. • Hardware needs to be reflashed with new version of firmware. • Hardware cannot be used.
2	If reflashing of the firmware is required, the system restarts itself automatically while going to a specific *Update Mode*. All hardware that requires firmware update is reflashed in the same restart.	During the Update Mode, an attempt is made to download appropriate firmware to the hardware while a message is very briefly displayed on the FlexPendant.
3	Was the reflashing successful? If NO, an event log error message is logged.	A message is very briefly displayed on the FlexPendant and stored in the event log. The actual reflashing can take a few seconds or up to a few minutes, depending on the hardware to be reflashed.
4	After performing a successful reflash of all required hardware, the system performs a normal restart.	
5	Another check is made for any additional hardware/firmware mismatches.	
6	Was any additional mismatches found? If YES, the process is repeated once again. If NO, the process is complete.	If the reflashing fails twice, an error is logged.

6.5 Backup and restore systems

6.5.1 What is saved on backup?

Introduction to backups

When creating a backup, or restoring a previously made backup, not all data is included.

What *is* saved?

The backup function saves all system parameters, system modules, and program modules in a context.

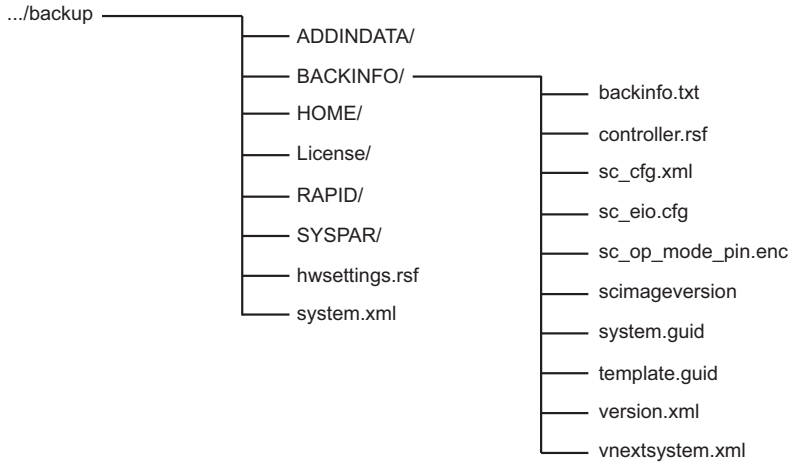
The data is saved in a directory specified by the user.

The directory is divided into the following five subdirectories:

- ADDINDATA
- BACKINFO
- HOME
- License
- RAPID
- SYSPAR

The files hwsettings.rsf and system.xml are also saved in the ../backup (root directory):

- hwsettings.rsf contains the serial number of the controller and the controller type.
- system.xml contains information about installed SW and optional features selected in the RobotWare System.



xx1900000318

ADDINDATA

ADDINDATA contains a number of sub-directories used for the RobotWare add-ins.

Continues on next page

BACKINFO

BACKINFO consists of the files *backinfo.txt*, *program.id*, and *system.guid*, *template.guid*, and *version.xml*.

- *backinfo.txt* is used when the system is restored. The backup must **never** be edited by the user!
- *controller.rsf* contains information about installed software and selected optional features in the backed up system.
- *sc_cfg.xml* is the safety configuration.
- *sc_eio.cfg* is the eio configuration connected to the safety configuration.
- *sc_op_mode_pin.enc* contains information on the pin code used for locking the operating mode.
- *scimageversion* contains information on which safety version the backup was created.
- *system.guid* is used to identify the unique system the backup was taken from.
- *system.guid* and/or *template.guid* is used in the restore to check that the backup is loaded to the correct system. If the *system.guid* and/or *template.guid* do not match, the user will be informed.
- *vnextsystem.xml* holds configuration data specific to the FlexPendant, such as programmable keys and start app on system event.

HOME

HOME is a copy of the files in the HOME directory.

License

License directory contains a copy of the license file that are used and valid only on the robot controller where the backup has been created. The license files can be used when creating a new RobotWare system in RobotStudio Installation Manger.

RAPID

RAPID consists of a subdirectory for each configured task. Each task has one directory for program modules and one for system modules. The module directory will keep all installed modules. More information on loading modules and programs is described in *Technical reference manual - System parameters*.

SYSPAR

SYSPAR contains the configuration files (that is, system parameters).

Limitations

It is not possible to move backups between controllers.

What *is not* saved?

A few things are not saved on backup, but can be useful to save separately:

- The current value of a PERS object in a installed module is not stored in a backup.
- UAS, Certificates and SW installed in RobotWare System are not included in the backup.

Continues on next page

6.5.1 What is saved on backup?
Continued

Related information

Technical reference manual - System parameters.

Operating manual - RobotStudio.

6.5.2 Backup the system

When do I need this?

We recommend performing a backup:

- *Before* installing new RobotWare.
- *Before* making any major changes to instructions and/or parameters to make it possible to return to the previous setting.
- *After* making any major changes to instructions and/or parameters and testing the new settings to retain the new successful setting.

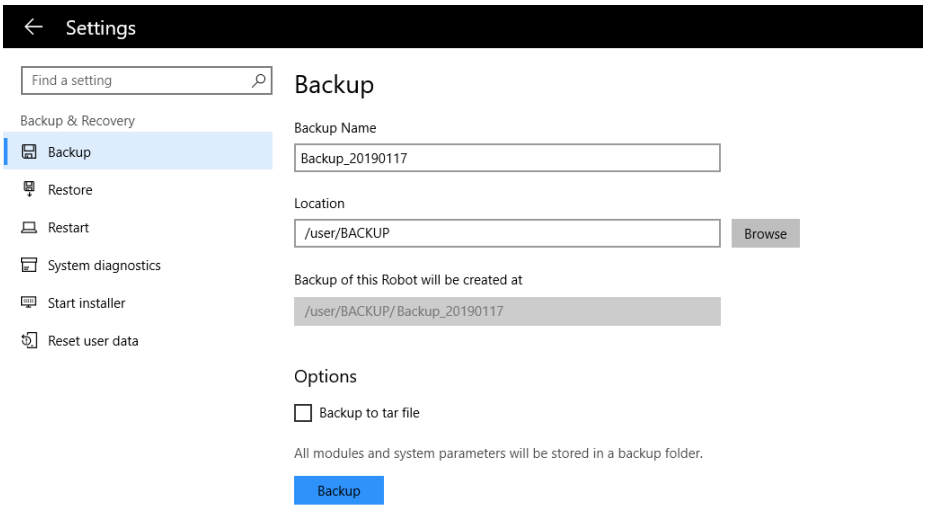
Backup the system

This section describes how to backup the system.

1 On the start screen, tap **Settings**, and then **Backup & Recovery**.

2 On the **Backup & Recovery** menu, tap **Backup**.

The backup dialog is displayed.



xx1900000152

3 In the **Backup Name** field, type a name for the backup.

Note

By default, a backup name is suggested, but you can edit it. While renaming, ensure that the name does not start with a space.

If the backup name starts with a space, a warning dialog appears.

Note

The backup name can have different formats depending on the parameter setup. The name may contain the date, or the system name etc. See information about backups in section **Type System Input** in *Technical reference manual - System parameters* for detailed information.

Continues on next page

4 In the **Location** field, tap the **Browse** button and select a location for saving the backup.

5 Select the **Backup to tar file** checkbox if the backup need to be created as a TAR file.

> **Note**
>
> If the backup location is USB or another disc, the **Backup to tar file** checkbox is selected by default and you cannot change it.

6 Tap **Backup**.

A backup of the modules and system parameters is created in the selected location.

Disable or queue backup

Backing up the system during production can interfere with the RAPID execution. To avoid that a backup is taken during critical process steps or sensitive robot movements, a system input (*Disable Backup*, type *System Input*) can be set during these critical steps. When the critical steps are done, the input should be reset to allow backups again.

If needed, the backup can be queued while *Disable Backup* is set, using the system parameter *General RAPID*, with action value *QueueBackup* set to *TRUE*. Then the backup will be queued until the signal is reset.

Disable Backup and *QueueBackup* are described in *Technical reference manual - System parameters*.

6.5.3 Important when performing backups

BACKUP directory

A local default backup directory, BACKUP, is automatically created by the system. We recommend using this directory for saving backups.

Never change the name of the BACKUP directory.

Never change the name of the actual backup to BACKUP, since this will cause interference with this directory.

When is backup possible?

A backup of a system can be performed during program execution, with a few limitations:

- Do not create backups while performing critical process steps or sensitive robot movements. This may affect the accuracy and performance of the movement. To make sure that no backup is requested, use a system input with the action value `Disable Backup` (type *System Input*). When the critical steps are done, the input should be reset to allow backups again.

  If needed, the backup can be queued while `Disable Backup` is set, using the system parameter *General RAPID*, with action value `QueueBackup` set to *TRUE*. Then the backup will be queued until the signal is reset.

  (Queueing functionality available from RobotWare 7.1.)

`Disable Backup` and `QueueBackup` are described in *Technical reference manual - System parameters*.

The system input signal can be set from RAPID for the parts of the code that are critical for disturbances.

What happens during backup?

During the backup process, background tasks continue to execute.

Large data amount

Since the HOME directory is included in the backup, large files contained in this folder will make the backup larger. To avoid this situation, you should either clean the HOME directory on regular basis removing the unnecessary files, or keep large files in some other location.

Faults during backup

If a fault occurs during the backup, for example full disk or power failure, the whole current backup is deleted to make sure that only valid fully saved backups are present on the disk.

6.5.4 Restore the system

When do I need this?

We recommend performing a restore:

- If you suspect that the program file is corrupt.
- If any changes made to the instructions and/or parameters settings did not prove successful, and you want to return to the previous settings.

During the restore, all system parameters are replaced and all the modules from the backup directory are loaded.

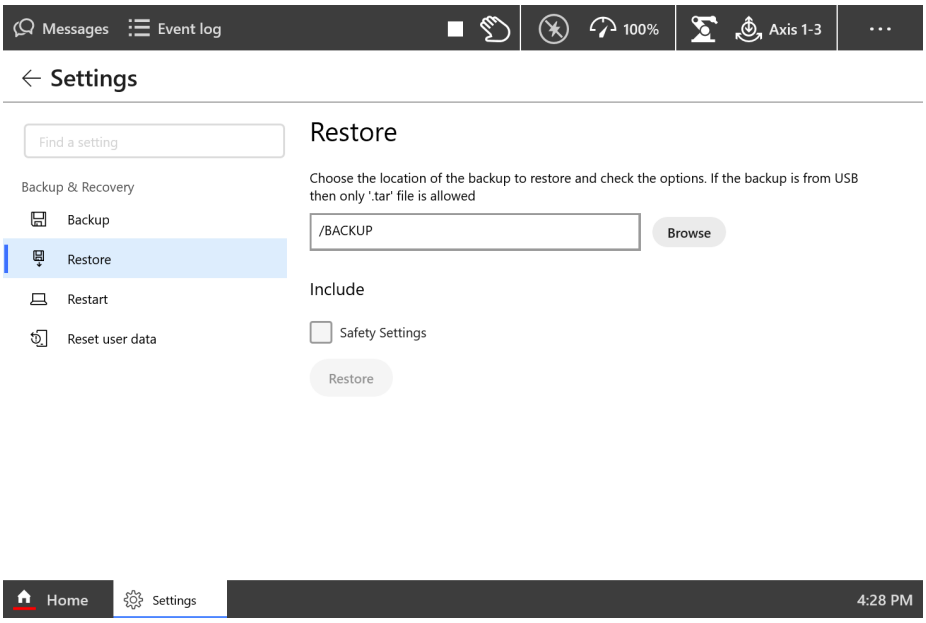
The Home directory is copied back to the new system's HOME directory during the restart.

Procedure

1 On the start screen, tap **Settings**, and then **Backup & Recovery**.

2 On the **Backup & Recovery** menu, tap **Restore**.

The **Restore** page is displayed.



xx1900000153

3 Browse for the location where the backup is stored and select the backup file.

> **Note**
>
> If the backup location is USB or another disc, the backup file is allowed only in the TAR format.

4 If required select the **Safety Settings** options.

5 Tap **Restore**.

The restore is performed, and the system is restarted.

6.6 Creating a system diagnostics file

When do I need this?

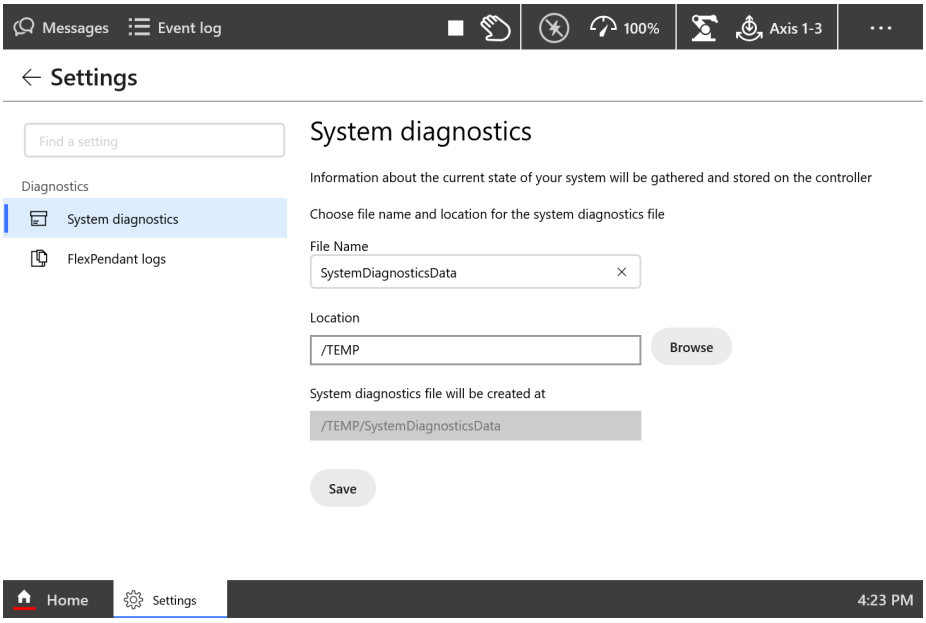
The diagnostic file can be useful when contacting ABB technical support personnel for troubleshooting. The diagnostic file contains the setup and a number of test results from your system. For more information, see *Technical reference manual - Event logs for RobotWare 7*, section *Instructions, how to correct faults - Filling an error report*.

Creating a system diagnostics file

1 On the start screen, tap **Settings**, and then **Diagnostics**.

2 On the left sidebar tap **System diagnostics**.

The **System diagnostics** page is displayed.



xx1900000151

Type a name for the file in the **File Name** section.

3 In the **Location** section, tap on the **Browse** button and select a folder to save the file.

4 Tap **Save**

A system diagnostics file is saved in the selected folder.

6.7 Manage configuration files

Loading configuration files

This section describes about loading a configuration file.

Use the following procedure to load a configuration file:

1. On the start screen, tap **Settings**, and then select **Controller Configuration**.
   The Controller Configuration page is displayed.
2. On the left sidebar tap **Load parameters**.
   The **Load parameters** page is displayed.
3. In the **Selected file** field, tap **Browse** and select the configuration file.
4. In the **Overwrite mode** section select a mode option. Following are the available options:
   - **Delete existing topic and load instances**: Configuration file will be loaded after deleting the existing parameters.
   - **Load instances but exclude duplicates**: Configuration file will be loaded only if there are no duplicates.
   - **Load instance and replace duplicates**: Configuration file will be loaded after replacing any duplicate that is found.
5. Tap **Load File**.
   The selected configuration file is loaded.
6. Tap **Restart Controller**.
   The Controller is restarted and the changes are implemented.

> **Note**
>
> Configuration files and backups shall not be loaded into the systems that has an older RobotWare version than in which they were created.
>
> Configuration files and backups are not guaranteed to be compatible between major releases of RobotWare and may need to be migrated after a RobotWare upgrade.

Saving configuration files

This section describes about saving the system parameter configurations. It is recommended to save the parameter configurations before making larger changes to the robot system. The parameters are saved automatically when performing backups.

Use the following procedure to save a configuration file:

1. On the start screen, tap **Settings**, and then select **Controller Configuration**.
   The **Controller Configuration** page is displayed.
2. On the left sidebar tap **Save parameters**.
   The **Save parameters** page is displayed.
3. In the **Location** field, tap **Browse** and select the location to which the file needs to be saved.

Continues on next page

4 In the **Selected topics** section select parameter topics that need to saved.
5 Tap **Save Files**.
The selected topics are saved to the selected location.

This page is intentionally left blank

7 RobotWare installation procedures

7.1 Introduction

Overview of the installation concept

The installation of a new RobotWare system, or the update of an existing RobotWare system, can be managed in the following ways:

- For installation, use the RobotStudio function **Modify Installation** to produce an installation package offline, put it on a USB-stick which later can be installed using RobotWare Installation Utilities from the FlexPendant.
- For update, use the function **Modify Installation** to connect directly to the robot controller online over the network or update package via FlexPendant, see *Apply Update overview on page 215*.

The following sections describe how to update an existing RobotWare system and how to create and install an installation package.

RobotWare system installation process

Follow these steps to install a new RobotWare system:

- Create a virtual controller, see *Creating virtual controllers on page 180*.
- Create an installation package and save it to a USB stick. See *Creating a new installation package on page 181*.
- Install the new package on the controller using RobotWare Installation Utilities from the FlexPendant. See *Installing a new RobotWare system on page 183*.

RobotWare system update process

Follow these steps to update an existing RobotWare system:

- Create a backup of the system (user data) and store it on an external storage media. See *Backup the system on page 161*.
- Create a snapshot of the current system state. See *Manually creating snapshot of current system state on page 188*.
- Update the RobotWare system. See *Updating an existing RobotWare system via RobotStudio on page 171*.

RobotWare system mass update process

Follow these steps to perform a mass software update on several controllers:

- Create an update package, see *Preparing an update package on page 215*.

  The package is generic and can be applied to multiple controllers, even those with differing configurations.
- Create a **Distribute Update Package** job in RobotStudio, see *Operating manual - RobotStudio*, section "Jobs".

  In the job, define the path to the update package to be distributed. Once the job is complete, the update can be applied to any controllers that have received it.

Continues on next page

- Apply the pending update using one of the following dialogs:
  - **Modify Installation** in RobotStudio, see *Updating an existing RobotWare system via RobotStudio on page 171*.
  - **Apply Update** in FlexPendant, see *Updating installed controller software using the Apply Update tool on page 215*

  If there is a pending update, the dialog will display a summary of the changes.

  The user can then choose to apply the update (which will restart the controller and install the changes), discard the update (which removes the pending update), or go back and ignore the update for now.

> **Note**
>
> Discarding an update will completely remove the pending update from the system.

Clarifications

To distinguish between the software system running on the robot controller, which manages the manipulators and the whole setup of the controller and its mechanical units, the following definitions are used:

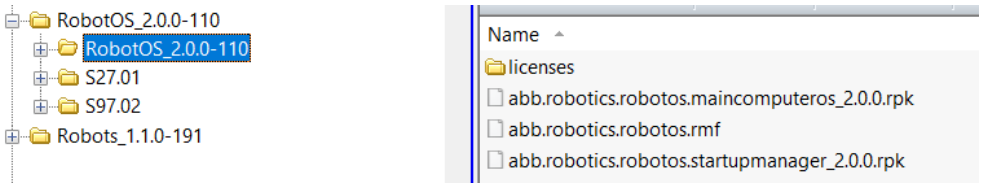
- RobotWare *system* - the software system running on the controller.
- Robot *system* - the controller and its mechanical units.

Installation media

When creating RobotWare systems, access to the original product installation media (rmf, rpk files) is always needed. The media files may, for example, be useful when creating a virtual controller from a real controller ("go-offline"), or when re-creating a system installation.

All files and folders from the product media package are preserved on the real controller during system installation or update. The files are stored in a product subdirectory, with the same name as its parent product installation folder, for example:

/products/RobotOS_2.0.0-110/RobotOS_2.0.0-110



xx2000000851

7.2 Updating an existing RobotWare system via RobotStudio

Description

The most frequent RobotWare system update use case is updating one or more software, for example, RobotWare and add-ins. This is a frequent operation during the commissioning time, especially on large installations.

> **Note**
>
> To perform a RobotWare system update, the controller must be in the RobotWare system mode.

System update changes the configuration of the currently installed RobotWare system. There are different types of configuration changes, such as:

- Adding or removing licenses
- Upgrading, removing installed software or adding new software
- Activating or deactivating optional features

Before performing a system update, it is recommended to:

- create a backup of the system (user data) and store it on an external storage media, see *Backup the system on page 161*.
- create a snapshot of the current system state, see *Manually creating snapshot of current system state on page 188*.

Accessing the Modify Installation tool

1. Start RobotStudio.
2. Select **Add Controller** > **Connect to Controller** in the **Controller** ribbon.
3. In the **Connect to Controller** window, select the controller and tap **OK**.
4. Request write access.
5. In the **Controller** ribbon, select **Installation** > **Modify Installation**, or right-click on the controller that you wish to modify.

Applying installation changes

When using the **Modify Installation** dialog to change a RobotWare system installation on a real controller, the state of the controller (RAPID program and system parameters) is automatically restored.

There are two ways of applying installation changes:

- Select **Apply** to apply changes and reload the current system state.
- Select **Apply and reset** to apply changes and start the system with a fresh new default state.

> **Note**
>
> Network and firewall settings are always preserved.

Continues on next page

Viewing installation change history

Changes that have been applied to the RobotWare system installation can be viewed in the **Update history** window:

- Select the **Show Update History** button in the **Summary** tab. The **Update history** window will display the ten most recent installation configuration changes.

Continues on next page

7.2.1 Updating software

Upgrading a software in the RobotWare system

The following procedure provides the steps involved during the update of the RobotWare system.

> **CAUTION**
>
> Do not turn off the controller while system update is in progress. Doing this may in worst case lead to data corruption in the RobotWare system, in which case it needs to be reinstalled.

1. Access the **Modify Installation** view in RobotStudio. See *Accessing the Modify Installation tool on page 171*.
2. Select **Software** > **Included**.
3. The **Included Software** window displays the software that is included in the current RobotWare system.
4. Select the product that should be upgraded and tap **Update**.
5. In the **Update Software** window, select the software version to be used and tap **OK**.
6. The **Summary** tab shows an overview of all the changes.
7. Continue to modify the system, or select **Apply**/**Apply and reset** to confirm and save the changes.

> **Note**
>
> The **Modify Installation** dialog will be closed during the controller update.
>
> When the update process is finished, check the event log for information about the update results. A successful update will be indicated in the event log, and if the update has failed, one or more error logs will be generated.

Adding/removing software

The following procedure provides the steps involved during the update of the RobotWare system.

> **CAUTION**
>
> Do not turn off the controller while system update is in progress. Doing this may in worst case lead to data corruption in the RobotWare system, in which case it needs to be reinstalled.

1. Access the **Modify Installation** view in RobotStudio. See *Accessing the Modify Installation tool on page 171*.
2. Select **Software** > **Included**.

Continues on next page

3 The **Included Software** window displays the software that is included in the current RobotWare system. Select one of the following:
   - Select the product box for the software that should be added to the system.
   - Deselect the product box to remove the product from the system.

> **Note**
>
> Products may have dependences to certain versions of other products. A product may only be removed if all products that are dependent on it are removed as well.

4 The **Summary** tab shows an overview of all the changes.

5 Continue to modify the system, or select **Apply**/**Apply and reset** to confirm and save the changes.

> **Note**
>
> The **Modify Installation** dialog will be closed during the controller update.
>
> When the update process is finished, check the event log for information about the update results. A successful update will be indicated in the event log, and if the update has failed, one or more error logs will be generated.

Adding/removing add-in packages

The following procedure provides the steps involved during the update of the RobotWare system.

> **CAUTION**
>
> Do not turn off the controller while system update is in progress. Doing this may in worst case lead to data corruption in the RobotWare system, in which case it needs to be reinstalled or recovered from a snapshot.

1 Access the **Modify Installation** view in RobotStudio. See *Accessing the Modify Installation tool on page 171*.

2 Select one of the following:
   - To add add-in packages, select **Software** > **Available** and tap **Include**.
   - To remove add-in packages, select **Software** > **Included** and tap **Remove**.

> **Note**
>
> Products may have dependences to certain versions of other products. A product may only be removed if all products that are dependent on it are removed as well.

Continues on next page

> **Note**
>
> RobotWare is mandatory and cannot be removed from the system.

3 The **Summary** tab shows an overview of all the changes.

4 Continue to modify the system, or select **Apply**/**Apply and reset** to confirm and save the changes.

> **Note**
>
> The **Modify Installation** dialog will be closed during the controller update.
>
> When the update process is finished, check the event log for information about the update results. A successful update will be indicated in the event log, and if the update has failed, one or more error logs will be generated.

Changing the software installation order when adding/removing RobotWare add-ins

When adding and removing RobotWare add-ins to/from the system, sometimes it is necessary to manually adjust the installation and initialization order or the included add-ins.

1 Access the **Modify Installation** view in RobotStudio. See *Accessing the Modify Installation tool on page 171*.

2 Select **Software** > **Included**.

3 In the **Included Software** window, tap the **Installation order** button to open the **Change Installation Order** window. Select a product and use the up and down arrows to change the installation order. Select **Done**.

4 The **Summary** tab indicates that the installation order has been updated.

5 Continue to modify the system, or select **Apply**/**Apply and reset** to confirm and save the changes.

> **Note**
>
> The **Modify Installation** dialog will be closed during the controller update.
>
> When the update process is finished, check the event log for information about the update results. A successful update will be indicated in the event log, and if the update has failed, one or more error logs will be generated.

7.2.2 Working with option selections

Overview

The following categories can be updated:

- System options
- Controllers
- Robots
- FlexPendant

> **Note**
>
> Some options extend, showing more options upon selection. For example, in the group controller variant, you get the option of choosing variant type only when a controller first is selected. The additional drive units work similarly, some are unavailable until you select a different drive system type. This means options can be locked behind selections.

Turning options on/off

> **CAUTION**
>
> Do not turn off the controller while system update is in progress. Doing this may in worst case lead to data corruption in the RobotWare system, in which case it needs to be reinstalled.

1. Access the **Modify Installation** view in RobotStudio. See *Accessing the Modify Installation tool on page 171*.
2. Select the tab **Options**.
3. Select the option category to be updated, and the corresponding **Options** that should be activated/deactivated for the system.

   > **Note**
   >
   > Linked options will be selected automatically.
   >
   > Conflicting options cannot be selected.

4. The **Summary** tab shows an overview of all the changes.
5. Continue to modify the system, or select **Apply**/**Apply and reset** to confirm and save the changes.

   > **Note**
   >
   > The **Modify Installation** dialog will be closed during the controller update.
   >
   > When the update process is finished, check the event log for information about the update results. A successful update will be indicated in the event log, and if the update has failed, one or more error logs will be generated.

Continues on next page

Adding licenses to enable additional option access

CAUTION

Do not turn off the controller while system update is in progress. Doing this may in worst case lead to data corruption in the RobotWare system, in which case it needs to be reinstalled.

1. Access the **Modify Installation** view in RobotStudio. See *Accessing the Modify Installation tool on page 171*.
2. Select the tab **Options**.
3. Select **Edit** to access the **Edit License files** window. Select one of the following:
   - Select **Add** to browse for a new license to be added.
   - Select an existing license and tap **Remove**.
4. The **Summary** tab shows an overview of all changes.
5. Continue to modify the system, or select **Apply**/**Apply and reset** to confirm and save the changes.

Note

The **Modify Installation** dialog will be closed during the controller update.

When the update process is finished, check the event log for information about the update results. A successful update will be indicated in the event log, and if the update has failed, one or more error logs will be generated.

Exporting and importing option selections

CAUTION

Do not turn off the controller while system update is in progress. Doing this may in worst case lead to data corruption in the RobotWare system, in which case it needs to be reinstalled.

1. Access the **Modify Installation** view in RobotStudio. See *Accessing the Modify Installation tool on page 171*.
2. Select the tab **Options**.
3. Select one of the following:
   - Select **Export** and browse to the location where the exported option selections should be saved. Select **Save**.

     The current option selections will be saved to an RSF file that can be imported or added to other systems.

Continues on next page

- Select **Import** and browse to the location of the configuration file, and then select **Open**.

  Note

  All current selections will first be cleared.

- Select **Add** and browse to the location of the configuration file, and then select **Open**.

  Note

  Existing selections are kept, and options that are not currently selected will be added.

4 Continue to modify the system, or select **Apply**/**Apply and reset** to confirm and save the changes.

Note

The **Modify Installation** dialog will be closed during the controller update.

When the update process is finished, check the event log for information about the update results. A successful update will be indicated in the event log, and if the update has failed, one or more error logs will be generated.

Drive system types

The following matrix describes the existing drive system types and some examples of compatible products:

Product		Power									
Manipulator	**Controller**	2.5kVA-310V	2.5kVA-370V	3.0kVA-370V	7.0kVA-370V	3.0kVA-370V	480VA-24V	1.2kVA-48V	1.5kVA-48V	13kVA-650V	7.5kVA-650V
IRB 1600 or smaller	C30	A1									
	C90XT										
	E10		B2								
IRB 14050	C30						C6				
CRB 15000 (5 kg)	C30							D7			
CRB 15000 (10 kg or 12 kg)	C30								D10		
IRB 390 IRB 2400 IRB 2600	V250XT				E4	E5					
	V400XT										
IRB 4600 or larger	V250XT									E8	E9
	V400XT										

7.3 Installing a new RobotWare system

Description

Before installing a new RobotWare system on the controller, it is required to:

- create a virtual controller, see *Creating virtual controllers on page 180*.
- create an installation package, see *Creating a new installation package on page 181*.

Continues on next page

7.3.1 Creating virtual controllers

Create a virtual controller

1 Start RobotStudio.
2 Select **Add Controller** > **Connect to Controller** in the **Controller** ribbon.
3 In the **Connect to Controller** window, select the **Virtual Controllers** tab.
4 Select **New Controller**.
5 In the **New Virtual Controller** dialog, select option **Create New** and complete the following:
   - **Name**

     Give the new system a valid name. If you enter an invalid name you will not be able to proceed.

     Note

     The system name can contain between 1 to 55 characters. Allowed characters are "A–Z", "a–z", "0–9", and "-" (hyphen). Hyphen "-" is only allowed between characters.

   - **Location**
   - **Robot model**
   - **RobotWare**
   - **Controller**

   Note

   Selecting option **Create from backup** can be used to create a system based on the configuration found in the selected Backup. This means that the same set of SW products (RobotWare and add-ins), licenses and options will be used.

   Note, however that the software referred to by the Backup is not included in the Backup itself and must be previously downloaded to your computer by using the RobotStudio **Add-Ins** page.

   Note also that this procedure will not automatically include RAPID programs and system parameters to your new system. If needed, they can be loaded to the new system by restoring the Backup once the new system is installed and started.

6 Select **OK** to continue.
7 Continue with creating an installation package, see *Creating a new installation package on page 181*.

7.3.2 Creating a new installation package

Overview

The installation package is a software package that consists of predefined directory structure and number of files, used for purpose of re-deploying RobotWare System on a robot controller. The installation package is created in RobotStudio and is deployed on the controller using RobotWare Installation Utilities on the FlexPendant.

RobotWare Installation Utilities is a small package of installation related utilities that is always present on each robot controller and cannot be removed. It is used to deploy and re-deploy RobotWare system which is the operating system of the robot controller. When in RobotWare Installation Utilities mode, the robot cannot be moved using the FlexPendant and robot programs cannot be written or executed.

Prerequisites

The following prerequisites must be met before you can start creating an installation package:

- Latest version of RobotStudio must be installed.
- License files for products to be installed must be available. Licenses are included in the RobotWare system at purchase, but can also be retrieved from a backup of the RobotWare system currently deployed on the controller, or exported from the controller via RobotWare Installation Utilities.

  Note

  Virtual licenses can also be used. See *Create installation package on page 181*.

- Product versions to be installed must be available in RobotStudio or in a custom location.

  These versions can be made available by selecting a RobotWare distribution package (.rspak file) from RobotStudio (tap **Install Package** in the **Add-Ins** tab). All products that are installed this way, have matched versions and correct dependencies to each other.
- A virtual controller must be created, see *Creating virtual controllers on page 180*.

Create installation package

1. Start RobotStudio.
2. Select **Add Controller** > **Connect to Controller** in the **Controller** ribbon.
3. In the **Connect to Controller** window, select the controller and tap **OK**.
4. Request write access.
5. Launch the **Modify Installation** dialog from the **Controller** ribbon.
6. Select the tab **Software**.

Continues on next page

7 Select **Create Package** to create an installation package based on the virtual controller configuration.

> **Note**
>
> If the virtual system has been built using virtual licenses, these will not be included in the installation package.
>
> If virtual licenses are used, the selected feature configuration will be matched against the real licenses present in the controller and the installation will stop if some licenses are missing. This situation can be avoided if real licenses from the controller are exported and imported into the virtual system when it is built.

8 In the **Create Installation Package** dialog, define the following:

- **Package Name**

  Enter a name for the installation package.
- **Location**

  Browse and select the output folder (for example, a USB-stick) for the installation package.

Select **OK**.

9 The window **Installation Package created** is displayed. The installation package for the selected system has been created. Select **OK**.

10 Continue with installing the package on the controller, see *Installing a new RobotWare system on page 183*.

7.3.3 Installing a new RobotWare system

Preparations

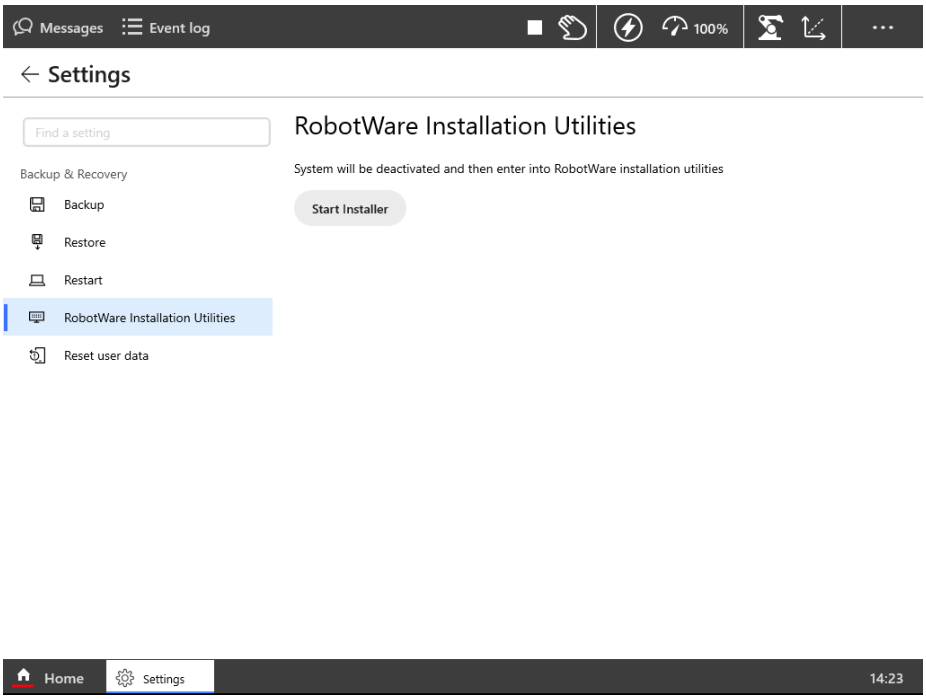
Create an installation package according to section *Creating a new installation package on page 181*.

Install RobotWare system

1 On the FlexPendant start screen, select **Settings**, and then **Backup & Recovery**.

2 On the **Backup & Recovery** menu, select **RobotWare Installation Utilities**.

The start installer dialog is displayed.

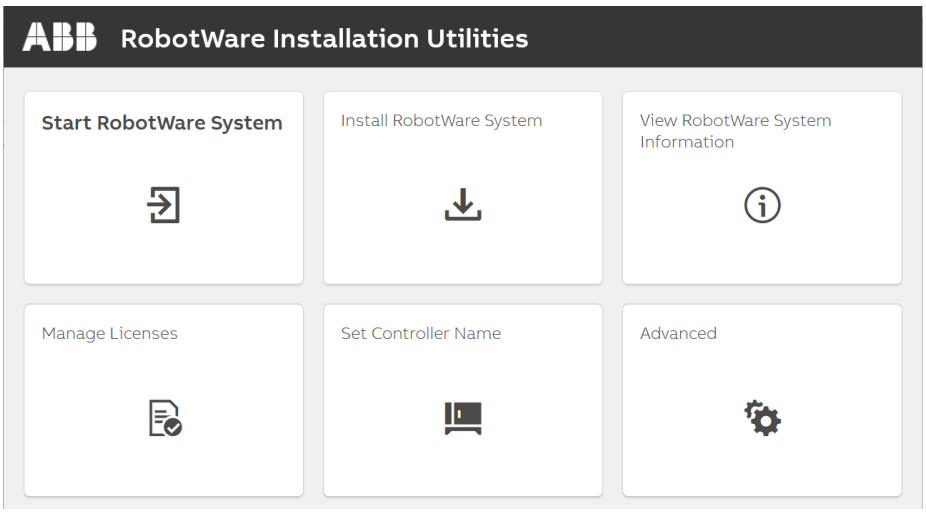


xx1900000109

Continues on next page

7.3.3 Installing a new RobotWare system
Continued

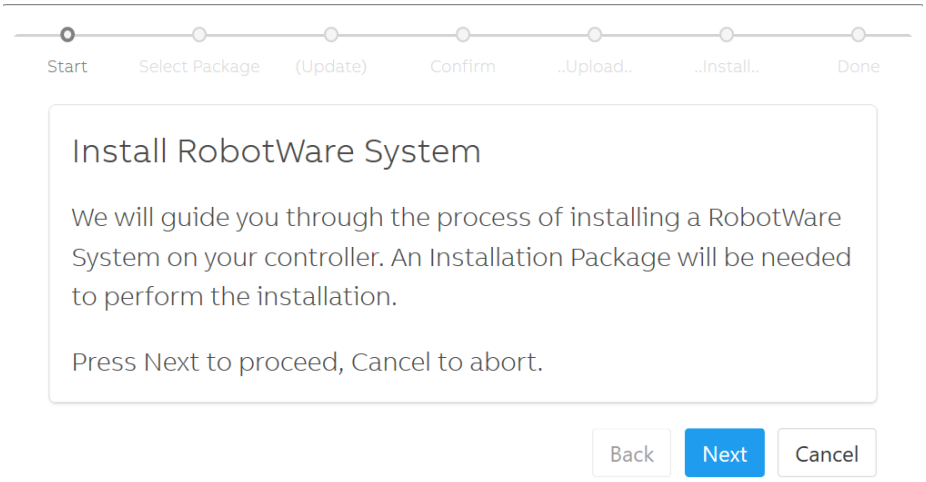
3 Select **Start installer** to access the RobotWare Installation Utilities. The controller will restart and the following menu is displayed:



xx1900000110

Tap **Install RobotWare System**.

The Install RobotWare System dialog is displayed.



xx1900000111

Note

If a system already is installed, the installation wizard will guide you in how to delete the current system before the new one can be installed.

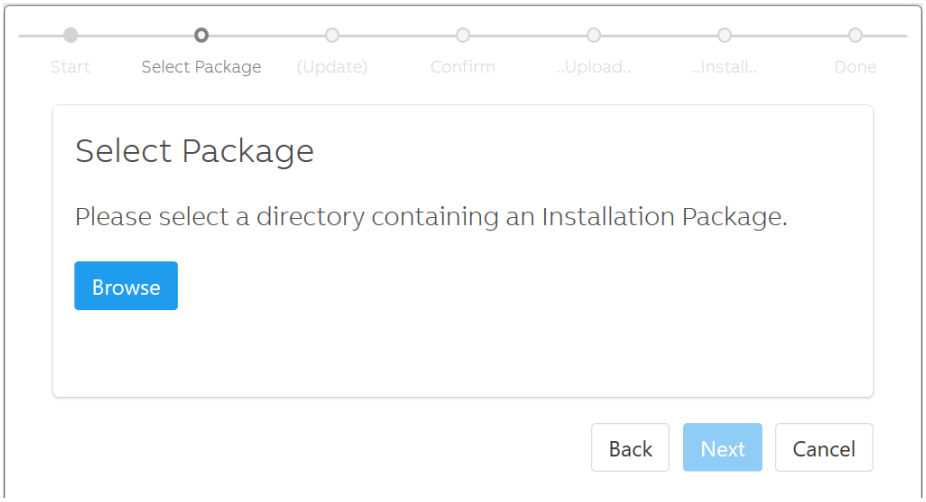
Tip

To access information about currently installed RobotWare System, it is possible to use dialog **View RobotWare System Information**.

Follow the directions in the Install RobotWare System wizard and tap **Continue**.

Continues on next page

4 Browse and select the installation package. Confirm the selection and tap **Continue**.



xx1900000112

The package is uploaded and unpacked.

5 Tap **Start RobotWare System** and then **Start**. This will close the RobotWare Installation Utilities and start the FlexPendant application.

7.4 Using RobotWare Installation Utilities

7.4.1 Manually updating RobotWare Installation Utilities

When do I need this?

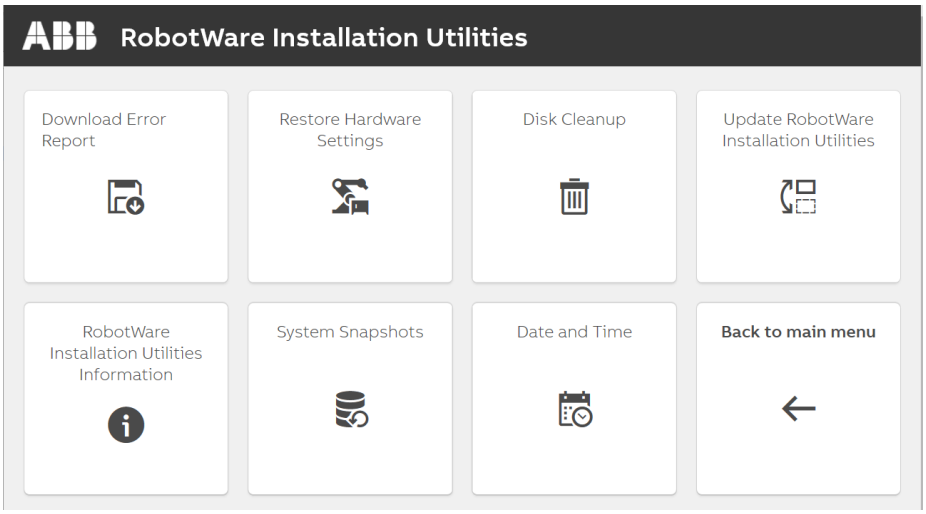
The RobotWare Installation Utilities can be manually updated if needed.

> **Note**
>
> RobotWare Installation Utilities is automatically updated when you install a RobotWare system if the version included in the installation package is of a newer version than the one currently installed, and if auto-update is possible.

Update RobotWare Installation Utilities manually

1 Access the RobotWare Installation Utilities. See *Start RobotWare Installation Utilities on page 151*.

2 Select **Advanced**, and then **Update RobotWare Installation Utilities**.



xx1900000120

3 Insert USB, and select **Browse** to navigate to the sub directory named **RobotWare Installer** containing the version to be used.

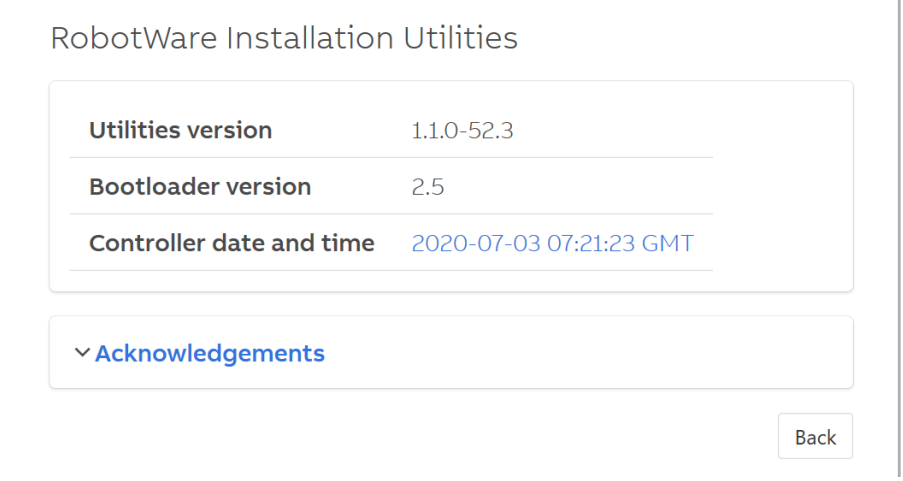
Select **Select folder** and then **OK**.

Continues on next page

The new version is now installed.

Tip

Access the RobotWare Installation Utilities Information to view information about the currently installed version.



xx1900000121

7.4.2 Managing system snapshots

When do I need this?

Snapshots are used to create a backup copy of the current system state. This is useful as a safety precaution before making changes to the RobotWare system. If something goes wrong during a system installation or when updating the system, it is possible to restore data from a selected snapshot.

Snapshots are automatically created by the controller before a system is modified through the **Modify Installation** function in RobotStudio, and the two latest snapshots are preserved and can be displayed via the RobotWare Installation Utilities. Snapshots can also be created manually.

> **Note**
>
> It is recommended to manually create a snapshot of the system before making changes to the RobotWare system.

Manually creating snapshot of current system state

Creating a snapshot will store a full copy of the current system state including installed software, user and system internal data.

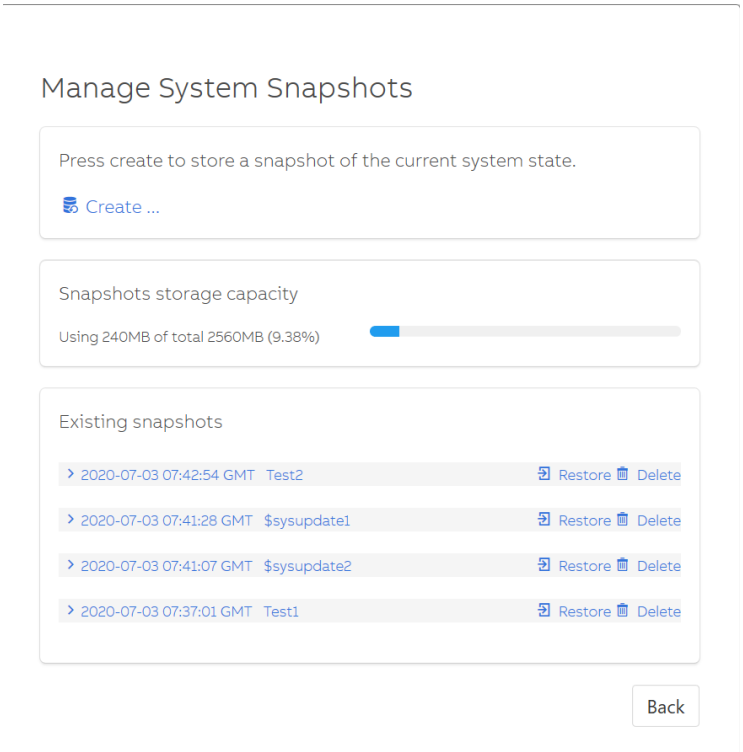
> **Note**
>
> Snapshots can only be created when a RobotWare system is installed.

> **Note**
>
> Snapshots do not affect the current system.

1 Access the RobotWare Installation Utilities. Select **Advanced** and then **System Snapshots**.

Continues on next page

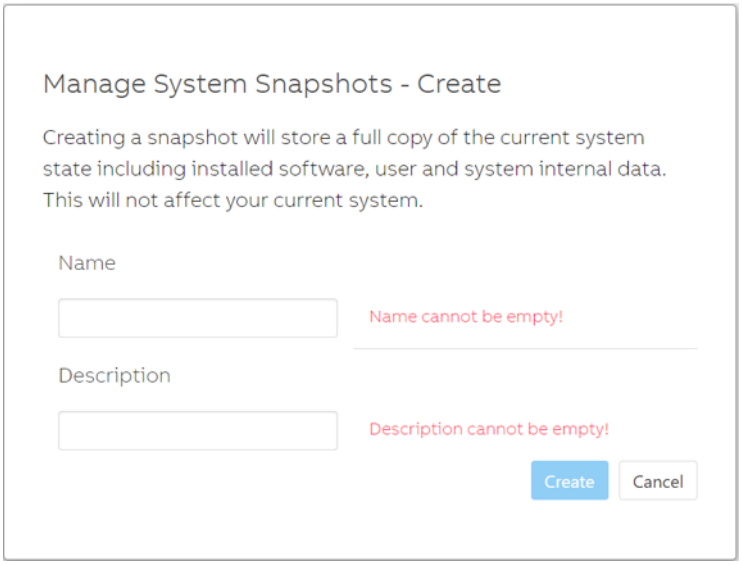
2 The **Manage System Snapshots** dialog is displayed:



xx2000001523

3 Select **Create** to start the snapshot creation process.

4 The **Manage System Snapshots - Create** window is displayed:



xx2000001524

5 Complete the following fields:

- **Name**

Continues on next page

Enter a name for the snapshot.

> **Note**
>
> The snapshot name can contain between 1 to 23 characters. Allowed characters are "A–Z", "a–z", "0–9", and "-" (hyphen). Hyphen "-" is only allowed between characters.

- **Description**

  Enter a description of the snapshot.

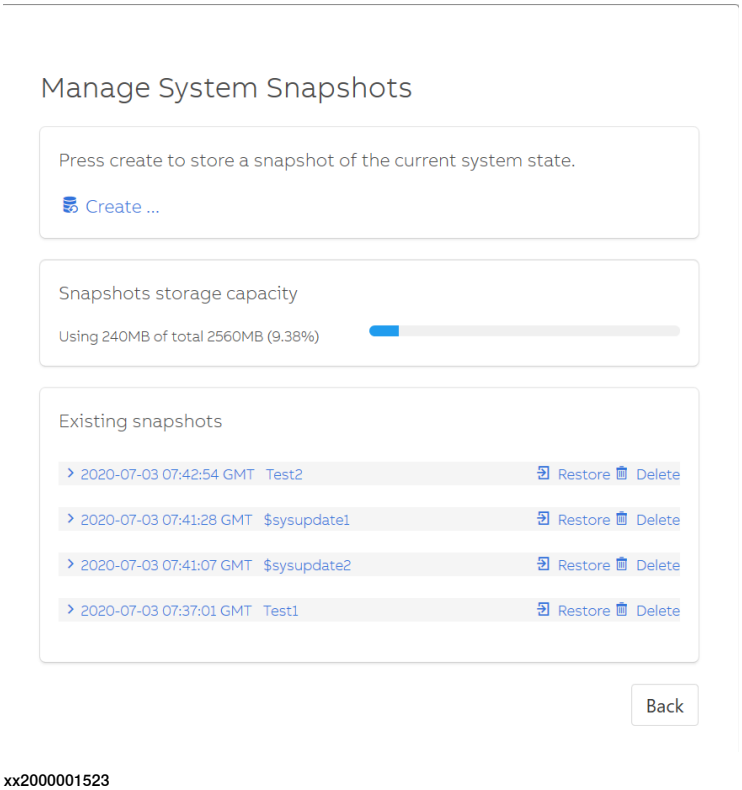
6 Select **Create**. A snapshot is created from the current system state.

Restoring system from snapshot

> **WARNING**
>
> This procedure will completely replace all installed software, user and system internal data in your current system.

1 Access the RobotWare Installation Utilities. Select **Advanced** and then **System Snapshots**.

2 The **Manage System Snapshots** dialog is displayed:

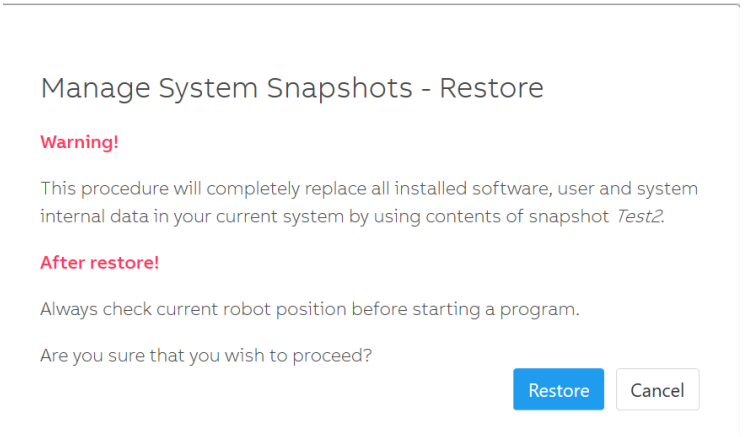


xx2000001523

3 Select an existing snapshot from the list and tap **Restore**.

Continues on next page

4 The **Manage System Snapshots - Restore** window is displayed:

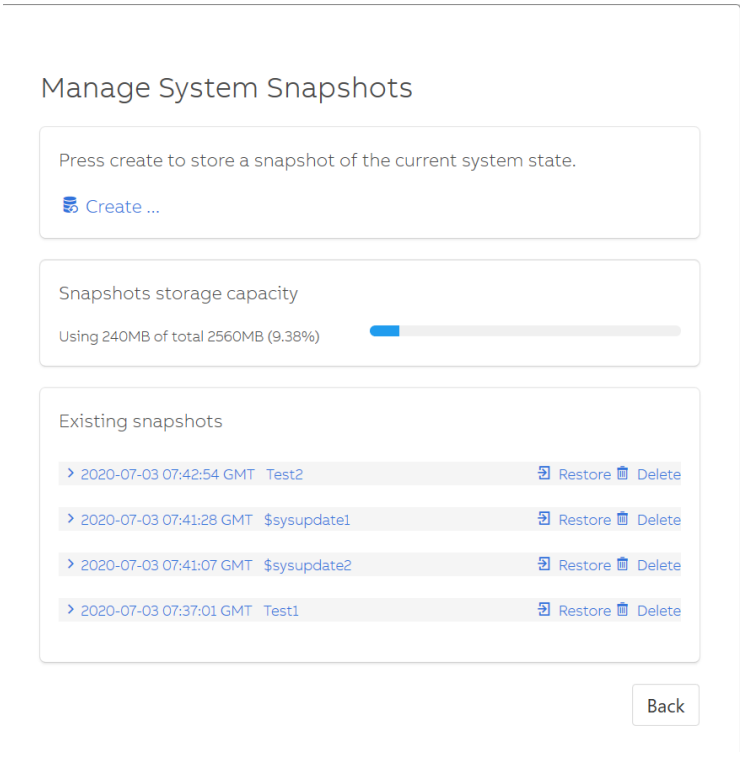


xx2000001525

5 Select **Restore** to replace all installed software, user and system internal data in your current system with the selected snapshot.

Deleting a snapshot

1 Access the RobotWare Installation Utilities. Select **Advanced** and then **System Snapshots**.
2 The **Manage System Snapshots** dialog is displayed:



xx2000001523

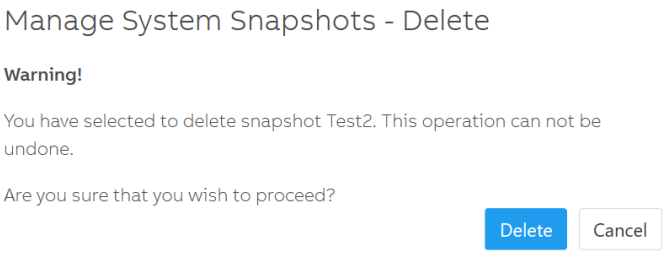
3 Select an existing snapshot from the list and tap **Delete**.

Continues on next page

7.4.2 Managing system snapshots

Continued

4 The **Manage System Snapshots - Delete** window is displayed:



xx2000001526

5 Select **Delete** to delete the snapshot.

7.4.3 Performing a controller disk cleanup

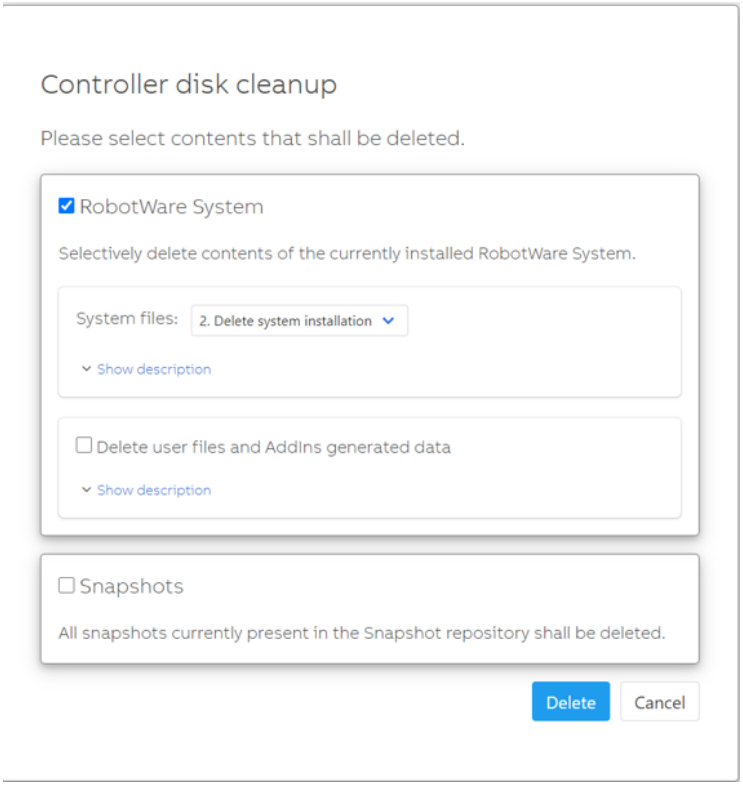
When do I need this?

The RobotWare system and/or Snapshots can be deleted if needed. User files and data generated from add-ins can also be deleted from this function.

Delete RobotWare System

1 Access the RobotWare Installation Utilities. Select **Advanced** and then **Disk Cleanup**.

2 The **Controller disk cleanup** dialog is displayed:



xx2000001521

3 In the **RobotWare System** window, field **System files**, define what data should be deleted by selecting one of the following options:

Option	Description
1. Delete system data	This option deletes the internal system data, but keeps the system installation.
2. Delete system installation	This option deletes the system installation, including the system data.

Continues on next page

7.4.3 Performing a controller disk cleanup
Continued

Option	Description
3. Full system delete	This option deletes *all* data and software installed in your system (RobotWare and Addins), including long-lived system files that are normally preserved when you are reinstalling the deployed system. **Note** This option is only recommended in exceptional circumstances.

4 Select the box **Delete user files and Addins generated data** if all files and directories stored in the user managed part of the disk should be deleted.

Note

This is data stored by the end user or generated by RAPID programs and Add-ins when running the system.

5 Select **Delete** to perform the disk cleanup.

Delete all existing snapshots

Using this function, all existing snapshots will be deleted.

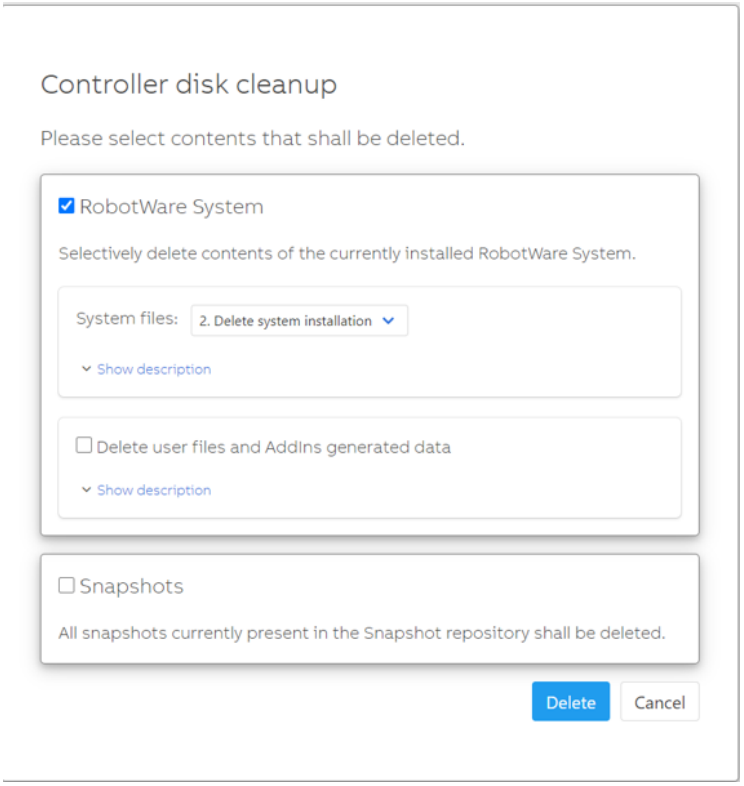
Note

To delete a selected snapshot only, see *Manually creating snapshot of current system state on page 188*.

1 Access the RobotWare Installation Utilities. Select **Advanced** and then **Disk Cleanup**.

Continues on next page

2 The **Controller disk cleanup** dialog is displayed:



xx2000001521

3 Select the box **Snapshots** if all snapshots that are currently present in the Snapshot repository should be deleted.

> **Note**
>
> If there are no snapshots, the message (**nothing to delete**) is displayed.

4 Select **Delete** to perform the disk cleanup.

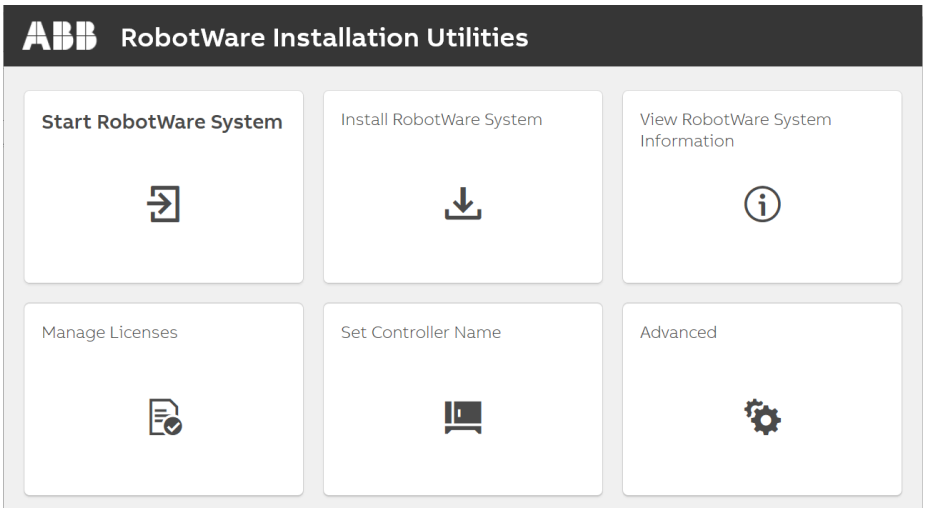
7.4.4 Downloading an error report

When do I need this?

In case of problems when using any functionality in RobotWare Installation Utilities, an error report can be downloaded. This report should be used when in contact with ABB.

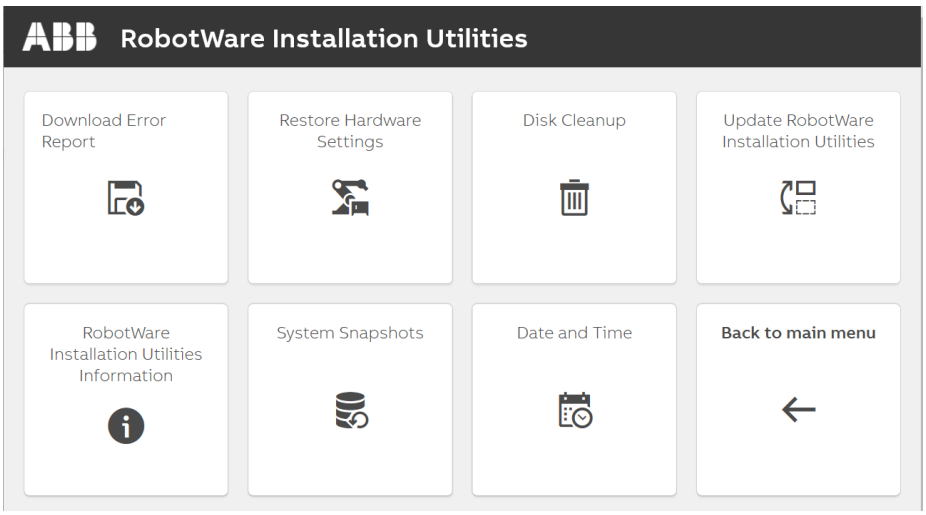
Download error report

1 Access the RobotWare Installation Utilities.



xx1900000110

Tap **Advanced**, and then **Download Error Report**.



xx1900000120

The Save as dialog is displayed.

2 Select a location for downloading of the report and tap **Save**.

7.4.5 Setting the controller name

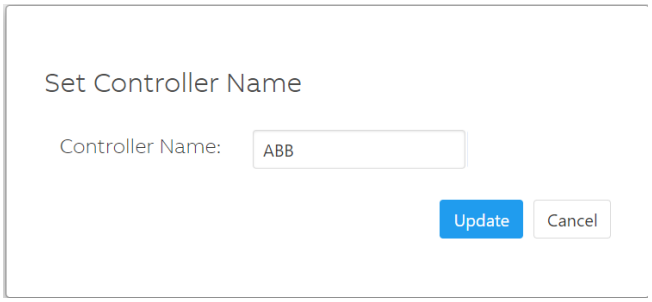
When do I need this?

In RobotWare, the system name is the same as the controller name. It is recommended to set the controller name before the RobotWare system is installed.

Set the controller name

1 Access the RobotWare Installation Utilities. See *Start RobotWare Installation Utilities on page 151*.

2 Select **Set Controller Name**. The **Set Controller Name** dialog is displayed:



xx2000001772

3 Enter the **Controller Name** and select **Update**.

Note

The controller name can contain between 1 to 63 characters. Allowed characters are "A–Z", "a–z", "0–9", and "-" (hyphen). Hyphen "-" is only allowed between characters.

7.4.6 Defining controller date and time

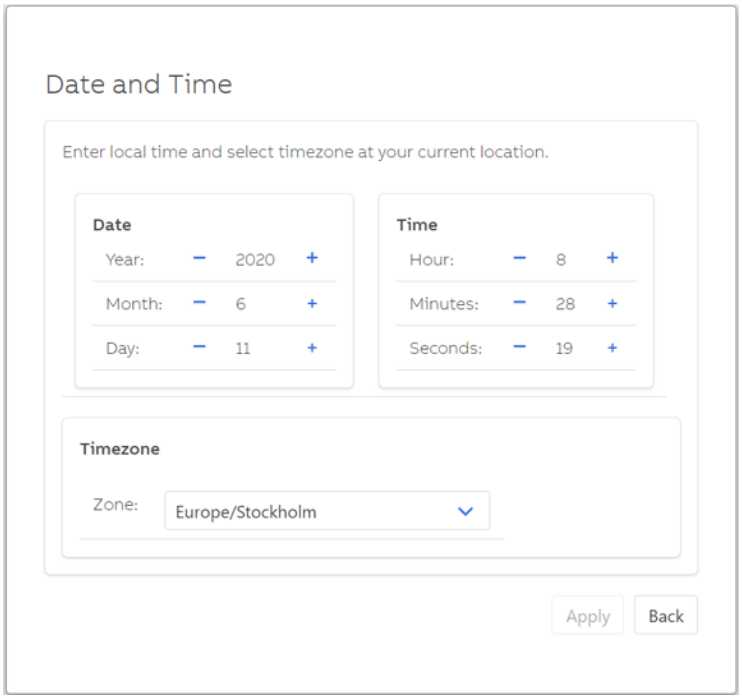
When do I need this?

Using this function, date and time zone can be defined for the controller. The date and time will be displayed in RobotWare Installation Utilities information.

Define controller date and time

1 Access the RobotWare Installation Utilities. Select **Advanced** and then **Date and Time**.

2 The **Date and Time** dialog is displayed:



xx2000001522

3 Complete the following fields:

- **Date**

  Enter the current date (Year/Month/Day).
- **Time**

  Enter the current time (Hour/Minutes/Seconds).
- **Timezone**

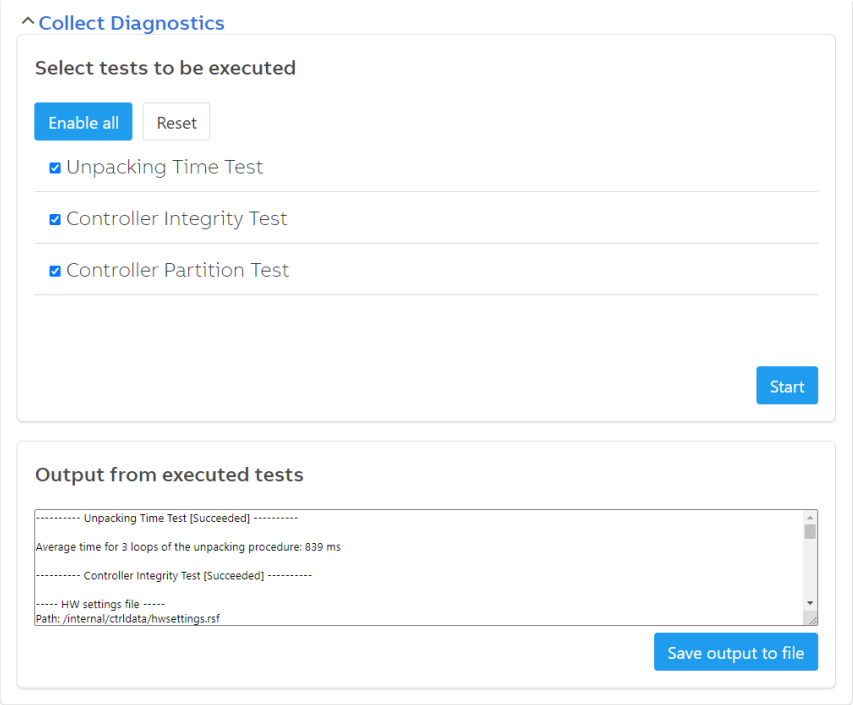
  Select the time zone for your location.

4 Select **Apply** to confirm the changes. The defined date and time will be displayed in RobotWare Installation Utilities information.

7.4.7 Collecting diagnostics

Collect diagnostics

1. Access the RobotWare Installation Utilities. Select **Advanced** > **RobotWare Installation Utilities Information**.
2. In the **RobotWare Installation Utilities** window, select **Collect Diagnostics**.
3. The **Collect Diagnostics** window is displayed:



xx2300001362

Select one the following tests, or select **Enable all**:

- **Unpacking Time Test**

  Checks the time it takes for a system to unpack an archive.

  > **Note**
  >
  > The test fails if the unpacking exceeds the expected time limit defined for the system.

- **Controller Integrity Test**

  Checks that a certain set of paths exist in the system. Some tests are optional and some are necessary.

  > **Note**
  >
  > The test fails if the necessary tests do not exist in the system.

Continues on next page

- **Controller Partition Test**

  Outputs the free space and checks the total size for each partition on the system.

  > **Note**
  >
  > The test fails if the partition size differs from the expected size.

4 Select **Start**. The selected tests will be executed and the test results are presented in the **Output from executed tests** window.

> **Tip**
>
> Select **Save output to file** to browse for a location and save a file with the test results.

7.5 Updating an existing RobotWare system via FlexPendant

Introduction

This chapter describes how an existing system can be modified using the FlexPendant.

> **Note**
>
> For information about how to update an existing RobotWare system using the **Modify Installation** function, see *Updating an existing RobotWare system via RobotStudio on page 171*.

Features provided by the FlexPendant are intentionally simpler and less capable compared to RobotStudio which is a more advanced engineering tool.

Two different tools can be used to modify the controller software installation, depending on the type of prepared input for the update:

- **Add-In Installer**:

  As input it uses add-ins that are packaged as standalone (rmf and RPKs) or further bundled in RSPAK format. Additional license files can be added through the Add-In Installer.

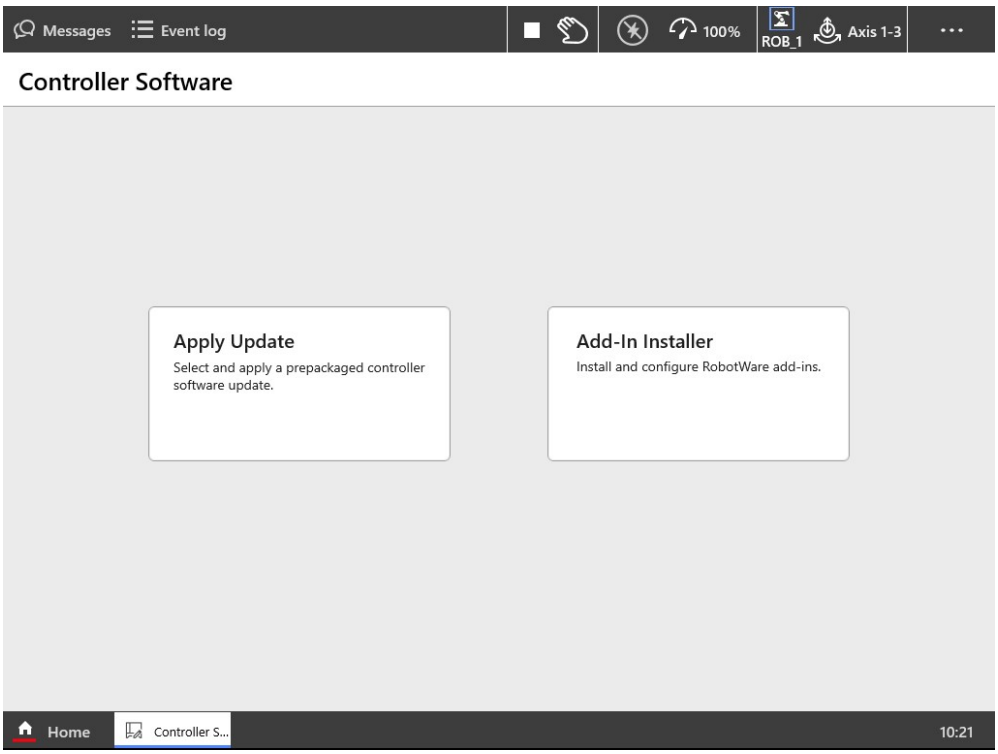
  See *Add-in installation via FlexPendant on page 203*.
- **Apply Update**:

  As input it uses an update package, defined by the user and created through a RobotStudio command-line tool **UpdatePackageCLI**.

  See *Apply Update overview on page 215*.

Continues on next page

The tools are accessed by selecting **Controller Software** on the FlexPendant start screen:



xx2300001392

Continues on next page

7.5.1 Add-in installation via FlexPendant

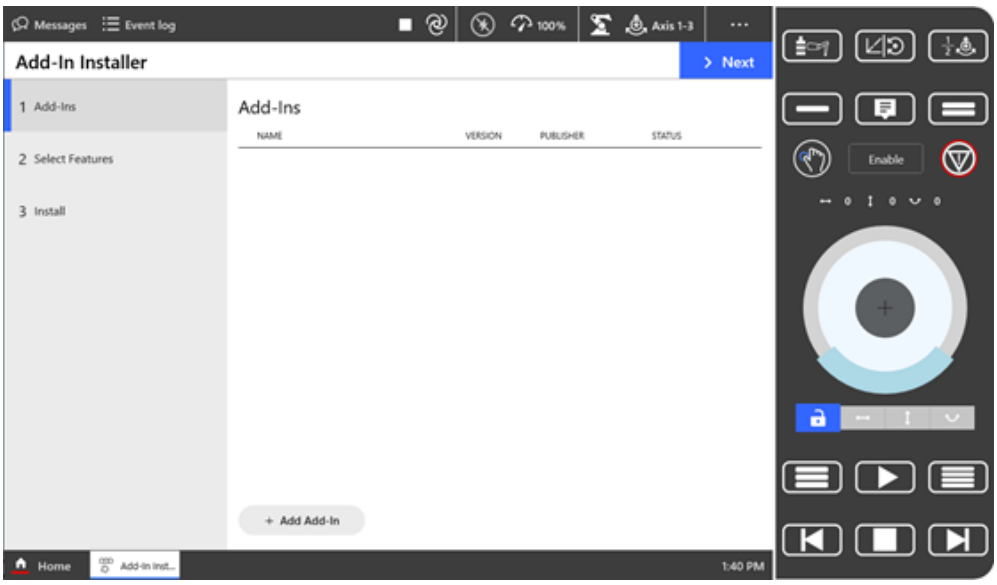
7.5.1.1 Introduction

About Add-In Installer

The Add-In Installer app provides means to install, update and uninstall add-ins on the controller. It can also be used to modify the selection of features for installed add-ins.

A developer of an add-in can create the package and distribute it on a USB storage device to the user. The user can, using the app, install the add-in directly from the FlexPendant, without having to connect to an external PC or network drive.

Add-ins and their features are governed by some rules and requirements, see *Application manual - RobotWare add-ins*. Knowledge of these rules is recommended for complex add-ins, as issues may arise if illegal combinations of feature selections are made. However, simple add-ins (without dependencies and rules) are easy to install using the app.



xx2200000749

Distribution of add-ins

The add-ins must be placed on a USB storage device so they can be used directly on the FlexPendant. It is expected that the add-in packages are distributed in the format that the Add-In Packaging Tool creates them, that is, a folder with the package (.rpk) file and the manifest (.rmf) file. Multiple add-ins can be placed on a USB device, and ordered into a folder hierarchy, as long as they conform to the requirements above.

Another way of packaging add-ins is to further package the rpk and rmf files into RobotStudio rspak format. In that case there is just one file to distribute, compared to a directory with rpk/rmf files. The rspak format can be particularly useful when

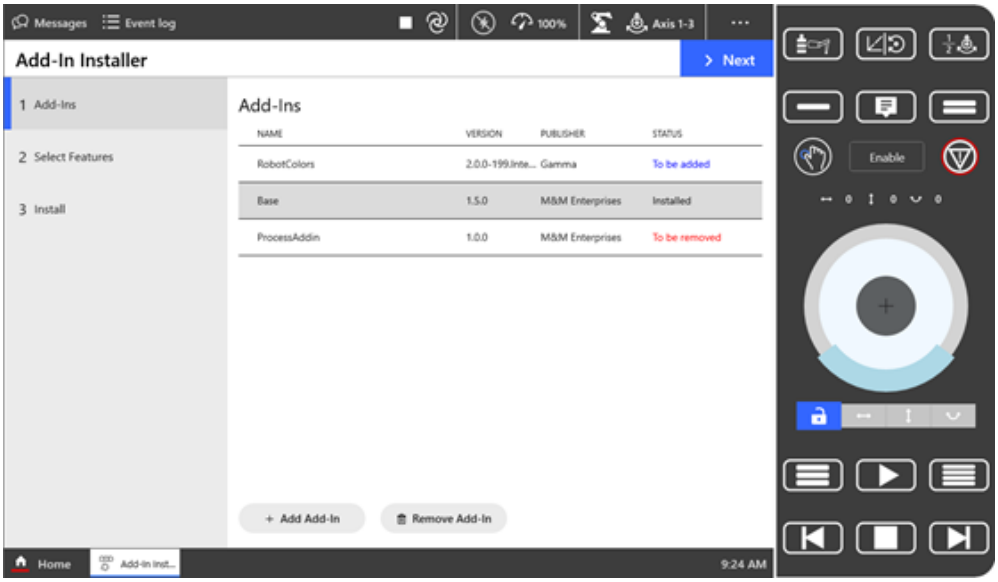
Continues on next page

7.5.1.1 Introduction
Continued

distributing multiple add-ins, since they can all be bundled to a single rspak file. For more information about creating rspak files, see RobotStudio Developer Center.

> **Note**
>
> If a folder contains several manifest files, it is not considered to be an add-in and cannot be installed by the tool.

Add-in statuses



xx2200000753

The add-ins that are included in the current configuration are listed in the main view of the Add-In Installer. The **Status** indicates what action will be taken when the installation is applied to the controller. To make it easier to identify changes, the status text is also color coded:

Status	Description	Status color
Installed	This add-in is installed on the controller. It will remain installed when changes are applied.	BLACK
Not installed	This add-in has been added to the configuration and is not installed on the controller. It will not be installed when changes are applied.	BLACK
To be added	This add-in has been added to the configuration and is not yet installed on the controller. It will be added to the controller when changes are applied.	BLUE
To be replaced	This add-in is installed on the controller and it will be replaced by another version of the add-in when changes are applied.	RED
To be removed	This add-in is currently installed on the controller but will be removed when changes are applied.	RED

Continues on next page

Prerequisites

The Add-In Installer app requires that add-ins are distributed on a USB storage device. Any other media, such as network connections or files on the controller are not supported. The USB storage device should be placed in the USB port on the bottom right side of the FlexPendant.

xx2200000751

Limitations

Add-In Installer only supports installation from USB devices (both RC and VC).

Other constraints

- Base add-ins from RobotWare distribution cannot be configured in the app.
- The user must have the UAS grant **Update a RobotWare system**.

Recommenced work process

The Add-In Installer consists of different configuration views between which you can navigate back and forth. The following steps describe the types of configuration that can be made:

1. **Configure the installation content** from the **Add-Ins** view:
   - Add new add-ins to the installation configuration, see *Adding a new add-in to the configuration on page 207*.
   - Remove add-ins from the installation configuration, see *Removing add-ins from the configuration on page 209*.
   - When add-ins are added, removed, upgraded and downgraded, this can cause missing dependencies and conflicting features. All these issues must be resolved before the installation can be verified and completed. See *Resolving conflicts on page 210*.
2. **Configure add-in features and resolve rule violations:**

   The **Select Features** view is used to modify the behavior of an add-in by enabling features it implements. See *Configure add-in features on page 211*.
3. **Verify the installation:**

Continues on next page

7.5.1.1 Introduction
Continued

In the **Install** view, a last verification of the changes is done and an overview is presented. If no issues are found the installation process can be completed by applying the changes made. See *Verifying an add-in installation on page 213*.

Note

Nothing is changed on the controller until the work is applied in the **Install** view. If the app is restarted (closed and opened again), the app will undo all previous changes.

7.5.1.2 Adding a new add-in to the configuration

Instructions

1 Tap **Add-In Installer** and then select **Add Add-In** to open the add-in browser.

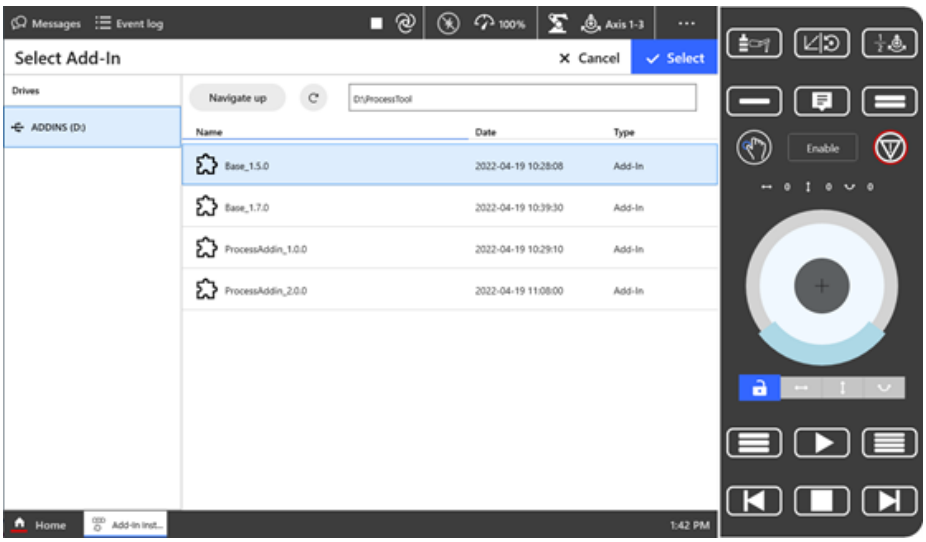
2 The **Select Add-In** view shows a filtered version of the file system on the USB device. It only show folders, add-ins and manifest (.rmf) files.

> **Note**
>
> The **Add-In Installer** uses an id assigned by the developer of the add-in to recognize the add-ins.

> **Note**
>
> Only one manifest file per add-in folder is supported.

> **Tip**
>
> Tap a folder in the list to navigate down into the folder. Tap **Navigate up** to return to the previous level in the hierarchy.
>
> If the content of the USB device has changed, tap the **Refresh** button.

3 Select the add-in to be added to the installation and tap **Select**.



xx2200000752

4 The add-in will now be added to the installation configuration, and its status will be changed to **To be added**.

Continues on next page

7.5.1.2 Adding a new add-in to the configuration
Continued

5 The new add-in may change the status of other versions of the same add-in. If the new addition causes a conflict, the earlier versions of the add-in must be removed. See *Resolving conflicts on page 210*.

> **Note**
>
> Only one version of an add-in can be installed at the same time.

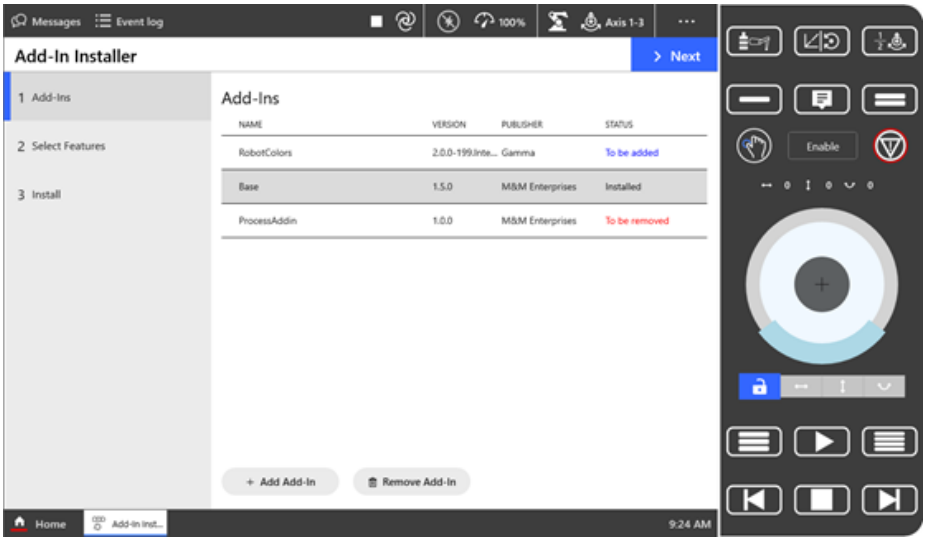
6 If there are no conflicts due to the new add-in, see *Verifying an add-in installation on page 213* for information about how to verify and finalize the installation.

7.5.1.3 Removing add-ins from the configuration

Instructions

1 On the start screen, tap **Add-In Installer** and then select **Add-Ins** from the menu.

2 In the **Add-Ins** view, select the add-in to be removed and tap **Remove Add-In**.



xx2200000753

The add-in selected for removal will get the status **To be removed**.

3 The add-in that was removed may affect dependencies to other add-ins. If the removed add-in causes a conflict, this must be resolved. See *Resolving conflicts on page 210*.

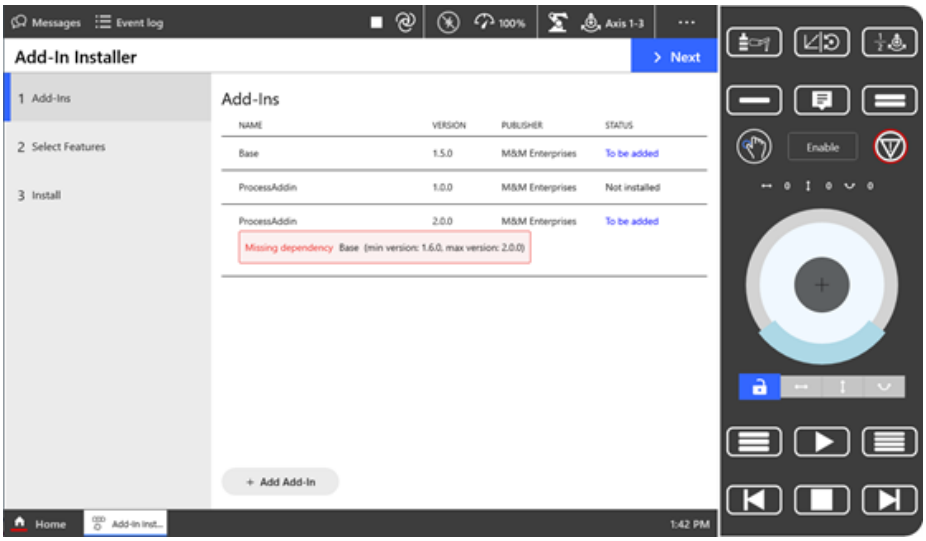
4 If there are no conflicts due to the removed add-in, see *Verifying an add-in installation on page 213* for information about how to verify and finalize the installation.

7.5.1.4 Resolving conflicts

Instructions

When add-ins are added, removed, upgraded and downgraded, this can cause missing dependencies and feature selection conflicts. If one add-in is dependent of another add-in that is not available in the system, a notification will be shown which identifies the required add-in and its expected version. All these issues must be resolved before the installation can be verified and completed.

1. On the start screen, tap **Add-In Installer** and then select **Add-Ins** from the menu.
2. If previous changes to the installation configuration have caused conflicts, this is indicated by the system:



xx2200000901

If a required add-in is missing, see *Adding a new add-in to the configuration on page 207*.

Note

All issues must be resolved, or it will not be possible to install the updates.

3. When all conflicts have been resolved, see *Verifying an add-in installation on page 213* for information about how to verify and finalize the installation.

7.5.1.5 Configure add-in features

Add-in features

Most add-ins have selectable features, also called options, that are used to set up the add-in. If any of the add-ins that will be installed on the system (whether newly added or already present) has features, these can be configured in the **Select Features** view.

Features can have dependencies and rules that regulate how they can be combined with other features (see *Application manual - RobotWare add-ins*). In these cases, the following issues are displayed:

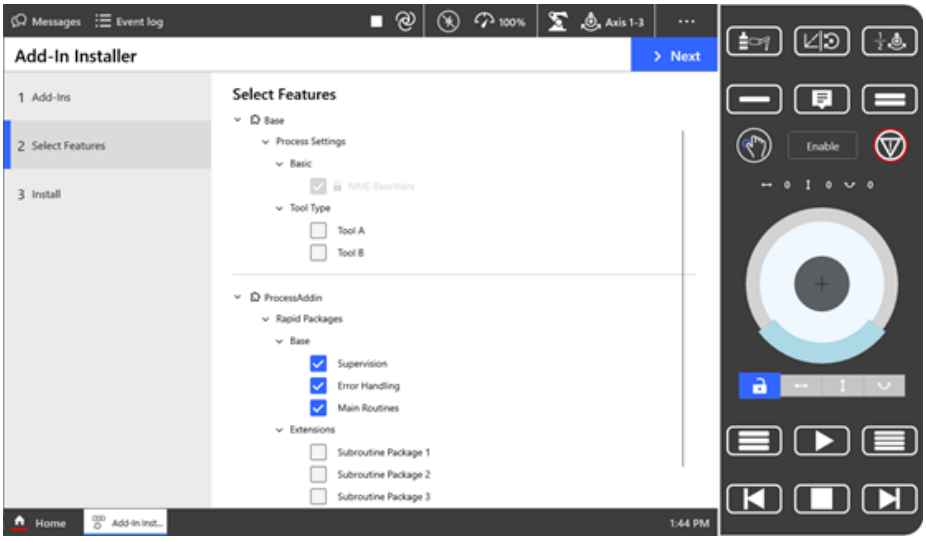
Issue	Description	Solution
Feature Conflict	The specified feature cannot be selected if this feature is selected.	Deselect the feature that is specified or the feature with the issue.
Feature Missing	The specified feature must be selected if this feature is selected.	Locate the missing feature and select it or deselect the feature with the issue. This may require adding another add-in.

Note

All issues must be resolved to be able to install updates.

Instructions

1. On the start screen, tap **Add-In Installer** and then select **Select Features** from the menu.
2. In the **Select Features** view, select all features that should be activated for the add-in.

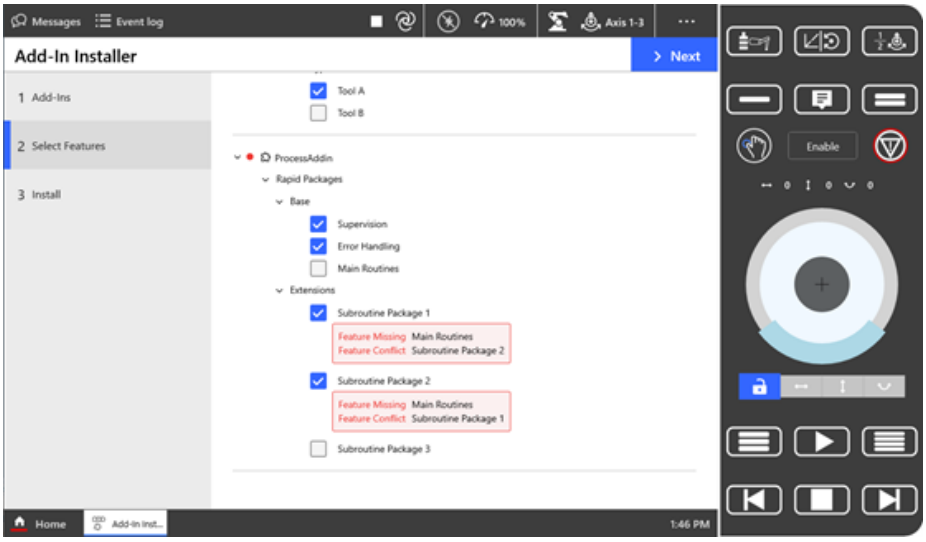


xx2200000902

Continues on next page

> **Note**
>
> Locked features cannot be changed.

3 If any issues arise when selecting a feature, the user will be notified with a message box below the feature. The description in the notification contains a reason and the name of the failing feature:



xx2200000903

> **Note**
>
> All issues must be resolved to be able to install updates.

4 If the app cannot find the name of the feature, its id will be given instead. Ids are written with the prefix *ID:* as to be distinguishable from names:

xx2200000904

5 When all feature conflicts are resolved, see *Verifying an add-in installation on page 213* for information about how to verify and finalize the installation.

7.5.1.6 Verifying an add-in installation

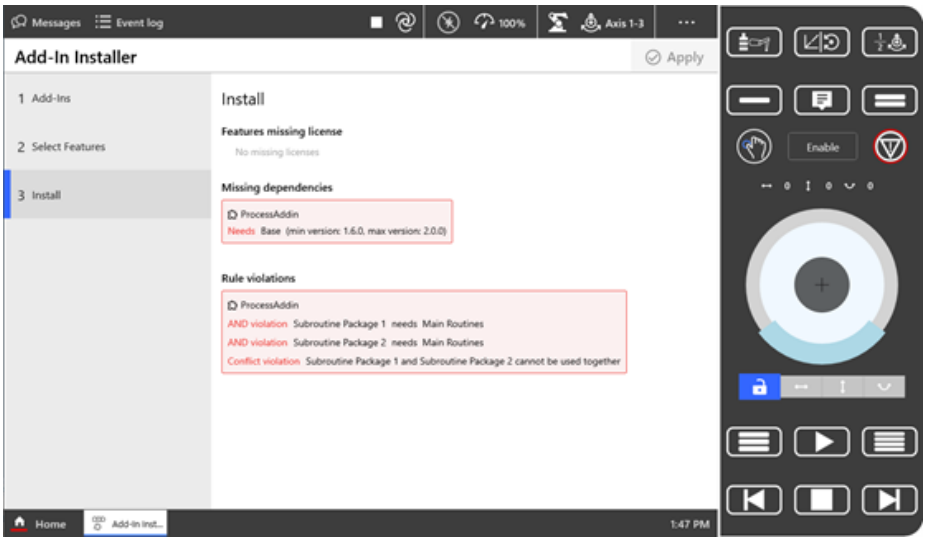
Prerequisites

Before the configuration can be applied to the controller, all dependencies and rule violations must be resolved. If not, the **Apply** button will be disabled.

- Missing dependencies are resolved in the **Add-Ins** view, see *Resolving conflicts on page 210*.
- Rule violations are resolved in the **Select Features** view, see *Configure add-in features on page 211*.

Instructions

1. On the start screen, tap **Add-In Installer** and then select **Install** from the menu.
2. The **Install** view presents a summary of any issues that prevent the changes to be applied to the controller:



xx2200000905

3. If there are any issues, these must be resolved before the installation can be completed. See *Resolving conflicts on page 210* and *Configure add-in features on page 211*.
4. When all issues have been resolved, tap **Apply** to confirm the installation.

> **Note**
>
> The **Apply** button is disabled if no changes have been made, or if there are missing dependencies or rule violations that prevent the installation.

Continues on next page

7.5.1.6 Verifying an add-in installation
Continued

When changes are applied to the controller, it will restart several times to complete the operation.

Note

For installation on virtual controller:

- When applying the installation, a Windows command prompt can briefly be seen as the installation is performed by an external application (VCInstaller.exe).
- The system will be reset when applying the update. Be sure to create a backup before applying the changes.

7.5.2 Updating installed controller software using the Apply Update tool

Apply Update overview

The **Apply Update** tool is used to update the installed controller software to a new version using a previously prepared update package.

An update package can contain any number of software products and distributions. for example, the update package can contain a subset, an exact match, or a superset of the software present on a controller.

When dealing with multiple controllers, to reduce number of different update packages, it is convenient to include a superset of products and make an update package applicable to several controllers. It is also possible to store several update packages on a single USB stick and put them in different folders.

Note

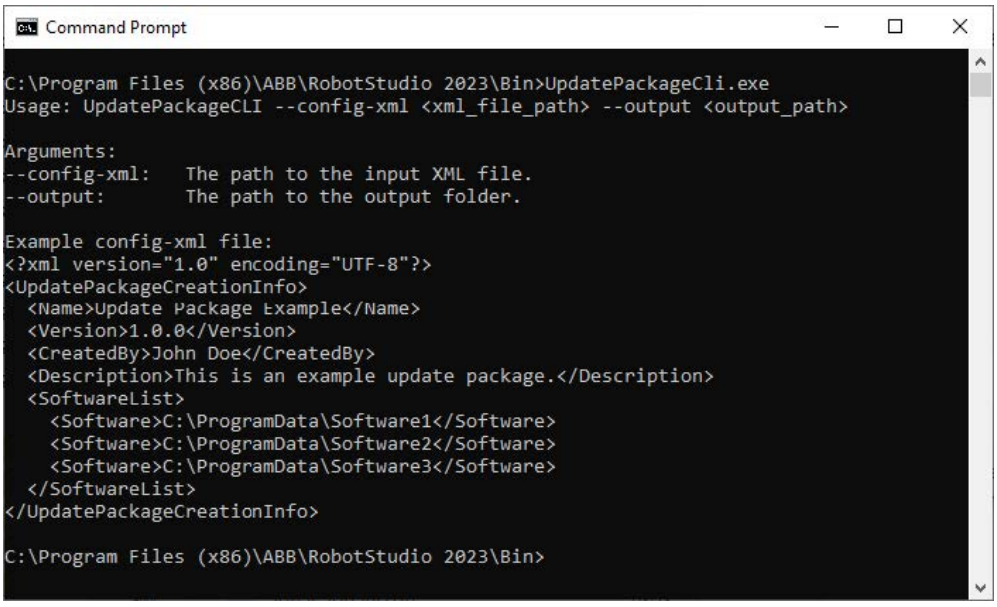
Both RobotWare and add-ins can be combined in a single update package. However, only one version of a software product can be included.

Preparing an update package

The update package is created using a RobotStudio command line tool called **UpdatePackageCLI.exe**. The tool can be accessed from the RobotStudio installation folder on your PC, for example:

```
C:\Program Files (x86)\ABB\RobotStudio 2023\Bin
```

To find out how to use the tool, run it without any arguments and help will be displayed. The tool uses a configuration file to specify what software shall be included in the update package, and for example package name.



xx2300001393

Continues on next page

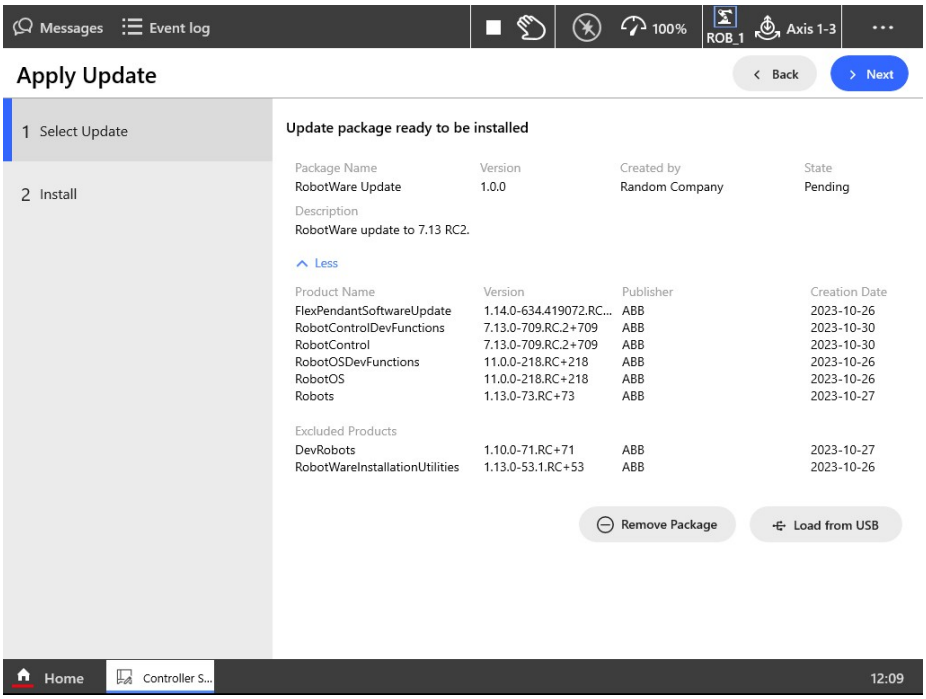
7.5.2 Updating installed controller software using the **Apply Update** tool
Continued

> **Note**
>
> In case of a software distribution, it is recommended to set up the software path <Software> to point to a root folder of a software distribution:
>
> ```
> <Software>
> C:\ProgramData\ABB\DistributionPackages\ABB.RobotWare-7.12.0
> </Software>
> ```

Applying the update package via FlexPendant

> **Note**
>
> There is a slight difference when working with virtual and real controllers. When working with virtual controllers, a backup of the system must be created by the user before the update and restored manually after. When working with real controllers, this is always automatically done by the controller itself.

1. On the FlexPendant start screen, tap **Controller Software**, and then **Apply Update**.
2. In the **Apply Update** view, select an update package from your USB stick.
3. The contents of the selected update package is matched against the current controller, and the software that will be updated is listed:

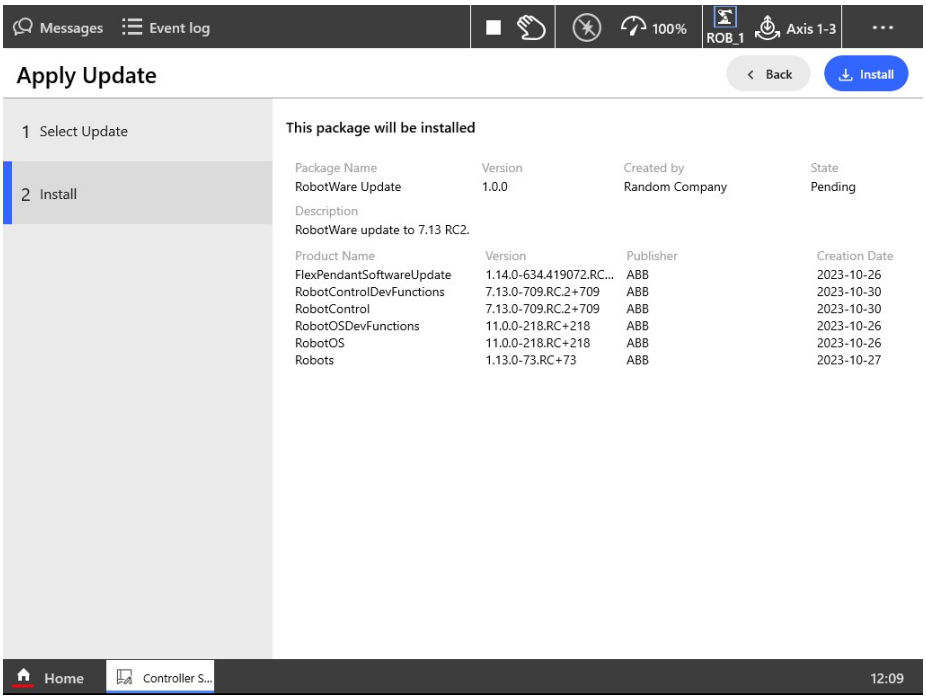


xx2300001394

4. Review the content, and select **Load from USB** to continue to confirm the changes.

Continues on next page

5 The **Install** view is displayed:



xx2300001395

6 Select **Install** to apply the changes.

7.6 Installing RobotWare add-ins

Installing RobotWare add-ins

The main steps required to correctly install a RobotWare add-in is described in *Operating manual - RobotStudio*.

For instruction on building RobotWare add-ins, visit the ABB Robotics Developer Center web site at *http://developercenter.robotstudio.com* or see *Application manual - RobotWare add-ins*.

8 Calibration

8.1 Robot calibration

About robot calibration

The procedures for fine calibration of a robot and updating revolution counters are different for different robots. For instructions, see the product manual for the robot.

8.2 How to check if the robot needs calibration

Check robot calibration status

Use the following procedure to check the calibration status of the robot:

1. On the start screen, tap **Calibrate**, and then **Calibration**.
2. The **Mechanical Unit** page is displayed.

> **Note**
>
> This step is required only if you are not already in the **Mechanical Unit** page when you open **Calibrate**.

> **Note**
>
> The **Mechanical Unit** page is displayed only if there are more than one mechanical unit available. Otherwise, the calibration summary page for the available mechanical unit is displayed.

3. Select the unit that needs to be calibrated from the **Mechanical Unit** list.

   The calibration summary page for the selected mechanical unit is displayed. The **Calibration Status** column displays the status of calibration for each axis.

What kind of calibration is needed?

If the calibration status is...	then...
Calibrated	Calibration is not needed.
Not calibrated	the robot must be fine calibrated by a qualified service technician. Performing a fine calibration is described in the product manual for the robot. **DANGER** Do not attempt to perform the fine calibration procedure without proper training and tools. Doing so may result in the incorrect positioning that may cause injuries and property damage. Always consult a qualified service technician.
Not updated	update the revolution counters or perform the calibration. **Note** For IRB 14050 when you select update the revolution counters, you are recommended to perform the calibration. Updating the revolution counters is described in the product manual for the robot.

Continues on next page

If the calibration status is...	then...
Not commutated	the robot must be fine calibrated by a qualified service technician. Performing a fine calibration is described in the product manual for the robot.  **DANGER** Do not attempt to perform the fine calibration procedure without the proper training and tools. Doing so may result in incorrect positioning that may cause injuries and property damage. Always consult a qualified service technician.

8.3 Update calibration data using the FlexPendant

Overview

This section describes how to load and update calibration data for using the FlexPendant.

The calibration data is normally stored in the robot memory of each robot, regardless of whether the robot runs an absolute measurement system (*Absolute Accuracy* option is installed, *AbsAcc*) or not. This data is normally transferred automatically to the controller when the system is powered up, and in such cases no action is required by the operator.

Verify that the correct robot memory (SMB) data has been loaded into the system as detailed below.

Load and update calibration data

Use the following procedure to update the calibration data:

1. On the start screen, tap **Calibrate**, and then **Calibration**.
2. The **Mechanical Unit** page is displayed.

   Note

   This step is required only if you are not already in the **Mechanical Unit** page when you open **Calibrate**.

   Note

   The **Mechanical Unit** page is displayed only if there are more than one mechanical unit available. Otherwise, the calibration summary page for the available mechanical unit is displayed.

3. Select the unit from the **Mechanical Unit** list.

   The calibration summary for the selected mechanical unit is displayed.
4. On the right pane tap **Calibration Methods**.
5. Tap **Robot Memory**.

   The status for controller and robot memory is displayed.

   Note

   If the status **Valid** is displayed for the data in **Controller Memory** and **Robot memory**, calibration data is correct.

Continues on next page

6 If the status is **Not Valid**, the data (on the SMB board or in the controller) must be replaced with the correct data as detailed below:

> **Note**
>
> If, for instance, the SMB board has been replaced, transfer the data from controller to SMB board. If the controller has been replaced, transfer the data from the SMB board to the controller.

- Select the option **Update controller with robot memory data** or **Update robot memory with controller data** and tap **Update**.

7 After loading the calibration data, proceed with updating the revolution counters.

8.4 Editing motor calibration offset

Editing motor calibration offset

This procedure should be used when no specific file with motor calibration data is available, but only the numerical values. These values are normally found on a sticker on the rear of the robot.

Entering motor calibration values can be done in three ways:

- From a disk, using the FlexPendant (as detailed in section *Load and update calibration data on page 222*).
- From a disk, using RobotStudio (as detailed in *Operating manual - RobotStudio*).
- Manually entering the values, using the FlexPendant (as detailed in section *Editing motor calibration offset on page 224*).

1 On the start screen, tap **Calibrate**, and then **Calibration**.

2 The **Mechanical Unit** page is displayed.

> **Note**
>
> This step is required only if you are not already in the **Mechanical Unit** page when you open **Calibrate**.

> **Note**
>
> The **Mechanical Unit** page is displayed only if there are more than one mechanical unit available. Otherwise, the calibration summary page for the available mechanical unit is displayed.

3 Select the unit from the **Mechanical Unit** list.

The calibration summary for the selected mechanical unit is displayed.

4 On the right pane tap **Calibration Methods**.

5 Tap **Calibration Parameters** and then tap **Edit Motor Calibration Offset**.

A confirmation window is displayed.

6 Tap **Yes**.

The **Edit Motor Calibration Offset** window is displayed.

7 For each axis tap in the value column and edit the offset value.

> **Note**
>
> The allowed value range is displayed at the bottom of the window.

8 Tap **Apply**.

Controller warm start required window is displayed.

9 Tap **Yes**.

The controller is restarted the motor calibration values are saved.

Continues on next page

After restarting, the calibration data in the controller and on the serial measurement board will differ. To update this follow the procedure *Load and update calibration data on page 222*.

8.5 Robot memory

Overview

The robot memory stores a number of data for the specific robot, for example, the resolver data from the motors. For most robots, the data is stored in the serial measurement board (SMB).

The data is used by the controller and can be transferred between the robot memory and the controller memory. Normally, the data is transferred automatically, but it can also be done manually.

The data in the memory is affected when:

- The robot is replaced.
- The SMB is replaced.
- The controller (or its main computer) is replaced.
- Updating with new calibration data.

The following data is stored in the robot memory:

- Serial number for the mechanical unit
- Joint calibration data
- Absolute accuracy data
- SIS data (Service Information System)

Robot memory data update

If...	then...
the **main computer** or complete **controller** is new or replaced by an unused spare part...	the data stored in the robot memory is automatically copied to the controller memory.
the **SMB** is replaced by a new, unused, spare part SMB...	the data stored in the controller memory is automatically copied to the robot memory.
the **main computer** or complete **controller** is replaced by a spare part, previously used in another system...	the data in the controller memory and the robot memory is different. You must manually **update the controller memory with robot memory data**.
the **SMB** is replaced by a spare part SMB, previously used in another system...	the data in the controller memory and the robot SMB memory is different. You must first **clear the data in the new robot memory**, and then **update the robot memory** with the data from the controller memory.
new **calibration data** has been loaded through FlexPendant or RobotStudio and the system has been restarted...	the data in the controller memory and the robot memory is different. You must manually **update the robot memory with controller memory**. Check that the new calibration values belong to a manipulator with the serial number defined in your system.

View robot memory data status

Use the following procedure to view the data status in the serial measurement board and the controller.

1 On the start screen, tap **Calibrate**, and then **Calibration**.

Continues on next page

2 The **Mechanical Unit** page is displayed.

> **Note**
>
> This step is required only if you are not already in the **Mechanical Unit** page when you open **Calibrate**.

> **Note**
>
> The **Mechanical Unit** page is displayed only if there are more than one mechanical unit available. Otherwise, the calibration summary page for the available mechanical unit is displayed.

3 Select the unit from the **Mechanical Unit** list.

The calibration summary for the selected mechanical unit is displayed.

4 On the right pane tap **Calibration Methods**.

5 Tap **Robot Memory**.

The **Robot Memory** window displays the status for controller and robot memory.

Update controller memory with the robot memory data

Use the following procedure to update controller memory with the robot memory data:

1 On the start screen, tap **Calibrate**, and then **Calibration**.

2 The **Mechanical Unit** page is displayed.

> **Note**
>
> This step is required only if you are not already in the **Mechanical Unit** page when you open **Calibrate**.

> **Note**
>
> The **Mechanical Unit** page is displayed only if there are more than one mechanical unit available. Otherwise, the calibration summary page for the available mechanical unit is displayed.

3 Select the unit from the **Mechanical Unit** list.

The calibration summary for the selected mechanical unit is displayed.

4 On the right pane tap **Calibration Methods**.

5 Tap **Robot Memory**.

The **Robot Memory** window displays the status for controller and robot memory.

6 In the **Update** section, select the option **Update controller with robot memory data**.

7 Tap **Apply**.

Continues on next page

A confirmation window is displayed.

8 Tap **Yes**.

The **Controller warm start required** window is displayed.

9 Tap **Yes**.

The controller is restarted and the changes are applied.

Update robot memory with the controller memory data

Usually you need to update the robot memory data when SMB has been replaced or when the calibration data has been loaded to the controller through RobotStudio or FlexPendant.

Use the following procedure to update the robot memory data with the controller memory data:

1 On the start screen, tap **Calibrate**, and then **Calibration**.

2 The **Mechanical Unit** page is displayed.

> **Note**
>
> This step is required only if you are not already in the **Mechanical Unit** page when you open **Calibrate**.

> **Note**
>
> The **Mechanical Unit** page is displayed only if there are more than one mechanical unit available. Otherwise, the calibration summary page for the available mechanical unit is displayed.

3 Select the unit from the **Mechanical Unit** list.

The calibration summary for the selected mechanical unit is displayed.

4 On the right pane tap **Calibration Methods**.

5 Tap **Robot Memory**.

The **Robot Memory** window displays the status for controller and robot memory.

6 In the **Update** section, select the option **Update robot memory with controller data**.

7 Tap **Apply**.

A confirmation window is displayed.

8 Tap **Yes**.

The robot memory is updated.

Delete the controller memory or robot memory data

This section describes how to delete the robot memory data or the controller memory data when creating spare parts.

Use the following procedure to delete the controller memory or robot memory data:

1 On the start screen, tap **Calibrate**, and then **Calibration**.

Continues on next page

2 The **Mechanical Unit** page is displayed.

> **Note**
>
> This step is required only if you are not already in the **Mechanical Unit** page when you open **Calibrate**.

> **Note**
>
> The **Mechanical Unit** page is displayed only if there are more than one mechanical unit available. Otherwise, the calibration summary page for the available mechanical unit is displayed.

3 Select the unit from the **Mechanical Unit** list.

The calibration summary for the selected mechanical unit is displayed.

4 On the right pane tap **Calibration Methods**.

5 Tap **Robot Memory**.

The **Robot Memory** window displays the status for controller and robot memory.

6 In the **Advanced** section, select the option **Clear Controller Memory** or **Clear Robot Memory**.

7 Tap **Clear**.

A confirmation window is displayed.

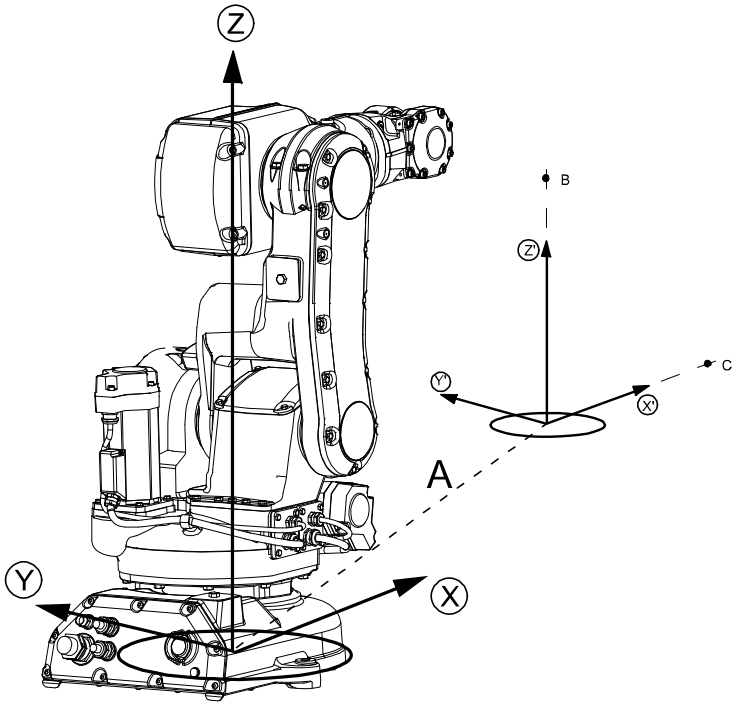
8 Tap **Yes**.

The data in the selected memory is deleted.

8.6 Base Frame calibration - 4 points XZ calibration

Overview

This section describes how to define the base frame using the 4 points XZ method. This method can move and rotate the base frame in relation to the world frame. Normally the base frame is centered and aligned with the world frame. The base frame is fixed to the base of the robot.



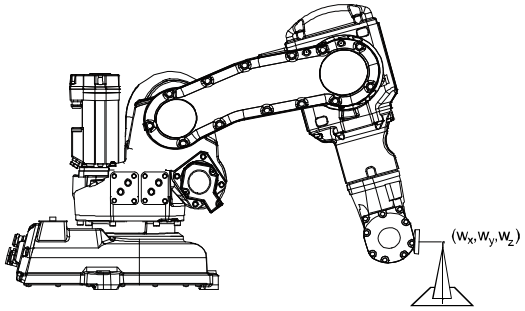
xx0400000782

A	Displacement distance between base frame and world frame
B	Elongator point Z’
C	Elongator point X’
X	X-axis in the base frame
Y	Y-axis in the base frame
Z	Z-axis in the base frame
X'	X-axis in the world frame
Y'	Y-axis in the world frame
Z'	Z-axis in the world frame

Continues on next page

Fixed reference Position

The calibration procedure requires that the tip of the tool is calibrated against a fixed reference position. The fixed position could be a manufactured World fixed tip device to facilitate finding the elongator points. The fixed reference position is the distance in meters (in (x,y,z)) between the fixed position and the world frame.



Calibrate_xx

Running the 4 points XZ calibration

1 On the start screen, tap **Calibrate**, and then **Calibration**.

2 The **Mechanical Unit** page is displayed.

> **Note**
>
> This step is required only if you are not already in the **Mechanical Unit** page when you open **Calibrate**.

> **Note**
>
> The **Mechanical Unit** page is displayed only if there are more than one mechanical unit available. Otherwise, the calibration summary page for the available mechanical unit is displayed.

3 Select the unit from the **Mechanical Unit** list.

The calibration summary for the selected mechanical unit is displayed.

4 On the right pane tap **Calibration Methods**.

5 Tap **Define Base Frame**.

The **Define Base Frame** wizard is displayed. The method **4 Points XZ** is selected by default.

6 Tap **Next**.

The **Define Base Frame** tab is displayed.

7 In the **Set World Reference Point (X Y Z)** field define the reference points. The default value is (0, 0, 0).

Continues on next page

8 If the calibration positions exists in a file, follow the below instruction. Otherwise proceed to the next step.
   - Tap the **Positions** button and then tap the **Load** option to load the file containing the base frame calibration values.

     Once this is successfully done, proceed to step 12.

9 Select Point 1, jog the robot to the desired position, and tap **Modify**.

   The selected point is modified and and the status is updated.

10 Repeat the previous step for the other 3 points.

   The selected points are modified and the status is updated.

11 Select Elongator X, jog the robot to a position where the tool center point (TCP) touches an imaginary extension of the X-axis, and tap **Modify**.

   The selected point is modified and the status is updated.

12 Select Elongator Z, jog the robot to a position where the tool center point (TCP) touches an imaginary extension of the Z-axis, and tap **Modify**.

   The selected point is modified and the status is updated.

> **Note**
>
> You can use the **Positions** button to perform the following tasks:
> - **Reset All**: Resets the current modified points
> - **Load**: Loads the saved base frame calibration points.
> - **Save**: Saves the selected calibration points to a RAPID module.

13 Tap **Next**.

   The **Calibration Results** page is displayed.

14 Tap **Finish**.

   The restart confirmation window is displayed

15 Tap **Restart Now**.

   The controller is restarted and the base frame is calibrated.

9 Descriptions of terms and concepts

9.1 What is a tool?

Tool

A tool is an object that can be mounted directly or indirectly on the robot turning disk or fitted in a fixed position within the robot working range.

Note

A fixture (jig) is not a tool.

All tools must be defined with a TCP (Tool Center Point).

Each tool that can be used by the robot must be measured and its data stored in order to achieve accurate positioning of the tool center point.

WARNING

It is important to always define the actual tool load and, when used, the payload of the robot (for example, a gripped part). Incorrect definitions of load data can result in overloading of the robot mechanical structure. There is also a risk that the speed in manual reduced speed mode can be exceeded.

When incorrect load data is specified, it can often lead to the following consequences:

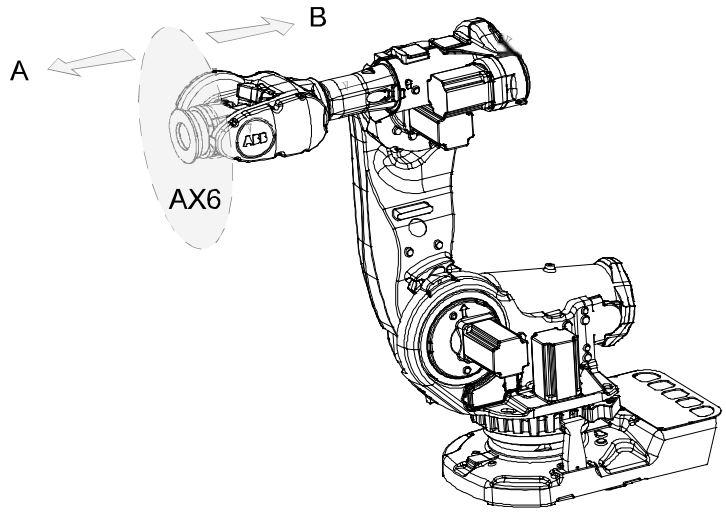
- The robot may not use its maximum capacity.
- Impaired path accuracy including a risk of overshooting.
- Risk of overloading the mechanical structure.

The controller continuously monitors the load and writes an event log if the load is higher than expected. This event log is saved and logged in the controller memory.

Continues on next page

9.1 What is a tool?
Continued

Illustration



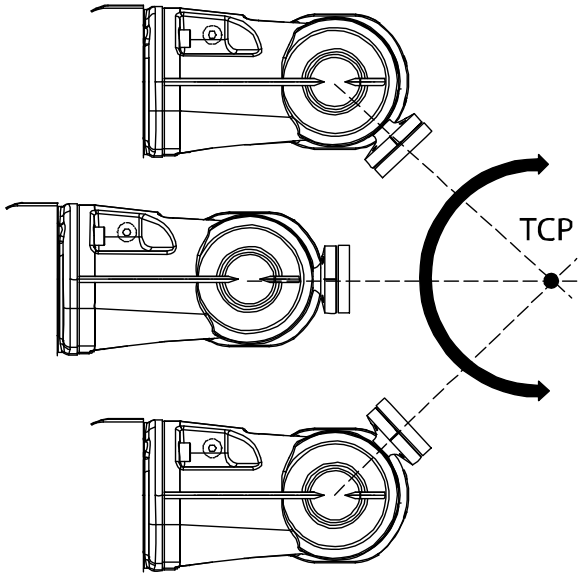
en0400000803

A	Tool side
B	Robot side

9.2 What is the tool center point?

Illustration

The illustration shows how the tool center point (TCP) is the point around which the orientation of the tool/manipulator wrist is being defined.



xx0300000604

Description

The tool center point (TCP) is the point in relation to which all robot positioning is defined. Usually the TCP is defined as relative to a position on the manipulator turning disk.

> **CAUTION**
>
> Incorrect settings for the TCP will result in incorrect speed. Always verify the speed after changing the settings.

The TCP will be jogged or moved to the programmed target position. The tool center point also constitutes the origin of the tool coordinate system.

The robot system can handle a number of TCP definitions, but only one can be active at any one time.

There are two basic types of TCPs: moveable or stationary.

Moving TCP

The vast majority of all applications deal with moving TCP, i.e. a TCP that moves in space along with the manipulator.

A typical moving TCP can be defined in relation to, for example the tip of a arc welding gun, the center of a spot welding gun, or the end of a grading tool.

Continues on next page

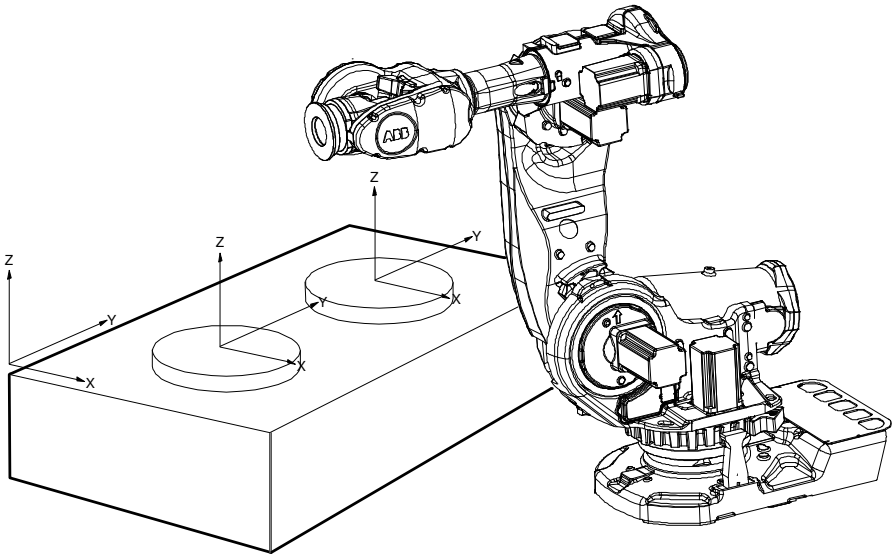
9.2 What is the tool center point?
Continued

Stationary TCP

In some applications a stationary TCP is used, for example when a stationary spot welding gun is used. In such cases the TCP can be defined in relation to the stationary equipment instead of the moving manipulator.

9.3 What is a work object?

Illustration



en0400000819

Description

A work object is a coordinate system with specific properties attached to it. It is mainly used to simplify programming when editing programs due to displacements of specific tasks, objects processes etc.

The work object coordinate system must be defined in two frames, the user frame (related to the world frame) and the object frame (related to the user frame).

Work objects are often created to simplify jogging along the object's surfaces. There might be several different work objects created so you must choose which one to use for jogging.

Payloads are important when working with grippers. In order to position and manipulate an object as accurate as possible its weight must be accounted for. You must choose which one to use for jogging.

9.4 What is a coordinate system?

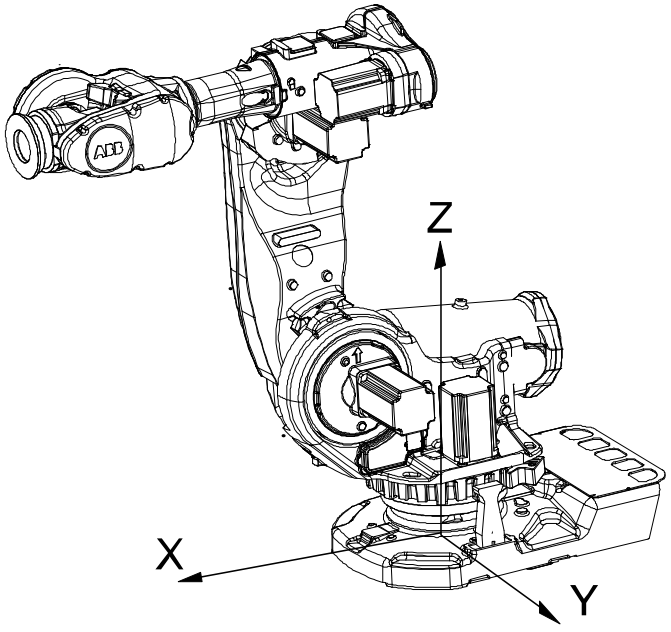
Overview

A coordinate system defines a plane or space by axes from a fixed point called the origin. Robot targets and positions are located by measurements along the axes of coordinate systems.

A robot uses several coordinate systems, each suitable for specific types of jogging or programming.

- The *base coordinate system* is located at the base of the robot. It is the easiest one for just moving the robot from one position to another. See *The base coordinate system on page 238* for more information.
- The *world coordinate system* that defines the robot cell, all other coordinate systems are related to the world coordinate system, either directly or indirectly. It is useful for jogging, general movements and for handling stations and cells with several robots or robots moved by external axes. See *The world coordinate system on page 239* for more information.
- The *user coordinate system* is useful for representing equipment that holds other coordinate systems, like work objects. See *The user coordinate system on page 240* for more information.
- The *work object coordinate system* is related to the work piece and is often the best one for programming the robot. See *The work object coordinate system on page 241* for more information.
- The *tool coordinate system* defines the position of the tool the robot uses when reaching the programmed targets. See *The tool coordinate system on page 242* for more information.

The base coordinate system



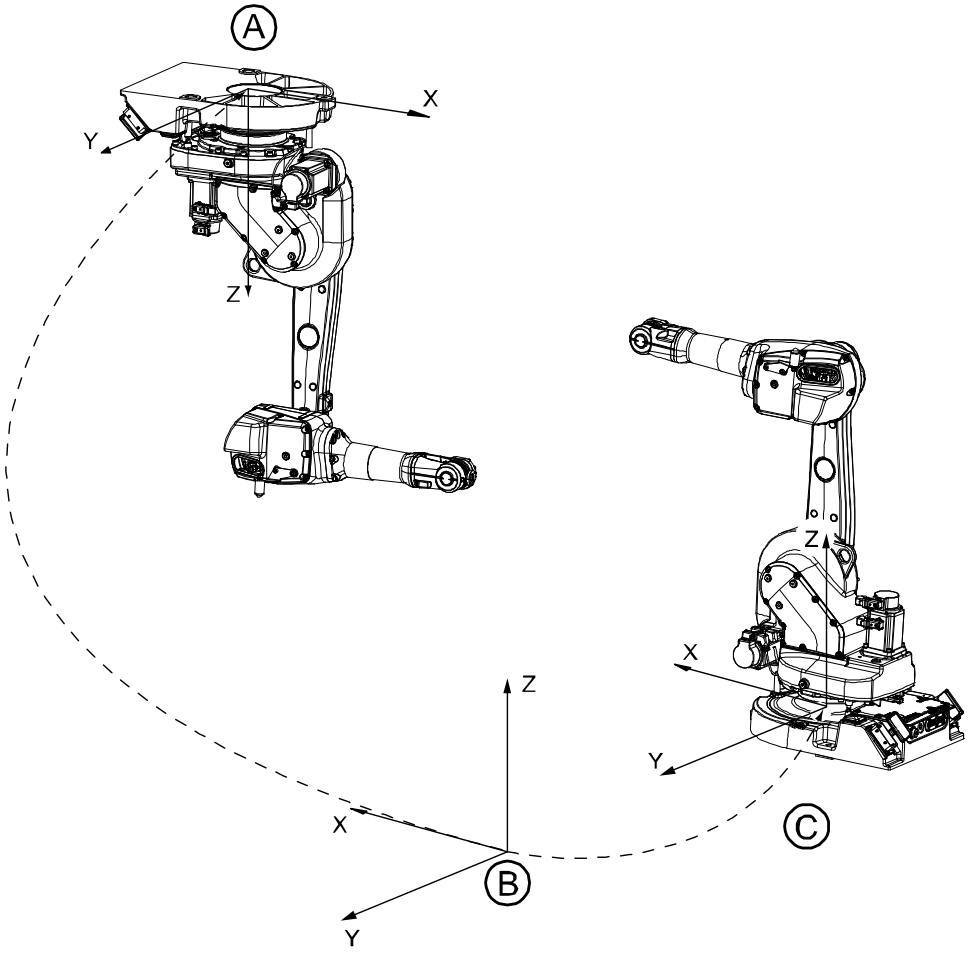
xx0300000495

Continues on next page

The base coordinate system has its zero point in the base of the robot, which makes movements predictable for fixed mounted robots. It is therefore useful for jogging a robot from one position to another. For programming a robot, other coordinate systems, like the work object coordinate system are often better choices. See *The work object coordinate system on page 241*for more information.

When you are standing in front of the robot and jog in the base coordinate system, in a normally configured robot system, pulling the joystick towards you will move the robot along the X axis, while moving the joystick to the sides will move the robot along the Y axis. Twisting the joystick will move the robot along the Z axis.

The world coordinate system



en0300000496

A	Base coordinate system for robot 1
B	World coordinate
C	Base coordinate system for robot 2

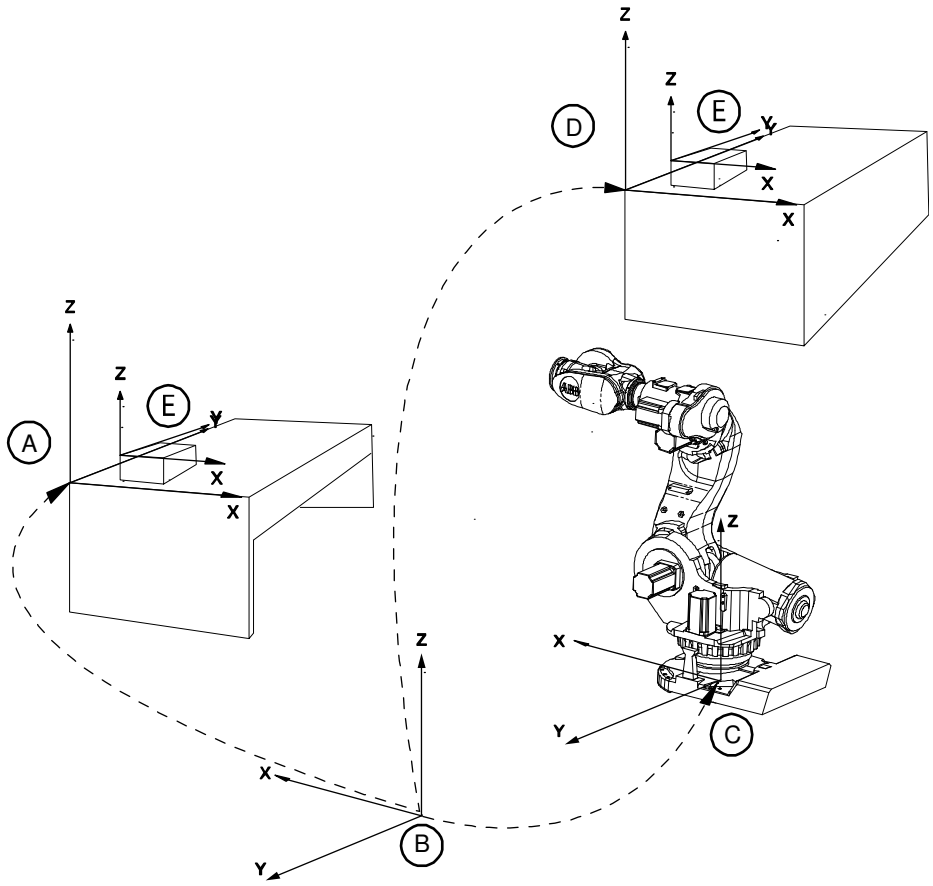
The world coordinate system has its zero point on a fixed position in the cell or station. This makes it useful for handling several robots or robots moved by external axes.

By default the world coordinate system coincides with the base coordinate system.

Continues on next page

9.4 What is a coordinate system?
Continued

The user coordinate system



en0400001225

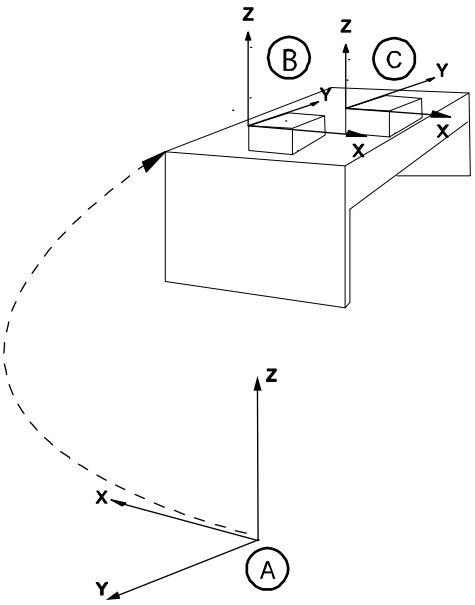
A	User coordinate system
B	World coordinate system
C	Base object coordinate system
D	Moved user coordinate system
E	Work object coordinate system, moved with user coordinate system

The user coordinate system can be used for representing equipment like fixtures, workbenches. This gives an extra level in the chain of related coordinate systems, which might be useful for handling equipment that hold work objects or other coordinate systems.

For information on how to define the user coordinate system, see information about the data type `wobjdata` in *Technical reference manual - RAPID Instructions, Functions and Data types*.

Continues on next page

The work object coordinate system



xx0600002738

A	World coordinate system
B	Work Object coordinate system 1
C	Work Object coordinate system 2

The work object coordinate system corresponds to the work piece: It defines the placement of the work piece in relation to the world coordinate system (or any other coordinate system).

A robot can have several work object coordinate systems, either for representing different work pieces or several copies of the same work piece at different locations.

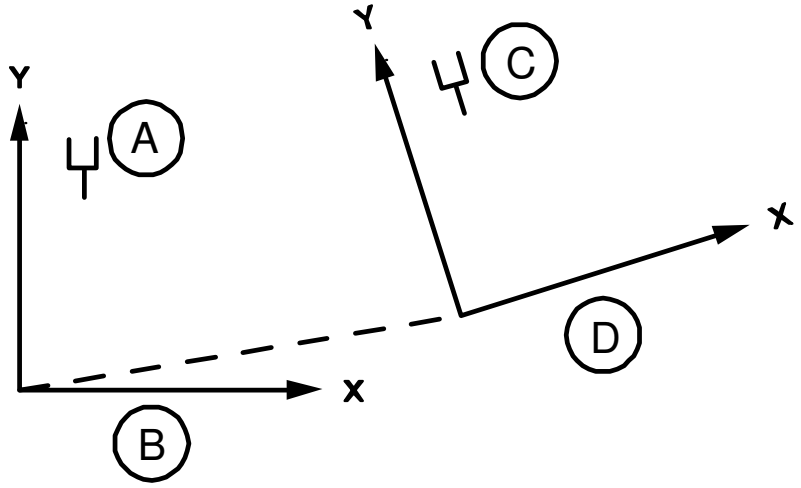
It is in work object coordinate systems you create targets and paths when programming the robot. This gives a lot of advantages:

- When repositioning the work piece in the station you just change the position of the work object coordinate system and all paths are updated at once.
- Enables work on work pieces moved by external axes or conveyor tracks, since the entire work object with its paths can be moved.

For information on how to define the work object coordinate system, see information about the data type `wobjdata` in *Technical reference manual - RAPID Instructions, Functions and Data types*.

Continues on next page

The displacement coordinate system



en0400001227

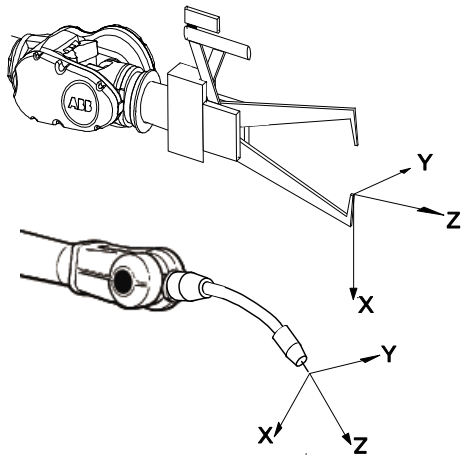
A	Original position
B	Object coordinate system
C	New position
D	Displacement coordinate system

Sometimes, the same path is to be performed at several places on the same object, or on several work pieces located next to each other. To avoid having to reprogram all positions each time a displacement coordinate system can be defined.

This coordinate system can also be used in conjunction with searches, to compensate for differences in the positions of the individual parts.

The displacement coordinate system is defined based on the work object coordinate system.

The tool coordinate system



en0300000497

The tool coordinate system has its zero position at the center point of the tool. It thereby defines the position and orientation of the tool. The tool coordinate system

Continues on next page

is often abbreviated TCPF (Tool Center Point Frame) and the center of the tool coordinate system is abbreviated TCP (Tool Center Point).

It is the TCP the robot moves to the programmed positions, when executing programs. This means that if you change the tool (and the tool coordinate system) the robot's movements will be changed so that the new TCP will reach the target.

All robots have a predefined tool coordinate system, called `tool0`, located at the wrist of the robot. One or many new tool coordinate systems can then defined as offsets from `tool0`.

When jogging a robot the tool coordinate system is useful when you don't want to change the orientation of the tool during the movement, for instance moving a saw blade without bending it.

For information on how to define the tool coordinate system, see information about the data type `tooldata` in *Technical reference manual - RAPID Instructions, Functions and Data types*.

9.5 What is mirroring?

Description

Mirroring creates a copy of a program, module, or routine in a specific mirror plane. The mirror function can be applied to any program, module, or routine.

Mirroring can be performed in two different ways:

- Default against the base frame coordinate system. The mirror operation will be performed across the xz-plane in the base frame coordinate system. All positions and work object frames that are used in an instruction in the selected program, module or routine are mirrored. The position orientation axes x and z will be mirrored.
- Advanced against a specific mirror frame. The mirror operation will be performed across the xy-plane in a specified work object frame, mirror frame. All positions in the selected program, module or routine are mirrored. If the work object argument in an instruction is another work object than specified in the mirror dialog, the work object in the instruction is used in the mirror operation. It is also possible to specify which axis in the position orientation that will be mirrored, x and z or y and z.

Note

The mirroring function recognizes the used workobject in all predefined motion instructions and in user made procedures with the same argument declaration:

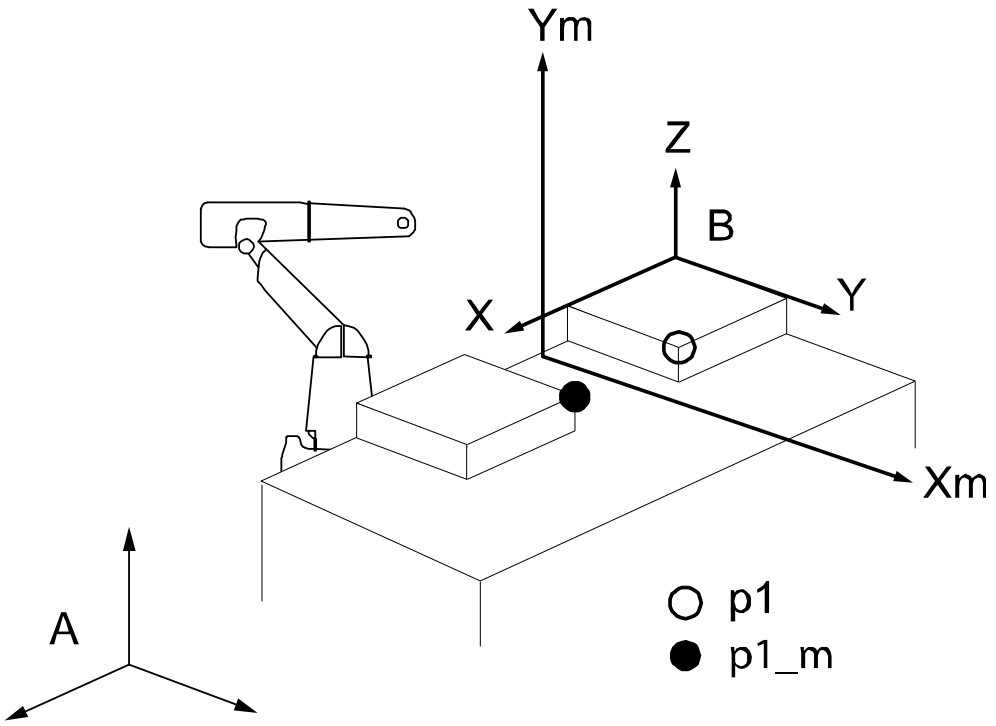
- an argument for the robtarget,
- an argument for the tooldata with name **'Tool'** and
- an optional argument for the wobjdata with the name **'Wobj'**.

The following descriptions of mirroring describes advanced mirroring.

Continues on next page

Mirror plane

The mirror function will mirror all positions (robtargets) in the mirror plane, i.e. the mirrored position will be located symmetrically on the other side of the plane, relative to the original position. The mirror plane is always the xy-plane of an object frame, used for mirroring. This object frame is defined by a work object data, e.g. with the name MIRROR_FRAME.



xx0600002815

Ym, Xm	Mirror plane
A	World frame
B	Work object frame
p1	Original point
p1_m	Mirrored point

Mirroring routines

Mirroring creates a copy of a routine with all positions (robtargets) mirrored in a specific mirror plane. In general, all data of the type robtarget used in the routine, both local and global, will be mirrored. It makes no difference whether the robtarget data is declared as a constant (which it should be), as a persistent, or as an ordinary variable. Any other data, e.g. of type pos, pose, orient, etc., will not be mirrored.

Mirroring data only affects the initialization value, i.e. any current value will be ignored. This means that if a robtarget variable has been defined without an init value, this variable will **not** be mirrored.

The new, mirrored routine will be given a new name (a default name is proposed). All stored data of type robtarget, used in the routine, will be mirrored and stored

Continues on next page

with a new name (the old name ending with “_m”). All immediate robtarget data, shown with an “*”, in movement instructions will also be mirrored.

Mirrored values and arguments

When mirroring a routine, the new routine is scanned for any local robtarget data, declared inside the routine with an init value. All init values of such data are mirrored. Then the new routine is scanned for statements with one or more arguments of type robtarget.

When such a statement is found, the following actions will take place:

- If the argument is programmed with a reference to a local variable or a constant, this argument will be ignored, since it has already been mirrored as described above.
- If the argument is programmed with an immediate robtarget data, shown with an asterisk“ *”, then this value will be mirrored directly.
- If the argument is programmed with a reference to a global variable, persistent or a constant, defined outside the routine with an init value, then a duplicate is created and stored in the module with a new name (the old name ending with “_m”). The init value of this new data is mirrored, and then the argument in the statement is changed to the new name. This means that the module data list will expand with a number of new mirrored robtarget data.

Error handlers or backward handlers in the routine are not mirrored.

Work object frame

All positions which are to be mirrored are related to a specific work object frame (B in figure above). This means that the coordinates of the robtarget data are expressed relative to this work object frame. Furthermore, the mirrored position will be related to the same work object frame.

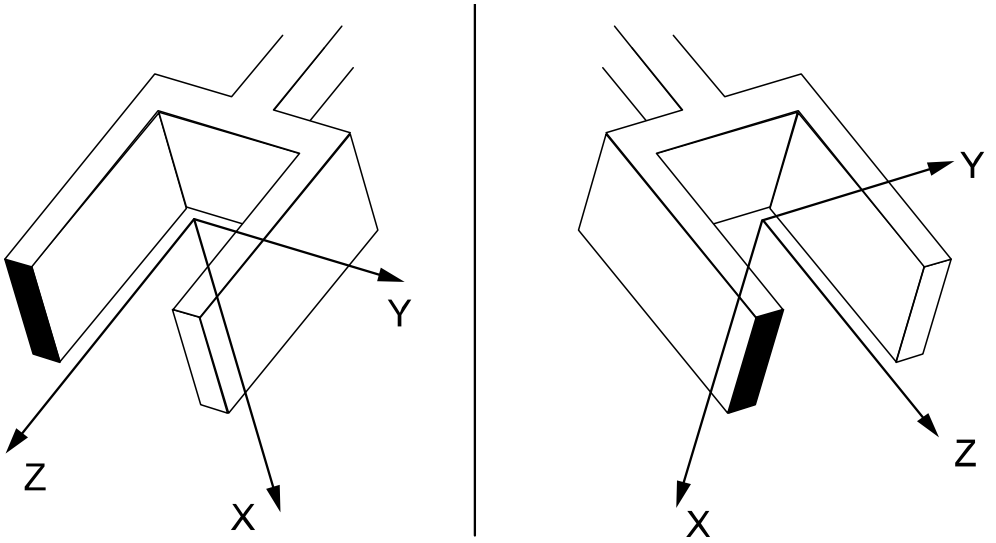
Before mirroring, this specific work object must be stated. This work object will be used as the reference frame for all variables that are to be mirrored.

Make sure to state the same work object as was originally used when defining the robtarget data, and which was used as a parameter in the movement instructions. If no work object was used, the wobj0 should be stated.

Continues on next page

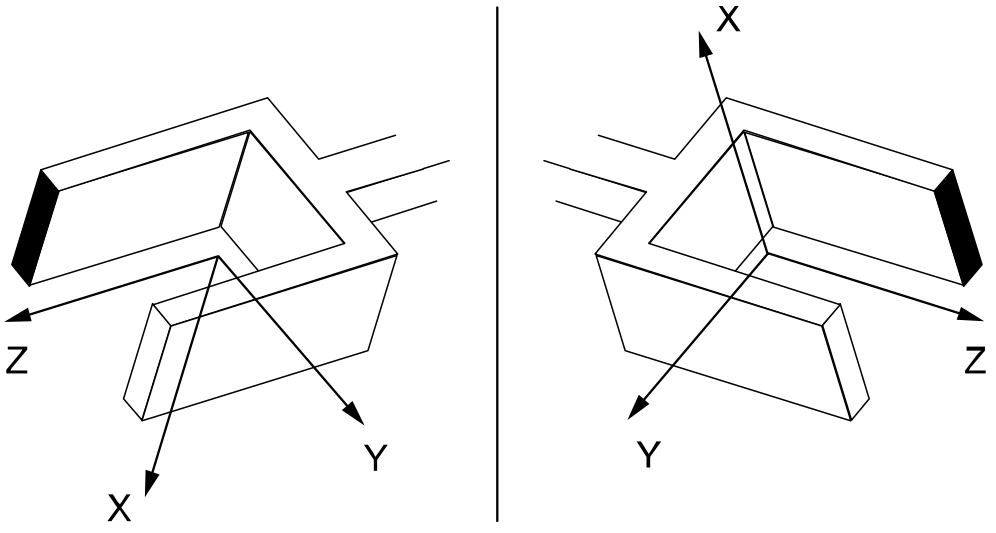
Orientation of mirrored positions

The orientation of the robtarget position is also mirrored. This mirroring of the orientation can be done in two different ways, where either the x and z axes are mirrored or the y and z axes. The method used, x or y axis (the z axis is always mirrored), is dependent on the tool used and how the tool coordinate system is defined.



xx0600002816

	Mirroring of x and z axes.



xx0600002817

	Mirroring of y and z axes.

Arm configurations

The arm configuration will not be mirrored, which means that after mirroring, it has to be carefully checked by executing the path in manual mode. If the arm configuration has to be changed, this must be done manually and the position corrected with a modpos command.

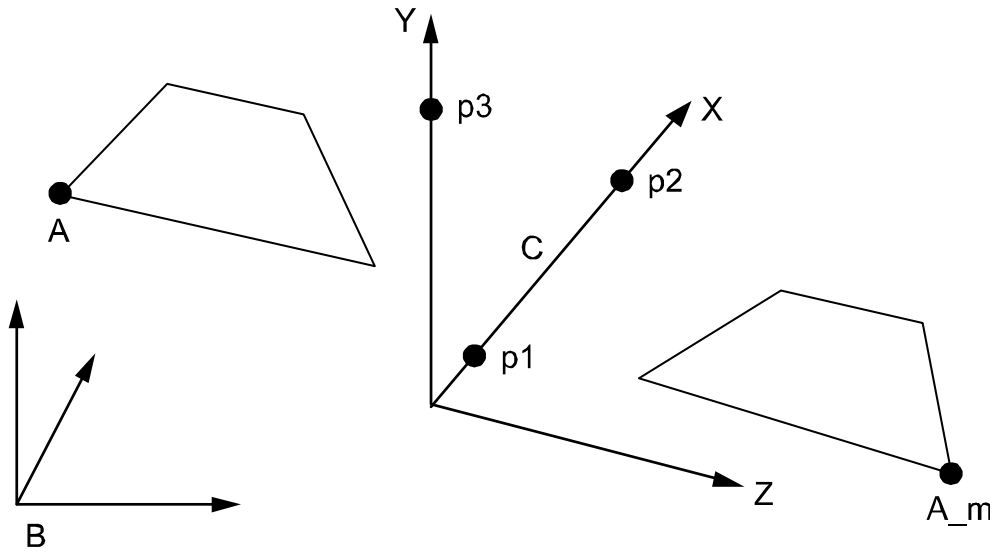
Continues on next page

9.5 What is mirroring?
Continued

Example 1: Mirroring with one robot

A mirrored copy of the routine `org` is to be created and stored with the name `mir`. All positions are related to the work object, wobj3. The mirror plane is known from three positions in the plane, p1, p2, and p3.

An original position in `org`, A, is mirrored to A_m.



xx0600002818

A	Original position
A_m	Mirrored position
B	Object frame wobj3
C	Mirror plane

To perform this mirroring, the mirror frame must first be defined. To do this, create a new work object and name it (e.g. `mirror`). Then, use the three points, p1 to p3, to define the object coordinate system by using the robot. This procedure is described in *Defining the work object coordinate system on page 123*.

After this, the routine, `org`, can be mirrored using wobj3 and mirror as input data.

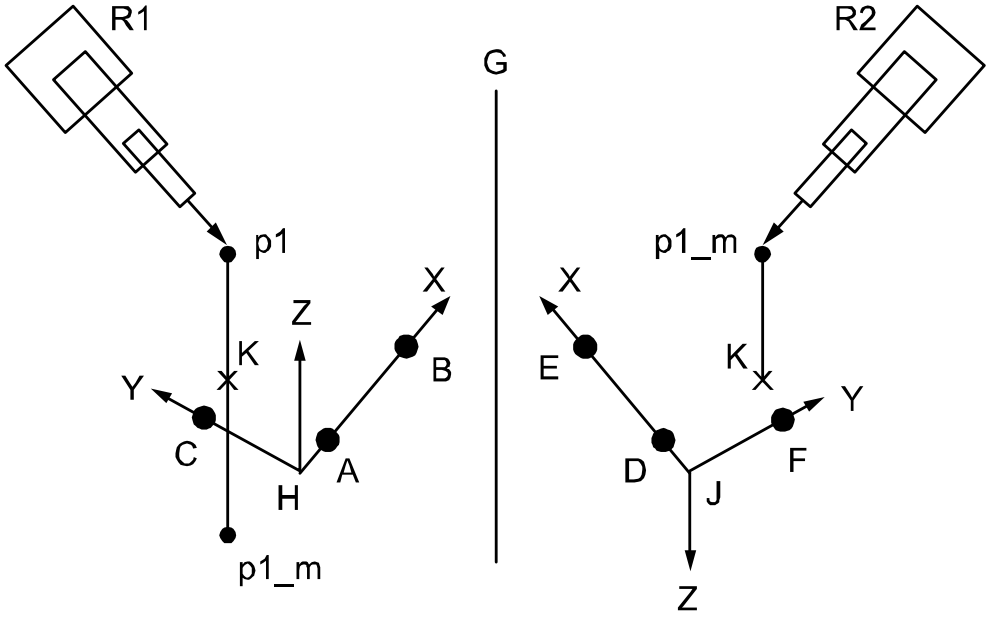
Example 2: Mirroring with two robots

The routine `org` was created on one robot and should be mirrored and used on another robot. Suppose that a spot welding robot, robot 1, is used for the left side of a car body. When the program for the left side is done, it should be mirrored and used again for the right side by robot 2.

The original program, `org`, is programmed relative to a work object, wobj1, which is defined with the help of three points, A, B and C on the left side of the car body. The mirrored program, `mir`, is to be related to a corresponding work object, wobj1, defined by the corresponding points D, E and F on the right side of the car body. Wobj1 for robot 2 is defined with robot 2.

Continues on next page

Note that since the points D, E, F are mirrored images of points A, B, and C, the wobj1 for robot 2 will also be mirrored. One of the consequences of this is that the z-axis will point downwards.



xx0600002819

R1	Robot 1
R2	Robot 2
G	Virtual mirror plane
H	wobj1 = mirror frame
J	wobj1 for robot 2
K	Projection of p1 in xy-plane
p1	Original position
p1_m	Mirrored position

After the work object, wobj1, has been defined, all programming is done in this frame. Then the program is mirrored using the same wobj1 frame as the mirroring frame. A position, p1, will be mirrored to the new position p1_m.

After this, the mirrored program is moved to robot 2, using the work object wobj1, as described above. This means that the mirrored position, p1_m, will be “turned up” as if it were mirrored in a “virtual” mirror plane between the two robots.

This page is intentionally left blank

10 OmniCore cyber security

10.1 Introduction

Overview

This chapter gives an overview of the security aspects of a network installation with ABB OmniCore systems, and of the communication with other products typically deployed in a network installation. It also addresses topics such as how to identify the most critical assets and security threats targeting them, plus how to reduce these security threats.

Cyber security features

The OmniCore cyber security comprises several combined security features for the protection against threats such as viruses, malware, and other exploits. Some of the included features are:

- **Certificate handling**

  Certificates are used to provide secure communication over the network. See section *Certificate handling on page 271* for more information.
- **Secure boot**

  A secure boot chain has been implemented in order to ensure that only trusted ABB software is used, thus minimizing the security risks. When the PC starts it validates the digital signatures of the software before executing it.
- **Secure storage**

  The secure storage functionality is used to store sensitive data (i.e. private keys, credentials etc.), thereby providing confidentiality and integrity for the user.
- **Firewall management**

  The objective of the firewall management is to protect the OmniCore controller from threats originating in the office network. By defining firewall settings, you can monitor the selection of enabled Network Services. See section *Firewall on page 263* for more information about firewall requirements.
- **User authentication system**

  The User Authentication System (UAS) provides authentication and authorization functionality to the OmniCore controllers. See section *User Authentication System on page 260* for more information.

Disclaimer

The intent of this chapter is to raise awareness about security threats and to provide guidance to address them as well as to inform how ABB is working on security assurance. However, due to the high number of different security risks and complex dependencies within actual installations, this document can neither cover all possible security risks, nor guarantee the success of the presented security mechanisms.

Continues on next page

The benefits and risks of using open networking technology for robot controllers

ABB OmniCore products use standard Internet transport protocols, TCP and UDP. This way, the products can be connected to a TCP/IP/Ethernet network, thus reducing costs and unifying network management. Furthermore, the interconnection of control systems and office systems enable a wide range of new applications, which take advantage of such vertical integration from the shop floor up to the enterprise systems. Section *Network architecture and communication on page 253* describes a typical OmniCore robot network.

However, the direct connection of control systems to the plant network also creates security risks (for example, malware infections (viruses, worms, Trojans), denial of service, disclosure of confidential data). Section *Security analysis on page 259* discusses these security threats in detail.

Mitigating the risks through a comprehensive security policy and architecture

It is generally accepted that the security features of a product or system are only one part in a successful protection strategy. It is equally important to define, implement, and maintain an effective security policy, which covers risk analysis, procedures, responsibilities, and regular auditing. Section *Security policy on page 261* discusses requirements for a security policy and shows, how such a security policy can be used to mitigate security threats targeting a robot control system. It is important to note though that security cannot be achieved by a one-time investment in a product or process but requires continuous effort to operate and maintain.

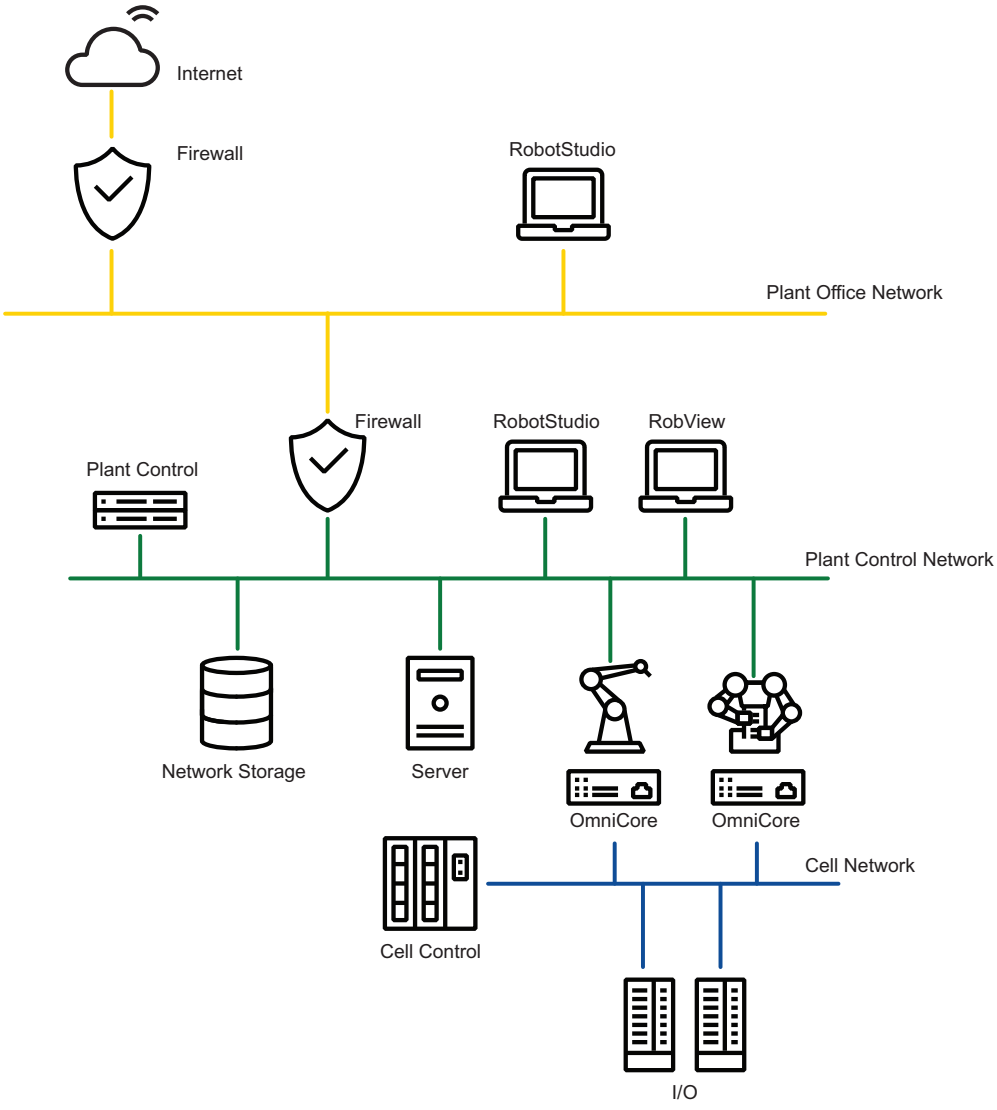
10.2 Network architecture and communication

About this section

This section gives an overview about typical components of an OmniCore installation, which are attached to the network, and the communication between them. Furthermore, this section serves as basis for the threat analysis (*Security analysis on page 259*) and the requirements to a security policy (*Security policy on page 261*).

Simplified example of network with OmniCore products

The following figure shows a simplified network with OmniCore products. It contains examples of typical components which may be part of a robot control system installation, and shows where they may be installed.



xx1800003061

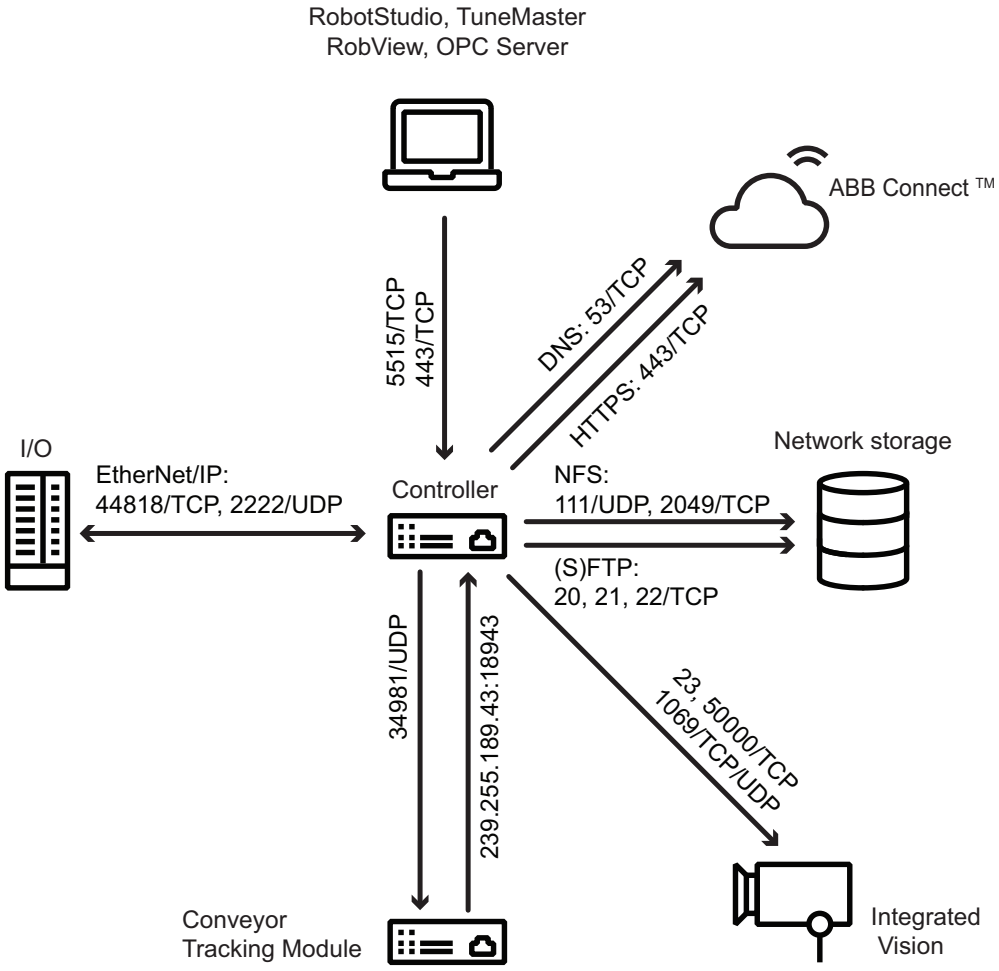
The objective of the network security architecture is to protect the plant control network from threats originating in the plant office network. Especially remote

Continues on next page

networks might be exposed to viruses and other threats. Therefore, it is strongly suggested to separate the plant control network and the plant office network with a protection device, such as a firewall.

Communication protocols between OmniCore related products

The following figure shows all components, which are involved in communication around OmniCore products, and the communication between them.



xx2300000054

The components are clustered according to where they reside: remote/plant office network or plant control network. The lines between the components indicate the used communication protocol.

The secure communication protocol Robot Web Services is used for communication between the controller and RobotStudio and PC SDK Application. For more information about Robot Web Services, see developercenter.robotstudio.com/webservice.

Continues on next page

Communication setup with Connected Services

The Connected Services Gateway is used for Connected Services connectivity. It may be connected over a secure Cellular, Wi-fi or Wired connection to ABB Connect Cloud. It can be connected to the Plant Network as it is firewalled to prevent unwanted and unneeded inbound access from internet.

Note

For OmniCore E10, only the wired connection is available.

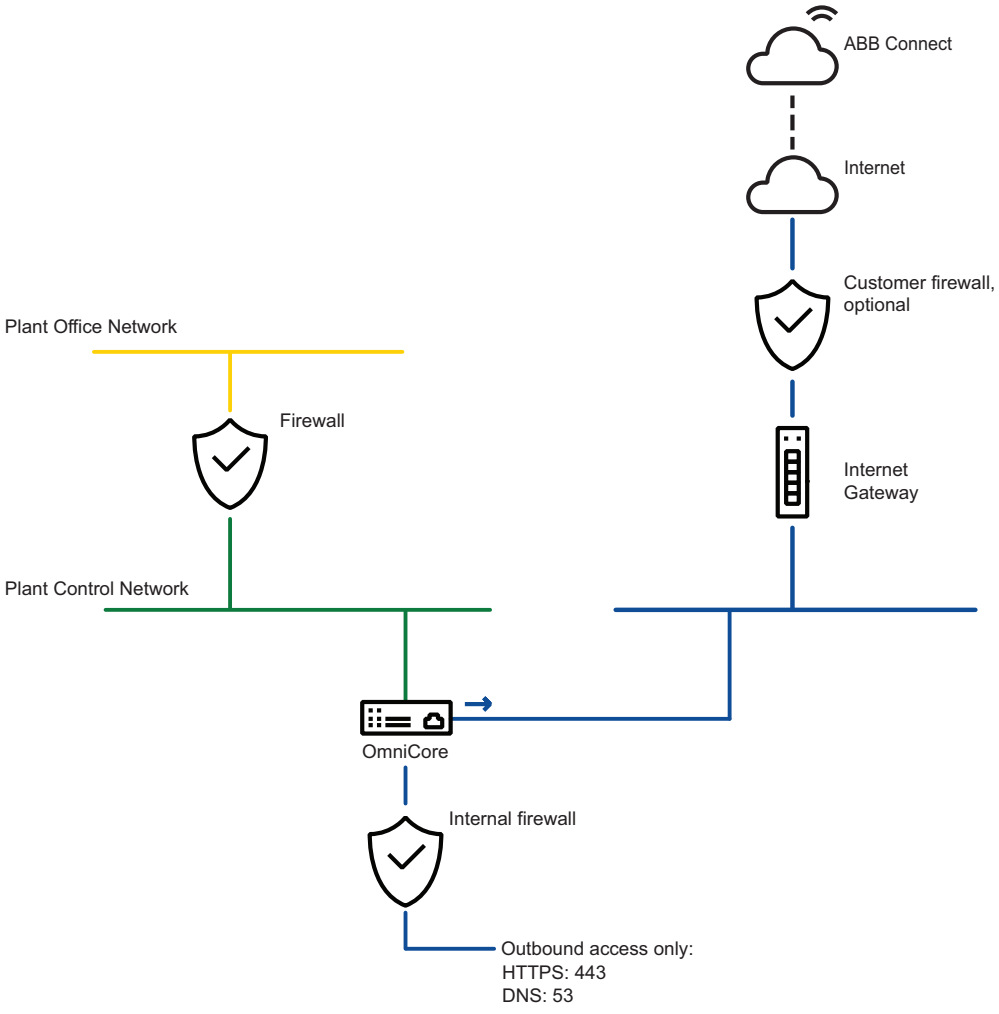
Connected Services Embedded is used for the embedded software Connected Services. It may be connected over a Wired connection or through the Connected Services Gateway to ABB Connect Cloud. It may be connected to the Plant Network if it is firewalled to prevent untrusted and unneeded inbound access from internet.

Continues on next page

Connected Services Embedded requires only HTTPS:443 and DNS:53 outbound accesses on the firewall.

Connected Services setup, wired connection

The following image illustrates the Connected Services setup with a wired connection.



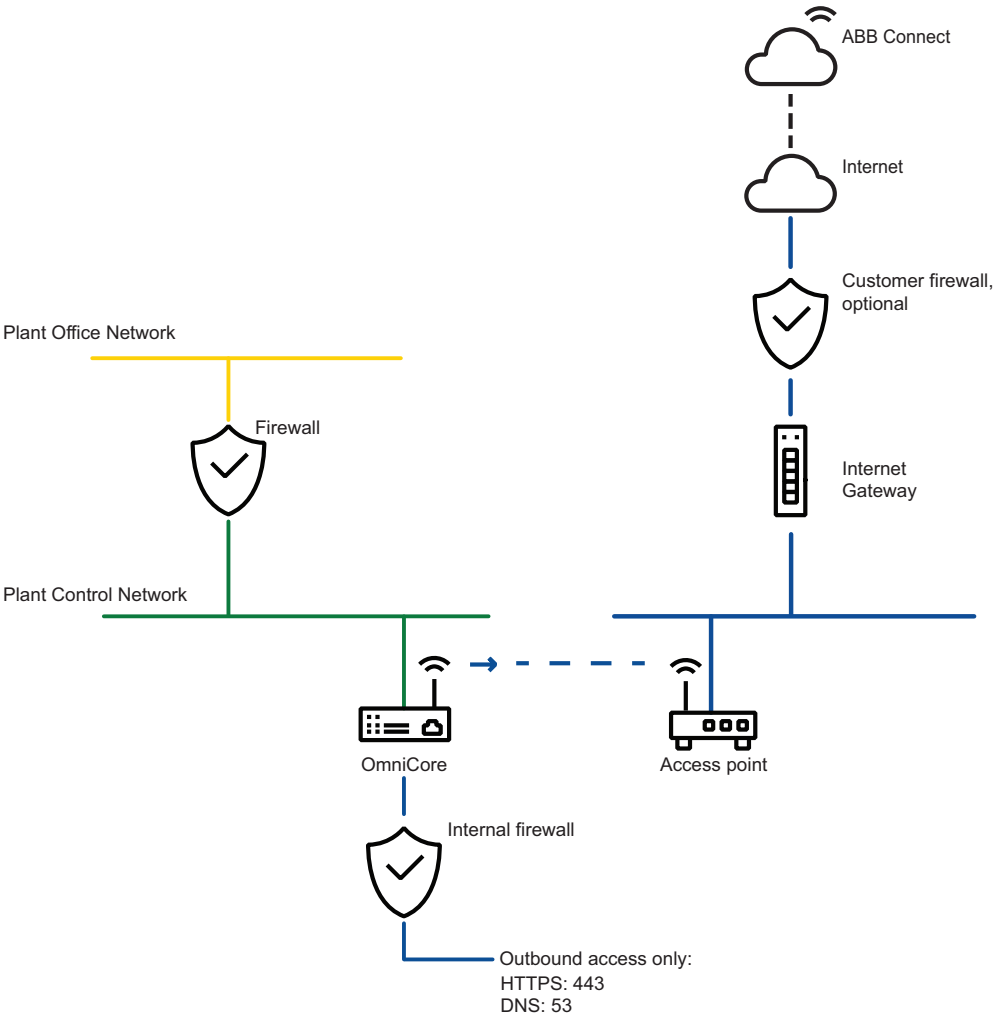
xx1900000618

The wired connectivity is done through the Connected Services Gateway Wired using outbound HTTPS protocol secured by a client certificate to identify the OmniCore controller on ABB Connect Cloud.

Continues on next page

Connected Services setup, Wi-Fi connection

The following image illustrates the Connected Services setup with a Wi-Fi connection.



xx1900000619

The Wi-Fi connectivity is done through the Connected Services Gateway Wi-Fi using outbound HTTPS protocol secured by a client certificate to identify the OmniCore controller on ABB Connect Cloud.

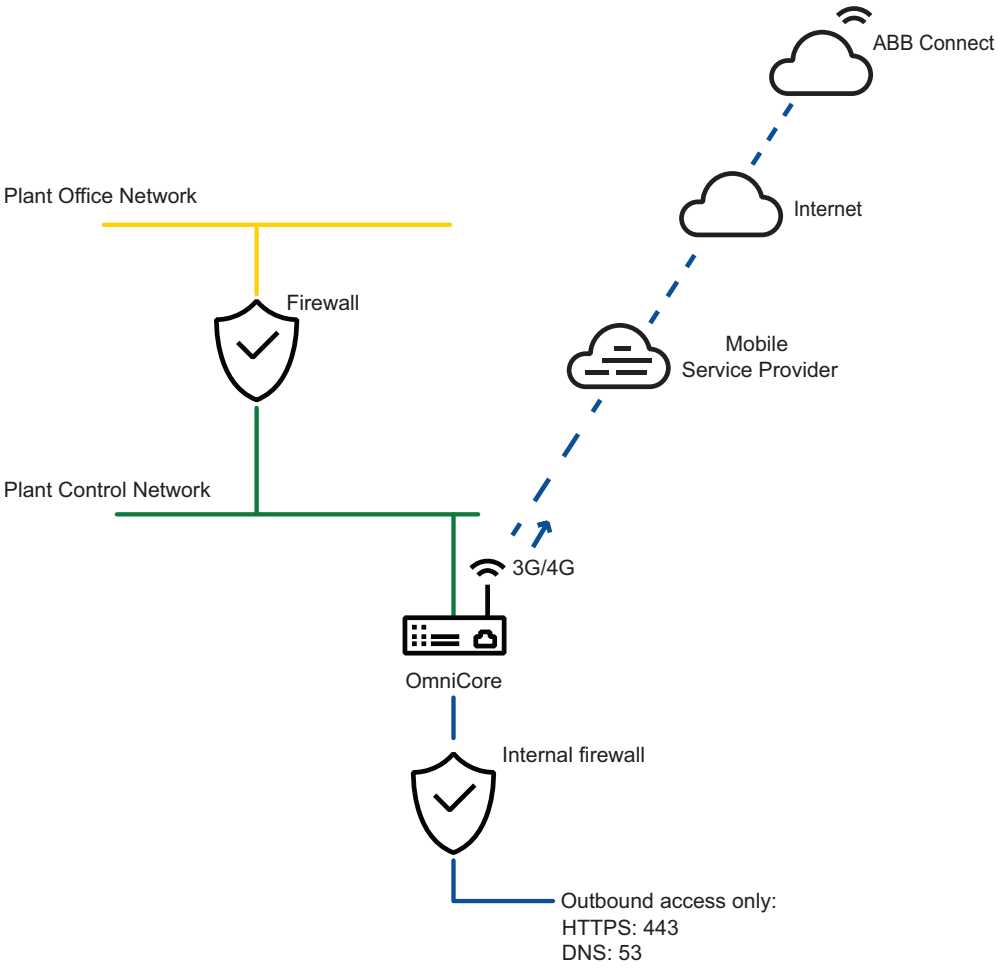
Note

Wi-Fi is not supported natively by the E10 controller.

Continues on next page

Connected Services setup, mobile network connection

The following image illustrates the Connected Services setup with a mobile network (3G/4G) connection.



xx1900000620

The mobile network connectivity is done through the Connected Services Gateway using outbound HTTPS protocol secured by a client certificate to identify the OmniCore controller on ABB Connect Cloud.

Note

3G/4G is not supported natively by the E10 controller.

10.3 Security analysis

Why do you need a security analysis?

Network architectures, in which the plant control network is connected to an office network, which in turn is connected to the Internet, potentially expose the plant control network to cyber attacks. Therefore, it is important to identify the security threats and to implement security mechanisms to prevent them.

Critical assets to protect

The main asset to be protected is the main computer of the OmniCore robot controller. Any reduced availability or unauthorized access may cause significant financial loss due to damaged semi-finished goods or loss of production. Unauthorized changes to configuration files on the devices can have a direct impact on the correct functioning and availability of the controlled robots and processes.

Further critical assets are the RobotStudio PCs, since they communicate directly with the OmniCore robot controllers and, if compromised, may be used as entry points to the devices.

Security threats

The following table summarizes security threats specific to the ABB Robotics OmniCore products, the attacks causing them, and the targeted assets.

Threat	Attacks causing the threat	Targeted assets
Disturbance / DoS	Sending of large amount of data Upload of invalid RAPID programs	OmniCore robot controller
Unauthorized access Unauthorized control	Brute force password attack Network Sniffing	OmniCore robot controller RobotStudio PC

Apart from the threats listed above, there are also a number of other ways the security might be threaten. The section *Security policy on page 261* covers both generic and ABB Robotics specific requirements to prevent/mitigate such security risks.

10.4 User Authentication System

About the User Authentication System

The User Authentication System (UAS), which provides authentication and authorization functionality to the OmniCore controllers, is a security feature that is implemented in every OmniCore controller.

UAS limits which individuals (with specific roles) can perform which operations on the controller by defining the users and roles that can access the controller and the functionality. Depending on which role a user is assigned to, the user is granted access to certain functionality, while other functionality will be inaccessible. Which functionality the members of a certain role have access to, is controlled by assigning a list of grants to that role.

Two types of grants exist: controller grants and application grants. Controller grants are predefined by ABB, are validated by the robot controller, and apply to all tools and devices, which access the controller. Application grants may be added by application developers and are used and valid only within a specific application (for example, FlexPendant).

Default User

The ABB OmniCore system is delivered with a default configured user named "Default User". This user has a number of grants and belongs to the role "Operator" by default.

If a new user is created with specific grants, the "Default User" can be removed.

If the "Default User" is active, but all grants are removed, there are still reading rights. Therefore, if unauthorized personnel should be prevented from viewing any content on the OmniCore controller, the "Default User" must be deleted.

Admin

The ABB OmniCore system is delivered with a default configured user named "Admin". This user user has full grants to manage the system, such as adding, removing and modifying users. The "Admin" user belongs to the role "Administrator" by default.

The "Admin" user can be removed, but the system must always contain at least one user with the grant *Manage UAS settings*.

Connected Services

The ABB OmniCore system is delivered with a predefined user dedicated to service data collection for Connected Services features. This user has the grants to collect service information and is available only internally. This service data collection can be disabled by disabling Connected Services in Configuration.

How to use the role based access control

For information about how to configure and use the role based access control, see *Operating manual - RobotStudio*.

10.5 Security policy

10.5.1 Introduction

Overview

Vulnerabilities in the industrial control systems can be found and exploited if the security policy is not well-defined, accurate, and enforced. Therefore, the security policy plays an essential role in the reduction of exploits of vulnerabilities and the defense against and mitigation of security threats.

The security policy must be defined according to requirements such as: how to identify users (authentication), who is allowed to access what (authorization), and what should be audited regularly (audit). Once the security policy is defined, it has to be implemented and applied to all covered software, hardware, systems, data, networks, and personnel within the control system owner's organization resulting in a security architecture consisting of technical and procedural means. The security policy and its implementation have to be maintained continuously, since organizational changes, upcoming and evolved regulations, and new technologies have all an impact on the security policy. Therefore, security is not a one-time initiative, but an on-going process.

This section describes the security requirements that should be addressed by the owner of the security policy of the control system. The proposed requirements are grouped into two categories: the first one is generic and the second one is specific to ABB Robotics products. Note that the listed requirements are not exhaustive and that they should be tailored to the specific requirements, the size, and available resources of the control system owner's organization.

10.5.2 General security requirements

About general security requirements

This section lists a number of things to consider in any computer network.

Physical security

All operations shall be tracked and servers, backup media, and other associated equipment shall be placed in locked machine rooms or cabinets. Access to these areas shall be restricted to dedicated employees. All access levels and responsibilities shall be explicitly documented.

Further physical security requirements:

- Physical data interfaces, such as USB ports, are locked or disabled.
- Network components, like switches and routers, shall be enclosed in locked cabinets.
- Tamper detection of unauthorized access (for example, inspecting the sealings).
- Only authorized personnel gets access to network components and cables.
- Unless necessary, wireless devices should not be connected to the plant control network. A rationale shall be documented for each exception.

Account management and network/system access

Account management and access control requirements define how user accounts and passwords are managed and how user rights and access to the network and the control system are controlled.

Procedures for account management (e.g., adding of a new account, removal of an existing account, and changing or assigning of a new password) shall be documented. Employees, who are assigned to this function, shall be trained in those procedures.

Further requirements:

- There must be a process for removal of inactive accounts, i.e. at termination of employment etc.
- Only authorized personnel get access to PCs and its interfaces, especially to PCs in the plant control network.
- Use the principle of "least privilege", i.e. use roles/grants with only the necessary rights to perform the task.
- Replace default accounts with personal accounts.
- Only controlled and protected computers are allowed to be connected to the plant control network.
- The definitions of authorized users, user roles, and access rights are continuously maintained to reflect properly the current authorities.
- Users lock the screen or log off before leaving the workplace.
- A password-protected screen-saver is activated after a certain idle time.
- After a specified number of consecutive log-on failures, a user account is locked for a certain time or until it is unlocked by a system administrator.

Continues on next page

Passwords

A password policy should be in place. This may include the following rules:

- Passwords shall expire, forcing users to change passwords regularly.
- All password authentications within the network shall be encrypted.
- Only "strong" passwords are used.
- User and password lists are protected from unauthorized access.
- Passwords are not written down.
- Passwords are not shared between users.

Administration and patching of servers and workstations

All systems shall be kept updated according to the vendor's recommendations (e.g., new patches addressing critical vulnerabilities shall be applied as soon as they have been successfully tested and verified).

The hardening of operating systems and applications shall be regularly reviewed (e.g., unused ports shall be closed, unused services uninstalled, unused application features disabled, and demo or default application data moved or deleted).

Only authorized personnel are allowed to change system configuration and to install new software.

Virus protection

Anti-virus software shall be installed on every workstation and server, for which it is available. Anti-virus software and virus definitions shall be updated regularly.

All software to be installed on systems in the plant control network, shall be first checked for malware (viruses, worms) on a separate virus scanning PC, which is not connected to the plant control network.

E-mail

Internet and e-mail services may serve as carriers for viruses, worms, spyware, and other kind of malware to penetrate the plant control network. Therefore, systems in the plant control network shall not be allowed to access arbitrary Internet sites and e-mail services.

Firewall

Firewalls that protect plant control networks shall provide stateful filtering and preferably offer application level support for the forwarded protocols (e.g., deep packet inspection). To protect plant control networks from flooding and denial-of-service attacks, the firewalls shall offer rate-limiting functionality.

If a firewall is used as VPN endpoint, it shall support state-of-the-art VPN protocols.

The firewalls shall be configured to allow only authorized traffic from dedicated source addresses and source ports to dedicated destination addresses and destination ports. Furthermore, the firewalls shall be locked down and need to be regularly maintained (i.e., patched, upgraded, and proper change management including regular audits for access rules).

Continues on next page

10.5.2 General security requirements
Continued

Backup and recovery

All critical data shall be backed-up periodically and stored in a secure place. For ABB Robotics OmniCore products, backups should especially cover the data stored on the controller including RAPID and configuration files.

A good backup policy should also include the configurations and parameters from Conveyor Tracking Module as well as the host Window PC configuration.

Note

An OmniCore backup contains unencrypted information. Make sure that all backup files are stored in a secure location.

Note

No UAS information is included in the backup.

Recovery, or restore, can be a solution when the program file seems to have become corrupt. In that case, it can be restored to previous settings. All system parameters will be replaced and all the modules from the backup directory will be loaded.

Vulnerability scanning and risk assessment

In many IT environments, the use of vulnerability scanners, such as Nessus, NMAP, Metasploit, etc, are used to find and assess the potential vulnerabilities in IT equipment. These tools perform extensive probing and conduct a representative set of attacks on equipment. Because of the potential for disruption to the OmniCore controller, the recommended best-practice in the industry is not to perform these types of tests on production equipment but only on equipment in a controlled laboratory environment. Performing these types of scans and tests on the OmniCore controller while in production has the potential of disrupting the normal operation of the OmniCore and its communications to other devices on the network.

Cyber security procedures and policies

- Background checks, instructions and training of personnel and subcontractors.
- Guide on what is allowed to do, using which tools, by whom, and when.
- Logging and cyber security monitoring methods in automation systems and networks.

Maintenance and audits

Perform maintenance of the ongoing responsibilities and actions to ensure that the security policy is followed, kept up-to-date, and adapted to organizational changes.

Periodic testing and reviews of the security policy are required.

Continues on next page

Disposal

Before disposal of any storage equipment, make sure all sensitive user data and system data has been deleted.

10.5.3 ABB Robotics product specific requirements

Remote access/client

Remote access allows users to access company networks and systems from computers that are located outside of the protected company. In the context of an ABB robot control network, these are hosts running RobotStudio. Since the computers running RobotStudio may directly access systems in the plant control network, they extend the perimeter of the plant control network and may therefore create security risks.

To mitigate these risks, the following actions are suggested (in descending order of preference):

1. Avoid remote clients
2. Use remote terminal services, which are protected within a secure tunnel
3. Tunnel communication protocol(s) through a VPN and authenticate communication partners

Since remote clients represent security risks, they should be avoided whenever possible. Any remote connection shall be justified by business reasons.

Any remote host, which has been identified to require connection to the protected network, has to be hardened according to security host hardening best practices, which include, but are not limited to:

- Dedicated machine for remote access (i.e., not the same as for daily business)
- No other simultaneous network connections
- Only required services and processes are installed and running
- Only required network ports are active
- Restrictive access control
- Up-to-date patches / services packs / upgrades are applied
- Up-to-date anti malware software, such as a virus scanner, is running
- Regular maintenance intervals

Clients in Plant Office Network

RobotStudio and other clients may also exist on hosts in the plant office network. Therefore, the same suggestions as for the remote clients (see *Remote access/client on page 266*) also apply to the clients in the plant office network.

Although, the plant office network is usually already protected by firewalls against the Internet and other networks, it still represents a security threat to the plant control network, since applications with high security risks, such as e-mail and Web browsers, are run within the plant office network. Therefore, a separation of the plant control network from all other networks, including the plant office network, using firewalls and preferably, also a DMZ, is strongly suggested.

The use of terminal services for access from clients in the plant office network to systems in the plant control network is, with respect to security, still preferred against the pure tunneling of communication protocols. However, since the risk caused by systems in the plant office network is lower than that caused by remote systems, tunneling may represents an acceptable alternative. It is still strongly

Continues on next page

suggested to deny any unprotected communication between the plant office network and the plant control network.

Robot Web Services

The Robot Web Services lets you access comprehensive and powerful programming interfaces to interact with the robot controller allowing you to develop your own apps and user screens. Robot Web Services exposes RESTful APIs that leverages the HTTPS protocol and the messages are composed of XHTML and JSON. Robot Web Services facilitates platform independent and language independent communication with the robot controller allowing you to create apps that run on any device including the FlexPendant, smart phones, PC etc.

UAS administration

UAS provides access control to the controller (as described in section *User Authentication System on page 260*). There are two requirements concerning UAS administration:

- In the factory configuration, UAS has a built-in "Default User" account that is assigned to the Operator role, which does not hold administrative grants.
- In the factory configuration, UAS has a built-in "Admin" account that is assigned to the Administrator role.
- There must always be at least one administrator in the system. If the admin user is the last user with administrative rights, the admin user cannot be deleted.

10.6 OmniCore application protocols

Overview

The OmniCore services and application protocols are presented in separate sections for default and configured/enabled protocols. The tables also define the network segments that may be used for each service/application. For detailed information about all network segments, see *Network segment overview on page 35*.

For more information about port number assignation, see *www.iana.org*.

Default services and application protocols

Service, or Application	Port number	Transport protocol	Network segment	Usage/Comments
Bonjour	5353	UDP	Public Network Private Network I/O Network	Multicast DNS (MDNS) for Bonjour Zero conf. Discovery service.
DHCP server	68	UDP	Private Network	Dynamic Host Configuration
DHCP client	67	UDP	Public Network	Dynamic Host Configuration.
NetScan	5512 5513 5514	UDP	Public Network Private Network I/O Network	Detection of available OmniCore robot controllers on the network.
Robot Network Protocol (RNP/RobAPI) over TLS	5515	TCP	Public Network Private Network I/O Network	Communication with OmniCore robot controller.
Robot Web Services (HTTPS)	80	TCP	Public Network Private Network I/O Network	Communication with OmniCore robot controller.
RobotWare Installation Utilities (HTTP)	80	TCP	Private Network	Used for troubleshooting and installation.

Configured/enabled services and application protocols

Service, or Application	Port number	Transport protocol	Network segment	Usage/Comments
CC-Link IE Field Basic Cyclic data	61450	UDP	Public Network Private Network I/O Network	Enabled by configuration. Requires option *3066-2 CC-Link IE Field Basic Device*.
CC-Link IE Field Basic SLMP	61451	UDP	Public Network Private Network I/O Network	Enabled by configuration. Requires option *3066-2 CC-Link IE Field Basic Device*.
DCP (Discovery and Configuration Protocol)		UDP	Public Network I/O Network	Used in combination with PROFINET.
DNS client	53	TCP	ABB Connect Network	ABB Connect™ server name resolution. Enabled by configuration.

Continues on next page

Service, or Application	Port number	Transport protocol	Network segment	Usage/Comments
EGM (Google Protocol Buffers)	Ports used are defined in the configuration for the UdpUc device.	UDP	Public Network Private Network I/O Network	Externally Guided Motion. Enabled by configuration.
EtherNet/IP messaging	44818	TCP UDP	Public Network Private Network I/O Network	Enabled by configuration. Requires option *3024-1 EtherNet/IP Scanner* or *3024-2 EtherNet/IP Adapter*.
EtherNet/IP I/O	2222	TCP UDP	Public Network Private Network I/O Network	Enabled by configuration. Requires option *3024-1 EtherNet/IP Scanner* or *3024-2 EtherNet/IP Adapter*.
FTP client	20 21	TCP	Public Network Private Network I/O Network	Remote disk mounting with FTP. Enabled by configuration.
HTTPS	443	TCP	ABB Connect Network	Secure connection to ABB Connect™ Cloud. Enabled by configuration.
IEEE1588/ PTP v1 or v2	319 320	UDP	Public Network Private Network I/O Network	Time synchronization used by the RobICI protocol. Requires the option *Conveyor Tracking Interface*.
Integrated Vision Telnet client	23 50000 1069	TCP, UDP	Private Network I/O Network	Communication between controller and Cognex cameras. Enabled by configuration.
Link Layer Discovery Protocol (LLDP)			Public Network I/O Network	Used in combination with PROFINET. Requires option 3020-1 *PROFINET Controller* or 3020-2 *PROFINET Device*.
NFS client	111 2049	TCP, UDP	Public Network Private Network I/O Network	Remote disk mounting. Enabled by configuration.
PROFINET RT	34962	UDP	Public Network I/O Network	Enabled by configuration. Requires option 3020-1 *PROFINET Controller* or 3020-2 *PROFINET Device*.
PROFINET RTM	34963	UDP	Public Network I/O Network	Enabled by configuration. Requires option 3020-1 *PROFINET Controller* or 3020-2 *PROFINET Device*.
PROFINET CM	34964	UDP	Public Network I/O Network	Enabled by configuration. Requires option 3020-1 *PROFINET Controller* or 3020-2 *PROFINET Device*.
PROFINET RPC	49152	UDP	Public Network I/O Network	Enabled by configuration. Requires option 3020-1 *PROFINET Controller* or 3020-2 *PROFINET Device*.

Continues on next page

Service, or Application	Port number	Transport protocol	Network segment	Usage/Comments
RobICI	239.255.189.43:18943 34981	TCP, UDP	Public Network Private Network I/O Network	I/O signals, command, and response. Required for conveyor tracking module (CTM). Enabled by configuration. Requires option *1550-1 Conv.Tracking unit Int.* or *1551-1 Conv.Tracking unit Ext.*
SFTP client	22	TCP	Public Network Private Network I/O Network	Secure remote disk mounting with FTP over SSH. Enabled by configuration.
SNMP v2c	161 162	UDP	Public Network I/O Network	Used in combination with PROFINET.

NFS

The Network File System (NFS) is the de facto standard for file sharing among UNIX hosts and also supported by Microsoft, e.g. Windows Services for UNIX (SFU). The OmniCore robot controller implements an NFS client. The supported NFS version is version 2 as defined in RFC 1094.

RobICI

RobICI is an internal ABB application protocol that is used for high speed communication of I/O signals and other data between ABB products, for example, Conveyor Tracking Modules, OmniCore robot controllers and RobotStudio.

10.7 Certificate handling

Default self-signed certificates

Robot controllers support the use of X.509 certificates for secure communication over the network. The robot controller generates self-signed X.509 certificates by default for Robot Web Services and Robot Network Protocol (RobAPI). The generated self-signed certificate has an RSA key pair with a key length of 2048 bits.

Certificate replacement

To enhance the security of the system and to assure that data is being transmitted over a secure connection, it is recommended to replace the self-signed certificates on the robot controller with your own X.509 certificates. These can either be created from your own Public Key Infrastructure (PKI) or from a Certificate Authority (CA) of your choice. This provides added security and the ability to use your own trusted certificate chain.

To replace a self-signed certificate, export your desired certificate and private key in PEM format (.pem) and replace the certificate in RobotStudio. It is important to follow the proper procedures for certificate replacement in order to ensure seamless and secure communication. See *Operating manual - RobotStudio* for information about how to access the **Manage Certificates** function in RobotStudio.

Connected Services certificate

Connected Services installs a public certificate as part of the ABB Connect™ registration process. The certificate is stored in hardware-based secure storage and can be removed or replaced.

This page is intentionally left blank

Index

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

3HAC065037-001, Rev R, en