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

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

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

This manual and parts thereof must not be reproduced or copied without ABB’s written permission, and contents thereof must not be imparted to a third party nor be used for any unauthorized purpose. Contravention will be prosecuted.

Additional copies of this manual may be obtained from ABB at its then current charge.
# Table of Contents

Overview ........................................................................................................... 10
Product documentation, M2004 ................................................................. 12
Safety .................................................................................................................. 14

## 1 Introduction

1.1 Terms and Concepts .................................................................................. 15
  1.1.1 Hardware concepts ............................................................................. 15
  1.1.2 RobotWare concepts .......................................................................... 17
  1.1.3 RAPID concepts ................................................................................ 19
  1.1.4 Concepts of programming ................................................................... 20
  1.1.5 Targets and paths .............................................................................. 21
  1.1.6 Coordinate systems ........................................................................... 22
  1.1.7 Robot axis configurations ................................................................. 24
  1.1.8 Libraries, geometries and CAD files ................................................. 26
  1.1.9 VSTA as the IDE ................................................................................ 29
1.2 Installing and Licensing RobotStudio ....................................................... 30
1.3 The Graphical User Interface .................................................................... 33
  1.3.1 The Getting Started window .............................................................. 33
  1.3.2 The Layout browser .......................................................................... 34
  1.3.3 The Paths & Targets browser ............................................................. 35
  1.3.4 The Modeling browser ....................................................................... 37
  1.3.5 The Offline and Online browsers ....................................................... 38
  1.3.6 The Output window .......................................................................... 41
  1.3.7 Operator Window ............................................................................. 44
  1.3.8 Using a mouse ................................................................................... 46
  1.3.9 Selecting an item ................................................................................ 47
  1.3.10 Attaching and detaching objects ...................................................... 48
  1.3.11 Keyboard Shortcuts ......................................................................... 49
1.4 Manually setting up a system with track motion .......................................... 62
  1.4.1 Setting up a system with track motion of type RTT or IRBTx003 manually .................................................. 62
  1.4.2 Setting up a system with track motion of type IRBTx004 manually .................................................. 63
1.5 The VC ....................................................................................................... 64
  1.5.1 Starting a VC .................................................................................... 64
  1.5.2 Restarting a VC ................................................................................ 66
1.6 Station components .................................................................................... 68
  1.6.1 Importing a station component ........................................................... 68
  1.6.2 Converting CAD formats ................................................................... 70
  1.6.3 Troubleshooting and optimizing geometries ..................................... 71
1.7 Modeling ..................................................................................................... 73
  1.7.1 Objects ............................................................................................ 73
  1.7.2 Mechanisms ...................................................................................... 75
  1.7.3 Tools and tooldata ............................................................................ 76
  1.7.4 Setting the local origin of an object ................................................ 77
1.8 Placement .................................................................................................... 78
  1.8.1 Placing objects .................................................................................. 78
  1.8.2 Placing external axes ....................................................................... 79
# Table of Contents

## 3 How to program robots
- 3.1 Workflow for programming a robot .......................................................... 82
- 3.2 Workobjects ................................................................................................. 83
- 3.3 Jogging mechanisms .................................................................................... 84
- 3.4 Targets .......................................................................................................... 85
- 3.5 Paths .............................................................................................................. 87
- 3.6 Orientations .................................................................................................. 91
- 3.7 RAPID Instructions ...................................................................................... 94
- 3.8 Testing positions and motions .................................................................... 100
- 3.9 Programming MultiMove systems ............................................................. 102
  - 3.9.1 About programming MultiMove ............................................................ 102
  - 3.9.2 Setting up the MultiMove ...................................................................... 104
  - 3.9.3 Testing the MultiMove .......................................................................... 105
  - 3.9.4 Tuning the motion behavior ................................................................. 106
  - 3.9.5 Creating paths ....................................................................................... 108
  - 3.9.6 Programming external axes .................................................................. 109
- 3.10 Loading and saving programs and modules ............................................. 111
- 3.11 Synchronization ......................................................................................... 112
- 3.12 Using the RAPID editor .......................................................................... 113

## 4 How to simulate programs
- 4.1 Simulation Overview ................................................................................... 117
- 4.2 Detecting collisions ..................................................................................... 119
- 4.3 Creating an event ......................................................................................... 122
- 4.4 Simulating I/O signals ................................................................................ 123
- 4.5 Enabling simulation monitoring ................................................................. 124
- 4.6 Measuring process time ............................................................................. 125

## 5 Deployment and distribution
- 5.1 Copying programs ....................................................................................... 128
- 5.2 Pack & Go / Unpack & Work .................................................................... 129
- 5.3 Screen Capture ............................................................................................ 130

## 6 Working online
- 6.1 Connecting a PC to the service port ............................................................ 132
- 6.2 Network settings ........................................................................................ 135
- 6.3 User Authorization ..................................................................................... 137
- 6.4 The System Builder .................................................................................... 139
  - 6.4.1 System Builder Overview ..................................................................... 139
  - 6.4.2 Viewing system properties .................................................................... 141
  - 6.4.3 Building a new system ........................................................................... 142
  - 6.4.4 Modifying a system .............................................................................. 145
  - 6.4.5 Copying a system ................................................................................ 149
  - 6.4.6 Creating a system from backup ............................................................ 150
  - 6.4.7 Downloading a system to a controller ................................................. 151
  - 6.4.8 Creating boot media ............................................................................ 152
  - 6.4.9 Examples using the System Builder Offline ........................................ 153
    - 6.4.9.1 A MultiMove system with two coordinated robots ....................... 153
    - 6.4.9.2 A system with support for one robot and one positioner external axis. 155
    - 6.4.9.3 Options settings for systems with positioners ............................. 157
- 6.5 Handle I/O .................................................................................................. 159
- 6.6 Configure systems ...................................................................................... 160
- 6.7 Handle events ............................................................................................. 165
- 6.8 Handle devices ............................................................................................ 168
# 8 The Home Tab

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Overview</td>
<td>185</td>
</tr>
<tr>
<td>8.2 ABB Library</td>
<td>186</td>
</tr>
<tr>
<td>8.3 Import Library</td>
<td>187</td>
</tr>
<tr>
<td>8.4 Robot System</td>
<td>188</td>
</tr>
<tr>
<td>8.5 Import Geometry</td>
<td>191</td>
</tr>
<tr>
<td>8.6 Frame</td>
<td>192</td>
</tr>
<tr>
<td>8.6.1 Frame</td>
<td>192</td>
</tr>
<tr>
<td>8.6.2 Frame from Three Points</td>
<td>193</td>
</tr>
<tr>
<td>8.7 Workobject</td>
<td>195</td>
</tr>
<tr>
<td>8.8 Tooldata</td>
<td>196</td>
</tr>
<tr>
<td>8.9 Target</td>
<td>197</td>
</tr>
<tr>
<td>8.9.1 Teach Target</td>
<td>197</td>
</tr>
<tr>
<td>8.9.2 Create Target</td>
<td>198</td>
</tr>
<tr>
<td>8.9.3 Create JointTarget</td>
<td>200</td>
</tr>
<tr>
<td>8.10 Empty Path</td>
<td>201</td>
</tr>
<tr>
<td>8.11 Path from Curve</td>
<td>202</td>
</tr>
<tr>
<td>8.12 MultiMove</td>
<td>205</td>
</tr>
<tr>
<td>8.13 Teach Instruction</td>
<td>213</td>
</tr>
<tr>
<td>8.14 Move Instruction</td>
<td>214</td>
</tr>
<tr>
<td>8.15 Action Instruction</td>
<td>215</td>
</tr>
<tr>
<td>8.16 Instruction Template Manager</td>
<td>216</td>
</tr>
<tr>
<td>8.17 The Freehand Group</td>
<td>219</td>
</tr>
<tr>
<td>8.17.1 Move</td>
<td>219</td>
</tr>
<tr>
<td>8.17.2 Rotate</td>
<td>220</td>
</tr>
<tr>
<td>8.17.3 Jog Joint</td>
<td>221</td>
</tr>
<tr>
<td>8.17.4 Jog Linear</td>
<td>222</td>
</tr>
<tr>
<td>8.17.5 MultiRobot Jog</td>
<td>223</td>
</tr>
<tr>
<td>8.18 Viewpoint</td>
<td>224</td>
</tr>
</tbody>
</table>

# 9 The Modeling Tab

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1 Overview</td>
<td>225</td>
</tr>
<tr>
<td>9.2 Component Group</td>
<td>226</td>
</tr>
<tr>
<td>9.3 Empty Part</td>
<td>227</td>
</tr>
<tr>
<td>9.4 Solid</td>
<td>228</td>
</tr>
<tr>
<td>9.5 Surface</td>
<td>232</td>
</tr>
<tr>
<td>9.6 Curve</td>
<td>234</td>
</tr>
<tr>
<td>9.7 Border</td>
<td>239</td>
</tr>
<tr>
<td>9.8 Intersect</td>
<td>241</td>
</tr>
<tr>
<td>9.9 Subtract</td>
<td>242</td>
</tr>
<tr>
<td>9.10 Union</td>
<td>243</td>
</tr>
<tr>
<td>9.11 Extrude Surface or Curve</td>
<td>244</td>
</tr>
<tr>
<td>9.12 Line from Normal</td>
<td>245</td>
</tr>
</tbody>
</table>
# Table of Contents

## 9.13 The Measure Group .......................................................... 246
## 9.14 Create Mechanism .............................................................. 247
## 9.15 Create Tool ................................................................. 253

### 10 The Simulation Tab .......................................................... 255
  - 10.1 Overview ........................................................................... 255
  - 10.2 Create Collision Set ........................................................ 256
  - 10.3 Simulation Setup ............................................................. 257
  - 10.4 Event Manager ............................................................... 258
  - 10.5 Activate Mechanical Units .................................................. 264
  - 10.6 Simulation Control ........................................................... 265
  - 10.7 I/O Simulator .................................................................... 266
  - 10.8 Monitor ............................................................................ 268
  - 10.9 Record Movie .................................................................... 269

### 10.10 Conveyor Tracking Mechanism ........................................... 270
  - 10.10.1 Conveyor Tracking ......................................................... 270
  - 10.10.2 Conveyor Simulation ......................................................... 271

### 11 The Offline Tab ..................................................................... 273
  - 11.1 Overview ........................................................................... 273
  - 11.2 Synchronize to Station ......................................................... 274
  - 11.3 Synchronize to VC ............................................................. 275
  - **11.4 Backup ................................................................. 276**
    - 11.4.1 Backing up a system ...................................................... 276
    - 11.4.2 Restoring a system from backup ...................................... 277
  - 11.5 Events ............................................................................. 279
  - 11.6 RAPID editor ................................................................... 280
  - 11.7 Inputs / Outputs ................................................................ 282
  - 11.8 System Builder .................................................................. 283
  - 11.9 Rapid Tasks ...................................................................... 284
  - 11.10 Restart ............................................................................ 287
  - 11.11 Shutdown ........................................................................ 288
  - 11.12 New Module .................................................................... 289
  - 11.13 Load Module .................................................................... 290
  - 11.14 Load Program ................................................................... 291
  - 11.15 Add signals ...................................................................... 292
  - 11.16 Set Task Frames ............................................................... 293
  - 11.17 System Configuration ....................................................... 294
  - 11.18 Encoder Unit .................................................................... 296
  - 11.19 Configuration editor .......................................................... 297
  - 11.20 Load Parameters .............................................................. 299
  - 11.21 Save System Parameters .................................................... 300

### 12 The Online Tab .................................................................... 301
  - 12.1 Overview ........................................................................... 301
  - 12.2 Add Controller ................................................................... 302
  - 12.3 Request Write Access .......................................................... 303
  - 12.4 Release Write Access ........................................................... 304
  - 12.5 Import Options ................................................................... 305
  - 12.6 User Accounts .................................................................... 306
  - 12.7 UAS Grant Viewer .............................................................. 311
  - 12.8 FlexPendant Viewer ............................................................ 315
  - 12.9 Device Browser ................................................................... 316
# Table of Contents

## 13 The Add-Ins Tab

13.1 Overview ................................................. 317  
13.2 Visual Studio Tools for Applications ...................... 318

## 14 The Context Menus

14.1 Add to Path .................................................. 320  
14.2 Align Frame Orientation ................................... 321  
14.3 Align Target Orientation ................................... 322  
14.4 Attach to ................................................... 323  
14.5 Auto Configuration ......................................... 324  
14.6 Check Reachability ......................................... 325  
14.7 Configurations ............................................. 326  
14.8 Control Panel .............................................. 327  
14.9 Convert Frame to Workobject .............................. 328  
14.10 Copy / Move Circular ...................................... 329  
14.11 Copy / Apply Orientation ................................. 330  
14.12 Detach ..................................................... 331  
14.13 Execute Move Instruction .................................. 332  
14.14 Graphic Appearance ....................................... 333  
14.15 Interpolate Path .......................................... 335  
14.16 Invert ..................................................... 336  
14.17 Jump to Target ............................................. 337  
14.18 Linked Geometry .......................................... 338  
14.19 The Library Group ......................................... 339  
14.20 Mechanism Joint Jog ...................................... 340  
14.21 Mechanism Linear Jog ..................................... 342  
14.22 Mirror Path ................................................. 343  
14.23 Modify Curve .............................................. 344  
14.24 Modify External Axis ...................................... 349  
14.25 Modify Instruction ........................................ 350  
14.26 Modify Mechanism ........................................ 351  
14.27 Modify Tooldata ........................................... 352  
14.28 Modify Workobject ........................................ 353  
14.29 Move Along Path .......................................... 354  
14.30 Move to Pose ............................................... 355  
14.31 Place ....................................................... 356  
14.32 Remove Unused Targets ................................... 358  
14.33 Rename Targets ........................................... 359  
14.34 Reverse Path .............................................. 360  
14.35 Rotate ....................................................... 361  
14.36 Rotate Path ................................................ 362  
14.37 Save Module As ............................................ 363  
14.38 Save Program As .......................................... 364  
14.39 Set Local Origin .......................................... 365  
14.40 Set Normal to Surface .................................... 366  
14.41 Set Position ............................................... 367  
14.42 Tool Compensation ........................................ 368  
14.43 Translate Path ............................................ 369  
14.44 View Robot at Target ..................................... 370  
14.45 View Tool at Target ...................................... 371

## Index

Index 373
Overview

About This Manual

This manual describes how to create, program and simulate robot cells and stations using RobotStudio. For online programming, this manual describes how to supervise, install, configure and program a real robot controller. Terms and concepts related to offline and online programming are also explained.

RobotStudio offers the following installation options:

- Full
- Custom, allowing user-customized contents and paths
- Minimal, running RobotStudio in Online mode only

Usage

This manual should be used when working with either the offline or online functions of RobotStudio.

Who Should Read This Manual?

This manual is intended for RobotStudio users, proposal engineers, mechanical designers, offline programmers, robot technicians and service technicians.

Prerequisites

The reader should have basic knowledge of:

- Robot programming
- Generic Windows handling
- 3D CAD programs

Organization of Chapters

The manual is organized into six main chapters containing procedures and explanations.

These are followed by descriptions of the commands of the Graphical User Interface (GUI), loosely arranged—in order to be read from the application as online help files—according to menu or tab.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>Contains installation instructions, basic explanations of the terms and concepts related to robotics and programming, and a description of the GUI.</td>
</tr>
<tr>
<td>2. How to build stations</td>
<td>Describes how to build stations in RobotStudio. This includes importing and configuring the equipment to be simulated, as well as testing the reachability for finding the optimal station layout.</td>
</tr>
<tr>
<td>3. How to program robots</td>
<td>Describes how to create robot movements, I/O signals, process instructions and logics in a RAPID program for the robots. It also describes how to run and test the program.</td>
</tr>
<tr>
<td>4. How to simulate programs</td>
<td>Describes how to simulate and validate robot programs.</td>
</tr>
</tbody>
</table>
Chapter 5. Deployment and distribution
Describes how to transfer systems between RobotStudio’s virtual controllers and real IRC5 controllers, how to copy programs, how to package an active station for moving between RobotStudio PCs, and how to capture a screen.

Chapter 6. Working online
Covers the functionality of the Minimal Installation, describing such online functions as building systems (with offline examples), handling I/O and events, and configuring systems.

References

<table>
<thead>
<tr>
<th>Reference</th>
<th>Document Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product manual - IRC5</td>
<td>3HAC021313-001</td>
</tr>
<tr>
<td>Operating manual - IRC5 with FlexPendant</td>
<td>3HAC16590-1</td>
</tr>
<tr>
<td>Technical reference manual - RAPID overview</td>
<td>3HAC16580-1</td>
</tr>
<tr>
<td>Technical reference manual - System parameters</td>
<td>3HAC17076-1</td>
</tr>
<tr>
<td>Application manual - MultiMove</td>
<td>3HAC021272-001</td>
</tr>
<tr>
<td>Application manual - Conveyor tracking</td>
<td>3HAC16587-1</td>
</tr>
</tbody>
</table>

Revisions

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>First revision, called RobotStudio 2008, released for Partner Days. The entire manual has been adapted to the new GUI, in which RobotStudioOnline has been integrated.</td>
</tr>
<tr>
<td>B</td>
<td>Updated the manual with information on Conveyor Tracking, Operator Window, Station Viewer, Linked Geometry, Viewpoints and Recorder simulation.</td>
</tr>
</tbody>
</table>
Categories for robot documentation

The robot documentation 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.

All documents listed can be ordered from ABB on a DVD. The documents listed are valid for M2004 robot systems.

Product manuals

All hardware, robots and controllers, will be delivered with a Product manual that contains:

- Safety information.
- Installation and commissioning (descriptions of mechanical installation, electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Additional procedures, if any (calibration, decommissioning).
- Reference information (article numbers for documentation referred to in Product manual, procedures, lists of tools, safety standards).
- Part list.
- Foldouts or exploded views.
- Circuit diagrams.

Technical reference manuals

The technical reference manuals describe the robot software in general and contain relevant reference information.

- **RAPID Overview**: An overview of the RAPID programming language.
- **RAPID Instructions, Functions and Data types**: Description and syntax for all RAPID instructions, functions, and data types.
- **RAPID Kernel**: A formal description of the RAPID programming language.
- **System parameters**: Description of system parameters and configuration workflows.

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, CD with PC software).
- How to use the application.
- Examples of how to use the application.

Continues on next page
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 trouble shooters.

The group of manuals includes (among others):

- Emergency safety information
- General safety information
- Getting started, IRC5 and RobotStudio
- IRC5 with FlexPendant
- RobotStudio
- Introduction to RAPID
- Trouble shooting, for the controller and robot.
Safety

Safety of personnel

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

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

Safety regulations

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

1.1 Terms and Concepts

1.1.1. Hardware concepts

Overview

This section introduces the hardware in a typical IRC5 robot cell. For detailed explanations, see the manuals related to IRC5 robots specified in References on page 11.

Standard hardware

The table below describes the standard hardware in an IRC5 robot cell.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot manipulator</td>
<td>An ABB industrial robot.</td>
</tr>
<tr>
<td>Control module</td>
<td>Contains the main computer that controls the motion of the manipulator. This includes RAPID execution and signal handling. One control module can be connected to 1 – 4 drive modules.</td>
</tr>
<tr>
<td>Drive module</td>
<td>A module containing the electronics that power the motors of a manipulator. The drive module can contain up to nine drive units, each controlling one manipulator joint. Since the standard robot manipulators have six joints, you usually use one drive module per robot manipulator.</td>
</tr>
<tr>
<td>FlexController</td>
<td>The controller cabinet for the IRC5 robots. It consists of one control module and one drive module for each robot manipulator in the system.</td>
</tr>
<tr>
<td>FlexPendant</td>
<td>The programming pendant, connected to the control module. Programming on the FlexPendant is referred to as “online programming”.</td>
</tr>
<tr>
<td>Tool</td>
<td>A device usually mounted on the robot manipulator to allow it to perform specific tasks, such as gripping, cutting or welding. The tool can also be stationary, see below for more information.</td>
</tr>
</tbody>
</table>

Optional hardware

The table below describes the optional hardware for an IRC5 robot cell.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track manipulator</td>
<td>A moving stand holding the robot manipulator to give it a larger work space. When the control module controls the motion of a track manipulator, it is referred to as a “Track External Axis”.</td>
</tr>
<tr>
<td>Positioner manipulator</td>
<td>A moving stand normally holding a work piece or a fixture. When the control module controls the motion of a positioner manipulator, it is referred to as an “External Axis”.</td>
</tr>
<tr>
<td>FlexPositioner</td>
<td>A second robot manipulator acting as a positioner manipulator. It is controlled by the same control module as the positioner manipulator.</td>
</tr>
<tr>
<td>Stationary tool</td>
<td>A device that stands in a fixed location. The robot manipulator picks up the work piece and brings it to the device to perform specific tasks, such as gluing, grinding or welding.</td>
</tr>
<tr>
<td>Work piece</td>
<td>The product being worked on.</td>
</tr>
</tbody>
</table>

Continues on next page
1 Introduction

1.1.1. Hardware concepts

*Continued*

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixture</td>
<td>A construction holding the work piece in a specific position so that the repeatability of the production can be maintained.</td>
</tr>
</tbody>
</table>
1.1.2. RobotWare concepts

Overview

This section introduces terminology regarding RobotWare. For detailed explanations, see the manuals related to IRC5 robots specified in References on page 11.

RobotWare

The table below describes the RobotWare terminology and concepts that can be useful when working with RobotStudio.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RobotWare</td>
<td>As a concept, refers to both the software used to create a RobotWare System and the RobotWare systems themselves.</td>
</tr>
<tr>
<td>RobotWare DVD</td>
<td>Delivered with each control module. On the DVD you will find the RobotWare installation and some other useful software. Check the Release Notes on your DVD for specifications.</td>
</tr>
<tr>
<td>RobotWare installation</td>
<td>When installing RobotWare on a PC, you install into the mediapool the specific versions of the files from which RobotStudio uses to create the RobotWare system. When installing RobotStudio, only one version of RobotWare will be installed. To simulate a specific RobotWare system, the RobotWare version used for this particular RobotWare system must be installed on your PC.</td>
</tr>
<tr>
<td>RobotWare Key</td>
<td>Used when you create a new RobotWare system or upgrade an existing system. The RobotWare keys unlock the RobotWare options included in the system, and determine the RobotWare version from which the RobotWare system will be built. For IRC5 systems there are three types of RobotWare keys: • The controller key, which specifies the controller and software options. • The drive keys, which specify the robots in the system. The system has one drive key for each robot it uses. • Additional option keys, which specify additional options, like positioner external axes. A virtual key allows you to select any RobotWare options you wish, but a RobotWare system created from a virtual key can only be used in a virtual environment such as RobotStudio.</td>
</tr>
<tr>
<td>RobotWare system</td>
<td>A set of software files that, when loaded into a controller, enables all functions, configurations, data and programs controlling the robot system. RobotWare systems are created in the RobotStudio software. The systems can be stored and saved on a PC, as well as on the control module. RobotWare systems can be edited by RobotStudio or the FlexPendant.</td>
</tr>
<tr>
<td>RobotWare version</td>
<td>Each RobotWare is released with a major and a minor version number, separated by a dot. The RobotWare version for IRC5 is 5.xx, where xx identifies the minor version. When ABB releases a new robot model, a new RobotWare version will be released with support for the new robot.</td>
</tr>
</tbody>
</table>
1 Introduction

1.1.2. RobotWare concepts

<table>
<thead>
<tr>
<th>Concept</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mediapool</td>
<td>The mediapool is a folder on the PC in which each RobotWare version is stored in a folder of its own. The files of the mediapool are used to create and implement all the different RobotWare options. Therefore, the correct RobotWare version must be installed in the mediapool when creating RobotWare systems or running them on virtual controllers.</td>
</tr>
</tbody>
</table>
1.1.3. RAPID concepts

Overview

This section introduces the basic terminology of RAPID. For detailed explanations, see the manuals related to RAPID and programming specified in References on page 11.

Terminology of the RAPID structure

The table below describes the RAPID terminology that you may come across when working with RobotStudio. The concepts are listed by size, from most basic to increasingly large.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data declaration</td>
<td>Used to create instances of variables or data types, like num or tooldata.</td>
</tr>
<tr>
<td>Instruction</td>
<td>The actual code commands that make something happen, for example, setting data to a specific value or a robot motion. Instructions can only be created inside a routine.</td>
</tr>
<tr>
<td>Move instructions</td>
<td>Create the robot motions. They consist of a reference to a target specified in a data declaration along with parameters that set motion and process behavior. If inline targets are used, the position is declared in the move instructions.</td>
</tr>
<tr>
<td>Action instruction</td>
<td>Instructions that perform other actions than moving the robot, such as setting data or sync properties.</td>
</tr>
<tr>
<td>Routine</td>
<td>Usually a set of data declarations followed by a set of instructions implementing a task. Routines can be divided into three categories: procedures, functions and trap routines.</td>
</tr>
<tr>
<td>Procedure</td>
<td>A set of instructions that does not return a value.</td>
</tr>
<tr>
<td>Function</td>
<td>A set of instructions that returns a value.</td>
</tr>
<tr>
<td>Trap</td>
<td>A set of instructions that is triggered by an interrupt.</td>
</tr>
<tr>
<td>Module</td>
<td>A set of data declarations followed by a set of routines. Modules can be saved, loaded and copied as files. Modules are divided into program modules and system modules.</td>
</tr>
<tr>
<td>Program module (.mod)</td>
<td>Can be loaded and unloaded during execution.</td>
</tr>
<tr>
<td>System module (.sys)</td>
<td>Used mainly for common system-specific data and routines, for example, an arcware system module that is common for all arc robots.</td>
</tr>
<tr>
<td>Program files (.pgf)</td>
<td>In IRC5 a RAPID program is a collection of module files (.mod) and the program file (.pgf) that references all the module files. When loading a program file, all old program modules are replaced by those referenced in the .pgf file. System modules are unaffected by program load.</td>
</tr>
</tbody>
</table>
1 Introduction

1.1.4. Concepts of programming

Overview

This section introduces the terminology regarding programming. For detailed explanations, see the manuals related to programming and IRC5 Robots specified in References on page 11.

Programming concepts

The table below describes the terminology and concepts that are used in robot programming.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online programming</td>
<td>Programming connected to the control module. This expression also implies using the robot to create positions and motion.</td>
</tr>
<tr>
<td>Offline programming</td>
<td>Programming without being connected to the robot or the control module.</td>
</tr>
<tr>
<td>True offline programming</td>
<td>Refers to the ABB Robotics concept of connecting a simulation environment to a virtual controller. This enables not only program creation, but also program testing and optimizing offline.</td>
</tr>
<tr>
<td>Virtual controller</td>
<td>A software that emulates a FlexController to allow the same software (the RobotWare system) that is controlling the robots to run on a PC. This gives the same behavior of the robots offline as you get online.</td>
</tr>
<tr>
<td>MultiMove</td>
<td>Running multiple robot manipulators with the same control module.</td>
</tr>
<tr>
<td>Coordinate systems</td>
<td>Used to define positions and orientations. When programming a robot, you can take advantage of using different coordinate systems to more easily position objects relative to each other.</td>
</tr>
<tr>
<td>Frame</td>
<td>A synonym for coordinate system.</td>
</tr>
<tr>
<td>Workobject calibration</td>
<td>If all your targets refer to workobjects, you only need to calibrate the workobjects when deploying offline programs.</td>
</tr>
</tbody>
</table>
1.1.5. Targets and paths

Overview

Targets (positions) and paths (sequences of move instructions to targets) are used when programming robot motions in RobotStudio.

When you synchronize the RobotStudio station to the virtual controller, RAPID programs are created from the paths.

Targets

A target is a coordinate that the robot shall reach. It contains the following information:

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>The position of the target, defined in a workobject coordinate system, see Coordinate systems on page 22.</td>
</tr>
<tr>
<td>Orientation</td>
<td>The orientation of the target, relative to the orientation of the workobject. When the robot reaches the target, it will align the TCP's orientation with the target's orientation, see Coordinate systems on page 22.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Configuration values that specify how the robot shall reach the target. For more information, see Robot axis configurations on page 24.</td>
</tr>
</tbody>
</table>

Targets are converted to instances of the data type robtarget when synchronized to the virtual controller.

Paths

A sequence of move instructions, paths are used to make the robot move along a sequence of targets.

Paths are converted to procedures when synchronized to the virtual controller.

Move instructions

A move instruction consists of:

- a reference to a target
- motion data, such as motion type, speed and zone
- a reference to a tooldata
- a workobject reference

Action instructions

An action instruction is a RAPID string that can be used for setting and changing parameters. Action instructions can be inserted before, after or between instruction targets in paths.
1 Introduction

1.1.6. Coordinate systems

Overview

This section gives a short introduction to coordinate systems that are often used for offline programming. In RobotStudio, you can use all coordinate systems described below, as well as user-defined coordinate systems, for relating objects and elements to each other.

Hierarchy

The coordinate systems relate to each other hierarchically, where the origin of each coordinate system is defined as a position in one of its ancestries. Below are descriptions of commonly used coordinate systems, starting at the top of the hierarchy.

World coordinate system

The world coordinate system represents the entire station or robot cell. This is the top of the hierarchy to which all other coordinate systems are related.

Task frame coordinate system

The task frame coordinate system is useful to define within a station or robot cell, in particular with multiple robots or mechanisms. One task frame can coordinate the placement of several mechanisms, whereas several task frames are suitable when working with MultiMove Independent.

Base coordinate system

Each robot in the station has a base coordinate system, which is always located at the base of the robot.

Tool Center Point coordinate system

The tool center point coordinate system, also called TCP, is the center point of the tool. Several different TCPs may be defined for one robot. All robots have one predefined TCP at the robot’s tool mounting point, called tool0.

When a program runs, the robot moves the TCP to the programmed position.

Continues on next page
1 Introduction

1.1.6. Coordinate systems

WorkObject coordinate system

The workobject normally represents the physical work piece. It is composed of two coordinate systems: the User frame and the Object frame, where the latter is a child to the former. When programming a robot, all targets (positions) are related to the object frame of a workobject. If no other workobject is specified, the targets will be related to the default Wobj0, which always coincides with the base frame of the robot.

Using workobjects provides the chance to easily adjust robot programs with an offset, if the location of the work piece has been changed. Thus, workobjects can be used for calibrating offline programs. If the placement of the fixture/work piece relative to the robot in the real station does not completely match the placement in the offline station, you simply adjust the position of the workobject.

Workobjects are also used for coordinated motions. If a workobject is attached to a mechanical unit (and the system uses the option for coordinated motions), the robot will find the targets in the workobject even when the mechanical unit moves the workobject.

In the picture below the grey coordinate system is the world coordinate system, and the black ones are the object frame and the user frame of the workobject. Here the user frame is positioned at the table/fixture and the object frame at the workpiece.

User Coordinate Systems

User Coordinate Systems (UCSs) are used for creating reference points of your choice. For example, you can create UCSs at strategic points in the work piece to facilitate programming.
1 Introduction

1.1.7. Robot axis configurations

Axis configurations

Targets are defined and stored as coordinates in a WorkObject coordinate system. When the controller calculates the position of the robot axes for reaching the target, it will often find more than one possible solution to configuring the robot axes.

To distinguish between the different configurations, all targets have a configuration value that specifies the quadrant in which each axis shall be located.

Storing axis configurations in targets

For targets that are taught after jogging the robot to the position, the used configuration will be stored in the target.

Targets created by specifying or calculating positions and orientations get a default configuration value (0,0,0,0), which might not be valid for reaching the target.

Common problems related to robot axis configurations

It is most likely that targets created by other ways than jogging cannot be reached at their default configuration.

Even if all targets in a path have validated configurations, you might encounter problems when running the path if the robot cannot move from one configuration to the other. This is likely to occur where an axis shifts greater than 90 degrees during linear movements.

Repositioned targets keep their configuration, but the configurations are no longer validated. As a result, the problems described above might occur when moving targets.

Common solutions for configuration problems

To resolve the problems described above, you can assign a valid configuration to each target and verify that the robot can move along each path. You can also turn configuration monitoring off, which means that you ignore the stored configurations and let the robot find working configurations at runtime. If this is not done the proper way, you might get unexpected results.

In some cases there might not be any working configurations. Possible solutions might then be to reposition the work piece, reorient targets (if acceptable for the process) or add an external axis that either moves the work piece or the robot for increasing reachability.

Continues on next page
How configurations are denoted

The robot’s axis configurations are denoted by a series of four integers, specifying in which quadrant of a full revolution significant axes are located. The quadrants are numbered from zero for positive (counterclockwise) rotation and from -1 for negative (clockwise) rotation.

For a linear axis, the integer specifies the range (in meters) from the neutral position in which the axis is located.

A configuration for a six-axis industrial robot (like IRB 140) may look like:

\[ [0, -1, 2, 1] \]

The first integer (0) specifies the position of axis 1: somewhere in the first positive quadrant (between 0 and 90 degrees rotation).

The second integer (-1) specifies the position of axis 4: somewhere in the first negative quadrant (between 0 and -90 degrees rotation).

The third integer (2) specifies the position of axis 6: somewhere in the third positive quadrant (between 180 and 270 degrees rotation).

The fourth integer (1) specifies the position of axis x, a virtual axis used for specifying the wrist center in relation to other axes.

Configuration monitoring

When executing a robot program, you can choose whether to monitor configuration values. If configuration monitoring is turned off, configuration values stored with the targets are ignored, and the robot will use the configuration closest its current configuration for reaching the target. If turned on, it will only use the specified configuration for reaching the targets.

Configuration monitoring can be turned off and on for joint and linear movements independently and is controlled by the ConfJ and ConfL action instructions.

Turning configuration monitoring off

Running a program without configuration monitoring may result in different configurations each time a cycle is executed: When the robot returns to the start position after completing a cycle, it may choose a different configuration then the original.

For programs with linear move instructions this might cause a situation where the robot gets closer and closer its joint limits and eventually will not be able to reach the target.

For programs with joint move instructions this might cause sweeping, unpredictable movements.

Turning configuration monitoring on

Running a program with configuration monitoring forces the robot to use the configurations stored with the targets. This results in predictable cycles and predictable motions. In some situations, however, like when the robot moves to a target from an unknown position, using configuration monitoring may limit the robot’s reachability.

When programming offline, you must assign a configuration to each target if the program shall be executed with configuration monitoring.
1.1.8. Libraries, geometries and CAD files

Overview

For programming or simulating in RobotStudio, you need models of your work pieces and equipment. Models for some standard equipment are installed as libraries or geometries with RobotStudio. If you have CAD models of your work pieces and custom equipment, these can be imported as geometries to RobotStudio. If you do not have CAD models, you can create them in RobotStudio.

Difference between geometries and libraries

The objects you import to a station can be either geometries or libraries. Geometries are basically CAD files, which, when imported, are copied to the RobotStudio station. Libraries are objects that have been saved in RobotStudio as external files. When you import a library, a link from the station to the library file is created. Accordingly, the station file does not grow in the same way as when importing geometries. Furthermore, besides the geometrical data, library files can contain RobotStudio-specific data. For example, if a tool is saved as a library, the tool data is saved together with the CAD data.

How geometries are constructed

An imported geometry is displayed as one part in the Objects browser. From RobotStudio’s Modeling tab, you can see the components of the geometry. The top node of the geometry is called a Part. The part contains Bodies, which can be of the types solid, surface or curve.

Solid bodies are 3D objects, made up of Faces. You recognize a true 3D solid by this one body containing multiple faces.

Surface bodies are 2D objects of just one face. A part that contains several bodies with one face each that together constitute a 3D object is created from 2D surfaces, and is therefore not a true 3D solid. If these parts are not created correctly, they might cause problems both in their display and graphical programming. see Troubleshooting and optimizing geometries on page 71.

Curved bodies, represented by the body node alone in the Modeling browser, do not contain any child nodes.

From the Modeling tab, you can edit the parts by adding, moving, rearranging or deleting bodies. Thus, you can optimize existing parts by removing unnecessary bodies, as well as create new parts by grouping bodies.

Importing and converting CAD files

For importing geometries from single CAD files, you use RobotStudio’s import function, see Importing a station component on page 68.

If you need to convert CAD files to other formats or want to change the default settings for the conversion before making the import, you can use the CAD converter installed with RobotStudio before making the import, see Converting CAD formats on page 70.

Continues on next page
### Supported CAD formats

Some of the CAD formats require a separate license for being imported or converted by RobotStudio. The table below shows the supported CAD formats and whether a license is required:

<table>
<thead>
<tr>
<th>Format</th>
<th>File extensions</th>
<th>License requirement</th>
<th>Default target formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acis, reads/writes versions v6 to R18</td>
<td>sat</td>
<td>No</td>
<td>Iges, Step, Vdafs</td>
</tr>
<tr>
<td>Iges, reads to version 5.3, writes version 5.3</td>
<td>igs, iges</td>
<td>Yes</td>
<td>Acis, Step, Vdafs</td>
</tr>
<tr>
<td>Step, reads versions AP203 and AP214 (geometry only), writes version AP 214</td>
<td>stp, step, p21</td>
<td>Yes</td>
<td>Acis, Step, Vdafs</td>
</tr>
<tr>
<td>Vdaf, reads to 2.0, writes 2.0</td>
<td>vda, vdaf</td>
<td>Yes</td>
<td>Acis, Iges, Step</td>
</tr>
<tr>
<td>Catia V4, reads versions 4.1.9 to 4.2.4</td>
<td>model, exp</td>
<td>Yes</td>
<td>Acis, Iges, Step, Vdafs</td>
</tr>
<tr>
<td>Catia V5, reads versions R2 - R18</td>
<td>CATPart, CATProduct</td>
<td>Yes</td>
<td>Acis, Iges, Step, Vdafs</td>
</tr>
<tr>
<td>Pro/Engineer, reads versions 16 to Wildfire3</td>
<td>prt, asm</td>
<td>Yes</td>
<td>Acis, Iges, Step, Vdafs</td>
</tr>
<tr>
<td>Inventor, reads versions 6 to 12</td>
<td>ipt</td>
<td>Yes</td>
<td>Acis, Iges, Step, Vdafs</td>
</tr>
<tr>
<td>Vrml</td>
<td>wrl, vrml, vrml1, vrml2</td>
<td>No</td>
<td>RsGfx</td>
</tr>
<tr>
<td>Jupiter, up to 6.4</td>
<td>jt</td>
<td>No</td>
<td>RsGfx</td>
</tr>
<tr>
<td>STL</td>
<td>stl</td>
<td>No</td>
<td>RsGfx</td>
</tr>
<tr>
<td>PLY</td>
<td>ply</td>
<td>No</td>
<td>RsGfx</td>
</tr>
<tr>
<td>3DStudio</td>
<td>3ds</td>
<td>No</td>
<td>RsGfx</td>
</tr>
</tbody>
</table>

You need licenses for both the source format and the target format when converting licensed formats.
1 Introduction

1.1.8. Libraries, geometries and CAD files

Continued

Mathematical versus graphical geometries

A geometry in a CAD file always has an underlying mathematical representation. Its graphical representation, displayed in the graphics window, is generated from the mathematical representation when the geometry is imported to RobotStudio, after which the geometry is referred to as a part.

For this kind of geometry, you can set the detail level of the graphical representation, thus reducing the file size and rendering time for large models and improving the visual display for small models you might want to zoom in on. The detail level only affects the visual display; paths and curves created from the model will be accurate both with coarse and fine settings.

A part can also be imported from a file that simply defines its graphical representation; in this case, there is no underlying mathematical representation. Some of the functions in RobotStudio, such as snap mode and creation of curves from the geometry, will not work with this kind of part.

To customize the detail level settings, see RobotStudio Options on page 178.
1.1.9. VSTA as the IDE

Overview

RobotStudio uses Microsoft Visual Studio Tools for Applications (VSTA) as its Integrated Development Environment (IDE), enabling advanced users to extend and customize its functionality. You can, for example, write an add-in in C# or VB.Net to create a toolbar or macro, debug code or inspect variable values during execution.

In addition, the Add-In browser acts as a single window for both VSTA add-ins and non-VSTA, RobotStudio-specific add-ins known as PowerPacs.

For a procedure, see Visual Studio Tools for Applications on page 318.

Types of add-ins

The following add-ins may be available in the Add-In browser:

<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerPac</td>
<td>An add-in that accords with RobotStudio specifications, but is not part of the VSTA system itself.</td>
</tr>
<tr>
<td>VSTA Station Add-In</td>
<td>A VSTA add-in that has been attached to a station, and saved to the station file.</td>
</tr>
<tr>
<td>VSTA User Add-In</td>
<td>A VSTA add-in that has not been attached to a station and is active only on the current user’s computer.</td>
</tr>
</tbody>
</table>

Using the shortcut menus of the Add-In browser, PowerPacs may be loaded or autoloaded, while VSTA add-ins may be added, loaded, unloaded, reloaded, autoloaded, edited, deleted, or removed from the station.
1 Introduction

1.2. Installing and Licensing RobotStudio

1.2. Installing and Licensing RobotStudio

Installing RobotStudio

RobotStudio offers the following installation options:

- **Minimal** - Installs only program features required to run RobotStudio in Online mode
- **Complete** - Installs all program features required to run the complete RobotStudio
- **Custom** - Installs user-customized program features

How to install RobotStudio on a PC

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
</table>
| 1. Insert the robot software DVD in the PC.  
If a menu for the DVD is opened automatically, continue with step 5.  
If no menu for the DVD is opened, continue with step 2.  |
| 2. On the **Start** menu, click **Run**.  |
| 3. In the **Open** box, type the drive letter for your DVD drive followed by:  
\`:\launch.exe`  
**Example:** If your DVD drive has the letter D, then type: `D:\ launch.exe`  |
| 4. Click **OK**.  |
| 5. Select language for the DVD menu.  |
| 6. On the DVD menu, click **Install**.  |
| 7. On the installation menu, click **RobotStudio**. This opens the installation wizard, which will guide you through the rest of the software installation.  |
| 8. Follow the instructions in the installation wizard.  |
**NOTE!** For a trial period of 30 days, RobotStudio will work without activation.  |

Activate RobotStudio

To continue using the product with all of its features, you must activate it. RobotStudio Product Activation is based on Microsoft anti-piracy technology and designed to verify that software products are legitimately licensed.

Activation works by verifying that the Activation Key is not in use on more personal computers than are permitted by the software license.

How do I activate RobotStudio?

When you start RobotStudio for the first time after installation, you are prompted to enter your 25-digit Activation Key (xxxxx-xxxxx-xxxxx-xxxxx-xxxxx).  

Trial period

Before entering a valid Activation Key, you can run the software, in Premium functionality mode, with all the features enabled, for a trial period of up to 30 days.  

**NOTE!** Trial period starts immediately after installation.  
After entering a valid Activation Key, you will see only the features you have purchased.  

**NOTE!** If installed during the trial period, you will lose the trial period.  

Continues on next page
Basic Functionality mode

After the trial period, the software reverts to Reduced Functionality mode unless you have entered a valid Activation Key. In Basic functionality mode, RobotStudio allows only the use of the Online and basic Virtual Controller features. No existing files or stations are harmed in Reduced Functionality mode. After activating your software, you will have full functionality for the features you have purchased.

**NOTE!**

Activation is not required for the Online features for programming, configuring and monitoring a real controller connected over Ethernet.

### Activate automatically over the Internet or manually

The Activation Wizard gives you two choices on how to proceed.

**Automatic activation by using the Internet (recommended)**

Once you have selected the option *Activate RobotStudio over the Internet*, the Activation Wizard automatically contacts the ABB licensing servers over your Internet connection. If you are using a valid Activation Key that has not exceeded the number of installations allowed, your product is activated immediately.

When you activate over the Internet, your activation request is sent to ABB. Your license will then be automatically installed and your product ready for use. If you choose to activate over the Internet but are not currently connected, the wizard alerts you that there is no connection.

**Manual activation**

If the computer does not have an Internet connection, you must create a license file by selecting the option *Create a license request file*. Proceed through the wizard, enter your Activation Key and save the License Request File to your computer. Use a removable medium, such as a USB stick or floppy disk, to transfer the file to a computer with an Internet connection. Go to [www.robotstudio.com/community](http://www.robotstudio.com/community) to find a hyperlink to the manual activation page, click the link and then follow the instructions. The result will be a License File that should be saved and transferred back to the computer holding your product. Relaunch the Activation Wizard and select the option *Install a license file*. Proceed through the wizard, selecting the License File when requested. Upon completion, RobotStudio is activated and ready for use.

### How do I activate later?

If you do not want to activate your copy of the software at installation, you can do so later. The following steps will launch the Activation Wizard.

If you have a problem with your activation, contact your local ABB customer support representative at the e-mail address or telephone number provided at [www.abb.com/robotics](http://www.abb.com/robotics).

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Click <strong>RobotStudio Button</strong>, then click <strong>RobotStudio Options</strong> button and select <strong>Licensing</strong>.</td>
</tr>
<tr>
<td>2. Click <strong>Activation wizard</strong> to launch the activation wizard.</td>
</tr>
<tr>
<td>3. Proceed through the wizard to complete the activation. <strong>NOTE!</strong> If the RobotStudio installation is activated, you will have valid licenses for the features covered by your subscription.</td>
</tr>
</tbody>
</table>

Continues on next page
1 Introduction

1.2. Installing and Licensing RobotStudio

Continued

Which RobotStudio version is installed?

The version number of RobotStudio is displayed on the start page that appears when RobotStudio is started.

How can I tell if my RobotStudio installation is activated?

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Click <strong>RobotStudio Button</strong>, then click <strong>RobotStudio Options</strong> button, and select <strong>Licensing</strong>.</td>
</tr>
<tr>
<td>2. Click <strong>View Installed License Keys</strong> to see the status of your current license.</td>
</tr>
<tr>
<td>3. If the RobotStudio installation is activated, you will have valid licenses for the features covered by your subscription.</td>
</tr>
</tbody>
</table>
1.3 The Graphical User Interface

1.3.1. The Getting Started window

Overview

The Getting Started window contains the following tabs:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recent Stations</td>
<td>Use this tab to open recent stations (which are previewed), a new station or any saved station.</td>
</tr>
<tr>
<td>Offline</td>
<td></td>
</tr>
<tr>
<td>Recent Controllers</td>
<td>Use this tab to open a recent controller, connect to a controller with one click or add a controller.</td>
</tr>
<tr>
<td>Online</td>
<td></td>
</tr>
<tr>
<td>Information</td>
<td>Use this tab to access help files, browse tutorials, connect to the online community, manage your licenses or browse RobotStudio news.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1.3.2. The Layout browser

**Overview**

The layout browser is a hierarchical display of physical items, such as robots and tools.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Node</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Robot</td>
<td>The robot in the station. The red lock in the lower right corner of the icon indicates that the object is connected to a library.</td>
</tr>
<tr>
<td></td>
<td>Tool</td>
<td>A tool.</td>
</tr>
<tr>
<td></td>
<td>Link collection</td>
<td>Contains all the links of the objects.</td>
</tr>
<tr>
<td></td>
<td>Link</td>
<td>A physical object in a joint connection. Each link is made up of one or several parts.</td>
</tr>
<tr>
<td></td>
<td>Frames</td>
<td>Contains all the frames for an object.</td>
</tr>
<tr>
<td></td>
<td>Component group</td>
<td>A grouping of parts or other assemblies, carrying its own coordinate systems. It is used to structure a station.</td>
</tr>
<tr>
<td></td>
<td>Part</td>
<td>A physical object in RobotStudio. Parts with geometric information are made up of one or more 2D or 3D entities. Parts without geometric information (such as imported .jt files) are empty.</td>
</tr>
<tr>
<td></td>
<td>Collision set</td>
<td>Contains all collision sets. Each collision set includes two groups of objects.</td>
</tr>
<tr>
<td></td>
<td>Objects group</td>
<td>Contains references to the objects that are subject to collision detection.</td>
</tr>
<tr>
<td></td>
<td>Collision set mechanisms</td>
<td>The objects in the collision set.</td>
</tr>
<tr>
<td></td>
<td>Frame</td>
<td>The frames in the station.</td>
</tr>
</tbody>
</table>
1.3.3. The Paths & Targets browser

Overview
The paths & targets browser is a hierarchical display of non-physical items.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Node</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Station</td>
<td>Your station in RobotStudio.</td>
</tr>
<tr>
<td></td>
<td>Virtual Controller</td>
<td>The system for controlling the robots, just like a real IRC5 controller.</td>
</tr>
<tr>
<td></td>
<td>Task</td>
<td>Contains all logical elements in the station, such as targets, paths, workobjects, tooldata and instructions.</td>
</tr>
<tr>
<td></td>
<td>Tooldata Collection</td>
<td>Contains all tooldata.</td>
</tr>
<tr>
<td></td>
<td>Tooldata</td>
<td>A tooldata for a robot or a task.</td>
</tr>
<tr>
<td></td>
<td>Workobjects &amp; Targets</td>
<td>Contains all workobjects and targets for the task or robot.</td>
</tr>
<tr>
<td></td>
<td>Jointtarget Collection and Jointtarget</td>
<td>A specified position of the robot axes.</td>
</tr>
<tr>
<td></td>
<td>Workobject Collection and Workobject</td>
<td>The workobject collection node and the workobjects it contains.</td>
</tr>
<tr>
<td></td>
<td>Target</td>
<td>A defined position and rotation for a robot. A target equals a RobTarget in a RAPID program.</td>
</tr>
<tr>
<td></td>
<td>Target without assigned configuration</td>
<td>A target for which no axis configuration has been assigned, for example, a repositioned target or a new target created by means other than teaching.</td>
</tr>
<tr>
<td></td>
<td>Target without found configuration</td>
<td>An unreachable target, that is, for which no axis configuration has been found.</td>
</tr>
<tr>
<td></td>
<td>Path Collection</td>
<td>Contains all paths in the station.</td>
</tr>
</tbody>
</table>
## 1.3.3. The Paths & Targets browser

*Continued*

<table>
<thead>
<tr>
<th>Icon</th>
<th>Node</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="xd000001381" alt="Path Icon" /></td>
<td>Path</td>
<td>Contains the instructions for the robot movements.</td>
</tr>
<tr>
<td><img src="xd000001474" alt="Linear Move Instruction Icon" /></td>
<td>Linear Move Instruction</td>
<td>A linear TCP motion to a target. If the target has no valid configuration assigned, the move instruction gets the same warning symbols as the target.</td>
</tr>
<tr>
<td><img src="xd000001851" alt="Joint Move Instruction Icon" /></td>
<td>Joint Move Instruction</td>
<td>A joint motion to a target. If the target has no valid configuration assigned, the move instruction gets the same warning symbols as the target.</td>
</tr>
<tr>
<td><img src="xd000001475" alt="Action Instruction Icon" /></td>
<td>Action Instruction</td>
<td>Defines an action for the robot to perform at a specified location in a path.</td>
</tr>
</tbody>
</table>
1.3.4. The Modeling browser

Overview

The modeling browser is a display of editable objects and their building blocks.

Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Node</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="xx0600002704" alt="Part" /></td>
<td>Part</td>
<td>Geometric items corresponding to the objects in the Layout browser.</td>
</tr>
<tr>
<td><img src="xx0600002705" alt="Body" /></td>
<td>Body</td>
<td>Geometric building blocks that comprise the parts. 3D bodies contain several faces, 2D bodies one face, and curves no faces.</td>
</tr>
<tr>
<td><img src="xx0600002706" alt="Face" /></td>
<td>Face</td>
<td>The faces of the bodies.</td>
</tr>
</tbody>
</table>
1 Introduction

1.3.5. The Offline and Online browsers

Overview

The offline and online browsers is a hierarchical display of controller and configuration elements.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Node</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="xx0300000026" /></td>
<td>Controllers</td>
<td>Contains the controllers that are connected to the Robot View.</td>
</tr>
<tr>
<td><img src="image2" alt="xx0300000027" /></td>
<td>Connected Controller</td>
<td>Represents a controller with a working connection.</td>
</tr>
<tr>
<td><img src="image3" alt="xx0400000077" /></td>
<td>Connecting Controller</td>
<td>Represents a controller which is currently being connected.</td>
</tr>
<tr>
<td><img src="image4" alt="xx0300000028" /></td>
<td>Disconnected Controller</td>
<td>Represents a controller that has lost its connection. It might have been turned off or disconnected from the network.</td>
</tr>
<tr>
<td><img src="image5" alt="xx0600003287" /></td>
<td>Denied login</td>
<td>Represent a controller that denies you access to login. Possible reasons for denied access:</td>
</tr>
<tr>
<td><img src="image6" alt="xx0300000029" /></td>
<td>Configuration</td>
<td>Contains the configuration topics.</td>
</tr>
<tr>
<td><img src="image7" alt="xx0300000030" /></td>
<td>Topic</td>
<td>Each parameter topic is represented by a node:</td>
</tr>
<tr>
<td><img src="image8" alt="xx0300000037" /></td>
<td>Event Log</td>
<td>With the Event Log you can view and save controller events.</td>
</tr>
<tr>
<td><img src="image9" alt="xx0300000043" /></td>
<td>I/O System</td>
<td>Represents the controller I/O system. The I/O system consists of I/O buses and units.</td>
</tr>
</tbody>
</table>

Continues on next page
## 1 Introduction

### 1.3.5. The Offline and Online browsers

<table>
<thead>
<tr>
<th><strong>I/O Bus</strong></th>
<th>An I/O bus is a connector for one or several I/O units.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I/O Unit</strong></td>
<td>An I/O Unit is a board, panel or any other device with ports through which the I/O signals are sent.</td>
</tr>
<tr>
<td><strong>RAPID Tasks</strong></td>
<td>Contains the active tasks (programs) of the controller.</td>
</tr>
<tr>
<td><strong>Task</strong></td>
<td>A task is a robot program, which executes alone or together with other programs. A program is composed of a set of modules.</td>
</tr>
<tr>
<td><strong>Modules</strong></td>
<td>A container for either program modules or system modules.</td>
</tr>
<tr>
<td><strong>Program Module</strong></td>
<td>A program module contains a set of data declarations and routines for a specific task. Program modules contain data specific for this program.</td>
</tr>
<tr>
<td><strong>System Module</strong></td>
<td>A system module contains a set of type definitions, data declarations and routines. System modules contain data that applies to the robot system, regardless which program modules that are loaded.</td>
</tr>
<tr>
<td><strong>Encrypted Module</strong></td>
<td>A module that is locked for editing and viewing.</td>
</tr>
<tr>
<td><strong>Nostepin Module</strong></td>
<td>A module that cannot be entered during step-by-step execution. That is all instructions in the module are treated as one if the program is executed step-by-step.</td>
</tr>
<tr>
<td><strong>Procedure</strong></td>
<td>A routine that does not return any value. Procedures are used as subprograms.</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>A routine that returns a value of a specific type.</td>
</tr>
<tr>
<td><strong>Trap</strong></td>
<td>A routine that provides a means of responding to interrupts.</td>
</tr>
<tr>
<td><strong>Documents Folder</strong></td>
<td>Folder for links to documents and files.</td>
</tr>
</tbody>
</table>

*Continues on next page*
1 Introduction

1.3.5. The Offline and Online browsers

Continued

<table>
<thead>
<tr>
<th>Document Link</th>
<th>Link to a document. The icon for the document is the same as the Window icon for the document type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>xx03000000038</td>
<td></td>
</tr>
</tbody>
</table>
1.3.6. The Output window

Overview

The output window displays information about events that occur in the station, such as when simulations are started or stopped. This information is useful when troubleshooting stations.

Layout of the Output tab

The Output tab contains two columns: the first states the event, the second the time the message was generated. Each row is a message.

Event types

The three event types indicate the severity of the event:

<table>
<thead>
<tr>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>An information message is a normal system event, for example, starting and stopping programs, changing the operational mode, and turning motors on and off. Information messages never require an action from you. They can be useful for tracking errors, collecting statistics or monitoring user-triggered event routines.</td>
</tr>
<tr>
<td>Warning</td>
<td>A warning is an event of which you should be aware, but it is not so severe that the process or RAPID program needs to be stopped. Warnings must occasionally be acknowledged. Warnings often indicate underlying problems that at some point will need to be resolved.</td>
</tr>
<tr>
<td>Error</td>
<td>An error is an event that prevents the robot system from proceeding. The running process or RAPID program cannot continue and will be stopped. An error must occasionally be acknowledged. Some errors require some immediate action from you in order to be resolved. Double-click an error to display a detailed information box.</td>
</tr>
</tbody>
</table>

Some of the events are active. These are linked to an action for resolving the problem that generated the event. To activate the linked action, double-click the message.

Handling messages in the Output window

<table>
<thead>
<tr>
<th>Goal</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>To filter messages...</td>
<td>Right-click in the Output window and then click Show messages. From the options All Errors, Information, Warnings and Warnings and Errors, select the type of messages you want to display.</td>
</tr>
<tr>
<td>To save a message to file...</td>
<td>Select it, right-click and then click Save to file. Choose a name and location in the dialog box. Multiple messages can be selected by pressing SHIFT while clicking each.</td>
</tr>
<tr>
<td>To clear the Output window...</td>
<td>Right-click in the Output window and then click Clear.</td>
</tr>
</tbody>
</table>

Continues on next page
1.3.6. The Output window

The Controller tab

The controller status window shows the operational status of the controllers in your robot view.

1. The System Name column
   This column shows the name of the system that is running on the controller.

2. The Controller Name column
   This column shows the name of the controller.

3. The Controller State column
   This column shows the state of the controller:

   **When the controller is in state...** | **the robot is...**
   -- | --
   Initializing | starting up. It will shift to state motors off when it has started.
   Motors off | in a standby state where there is no power to the robot’s motors. The state has to be shifted to motors on before the robot can move.
   Motors on | ready to move, either by jogging or by running programs.
   Guard Stop | stopped because the safety runchain is opened. For instance, a door to the robot’s cell might be open.
   Emergency Stop | stopped because emergency stop was activated.
   Waiting for motors on after e-stop | ready to leave emergency stop state. The emergency stop is no longer activated, but the state transition isn’t yet confirmed.
   System Failure | in a system failure state. A warm start is required.

4. The Program Execution State column
   This column shows if the robot is running any program or not:

   **When the controller is in state...** | **the robot...**
   -- | --
   Running | is running a program.
   Ready | has a program loaded and is ready to run it when a PP (starting point in the program) has been set.
   Stopped | has a program loaded, with a PP, and is ready to run it.
   Uninitialized | has not initialized the program memory. This indicates an error condition.

5. The Operating Mode column
   This column shows the operating mode of the controller:

   **When the controller is in mode...** | **the robot is...**
   -- | --
   Initializing | starting up. It will shift to the mode selected on the controllers cabinet when it has started.
   Auto | ready to run programs in production. In Auto mode it is possible to get remote Write access to the controller, which is necessary for editing programs, configurations and other things with RobotStudio Online.
6. The **Logged on as** column

   This column shows the user name the PC is logged on to the controller with.

7. The **Access** column

   This column shows who has write access to the controller, or if it is available.

8. The **Connection Type** column

   This column shows the type of connection to the controller.

<table>
<thead>
<tr>
<th>When the controller is in mode...</th>
<th>the robot is...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>only able to move if the enabling device on the FlexPendant is activated. Furthermore, the robot can only moved with reduced speed in manual mode. In manual mode it is not possible to get remote Write access to the controller, unless it is configured for this and the remote Write access granted on the FlexPendant.</td>
</tr>
<tr>
<td>Manual full speed</td>
<td>only able to move if the enabling device on the FlexPendant is activated. In manual mode it is not possible to get remote Write access to the controller, unless it is configured for this and the remote Write access granted on the FlexPendant.</td>
</tr>
<tr>
<td>Waiting for acknowledge</td>
<td>about to enter Auto mode, but the mode transition has not yet been acknowledged.</td>
</tr>
</tbody>
</table>
1 Introduction

1.3.7. Operator Window

1.3.7. Operator Window

Overview

Operator Window is an alternative to the corresponding feature in the Virtual FlexPendant for communicating with the user during RAPID program execution. It displays the same output as displayed on the Virtual FlexPendant Operator Window.

When running in a Virtual Controller, the RAPID program communicates with the operator via messages on the FlexPendant screen. The Operator Window integrates this functionality and allows the user to run interactive RAPID programs without starting the Virtual FlexPendant.

Enabling Operator Window

To enable an operator window:

1. On the Application menu, click RobotStudio Options.
2. On the Navigation pane to the left, select Robotics:Virtual Controller.
3. On the Virtual Controller page to the right, select Show virtual Operator Window option.
4. Click Apply.

NOTE!

When the Show virtual Operator Window feature is enabled, an Operator Window is automatically created for each controller in the station. By default, the window is located in the tab area below the graphics window.

RAPID Instructions

List of RAPID instructions supported by the Operator Window:

• TPErase
• TPReadFK
• TPReadNum
• TPWrite
• UIAlphaEntry
• UIMsgBox
• UINumEntry

NOTE! When these instructions are executed, the behavior is similar to that of Virtual FlexPendant.

List of RAPID instructions not supported by the Operator Window:

• TPShow
• UIShow
• UINumTune
• UILListView

NOTE! When these instructions are executed, an error message is displayed in the Operator Window prompting you to use the Virtual FlexPendant instead.

Continues on next page
NOTE!

You should not run both the Virtual Flexpendant and Operator Window simultaneously.
1.3.8. Using a mouse

Navigating the graphics window using the mouse

The table below shows how to navigate the graphics window using the mouse:

<table>
<thead>
<tr>
<th>To</th>
<th>Use the keyboard / mouse combination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select items</td>
<td>xx0500002417 xx0500002421</td>
<td>Just click the item to select. To select multiple items, press CTRL key while clicking new items.</td>
</tr>
<tr>
<td>Rotate the station</td>
<td>CTRL + SHIFT + xx0500002424 xx0500002421</td>
<td>Press CTRL + SHIFT + the left mouse button while dragging the mouse to rotate the station. With a 3-button mouse you can use the middle and right buttons, instead of the keyboard combination.</td>
</tr>
<tr>
<td>Pan the station</td>
<td>CTRL + xx0500002422</td>
<td>Press CTRL + the left mouse button while dragging the mouse to pan the station.</td>
</tr>
<tr>
<td>Zoom the station</td>
<td>CTRL + xx0500002426</td>
<td>Press CTRL + the right mouse button while dragging the mouse to the left to zoom out. Dragging to the right zooms in. With a 3-button mouse you can also use the middle button, instead of the keyboard combination.</td>
</tr>
<tr>
<td>Zoom using window</td>
<td>SHIFT + xx0500002425</td>
<td>Press SHIFT + the right mouse button while dragging the mouse across the area to zoom into.</td>
</tr>
<tr>
<td>Select using window</td>
<td>SHIFT + xx0500002428</td>
<td>Press SHIFT + the left mouse button while dragging the mouse across the area to select all items that match the current selection level.</td>
</tr>
</tbody>
</table>
1.3.9. Selecting an item

Overview

Each item in a station can be moved to achieve the required layout, so you first have to determine its selection level. The selection level makes it possible to select only specific types of items, or specified parts of objects.

The selection levels are curve, surface, entity, part, mechanism, group, target/frame and path. The target/frame and path selection can be combined with any of the other selection levels. Objects may also be grouped together as component groups, see Component Group on page 226.

Selecting an item in the graphics window

To select items in the graphics window, follow these steps:
1. At the top of the graphics window, click the desired selection level icon.
2. Optionally, click the desired snap mode icon for the part of the item you wish to select.
3. In the graphics window, click the item. The selected item will be highlighted.

Multiple selection of items in the graphics window

To select multiple items in the graphics window, do the following:
1. Press the SHIFT key, and in the graphics window drag the mouse diagonally over the objects to select.

Selecting an item in the browsers

To select items in a browser, do the following:
1. Click the item. The selected item will be highlighted in the browser.

Multiple selection of items in the browsers

To select multiple items in a browser, follow these steps:
1. Make sure that all the items to be selected are of the same type and located in the same branch of the hierarchical structure; otherwise, the items will not be operable.
2. Do one of the following:
   • To select adjacent items: In the browser, hold down the SHIFT key and click the first and then the last item. The list of items will be highlighted.
   • To select separate items: In the browser, hold down the CTRL key and click the items you want to select. The items will be highlighted.
1 Introduction

1.3.10. Attaching and detaching objects

1.3.10. Attaching and detaching objects

Overview

You can attach an object (child) to another object (parent). Attachments can be created on part level and on mechanism level. When an object has been attached to a parent, moving the parent also moves the child.

One of the most common attachments is to attach a tool to a robot. For procedures, see Attach to on page 323 and Detach on page 331.
# 1.3.11. Keyboard Shortcuts

## Keyboard Shortcuts

The following table lists the keyboard shortcuts:

<table>
<thead>
<tr>
<th>Command</th>
<th>Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Activate menu bar</td>
<td>F10</td>
</tr>
<tr>
<td>Open API Help</td>
<td>ALT + F1</td>
</tr>
<tr>
<td>Open Help</td>
<td>F1</td>
</tr>
<tr>
<td>Open Virtual FlexPendant</td>
<td>CTRL + F5</td>
</tr>
<tr>
<td>Switch between windows</td>
<td>CTRL + TAB</td>
</tr>
<tr>
<td><strong>General Commands</strong></td>
<td></td>
</tr>
<tr>
<td>Add Controller System</td>
<td>F4</td>
</tr>
<tr>
<td>Open Station</td>
<td>CTRL + O</td>
</tr>
<tr>
<td>Take Screenshot</td>
<td>CTRL + B</td>
</tr>
<tr>
<td>Teach Move Instruction</td>
<td>CTRL + SHIFT + R</td>
</tr>
<tr>
<td>Teach Target</td>
<td>CTRL + R</td>
</tr>
<tr>
<td>Toggle View Representation</td>
<td>F7</td>
</tr>
<tr>
<td><strong>General Editing</strong></td>
<td></td>
</tr>
<tr>
<td>Copy</td>
<td>CTRL + C</td>
</tr>
<tr>
<td>Cut</td>
<td>CTRL + X</td>
</tr>
<tr>
<td>Paste</td>
<td>CTRL + V</td>
</tr>
<tr>
<td>Redo</td>
<td>CTRL + Y</td>
</tr>
<tr>
<td>Refresh</td>
<td>F5</td>
</tr>
<tr>
<td>Rename</td>
<td>F2</td>
</tr>
<tr>
<td>Select All</td>
<td>CTRL + A</td>
</tr>
<tr>
<td>Undo</td>
<td>CTRL + Z</td>
</tr>
<tr>
<td>Import Geometry</td>
<td>CTRL + G</td>
</tr>
<tr>
<td>Import Library</td>
<td>CTRL + J</td>
</tr>
<tr>
<td>New Station</td>
<td>CTRL + N</td>
</tr>
<tr>
<td>Save Station</td>
<td>CTRL + S</td>
</tr>
<tr>
<td><strong>Program Editor Intellisense</strong></td>
<td></td>
</tr>
<tr>
<td>Complete Word</td>
<td>CTRL + SPACEBAR</td>
</tr>
<tr>
<td>Parameter Info</td>
<td>CTRL + I</td>
</tr>
<tr>
<td>PickList</td>
<td>CTRL + SHIFT + SPACEBAR</td>
</tr>
<tr>
<td><strong>Program Editor Commands</strong></td>
<td></td>
</tr>
<tr>
<td>Start Program Execution</td>
<td>F8</td>
</tr>
<tr>
<td>Step In</td>
<td>F11</td>
</tr>
<tr>
<td>Step Out</td>
<td>SHIFT + F11</td>
</tr>
<tr>
<td>Step Over</td>
<td>F12</td>
</tr>
<tr>
<td>Stop</td>
<td>SHIFT + F8</td>
</tr>
</tbody>
</table>

*Continues on next page*
## 1.3.11. Keyboard Shortcuts

Continued

<table>
<thead>
<tr>
<th>Command</th>
<th>Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toggle Breakpoint</td>
<td>F9</td>
</tr>
<tr>
<td>Apply Changes</td>
<td>CTRL + SHIFT + S</td>
</tr>
<tr>
<td>Print</td>
<td>CTRL + P</td>
</tr>
</tbody>
</table>

### Program Editor Hotkeys

<table>
<thead>
<tr>
<th>Command</th>
<th>Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>CTRL + Insert</td>
</tr>
<tr>
<td>Cut</td>
<td>SHIFT + Delete</td>
</tr>
<tr>
<td>Cut marked lines</td>
<td>CTRL + L</td>
</tr>
<tr>
<td>Delete marked lines</td>
<td>CTRL + SHIFT + L</td>
</tr>
<tr>
<td>Delete to beginning of word</td>
<td>CTRL + BACKSPACE</td>
</tr>
<tr>
<td>Delete to end of word</td>
<td>CTRL + Delete</td>
</tr>
<tr>
<td>Indent</td>
<td>Tab</td>
</tr>
<tr>
<td>Make the selected text lowercase</td>
<td>CTRL + U</td>
</tr>
<tr>
<td>Make the selected text uppercase</td>
<td>CTRL + SHIFT + U</td>
</tr>
<tr>
<td>Move to beginning of document</td>
<td>CTRL + Home</td>
</tr>
<tr>
<td>Move to beginning of line</td>
<td>Home</td>
</tr>
<tr>
<td>Move to end of document</td>
<td>CTRL + End</td>
</tr>
<tr>
<td>Move to end of line</td>
<td>End</td>
</tr>
<tr>
<td>Move to next word</td>
<td>CTRL + Right</td>
</tr>
<tr>
<td>Move to previous word</td>
<td>CTRL + Left</td>
</tr>
<tr>
<td>Move to visible bottom</td>
<td>CTRL + Page Down</td>
</tr>
<tr>
<td>Move to visible top</td>
<td>CTRL + Page Up</td>
</tr>
<tr>
<td>Open line above</td>
<td>CTRL + Enter</td>
</tr>
<tr>
<td>Open line below</td>
<td>CTRL + SHIFT + Enter</td>
</tr>
<tr>
<td>Outdent</td>
<td>SHIFT + TAB</td>
</tr>
<tr>
<td>Paste</td>
<td>SHIFT + Insert</td>
</tr>
<tr>
<td>Redo</td>
<td>CTRL + SHIFT + Z</td>
</tr>
<tr>
<td>Scroll down</td>
<td>CTRL + Down</td>
</tr>
<tr>
<td>Scroll up</td>
<td>CTRL + Up</td>
</tr>
<tr>
<td>Select block down</td>
<td>ALT + SHIFT + Down</td>
</tr>
<tr>
<td>Select block left</td>
<td>ALT + SHIFT + Left</td>
</tr>
<tr>
<td>Select block right</td>
<td>ALT + SHIFT + Right</td>
</tr>
<tr>
<td>Select block to next word</td>
<td>CTRL + ALT + SHIFT + Right</td>
</tr>
<tr>
<td>Select block to previous word</td>
<td>CTRL + ALT + SHIFT + Left</td>
</tr>
<tr>
<td>Select block up</td>
<td>ALT + SHIFT + Up</td>
</tr>
<tr>
<td>Select down</td>
<td>SHIFT + Down</td>
</tr>
<tr>
<td>Select left</td>
<td>SHIFT + Left</td>
</tr>
<tr>
<td>Select page down</td>
<td>SHIFT + Page Down</td>
</tr>
<tr>
<td>Select page up</td>
<td>SHIFT + Page Up</td>
</tr>
<tr>
<td>Select right</td>
<td>SHIFT + Right</td>
</tr>
<tr>
<td>Select to beginning of document</td>
<td>CTRL + SHIFT + Home</td>
</tr>
<tr>
<td>Select to beginning of line</td>
<td>SHIFT + Home</td>
</tr>
<tr>
<td>Select to end of document</td>
<td>CTRL + SHIFT + End</td>
</tr>
</tbody>
</table>

*Continues on next page*
## 1.3.11. Keyboard Shortcuts

<table>
<thead>
<tr>
<th>Command</th>
<th>Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select to end of line</td>
<td>SHIFT + End</td>
</tr>
<tr>
<td>Select to next word</td>
<td>CTRL + SHIFT + Right</td>
</tr>
<tr>
<td>Select to previous word</td>
<td>CTRL + SHIFT + Left</td>
</tr>
<tr>
<td>Select to visible bottom</td>
<td>CTRL + SHIFT + Page Down</td>
</tr>
<tr>
<td>Select to visible top</td>
<td>CTRL + SHIFT + Page Up</td>
</tr>
<tr>
<td>Select up</td>
<td>SHIFT + Up</td>
</tr>
<tr>
<td>Select word</td>
<td>CTRL + SHIFT + W</td>
</tr>
<tr>
<td>Toggle overwrite mode</td>
<td>Insert</td>
</tr>
<tr>
<td>Transpose characters</td>
<td>CTRL + T</td>
</tr>
<tr>
<td>Transpose lines</td>
<td>CTRL + ALT + SHIFT + T</td>
</tr>
<tr>
<td>Transpose words</td>
<td>CTRL + SHIFT + T</td>
</tr>
</tbody>
</table>
1 Introduction

1.3.11. Keyboard Shortcuts
2 How to build stations
2 How to build stations

2.1. Workflow for building a new station

Overview
This is an overview of how to build a station by including the equipment necessary to create and simulate robot programs.

The first part of this overview describes the alternatives for creating a station with a system. The second part describes importing or creating the objects to work with. The third part describes the workflow for optimizing the station layout by finding the best placement of robots and other equipment.

In most cases, following the workflows from start to finish is recommended, even if other sequences are also possible.

Creating a station with a system
The table below shows the alternatives for creating a station with a system.

For procedures, see New Station on page 172.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a station with a template system</td>
<td>This is the simplest way to create a new station containing a robot and a link to a rudimentary system template.</td>
</tr>
<tr>
<td>Create a station with an existing system</td>
<td>This creates a new station containing one or more robots in accordance with an existing, built system.</td>
</tr>
<tr>
<td>Create a station with no system</td>
<td>An advanced user can build a station from scratch and then add a new or existing system to it.</td>
</tr>
</tbody>
</table>

Manually starting the VC
The table below shows the alternatives for manually starting with a system. Perform only those steps applicable to your station.

For procedures, see Starting a VC on page 64.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manually connecting a library to the VC</td>
<td>See Starting a VC on page 64.</td>
</tr>
<tr>
<td>Restarting the VC</td>
<td>See Restarting a VC on page 66.</td>
</tr>
</tbody>
</table>

Importing station components
The table below shows the workflow for importing station components. Perform only those steps applicable to your station.

For procedures, see Importing a station component on page 68.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import a robot model</td>
<td>See Robot System on page 188.</td>
</tr>
<tr>
<td>Import a tool</td>
<td>See Import Library on page 187.</td>
</tr>
<tr>
<td>Import a positioner</td>
<td>See ABB Library on page 186.</td>
</tr>
<tr>
<td>Import a track</td>
<td>See Import Library on page 187.</td>
</tr>
<tr>
<td>Import other equipment</td>
<td>If you have CAD models of the equipment, you can import them, see Import Library on page 187. Otherwise, you can create models in RobotStudio, see Mechanisms on page 75.</td>
</tr>
</tbody>
</table>
Placing objects and mechanisms

The table below shows the workflow for placing the objects in the station.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add work piece</td>
<td>If you have CAD models of the work piece, you can import them, see Workobject on page 195. Otherwise, you can create models in RobotStudio, see Objects on page 73.</td>
</tr>
<tr>
<td>Place objects</td>
<td>If you are building a model of a real station, start by placing all objects with known positions. For objects without known positions, find a suitable placement, see Placing objects and Placing external axes on page 79.</td>
</tr>
<tr>
<td>Attach tools</td>
<td>Attach the tools to the robot, see Attach to on page 323.</td>
</tr>
<tr>
<td>Attach robots to tracks</td>
<td>If track external axes are used, attach the robots to the tracks, see Attach to on page 323.</td>
</tr>
<tr>
<td>Attach work pieces to positioners</td>
<td>If positioner external axes are used, attach the work pieces to the positioners, see Attach to on page 323.</td>
</tr>
<tr>
<td>Test reachability</td>
<td>Test if the robot can reach critical positions on the work piece. If you are satisfied with how the robot reaches the positions, your station is ready for programming. Otherwise, continue adjusting the placement or trying other equipment as described below, see Testing positions and motions on page 100.</td>
</tr>
</tbody>
</table>
2 How to build stations

2.2 Setting up a conveyor tracking station with two robots working on the same conveyor

2.2.1. Two robot systems sharing the same task frame position

Overview

This section describes what happens when both the robot systems share the same task frame position. The baseframes of the mechanical units in both the robot systems have the same task frame position.

Prerequisite

- Two robot systems with conveyor tracking option (system 1 and system 2)
- A conveyor mechanism saved as library

NOTE! See Create Conveyor mechanism on page 247 to create conveyor tracking systems.

Setting up the conveyor tracking station

1. Add the existing system (system 1) to the station. See Robot System on page 188.

   NOTE! After starting the system, when asked to select the library, browse and select the already saved conveyor mechanism library.

2. Modify the baseframe positions of conveyor and robot.
   1. Move the mechanical unit (conveyor/robot) to its new location.
   2. See Updating the baseframe position on page 294 to update the baseframe position of the conveyor/robot.
   3. Repeat steps 1 and 2 to modify the baseframe position of the robot.
   4. In the System Configuration window, click OK. When asked if you want to restart the system, answer Yes. Close the System Configuration window.

3. Add the existing system (system 2) to the station. See Robot System on page 188.

   NOTE! After starting the system, when asked to select the library, browse and select the same library as the one selected for system 1 or any other library. Later, this conveyor library will be removed from the station since system 2 shall use the same conveyor library as system 1.

4. Refer both systems (system 1 and system 2) to the same conveyor library.
   1. In the Offline browser, click System Configuration to bring up a dialog box for system 2.
   2. Select the library node in the hierarchical tree.
   3. Select the option Select from Station. Click Change. The Select Library dialog box appears.
   4. Select the same conveyor library as the one selected for system 1. Click OK.

Continues on next page
2.2.1. Two robot systems sharing the same task frame position

**NOTE!** Now both systems (system 1 and system 2) use the same conveyor library and the library previously referenced by system 2 is removed from the station.

5. Modify the baseframe positions of robot (system 2).
   1. Move the mechanical unit (robot) to its new location.
   2. See *Updating the baseframe position on page 294* to update the baseframe position of the robot.
   3. Repeat steps 1 and 2 to modify the baseframe position of the robot.
   4. In the **System Configuration** window, click **OK**. When asked if you want to restart the system, answer **Yes**. Close the **System Configuration** window.

Continued
2 How to build stations

2.2.2. Two robot systems having different task frame positions

2.2.2. Two robot systems having different task frame positions

Overview

This section describes what happens when two robot systems have different task frame positions but uses the same sync switch. This means the Baseframes of the conveyor mechanical units in both the robot systems have different values.

Prerequisites

Two robot systems with conveyor tracking option (system 1 and system 2)

NOTE! See Create Conveyor mechanism on page 247 to create conveyor tracking systems.

Setting up the conveyor tracking station

1. Add the existing system (system 1) to the station. See Robot System on page 188.

   NOTE! After starting the system, when asked to select the library, browse and select the already saved conveyor mechanism library.

2. Modify the baseframe positions of conveyor and robot.

   1. Move the mechanical unit (conveyor/robot) to its new location.
   2. See Updating the baseframe position on page 294 to update the baseframe position of the conveyor/robot.
   3. Repeat steps 1 and 2 to modify the baseframe position of the robot.
   4. In the System Configuration window, click OK. When asked if you want to restart the system, answer Yes. Close the System Configuration window.

3. Add the existing system (system 2) to the station. See Robot System on page 188.

   NOTE! After starting the system, when asked to select the library, browse and select the same library as the one selected for system 1 or any other library. Later, this conveyor library will be removed from the station since system 2 shall use the same conveyor library as system 1.

4. Update both systems (system 1 and system 2) to use the same conveyor library.

   1. In the Offline browser, click System Configuration to bring up a dialog box for system 2.
   2. Select the library node in the hierarchical tree.
   3. Select the option Select from Station. Click Change. The Select Library dialog box appears.
   4. Select the same conveyor library as the one selected for system 1. Click OK.

   NOTE! Now both systems (system 1 and system 2) use the same conveyor library and the library previously referenced by system 2 is removed from the station.

5. Modify the task frame position of the conveyor mechanism. See Set Task Frames on page 293.

   NOTE! Before modifying the task frame, make a note of the current conveyor position in world coordinates. After modifying the task frame, move the conveyor back to the position it was before modifying the task frame.

Continues on next page
6. Modify the baseframe positions of robot (system 2).
   Repeat step 2 to modify the baseframe position of the robot (system 2)
   1. Move the mechanical unit (robot) to its new location.
   2. See *Updating the baseframe position on page 294* to update the baseframe position of
      the robot.
   3. Repeat steps 1 and 2 to modify the baseframe position of the robot.
   4. In the **System Configuration** window, click **OK**. When asked if you want to restart
      the system, answer **Yes**. Close the **System Configuration** window.

7. Modify the baseframe position of the conveyor (system 2).
   1. In the **Offline** browser, click **System Configuration** to bring up a dialog box for
      system 2.
   2. Select the conveyor in the hierarchical tree. The BaseFrame property list for the
      conveyor is now displayed.
   3. Select the option **Use Current Station Values** to update the baseframe value of the
      robot in the controller.
   4. Deselect the option **Check BaseFrame on Startup**.
   5. In the **System Configuration** window, click **OK**. When asked if you want to restart
      the system, answer **Yes**.

**NOTE!** By deselecting the option **Check BaseFrame on Startup**, RobotStudio will not
compare the BaseFrame values in the station and the controller every time the controller
is started. This avoids repositioning the conveyor library.

**NOTE!** If the two robot systems use the same part on the conveyor, the relation between
the part and the two conveyor workobjects should be the same.
2 How to build stations

2.3. Creating a system with external axes automatically

2.3. Creating a system with external axes automatically

**Automatically create a system with external axes**

1. Create a robot system from layout. See *Robot System on page 188*.

2. Import the desired robots, positioners, and track libraries while running through the wizard. See *Import Library on page 187*.

**NOTE!** If a robot and track are selected, attach the robot to the track. See *Attach to on page 323*.

**NOTE!** Robot system supports the following tracks with lengths 1.7 m to 19.7 m in a separate task or same robot task. Depending on the manipulator type, the system allows one to three tracks per task. However with IRBTx004, only one track of this type can be used per system.

- IRBT4003
- IRBT4004
- IRBT6003
- IRBT6004
- IRBT7003
- IRBT7004
- RTT_Bobin
- RTT_Marathon
- Paint Rail

**Supported external axes configuration**

The following table shows a combination of different external axes configurations:

<table>
<thead>
<tr>
<th>Combination</th>
<th>Positioner type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>One IRB (Positioner in same task)</td>
<td>Y</td>
</tr>
<tr>
<td>One IRB (Positioner in separate task)</td>
<td>Y</td>
</tr>
<tr>
<td>Two IRB (Positioner in separate task)</td>
<td>Y</td>
</tr>
<tr>
<td>One IRB on Track Motion (Positioner in same task)</td>
<td>Y</td>
</tr>
<tr>
<td>One IRB on Track Motion (Positioner in separate task)</td>
<td>Y</td>
</tr>
<tr>
<td>Two IRB on Track Motion (Positioner in separate task)</td>
<td>Y</td>
</tr>
</tbody>
</table>

- **Y** - Combination is supported
- **N** - Combination is not supported
- **YX** - Combination is supported and manual mapping of mechanical units and joints required

Continues on next page
2 How to build stations

2.3. Creating a system with external axes automatically

Continued

Manual mapping of mechanical units and joints

If the system contains more than one mechanical unit, the number of tasks and base frame positions of the mechanism should be verified in the System Configuration.

1. In the Offline browser, click System Configuration to bring up a dialog box.

2. Select the robot from the node in the hierarchical tree.
   The property page of this node contains controls for mapping and setting axes and joints.

3. Click Change to open a dialog box.

4. Manually map the mechanical unit and mechanism joints. Click Apply.

5. Modify the baseframe positions of the mechanical unit. See Updating the baseframe position on page 294.
2.4.1. Setting up a system with track motion of type RTT or IRBTx003 manually

Manually set up a system with track motion of type RTT or IRBTx003

Use this procedure to manually set up a system with track motion type RTT Bobin, RTT Marathon or IRBT4003, IRBT6003, or IRBT7003.

1. Build and start a new system. See Building a new system on page 142.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select the desired robot variant (IRB6600). On the Modify Options page of the System Builder, scroll down to Drive Module 1 &gt; Drive module application group and expand ABB Standard manipulator option and select Manipulator type (IRB6600).</td>
</tr>
<tr>
<td>2</td>
<td>Select Additional axes configuration. On the Modify Options page of the System Builder, scroll down to Drive Module 1 &gt; Additional axes configuration group and expand the Add axes IRB/drive module 6600 option and select the 770-4 Drive W in pos Y2 option. NOTE! The option 770-4 Drive W in pos Y2, the Drive module, and the Position varies depending on the Additional axes configuration selected. Make sure to select at least one drive in any position.</td>
</tr>
<tr>
<td>3</td>
<td>Click Finish. Close the Modify Options page.</td>
</tr>
</tbody>
</table>

2. Add the system to the station. See Adding a system on page 65

3. Add the corresponding track configuration file of the desired robot variant (IRB 6600) and the desired track model to the station. See Adding the track to the system on page 68.

NOTE! In the Select Library group, select either the existing track or import a different track

NOTE! The system may fail unless the correct additional axes configuration is selected.

4. Specify whether the baseframe is moved by another mechanism.

   1. In the Offline browser, click System Configuration to bring up a dialog box.
   2. Select ROB_1 node from the hierarchical tree.
   3. Select the option Track from the BaseFrame moved by list.
   4. Click OK. When asked if you want to restart the system, answer Yes. Close the System Configuration window.
2.4.2. Setting up a system with track motion of type IRBTx004 manually

Overview

For configuration of tracks IRBT4004, IRBT6004 or IRBT7004, the additional option mediapools are installed in the same Mediapool folder as that of RobotWare. Search path \ProgramFiles\ABB Industrial IT\Robotics IT\MediaPool\.

The following three Track mediapool versions are installed in RobotStudio:

- Track.5.09.0012, supports RobotWare 5.09
- Track.5.10.0003 and Track.5.10.0005, supports RobotWare 5.10
- Track.5.11.0001, supports RobotWare 5.11

Manually set up a system with track motion of type IRBTx004

1. Build and start a new system. See *Building a new system on page 142*.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add additional options for IRBTx004. See <em>Adding additional options on page 143</em>. <strong>NOTE!</strong> Browse and select the key file (.kxt) located in the mediapool Track 5.11.0001.</td>
</tr>
</tbody>
</table>

| 2      | Select the desired robot variant (IRB6600). On the Modify Options page of the System Builder, scroll down to Drive Module 1 > Drive module application group and expand ABB Standard manipulator option and select Manipulator type (IRB6600). |

| 3      | Select Additional axes configuration. On the Modify Options page of the System Builder, scroll down to Drive Module 1> Additional axes configuration group and expand the Add axes IRB/drive module 6600 option and select the 770-4 Drive W in pos Y2 option. **NOTE!** The option 770-4 Drive W in pos Y2, the Drive module, and the Position varies depending on the Additional axes configuration selected. Make sure to select at least one drive in any position. |

| 4      | Select the desired track motion (IRBT 6004). On the Modify Options page of the System Builder, scroll down to the TRACK and expand the Drive module for Track motion group. Select Drive Module 1 > Track Motion type > IRBT 6004 > Irb Orientation on Track > Standard carriage In Line > Select Track Motion Length > 1.7m (or any other variant). |

| 5      | Click Finish. Close the Modify Options page. |

2. Add the system to the station. See *Adding a system on page 65*.

3. Add the corresponding track configuration file of the desired robot variant (IRB 6600) and the desired track model to the station. See *Adding the track to the system on page 68*.

**NOTE!**

1. In the Select Library group, click Other to import a different track motion library.
2. Click OK. When asked if you want to restart the system, answer Yes. Close the System Configuration window.
2 How to build stations

2.5 The VC

2.5.1. Starting a VC

Overview

RobotStudio uses virtual controllers for running the robots. Virtual controllers can run both systems for real robots and specific virtual systems for testing and evaluation purposes. A virtual controller uses the same software as the controller to execute the RAPID program, to calculate robot motions and to handle I/O signals.

When starting a virtual controller, point out which system to run on it. Since the system contains information about the robots to use and important data such as robot programs and configurations, it is important to select the right system for the station.

Starting a VC

The table below describes the different ways a virtual controller may start:

<table>
<thead>
<tr>
<th>Startup</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic, when creating a station</td>
<td>In most cases, a VC is automatically started when you create a new station. Library files for the robots used by the system are then imported to the station.</td>
</tr>
<tr>
<td>Automatic, when adding a system to an existing station</td>
<td>If your station uses several systems or if you started with an empty station, you can add systems to an open station. Library files for the robots used by the the systems are then imported to the station.</td>
</tr>
</tbody>
</table>
| Manually, when connecting to an imported library | If you have manually imported a robot library you want to use with a system, instead of importing a new library at startup, you can connect this library to a controller.  
If you have manually imported a robot library you want to use with a system, instead of importing a new library at startup, you can connect this library to a controller.  
A library may only be connected to a single-robot system and may not be already connected to another VC. |
## 2 How to build stations

### 2.5.1. Starting a VC

**Continued**

### Adding a system

To start a system in a new virtual controller and automatically import the robots used by the system, follow these steps:

1. • If you have the Select system dialog box open, continue with the next step.
   • If you do not have the Select system dialog box open, go to the Controller menu and click Add System.

2. Depending on whether the system to start on the virtual controller is ready or not, do one of the following:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create and start a copy of an existing system</td>
<td>From the Select system pool list, select the folder on your PC that contains the system to use. Select the system to copy from the Systems Found list and click Copy. In the Copy System dialog box, enter a name for the system and click OK.</td>
</tr>
<tr>
<td>Start an existing system as is</td>
<td>From the Select system pool list, select the folder on your PC that contains the system to use.</td>
</tr>
<tr>
<td>Modify and start an existing system</td>
<td>From the Select system pool list, select the folder on your PC that contains the system to modify. Select the system to modify from the Systems Found list and click Modify. This starts the System Builder from which you make the modifications. For detailed information, see Modifying a system on page 145.</td>
</tr>
<tr>
<td>Build and start a new system</td>
<td>Click Create to start the System Builder from which you build the new system. For detailed information about creating the system, see Building a new system on page 142.</td>
</tr>
</tbody>
</table>

3. From the Systems Found list, select the system to use and click Start.
2 How to build stations

2.5.2. Restarting a VC

2.5.2. Restarting a VC

Overview

This section describes when and how to restart a VC in RobotStudio. See also Restart on page 287.

Prerequisites

The following are the prerequisites for restarting a controller:

- You must have Write access to the controller you are restarting.
- For the advanced restart methods X-start and C-start, you must have access to the controller's FlexPendant.

When to restart a virtual controller

Some operations require a restart of the controller to take effect. When working in RobotStudio, you will be notified when a restart is necessary.

Warm restart

Typically, you need to make an ordinary warm restart of a virtual controller when:

- you have changed the baseframe of any of the robots belonging to that virtual controller.
- you have changed the robot's configuration, either with the Configuration Editor or by loading new configuration files.
- you have added new options or hardware to the system.
- A system failure has occurred.

Cold restart

A cold restart is useful for restoring the system running on a virtual controller to the state it was created in. The cold restart deletes all RAPID programs, data and custom configurations that have been added to the system.

Advanced restarts

In addition to warm and cold restarts, several advanced restarts are available. See Advanced restart options on page 66.

Advanced restart options

The controller can be restarted with the following advanced restart options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-Start</td>
<td>Restarts the controller with the current system and the default settings. This method discards the changes made to the robot's configuration. It reverts the current system to the state it had when it was installed on the controller (an empty system).</td>
</tr>
</tbody>
</table>
## 2 How to build stations

### 2.5.2. Restarting a VC

Continued

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| P-Start | Restart the controller with the current system and reinstall RAPID.  
This method deletes all RAPID program modules. It can be useful if the system has changed in such a way that the programs no longer are valid, for instance if system parameters used by the program are changed. |
| X-Start | Saves the current system and starts the boot application.  
This method saves the current system, with the current settings, and starts the boot application from which you can choose a new system to start with. You can also configure the controller's network settings from the boot application. |
| C-Start | Deletes current system and start boot application.  
This method deletes the current system and starts the boot application from which you can choose a new system to start with. You can also configure the controller's network settings from the boot application. |
| B-Start | Restarts the controller with the current system and the last known good settings. This method restores changes made to the robot's configuration to a previously good state. |
2 How to build stations

2.6 Station components

2.6.1. Importing a station component

Importing a robot model

This is how to import a robot model without a controller to your station.

A robot which is not connected to a controller cannot be programmed. To import a robot connected to a virtual controller, configure a system for the robot and start it in a virtual controller, see Building a new system on page 142 and Starting a VC on page 64, respectively.

To import a robot model, in the Home tab, click Robot System and then select a robot model from the gallery.

Importing a tool

A tool is a special object, for example, an arc weld gun or a gripper, that operates on the work piece. For achieving correct motions in robot programs, the parameters of the tool have to be specified in the tool data. The most essential part of the tool data is the TCP, which is the position of the tool center point relative to the wrist of the robot (which is the same as the default tool, tool0).

When imported, the tool will not be related to the robot. So in order for the tool to move with the robot, you must attach it to the robot.

To import a tool, in the Home tab, click Tool and then select a tool from the gallery.

Importing a positioner

To import a tool, in the Home tab, click Positioner and then select a positioner from the gallery.

Adding the track to the system

To select the model of the external axis to use, follow these steps:

1. Start the system in a virtual controller, either in a new empty station or in an existing station, see Robot System on page 188.
2. In the Layout browser, select the system to add the track to.
3. On the Offline tab, click System Configuration.
4. Click **Add** to add parameters for the track to the system. Browse to the parameter file (.cfg) for the track to add and click **Open**.

If you have a specific parameter file for your track, use that one. Otherwise, parameter files for some standard tracks are delivered with the RobotStudio installation. These can be found in the folder *ABB Library/ Tracks* in RobotStudio’s installation folder. The *ABB Library* folder can also be opened from the Quick access pane at the left of the Open dialog box used for adding parameter files.

The file name of each parameter file tells which tracks it supports. The first part tells the length of the track and the second the number of tasks.

For example, the file TRACK_1_7.cfg supports all tracks with the length 1.7 meters in systems with one single task. For Multimove systems or other systems with several tasks, use the configuration file with the matching number of tasks.

For example, if the track length is 19.9 m and the robot attached to the track is connected to task 4 of the MultiMove system, then select TRACK_19_9_Task4.cfg file.

5. In the **System Configuration** window, click **OK**. When asked if you want to restart the system, answer **Yes**.

6. During the restart a list of all tracks compatible with the configuration file is displayed. Select the one to use and click **OK**.

After the restart the track appears in the station. Continue with attaching the robot to the track.

---

**Importing a library, geometry or piece of equipment**

A library component is a RobotStudio object that has been saved separately. Normally, components in a library are locked for editing.

A geometry is CAD data which you can import to use in RobotStudio. For a list of importable CAD formats, see *Libraries, geometries and CAD files on page 26*.

To import a library, geometry or piece of equipment, see *Import Library on page 187*. 
2.6.2. Converting CAD formats

Overview

A CAD converter is installed together with RobotStudio by default. In most cases you do not have to convert CAD files before importing them to RobotStudio, but the CAD converter might be useful for converting several files at once, or for converting with custom settings.

Prerequisites

Most of the file formats require separate licenses, see Libraries, geometries and CAD files on page 26 for more information.

Starting the CAD converter

Click Start menu, point to Programs, ABB Industrial IT, Robotics IT and click CAD Converter.

Converting CAD files

To convert CAD files, follow these steps:

1. Click Add files and select the files to convert. Optionally, click Add files, again to add more files from another location. Each file is now added to a row in the grid.

2. Optionally, change the suggested file name or target format by clicking that column for the file to change.

3. In the Target directory box, specify the folder in which to save the new files.

4. Optionally, click Settings and change the settings for the conversion. For details about the conversion settings, see Conversion settings on page 70.

5. Click Convert Files.

Conversion settings

The table below describes the settings for the conversion:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acis save file format</td>
<td>Select which version of ACIS to save to when using ACIS as target format.</td>
</tr>
<tr>
<td>Enable Healing</td>
<td>Controls whether the conversion engine attempts to heal geometric entities. Only supported for specified formats.</td>
</tr>
<tr>
<td>Translate hidden/no-show entities</td>
<td>Controls whether the hidden entities are translated or discarded. Only supported for specified formats.</td>
</tr>
<tr>
<td>VRML/STL Scale factors</td>
<td>VRML and STL are often created in units that RobotStudio does not expect; they thus need to be resized.</td>
</tr>
<tr>
<td>Delete all generated log files on exit</td>
<td>Makes the CAD converter delete log files when exiting.</td>
</tr>
</tbody>
</table>
2.6.3. Troubleshooting and optimizing geometries

Overview

The characteristics of the geometries and CAD models in the station may have great effect on your work in RobotStudio, both in aspects of making the objects easier to program as well as enhancing simulation performance.

Below are some guidelines for troubleshooting geometries.

<table>
<thead>
<tr>
<th>Trouble</th>
<th>Information</th>
</tr>
</thead>
</table>
| The pointer snaps to the wrong parts of the objects when selecting in the graphics window | This problem might be caused by wrong snap mode settings, imprecise selecting, hidden or lack of geometrical information. To resolve these problems, do the following:  
  • Check the selection level and snap mode settings. For more information, see *Selecting an item on page 47*.  
  • When making the selection, zoom and rotate the object so that you are sure to click *inside* the object.  
  • Check if the object has hidden details that might affect the snapping. Remove details that are not necessary for your programming or simulation. For more information, see *Modifying a part on page 74*.  
  • Some file formats only contain a graphical representation and no geometrical data. Import the geometry from a file format that also contains geometrical data. For more information, see *Libraries, geometries and CAD files on page 26*. |
| The graphics window redraws or updates slowly                          | This might be due to the performance of your computer not being high enough for the size of the geometry files in your station. To reduce the size of the geometry files, do any of the following:  
  • Use a lower detail level for rendering the geometry. For more information, see *Graphic Appearance on page 333*.  
  • Blends, chamfers and holes can be automatically simplified by using the Defeature function. This can greatly reduce graphic complexity, speed up simulations and reduce memory usage. For more information, see *To defeature an object*.  
  • Check if the object has unnecessary details. Remove details that are not necessary for your programming or simulation. For more information, see *Modifying a part on page 74*. |
# How to build stations

## 2.6.3. Troubleshooting and optimizing geometries

*Continued*

<table>
<thead>
<tr>
<th>Trouble</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts of the geometry are not visible</td>
<td>If parts of the geometry are not visible from some views, a probable cause is that the object is made up of 2D surfaces and the option Backface culling is on. Backface culling means that the faces of the object are only visible from the front, and if the object (or any of its faces) is oriented differently, they will not be visible. To correct the problem, do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Switch to modeling mode and invert the direction of the face that is not displayed correctly. This not only corrects the display, it also decreases the chance of faulty orientations during graphical programming. For more information, see Invert on page 336 or To invert the direction of all faces of a part on page 72.</td>
</tr>
<tr>
<td></td>
<td>• Turn backface culling off for the specific object. This makes the object display correctly, but does not affect the direction of the face, which might cause problems if the face will be used for graphical programming. For more information, see To deactivate backface culling for a single object on page 72.</td>
</tr>
<tr>
<td></td>
<td>• Turn backface culling off for all objects in the station. This makes the objects display correctly, but does not affect the direction of the face, which might cause problems if the face will be used for graphical programming. It also decreases the performance of the graphic handling. For more information, see To change the generic setting for backface culling on page 72.</td>
</tr>
</tbody>
</table>

### To invert the direction of all faces of a part

To invert the direction of all faces of a part, follow these steps:

1. Select the part on which faces you want to invert the directions.
2. On the Modify menu, click Graphic Appearance.
3. On the Rendering tab, click Flip normals and then click OK.

### To deactivate backface culling for a single object

To change the backface culling setting for a single object, follow these steps:

1. Select the part for which you want to change the backface culling setting.
2. On the Modify menu, click Graphic Appearance.
3. On the Rendering tab, clear the Backface culling check box and then click OK. The faces of the object will now be displayed even if the generic setting for backface culling is on.

### To change the generic setting for backface culling

The generic setting for backface culling affects all new objects and existing objects that do not have backface culling specifically deactivated.

1. On the Application menu, click RobotStudio Options.
3. On the Performance page, select or clear the Cull back-facing triangles check box and then click OK.
2 How to build stations

2.7 Modeling

2.7.1. Objects

Overview

This section describes how to create or modify geometrical objects.

Creating a frame

A frame is a generic coordinate system that you can use as reference when positioning objects. Generic frames can also be converted to special kinds of coordinate systems, like workobjects or tool center points.

For procedures, see Frame on page 192 and Frame from Three Points on page 193.

Creating a solid

With the create solids commands you can create and build models of objects you do not have CAD files or libraries for. With the create solids commands you create primitive solid bodies; these can later be combined to more complex bodies.

For procedures, see Solid on page 228.

Creating a surface

For procedures, see Surface on page 232.

Creating a curve

When creating paths with targets based on the object geometries, curves are the geometrical objects that RobotStudio uses. For example, if you want the robot to run along the edge of an object, you can first create a curve along the border and then generate a complete path along that curve, instead of manually finding and creating the necessary targets.

If the CAD model/geometry of the work piece does not already contain curves, you can create the curves in RobotStudio.

For procedures, see Curve on page 234.

Modifying a curve

When creating paths with targets based on the objects geometries, curves are the geometrical objects that RobotStudio uses. By optimizing the curves before starting programming, you reduce the touch-up of the generated paths.

For procedures, see Modify Curve on page 344.

Creating a border

For procedures, see Border on page 239.

Creating a line from normal

A line can be created as a new part and body perpendicular to a surface.

For a procedure, see Line from Normal on page 245.

Continues on next page
2 How to build stations

2.7.1. Objects

Continued

Extruding a surface or curve

Curves and surfaces and curves can also be extruded to 3D objects, which may then be converted to solids. You can extrude along either a vector or a curve.

For procedures, see Extrude Surface or Curve on page 244.

Modifying a part

When you import a geometry or create an object, it will be one part. A part can, however, contain several bodies. In RobotStudio’s modeling mode you can modify the parts by adding, moving and deleting the bodies.

To modify a part, follow this step:

1. In the Modeling browser, expand the node for the part to modify. Then modify the part by doing any of the following:

<table>
<thead>
<tr>
<th>To</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete a body</td>
<td>Select the body and press the DEL key.</td>
</tr>
<tr>
<td>Move a body from one part to another</td>
<td>Drag the body or use the Copy and Paste commands on the Edit menu.</td>
</tr>
<tr>
<td>Move one body relative to the others</td>
<td>Select the body and then move it using any of the ordinary commands for moving objects, see Placing objects.</td>
</tr>
</tbody>
</table>

Modifying a library component

As external files, libraries are merely linked from a station. Therefore, to modify an imported library component, the link must first be broken and later reestablished. For procedures, see The Library Group on page 339.
2.7.2. Mechanisms

Workflow

This information topic describes how to create a new mechanism, that is, a graphical representation of a robot, tool, external axis or device. The various parts of a mechanism move along or around axes.

Creating a mechanism is dependent upon skillful construction of the main nodes of the tree structure. Four of these—links, joints, frames/tools and calibration—are initially marked red. As each node is configured with enough subnodes to make it valid, the marking turns to green. As soon as all nodes have become valid, the mechanism will be considered compilable and can be created. For additional validity criteria, see the table below.

<table>
<thead>
<tr>
<th>Node</th>
<th>Validity criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Links</td>
<td>• It contains more than one subnode.</td>
</tr>
<tr>
<td></td>
<td>• The BaseLink is set.</td>
</tr>
<tr>
<td></td>
<td>• All link parts are still in the station.</td>
</tr>
<tr>
<td>Joints</td>
<td>• At least one joint must be active and valid.</td>
</tr>
<tr>
<td>Frame/tool Data</td>
<td>• At least one frame/tool data exists.</td>
</tr>
<tr>
<td></td>
<td>• For a device, no frames are needed.</td>
</tr>
<tr>
<td>Calibration</td>
<td>• For a robot, exactly one calibration is required.</td>
</tr>
<tr>
<td></td>
<td>• For an external axis, one calibration is required for each joint.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>• For a tool or device, calibrations are accepted, but not required.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>• None.</td>
</tr>
</tbody>
</table>

The modify mode of the Mechanism Modeler has two purposes: to enable modification of an editable mechanism in its tree structure, and to complete the modeling of a new or modified mechanism.

It is recommended to configure each main node in the tree structure from the top down. Depending on its current status, right-click or double-click a node or subnode to add, edit or remove it.

For procedures, see Create Mechanism on page 247.
2.7.3. Tools and tooldata

Overview
To simulate the robot tool, you need tooldata for the tool. If you import a predefined tool or if you create a tool using the Create Tool Wizard, the tooldata is automatically created; otherwise, you have to create the tooldata yourself.

The tooldata simplifies the programming work with respect to the different tools that may come in use. Defining separate sets of tooldata for different tools makes it possible to run the same robot program with different tools: only the new tooldata has to be defined. The tooldata contains the information required for moving and simulating the tool.

Two methods for manipulating tooldata in RobotStudio are as follows:

- Create or modify tooldata, see Tooldata on page 196 and Modify Tooldata on page 352, respectively. This will create all data necessary for programming, but there will be no visual tool during the simulation.
- Create tooldata for an existing geometry, Create Tool on page 253.

 Creating and setting up a stationary tool
This information topic describes how to create a stationary tool. For information about creating a robot hold tool, see Create Tool on page 253.

Using a stationary tool, the robot holds and moves the work piece in relation to the tool. Thus, both the tooldata and the workobject must be set up correctly.

To create the tooldata for a stationary tool, follow these steps:

1. Import the geometry or library that represents the tool, see Import Geometry on page 191.
   If you do not have the geometry or library at hand but know the position, you can skip this step. The tool will be programable, but not visible in the station.
2. Create the tooldata for the tool, see Tooldata on page 196. Make sure to set the Robot holds tool option to false.
3. Create a workobject that is moved by the robot. see Workobject on page 195. Make sure to set the Robot holds workobject option to true.
4. If you have a geometry or library component for the work piece, attach it to the robot, see Attach to on page 323.
2.7.4. Setting the local origin of an object

Overview

Each object has a coordinate system of its own called local coordinate system in which the object dimensions are defined. When the object’s position is referred from other coordinate system, it is the origin of this coordinate system that is used.

With the Set Local Origin command you reposition the object’s local coordinate system, not the object itself.

For a procedure, see *Set Local Origin on page 365*. 
### 2.8 Placement

#### 2.8.1. Placing objects

#### Overview

To achieve the required layout of your station, you need to import or create objects, place them accordingly and, if applicable, attach them to other objects.

Placing objects means setting their position and rotation. If the objects are to be attached to robots or other mechanisms, they will be placed at their attachment point automatically.

The following table describes the actions relating to placement:

<table>
<thead>
<tr>
<th>Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Placing an object</td>
<td>To place an object is to put the object in the required position in the station, see Place on page 356 and Set Position on page 367.</td>
</tr>
<tr>
<td>Rotating an object</td>
<td>The objects in the station can be rotated to achieve the required layout, see Rotate on page 361.</td>
</tr>
<tr>
<td>Measuring distance or angles</td>
<td>The measurement functions calculates distances, angles and diameters between points you select from the graphics window. When using measurements, results and instructions on how to proceed are displayed in the Output window, see The Measure Group on page 246.</td>
</tr>
<tr>
<td>Creating a component group</td>
<td>A component group groups related object in the browser, see Component Group on page 226.</td>
</tr>
<tr>
<td>Attaching or detaching an object</td>
<td>Objects that are to be used by the robots in any way, such as tools, need to be attached to the robot, see Attach to on page 323 and Detach on page 331.</td>
</tr>
<tr>
<td>Jogging a robot</td>
<td>Robots can be placed by jogging. The robot axes can also be positioned by jogging, see Jogging mechanisms on page 84.</td>
</tr>
<tr>
<td>Modifying the task frame</td>
<td>Modifying the task frame repositions a controller and all its robots and equipment in the station.</td>
</tr>
<tr>
<td></td>
<td>By default the controller world and the station world coordinate system coincide. This is convenient when building a station with one single controller. For a procedure, see Set Task Frames on page 293.</td>
</tr>
<tr>
<td></td>
<td>However, when you have several controllers in one station, or need to reposition a controller in an existing station, you need to modify the System Configuration on page 294.</td>
</tr>
<tr>
<td>Modifying the baseframe position</td>
<td>Modifying the baseframe position sets an offset between the controller's world coordinate system and the baseframe of the mechanical unit. This is necessary when having several mechanical units belonging to one controller, for example, several robots in MultiMove systems or when using positioner external axes. For a procedure, see System Configuration on page 294.</td>
</tr>
</tbody>
</table>
2.8.2. Placing external axes

Overview

When starting a system with a track or positioner external axis in a RobotStudio station, you have to set up the system to load a model for the track or positioner and get the motions to work properly.

Prerequisites

The system shall be created with support for track or positioner external axes, see *A system with support for one robot and one positioner external axis on page 155*.

Attaching the robot to the track

To attach the robot to the track, follow these steps:

1. In the Layout browser, drag the robot icon and drop it on the track icon.
2. On the question *Should the robot be coordinated with the track?*, answer **Yes** to be able to coordinate the track’s position with that of the robot in robot programs. To program the track and the robot independently, answer **No**.
3. When asked if you want to restart the system, answer **Yes**.

The track is now added to the system and ready to be programmed, see *Programming external axes on page 109* for more information on how to program the track.

**CAUTION!**

If the system is cold started, the setup is deleted and the procedures described here must be performed again.

Placing the positioner in the station

To place the positioner in the station, follow these steps:

1. Move the positioner to the desired position using any of the ordinary functions for placing and moving objects, see *Placing objects*.
2. Modify the baseframe position of each mechanical unit of the positioner except the INTERCH unit, if it exists. When asked if you want to restart the system, answer **Yes**. See *Baseframe position* for detailed instruction.

After the restart the system is updated with the positioner’s new location. Continue attaching fixtures and workobjects to the positioner.

Attaching objects to the positioner

To program robot motions on an object that is held by the positioner, the targets must be created in a workobject that is attached to the positioner. For a complete visual simulation, CAD models that are moved by the positioner should also be attached. To attach the objects, follow these steps:

1. Import the models of the fixture and the work piece if you do not have them in the station already, see *Importing a station component on page 68*.
2. Attach the fixture to the positioner, see *Attaching and detaching objects on page 48*. When asked whether to keep the current position, answer **No**.

If the positioner has several stations, you will be asked which one to attach the object to.
3. Attach the work piece to the fixture. When asked whether to keep the current position, answer No.

4. Attach the workobject in which you will program the work piece to either the fixture, the work piece or the positioner. If you have defined calibration positions on either the work piece or the fixture, it is a good practice to use that object. When asked whether to keep the current position, answer No.

The positioner is now set up and ready to be programed, see Programming external axes on page 109 for more information.

**TIP!**

If the positioner is of an interchangeable type with several stations, you can either attach individual fixtures, work pieces and workobjects to each station flange, or you can use one set of objects that you attach and detach to the different flanges by events.

**CAUTION!**

If the system is cold started, the setup is deleted and the procedures described here must be performed again.
3 How to program robots

2.8.2. Placing external axes
3 How to program robots

3.1. Workflow for programming a robot

Overview

In most cases, going through the workflow from start to finish is recommended, even if it possible to work in other sequences as well.

Synchronizing will save and load text files containing RAPID modules, and create RAPID programs from your station.

Prerequisites

Before creating a program for your robot, you should set up the station, including robots, work pieces and fixtures, in which your robot will work.

Programming a robot

The table below describes the workflow for programming a robot to perform the task you require.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create targets and paths</td>
<td>Create the targets and paths the robot requires to perform the work tasks. To create targets and paths, do one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Create a curve to match your required shape. Then use the Create path from curve command to generate a path, complete with targets, along the shape you have created. See Curve on page 234 and Path from Curve on page 202.</td>
</tr>
<tr>
<td></td>
<td>• Create targets at the requested positions, then create a path and insert the created targets into it. See Create Target on page 198, Teach Target on page 197 and Empty Path on page 201.</td>
</tr>
<tr>
<td>Check the target orientations</td>
<td>Make sure that the targets are oriented in the most efficient way for the tasks to be performed. If not, reorient the targets until you are satisfied. See Orientations on page 91.</td>
</tr>
<tr>
<td>Check reachability</td>
<td>Check that the robot and tool reach all targets in the path. See Testing positions and motions on page 100.</td>
</tr>
<tr>
<td>Synchronize the program to the virtual controller</td>
<td>Generates RAPID code from the RobotStudio items and enables the program to be simulated.</td>
</tr>
<tr>
<td>Perform text-based editing</td>
<td>If you need to edit the instructions or data created by RobotStudio, you can start the Program Editor. See Using the RAPID editor on page 113.</td>
</tr>
<tr>
<td>Collision detection</td>
<td>Check that the robot or tool does not collide with the surrounding equipment or the fixtures. If it does, adjust the placements or orientations until no collisions occur. See Detecting collisions on page 119.</td>
</tr>
<tr>
<td>Test the program</td>
<td>Test the program by moving along the paths. See Testing positions and motions on page 100.</td>
</tr>
</tbody>
</table>

© Copyright 2008-2009 ABB. All rights reserved.
3.2. Workobjects

Creating a workobject
A workobject is a coordinate system used to describe the position of a work piece. The workobject consists of two frames: a user frame and an object frame. All programmed positions will be related to the object frame, which is related to the user frame, which is related to the world coordinate system.

For a procedure, see Workobject on page 195.

Modifying a workobject
For a procedure, see Modify Workobject on page 353.

Converting a frame to a workobject
You can create a new workobject from an existing frame. The converted workobject gets the same name and position as the selected frame.
For a procedure, see Convert Frame to Workobject on page 328.

Creating a frame by points
You can create a frame by specifying points on the axes of the coordinate system and letting RobotStudio calculate the placement and orientation of the frame’s origin.
For a procedure, see Frame from Three Points on page 193.
3 How to program robots

3.3. Jogging mechanisms

3.3. Jogging mechanisms

Jogging a robot

To check if the robot can reach all positions on the work piece, you can jog the TCP or the joints of the robot, either with the freehand commands or through dialog boxes. Jogging the robot close to its boundaries is best done with the latter method.

<table>
<thead>
<tr>
<th>To</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jog the joints of a robot</td>
<td>For freehand, see <em>Jog Joint on page 221</em>. For a dialog box, see <em>Mechanism Joint Jog on page 340</em>.</td>
</tr>
<tr>
<td>Jog the TCP of a robot</td>
<td>For freehand, see <em>Jog Linear on page 222</em>. For a dialog box, see <em>Mechanism Linear Jog on page 342</em>.</td>
</tr>
</tbody>
</table>

Prerequisites

To jog the TCP of a robot, the robot’s VC must be running.

Jogging several mechanisms

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multirobot jog</td>
<td>When using multirobot jog, all selected mechanisms will follow the TCP of the one being jogged. Multirobot jog is available for all kinds of jogging. See <em>MultiRobot Jog on page 223</em>.</td>
</tr>
</tbody>
</table>
| Jogging with locked TCP | When jogging a mechanism that moves a robot (like a track external axis) with locked TCP, the robot will reposition so that the position of the TCP does not change, even though its baseframe is moved.  
When jogging an external axis that moves the work object with locked TCP, the robot will reposition so that its TCP follows the work object in the same way as when using multirobot jog.  
Locked TCP is available when jogging a mechanism that belongs to the same task as a robot. See *Mechanism Joint Jog on page 340*. |
3.4. Targets

Creating a target

You can create a new target manually either by entering the position for the target in the Create Target dialog box or by clicking in the graphics window. The target will be created in the active workobject. For a procedure, see Create Target on page 198.

Creating a jointtarget

A jointtarget is a specification of the position for the robot axes. For a procedure, see Create Jointtarget on page 200.

Teaching targets

You can create a new target by jogging the robot and teaching a target at the active TCP. Taught targets will be created with the axis configuration used when jogged to the target. The target will be created in the active workobject. For a procedure, see Teach Target on page 197.

Modifying a target position

By using the modify position command you can modify the position and rotation of a target. For procedures, see Set Position on page 367 and Rotate on page 361, respectively.

Modifying a target with ModPos

The position of an existing target can be modified by jogging the robot to a new, preferred position. By selecting a move instruction for the target in a path, the ModPos command can be used to move the target to the TCP of the active tool.

When ModPos is executed, the target, referenced to by the move instruction, will be updated with the following information:

- position and orientation corresponding to the TCP of the active tool
- the current configuration of the active robot
- the current position and orientation values of all active external axes for the active robot

NOTE: To jog a robot linearly, a virtual controller must be running for that robot. For detailed information, see Starting a VC on page 64.
3 How to program robots

3.4. Targets

Continued

Renaming targets

With this command you can change the name of several targets at once. You can either rename targets individually, or you can rename all targets in one or several paths at once.

The new target names will consist of an optional prefix, an incremental number and an optional suffix.

For a procedure, see Rename Targets on page 359.

When renaming targets, make sure that the new targets conform to the naming rules. The target names must:

- start with an alphabetical character in the ISO 8859-1 encoding (that is, an ordinary letter from the English alphabet)
- be shorter than 16 characters
- not be empty strings
- not contain any characters illegal in RAPID. See the RAPID reference manual for details.

Removing unused targets

If deleting or changing paths or move instructions during programming, you might end up with large numbers of targets that are no longer used in any instructions. To make the workobjects and their targets easier to grasp, you can delete all unused targets.

For a procedure, see Remove Unused Targets on page 358.
3.5. Paths

Creating an empty path
A path is a sequence of targets with move instructions that the robot follows. An empty path will be created in the active task.

For a procedure, see Empty Path on page 201.

Creating a path from curve
If the work piece has curves or contours that correspond to the path to be created, you can create the paths automatically. The create path from curve command generates paths, complete with targets and instructions along existing curves.

The path will be created in the active task.

The orientation of the targets that will be created will be according to the settings of the approach/travel vectors in the Options dialog box.

To create a path from a curve, the curve must have first been created in the station.

For a procedure, see Path from Curve on page 202.

Setting robot axis configuration for paths
The robot axis configuration specifies the position of the axes as the robot moves from target to target, when multiple solutions are possible. This is necessary for executing move instructions using configuration monitoring.

Taught targets have validated configurations, but targets created in any other way do not. Also, targets that are repositioned lose their configuration. In RobotStudio, targets without a valid configuration are marked with a yellow warning symbol. See Robot axis configurations on page 24 for more information about configurations.

To set a configuration for all targets in a path, see Auto Configuration on page 324.
To set a configuration for a single target, see Configurations on page 326.

Reversing paths
The reverse path commands change the sequence of targets in the path so that the robot moves from the last target to the first. When reversing paths, you can reverse either the target sequence alone or the entire motion process.

For procedures, see Reverse Path on page 360.
NOTE!
When reversing paths, the original paths are deleted. If you want to keep them, make copies before reversal.

NOTE!
When reversing paths, only move instructions are handled. Action instructions, if any exist, have to be inserted manually after the reversal.

Rotating paths
With the rotate path command you can rotate complete paths and move the targets used by the paths accordingly. When rotating paths, the included targets will lose their axis configurations, if any have been assigned.
A frame or target must exist at the position to rotate around before starting the rotate path command.
For a procedure, see Rotate Path on page 362.

Translating a path
The translate path function moves a path and all included targets.
For a procedure, see Translate Path on page 369.

Compensating paths for tool radius
You can offset a path so that it compensates for the radius of a rotating tool. Since the targets in the path are moved, they will lose their axis configurations, if any have been assigned.
For a procedure, see Tool Compensation on page 368.
### Interpolating a path

The interpolate functions reorient the targets in a path so that the difference in orientation between the start and end targets is distributed evenly among the targets in between. The interpolation can be either linear or absolute.

Linear interpolation distributes the difference in orientation evenly, based on the targets’ positions along the length of the path.

Absolute interpolation distributes the difference in orientation evenly, based on the targets’ sequence in the path.

Below are examples of the difference between linear and absolute interpolation.

The interpolate functions reorient the targets in a path so that the difference in orientation between the start and end targets is distributed evenly among the targets in between. The interpolation can be either linear or absolute.

For a procedure, see *Interpolate Path on page 335*.

### No interpolation

This is the path before any interpolation. Note that the last target is oriented differently than the others.

![No interpolation](image)

### Linear interpolation

This is the same path after linear interpolation.

![Linear interpolation](image)

Note that the targets are oriented based on their placement relative to the start and end targets.

If a target were moved and you reran the linear interpolation, it would be reoriented according to its new position.

If new targets were inserted between the existing ones and you reran the linear interpolation, it would not affect the orientation of the existing targets.
3 How to program robots

3.5. Paths

Continued

Absolute interpolation

This is the same path after absolute interpolation

Note that the targets are orientated based on their sequence in the path: each target has been reoriented equally, regardless of its place.

If a target were moved and you reran the absolute interpolation, it would not affect the orientation.

If new targets were inserted between the existing ones and you reran the absolute interpolation, it would change the orientation of all targets.

Mirroring a path

The mirror path function mirrors all motions instructions and their targets to a new path.

For a procedure, see Mirror Path on page 343.
3.6. Orientations

Overview
This is an overview of the tools for automating the modification of target orientations. When creating paths from curves in RobotStudio, the orientation of the targets depends on the characteristics of the curves and the surrounding surfaces. Below is an example of a path with unordered target orientations and examples of how the different tools have affected the targets.

Unordered orientations
In the path below, the target orientations are unordered. The function View tool at target has been used for illustrating how the targets point in different directions.

Effect of target normal to surface
In the picture below, the targets, which previously were orientated randomly, have been set normal to the flat round surface at the right side of the path. Note how the targets’ Z axis has been orientated normal to the surface; the targets have not been rotated in the other directions.
3 How to program robots

3.6. Orientations

Continued

Setting a target normal to surface

To set a target orientation normal to a surface is to make it perpendicular to the surface. The target can be oriented normal to the surface in two different ways:

- The entire surface can be used as a reference for the normal. The target will be oriented as the normal to the closest point at the surface. The entire surface is the default surface reference.
- A specific point on the surface can be used as the reference for the normal. The target will be orientated as the normal to this point, regardless of whether the normal to the closest point at the surface has another orientation.

Objects imported without geometry (for example, .jt files) can only refer to specific points on the surface.

For a procedure, see Set Normal to Surface on page 366.

Effect of align target orientation

In the picture below, the targets, which were previously orientated with the Z axis normal to the surface but with the X and Y axes orientated randomly, have been organized by aligning the targets’ orientation around the X axis with the Z axis locked. One of the targets in the path has been used as reference.

Aligning a target orientation

With the align target orientation command you align the rotation of selected targets around one axis without changing the rotation around the others.

For a procedure, see Align Target Orientation on page 322.

Continued on next page
TIP!
You can also align ordinary frames in the same way.

Effects of copy and apply orientation
In the picture below, the targets, which were previously oriented randomly, have been organized by copying the exact orientation of one target to all the others. This is a quick way to fix workable orientations for processes where variations in approach, travel, or spin directions either do not matter or are not affected, due to the shape of the work piece.

Copying and applying an orientation for objects
To transfer an orientation from one object to another is an easy way to align different frames for simplifying the programming of the robot. Target orientations may also be copied.

For procedures, see Copy / Apply Orientation on page 330.
3 How to program robots

3.7. RAPID Instructions

Move and action instructions

For RAPID programming, RobotStudio’s main advantage is in the area of motion programming.

A move instruction is an instruction for the robot to move to a specified target in a specified manner. With RobotStudio, you can create move instructions in three ways:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a move instruction based on an existing target</td>
<td>Creates move instructions based on one or several targets selected in the Paths &amp; Targets browser. For a procedure, see Add to Path on page 320.</td>
</tr>
<tr>
<td>Create a move instruction and a corresponding target</td>
<td>Creates a move instruction and a corresponding target at once. The position of the target can either be selected from the graphics window or typed numerically. For a procedure, see Move Instruction on page 214.</td>
</tr>
<tr>
<td>Teach a move instruction</td>
<td>Teaching a move instruction creates a move instruction and a corresponding target at the robot’s current position. Teaching a move instruction also stores the current configuration with the target. For a procedure, see Teach Instruction on page 213.</td>
</tr>
</tbody>
</table>

In addition to move instructions, you can also create and insert action instructions from RobotStudio. An action instruction is an instruction other than a move instruction that can, for example, set parameters, or activate or deactivate equipment and functions. The action instructions available in RobotStudio are limited to those commonly used for affecting the robot’s motions. For inserting other action instructions or another kind of RAPID code in the program, use the Program Editor. For a procedure, see Action Instruction on page 215.

The table below lists the action instructions that can be created. For details, see the RAPID Reference Manual.

<table>
<thead>
<tr>
<th>Action instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfL On/Off</td>
<td>ConfL specifies whether to monitor the robot’s configurations during linear movements. When ConfL is set to Off, the robot may use another configuration than the programed one for reaching the target during program execution.</td>
</tr>
<tr>
<td>ConfJ On/Off</td>
<td>ConfJ specifies whether to monitor the robot’s configurations during joint movements. When ConfJ is set to Off, the robot may use another configuration than the programed one for reaching the target during program execution.</td>
</tr>
<tr>
<td>Actunit UnitName</td>
<td>Actunit activates the mechanical unit specified by UnitName.</td>
</tr>
<tr>
<td>DeactUnit UnitName</td>
<td>Deactunit deactivates the mechanical unit specified by UnitName.</td>
</tr>
<tr>
<td>ConfJ On/Off</td>
<td>ConfJ specifies whether to monitor the robot’s configurations during joint movements. When ConfJ is set to Off, the robot may use another configuration than the programed one for reaching the target during program execution.</td>
</tr>
<tr>
<td>Actunit UnitName</td>
<td>Actunit activates the mechanical unit specified by UnitName.</td>
</tr>
<tr>
<td>DeactUnit UnitName</td>
<td>Deactunit deactivates the mechanical unit specified by UnitName.</td>
</tr>
</tbody>
</table>

Continues on next page
Modifying an instruction

Most instructions have arguments that specify how the instruction shall be carried out. For example, the MoveL instruction has arguments that specify the speed and accuracy with which the robot moves to the target.

For a procedure, see *Modify Instruction on page 350.*

**NOTE!**

Some arguments are read from the virtual controller. If the virtual controller has not been started, only the arguments stored in the station can be modified.

Converting to move circular

To create a circular motion to an instruction target, you must convert the motion type to circular motion (that is, MoveC in RAPID).

A circular motion is defined by two motion instructions, where the first is the via-point and the second contains the end point of the circular motion.

The circular motion can only be used for open circular arcs, not for closed circles. To create a path for a closed circle, use two circular motions.

For a procedure, see *Convert to Move Circular on page 329.*

Creating RAPID instructions for setting I/O signals

For controlling I/O signals in the robot program, you use RAPID commands that set the signals. These require that you first create instruction templates for the instructions that set the signals. See the *RAPID reference manual* for details about the instructions that control I/O signals.

To add RAPID instructions that set I/O signals, follow these steps:

1. Synchronize the system in which you want to add the instructions to the virtual controller, see *Synchronization on page 112.*

2. In programming mode, select the module for editing, right-click it and then click **Edit program.**

3. In the program editor, add the instructions for setting the signals.

4. When you are done adding instructions, synchronize the task and paths from the Virtual Controller back to the station.

Continued on next page
3 How to program robots

3.7. RAPID Instructions

Continued

Using cross-connections and groups for setting I/O signals

You can also create cross-connections and signal groups, which make one signal set the value of several other signals. See the System parameters reference manual for details about cross-connections and groups.

To make one signal set several others, follow these steps:

1. Request write access, and then open the configuration topic I/O in the configuration editor. Add configure instances for the cross-connections and groups to create.

Instruction templates

Instruction templates contain predefined sets of argument values that are applied to the instructions you create using the template. You can create templates for all instructions in the system running on the virtual controller. To see which instructions are available and what their arguments do, see the RAPID reference manual for your RobotWare version and the reference sections in manuals for software options, if you have any installed on the system.

Move instruction templates are always part of process templates. The process templates contain one instruction template for each type of move instruction that might be used by the process.

The process templates are instances of process definitions, which define the types of move instructions (move instruction definitions) that might be used by the process.

To create new move instruction templates, start by creating a new process template for a process that uses the type move instructions you want to create templates for. If such a process does not exist, you first have to create a new process definition.

If no move instruction definition for the type of instruction you want to create a template for exists, you must create it first.

When creating instruction descriptions, the virtual controller must be running, since the available instruction types are read from the system.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating a process template with move instruction templates on page 97</td>
<td>Move instructions are always related to processes.</td>
</tr>
<tr>
<td>Creating a move instruction description on page 98</td>
<td>To create templates for other instructions than the one that already exists in the tree view, you first have to create an instruction description that defines the arguments that belong to the instruction.</td>
</tr>
<tr>
<td>Editing an instruction template on page 98</td>
<td>XXX</td>
</tr>
<tr>
<td>Creating a process definition on page 99</td>
<td>XXX</td>
</tr>
<tr>
<td>Creating an action instruction template on page 99</td>
<td>XXX</td>
</tr>
<tr>
<td>Creating an action instruction description on page 99</td>
<td>To create templates for other instructions than the one that already exists in the tree view, you first have to create an instruction description that defines the arguments that belong to the instruction.</td>
</tr>
</tbody>
</table>

Continues on next page
Templates can be imported and exported on four levels: tasks, move instruction descriptions, action instruction descriptions and process definitions. The default directory for imported and exported template files is \textit{My Documents/RobotStudio}. Simply choosing another directory will then make that directory default. The default file format is .xml.

The validation procedure checks for duplicate names, incomplete process definitions and virtual controller equality. It is performed automatically, after a template file has been imported or a node renamed or deleted.

For procedures, see \textit{Instruction Template Manager}.

\textbf{Instruction Template Manager}

The Instruction Template Manager is used to add support for instructions other than the default set that comes with the RobotStudio.

For example, a robot controller system with the RobotWare Dispense option has specialized move instructions related to glueing like DispL and DispC. You can manually define the instruction templates for these using the Instruction Template Manager. The instruction templates are exported to XML format and reused later.

The instruction template supports the following Robotware options:
- Cap (Continuous Application Process)
- Disp (Dispense)
- Trigg (Fixed Position Events)
- Spot Pneumatic
- Spot Servo
- Spot Servo Equalizing
- Paint

RobotStudio has pre-defined XML files that are imported and used for robot controller systems with the appropriate RobotWare options. These XML files has both the Move and Action instructions.

\textbf{NOTE!} It is recommended to use RobotStudio ArcWelding PowerPac while using RobotWare Arc.

\textbf{Creating a process template with move instruction templates}

1. On the \textbf{Home} tab, from the active \textbf{Task} list, select the task for the robot for which you want to create the instruction template.

2. On the \textbf{Create} menu, click \textbf{Instruction Template Manager}. This opens the Instruction templates page in the work space.

3. In the \textbf{Instruction Templates} tree to the left, make sure there are move instruction definitions for the types of move instructions you want to create templates for. If not, follow the procedure in \textit{Creating a move instruction description on page 98} for creating them.

4. Make sure there is a process definition that uses the types of move instructions you want to create templates for. If not, follow the procedure in \textit{Creating a process definition on page 99} for creating it.

5. Right-click the \textit{process definition} for which you want to create a new templates and click \textbf{Create Process Definition}.

\textit{Continues on next page}
3 How to program robots

3.7. RAPID Instructions

Continued

6. In the Create Process Definition dialog box, enter a name, with characters from ASCII set, for the new template and click Create. A new process template node with a set of move instruction templates is now created.

7. Select each new template one at a time, and in the arguments grid to the right of the tree view, set the argument values that shall be applied when you create new instructions based on the template. Finish for each template by clicking Apply changes at the bottom of the grid.

For details about available arguments and what they do, see the RAPID reference manual for ordinary RAPID instructions and the option manual for software option instructions.

Creating a move instruction description

To create the instruction description, follow these steps:

1. Right-click the Move Instructions node and click Create Move Instruction Description. This opens the Create Move Instruction Description dialog box.

2. In the Controller Description list, select the instruction you want to make a description for. All action instructions installed on the controller, both through RobotWare and software options, are available.

3. In the Motion type list, select the motion type for the instruction.

4. Optionally, in the Information text box, enter a comment to the instruction.

5. Click Create. The instruction description appears in the tree view and its settings are displayed in the grid.

6. In the instruction grid, set the point type values. If necessary, also change the other settings.

7. After changing the settings, click Apply Changes in the bottom of the grid view.

Editing an instruction template

To edit an instruction template, follow these steps:

1. In the active task list, select the task for the robot for which you want to edit the instruction template.

2. On the Create menu, click Instruction Template Manager. This opens the Instruction templates page in the work space.

3. In the Instruction Templates tree to the left, browse to and select the template to edit.

4. In the arguments grid to the right of the tree view, set the argument values that shall be applied when you create new instructions based on the template. Finish by clicking Apply changes at the bottom of the grid.

For details about available arguments and what they do, see the RAPID reference manual for ordinary RAPID instructions and the option manual for software option instructions.
3 How to program robots

3.7. RAPID Instructions

Continued

Creating a process definition

To create a process definition, follow these steps:

1. Make sure there are move instruction definitions for the types of move instructions you want to create templates for. If not, follow the procedure above for creating them.

2. Right-click the Process definitions node and click Create Process Definition. This opens the Create Process Description dialog box.

3. In the Create Process Definitions dialog box, enter a name for the process definition, a name for its first process template and then select the move instruction types to use. Finish by clicking Create.

Creating an action instruction template

To create a template for an action instruction, follow these steps:

1. In the active task list, select the task for the robot for which you want to create the instruction template.

2. On the Home menu, click Instruction Template Manager.

3. In the Instruction Templates tree to the left, right-click the instruction description (corresponding to the instructions as described in the RAPID reference manual) for which you want to create a new template and click Create Action instruction Template. If the instruction description does not exist in the tree, create it by following the procedure described in Creating an action instruction description on page 99.

4. In the Create Action Instruction Template dialog box, enter a name for the new template and click Create. The new template is now created under the instruction description node it belongs to.

5. Select the new template and in the arguments grid to the right of the tree view, set the argument values that shall be applied when you create new instructions based on the template. Finish by clicking Apply changes at the bottom of the grid.

For details about available arguments and what they do, see the RAPID reference manual for ordinary RAPID instructions and the option manual for software option instructions.

Creating an action instruction description

To create the instruction description, follow these steps:

1. Right-click the Action Instructions node and click Create Action Instruction Description. This opens the Create Instruction Description dialog box.

2. In the Controller Description list, select the instruction you want to make a description for. All action instructions installed on the controller, both through RobotWare and software options, are available.

3. Optionally, in the Information text box, enter a comment to the instruction.

4. Click Create. The instruction description appears in the tree view, and its settings are displayed in the grid.

   After changing the settings (if necessary), click Apply Changes in the bottom of the grid view.

5. Continue with creating templates for the instruction description, as described in Creating an action instruction template on page 99.
3 How to program robots

3.8. Testing positions and motions

Overview

RobotStudio has several functions for testing how robots reach and move to targets. They are useful both for finding the optimal layout when building a station and during programming. Below are brief descriptions of the functions for testing reachability and motions.

Checking reachability

The check reachability function displays whether the robot can reach selected targets and motion instructions by changing the frames’ colors in the graphic view. Reachable frames are colored green, unreachable red, and frames with reachable positions but not with current orientation are colored yellow.

The reachability check is useful when building the station, since the reachability of several targets are displayed at once. For a procedure, see Check Reachability on page 325.

Jumping to target

Jump to target tests whether the robot can reach a specific position. This is useful when building the station: by creating targets at critical positions on the work piece and jumping the robot to them, you get an early indication of whether the items are positioned correctly or not. For a procedure, see Jump to Target on page 337.

Viewing a robot at target

When View robot at target is activated, the robot is automatically positioned with the tool at the target when one is selected. If several robot axis configurations are possible for reaching the target, the robot will use the one nearest the configuration it had before jumping to the target. For a procedure, see View Robot at Target on page 370.

Viewing tool at target

View tool at target displays the tool at target, without checking that the robot can reach it. This test is useful both when building the station and when programming the robot, since the orientation of targets both affects the reachability and the process performance. For a procedure, see View Tool at Target on page 371.

Executing move instructions

Execute move instruction tests if the robot can reach a specific position with the programed motion properties. This is useful for testing motions during programming. For detailed information, see Execute Move Instruction on page 332.

Moving along path

Move along path executes all move instructions in a path. It is thereby a more complete test than Execute move instructions, but not as complete as a full simulation, since it ignores RAPID code that is not move instructions. For a procedure, see Move Along Path on page 354.

Continues on next page
3 How to program robots

3.8. Testing positions and motions

Continued

Moving to a pose

Moving to a pose moves a mechanism to a predefined joint value at a predefined time without using the Virtual Controller. This is useful when movement of the external equipment (such as a clamp or conveyer) must be simulated. For a procedure, see Move to Pose on page 355.

Simulating programs

Simulating programs involves running a program on the virtual controller as it is run on a real controller. It is the most complete test whereby you can see how the robot interacts with external equipment through events and I/O signals. For a procedure, see Simulation Setup on page 257.

Improving the reachability

If the robot cannot reach the target, or if you are not satisfied with the motions, try the following for improving the reachability:

- Set ConfL or ConfJ to Off for enabling the robot to use new configurations for reaching the target.
- Change the orientation of the target.
- Change the position of either the robot or the work piece.
- Use a system with a track external axis for increasing the robot’s range.
- Use a system with a positioner external axis for enabling different work piece positions for different targets.
3 How to program robots

3.9.1. About programming MultiMove

3.9 Programming MultiMove systems

3.9.1. About programming MultiMove

About MultiMove

The MultiMove functions helps you create and optimize programs for MultiMove systems where one robot or positioner holds the work piece and other robots operate on it. Below is an outline of the main workflow for programming MultiMove systems with RobotStudio, with references to detailed instructions further down in the section.

Prerequisites

For using the MultiMove functions you must first have the following:

• A virtual controller running a MultiMove system started in RobotStudio, see A MultiMove system with two coordinated robots on page 153 for an example.

• All coordinate systems and tools used by the system.

• The paths along which the tool shall move. The paths must be created in a workobject that belongs to a tool robot and is attached to the work piece robot. A wizard will guide you through attaching the workobjects if this has not been done before starting the MultiMove functions.

For detailed information about MultiMove in RobotWare systems and RAPID programs, see the MultiMove application manual.

Normal workflow

This is the typical workflow for creating MultiMove programs using the MultiMove function:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up the MultiMove</td>
<td>Select the robots and paths to use in the program, see Setting up the MultiMove on page 104.</td>
</tr>
<tr>
<td>Testing the MultiMove</td>
<td>Execute the motion instructions along the paths, see Testing the MultiMove on page 105.</td>
</tr>
<tr>
<td>Tuning the motion behavior</td>
<td>Tune motion behavior, such as tolerances and constraints for TCP motions, see Tuning the motion behavior on page 106.</td>
</tr>
<tr>
<td>Creating the program</td>
<td>Generate the tasks for the robots, see Creating paths on page 108.</td>
</tr>
</tbody>
</table>
In addition to using the functions that calculate and create optimized MultiMove paths, you can program MultiMove manually using a combination of the ordinary programming tools in RobotStudio and a set of tools specific for MultiMove programming.

The main actions for programming MultiMove manually are outlined below. Not all actions might be necessary, but the order in which they shall be carried out depends on the contents of the station and your goals.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating Tasklists and Syncidents</td>
<td>This data specifies the tasks and paths that shall be synchronized with each other. See <a href="#">The Create Tasklist tool on page 212</a> and <a href="#">The Create Syncident tool on page 212</a>, respectively.</td>
</tr>
<tr>
<td>Adding and updating ID arguments to the instructions to synchronize</td>
<td>To add IDs to the instructions, you can use one of the following methods: Using <a href="#">The Recalculate ID tool on page 211</a> to add and update IDs for instructions in paths that already are synchronized. Using <a href="#">The Convert path to MultiMove path tool on page 212</a> to add IDs to instructions in paths that have not yet been synchronized.</td>
</tr>
<tr>
<td>Adding and adjusting Sync instructions to the paths.</td>
<td>Add SyncMove on/Off or WaitSyncTask instructions to the paths to synchronize and set their tasklist and Syncident parameters. See <a href="#">Creating an action instruction on page 215</a>.</td>
</tr>
<tr>
<td>Teaching MultiMove instructions</td>
<td>It is also possible to jog all robots to the desired positions and then teach instructions to new synchronized paths. See <a href="#">MultiTeach tab on page 208</a>.</td>
</tr>
</tbody>
</table>
3.9.2. Setting up the MultiMove

Selecting robots and paths

This procedure is for selecting the robots and paths in the station that shall be used for the MultiMove program. All robots for the MultiMove program must belong to the same system.

1. On the Home tab, click MultiMove. Click the Setup tab below the MultiMove work area.
2. In the work area, click the System config bar for expanding the system config section.
3. In the Select System box, select the system that contains the robots to program.

   The robots of the selected system are now displayed in the System grid below the Select system box.
4. For each robot that shall be used in the program, select the check box in the Enable column.
5. For each robot that shall be used in the program, specify whether it carries the tool or the work piece using the options in the Carrier column.
6. In the work area, click the Path config bar for expanding the path config section.
7. Select the Enable check box for the tool robot and click the expand button. This displays the paths of the robot.
8. Select the order of the paths to execute by specifying them in right order using Path name column.
9. For each path that shall be included in the program, select the check box in the Enable column.
10. When you have set up the robots and paths, continue testing the Multimove and then tune the motion properties, if necessary.
3.9.3. Testing the MultiMove

Overview

Testing the MultiMove executes the motion instructions along the paths according to the current settings on the setup and motions properties pages.

Testing Paths

This procedure is for setting the robots start position and testing the resulting movements along the path sequence.

1. Jog the robots to what seems to be a good start position.
2. On the Home tab, click MultiMove. Click the Test tab at the bottom of the MultiMove work area for displaying the test area.
3. Optionally, select the Stop at end check box to make the simulation stop after moving along the paths. Clearing this check box makes the simulation continue in a loop until you click Pause.
4. Click Play to simulate the motions along the paths based on the current start position.

   If you are satisfied with the motions, continue generating multimove paths. If the simulation cannot complete or if you are not satisfied with the motions, pause the simulation and perform any of the actions below to adjust the motions:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examine the robots’ positions for critical targets.</td>
<td>Click Pause and then use the arrow buttons to move to one target a time.</td>
</tr>
<tr>
<td>Jog the robots to new start positions.</td>
<td>New start positions might result in changed motions, since the robots will use different configurations. In most cases, positions near the robots’ joint limits shall be avoided.</td>
</tr>
<tr>
<td>Go to the Motion Behavior tab and remove constraints.</td>
<td>The default setting for the motion properties is no constraints. If this has been changed, constraints might exist that limit motions more than necessary.</td>
</tr>
</tbody>
</table>
3.9.4. Tuning the motion behavior

Overview

Tuning the motion behavior means to set up rules for the robot’s motions, for example, constraints on the position or orientation of the tool. Generally, the MultiMove program will obtain the smoothest motions with the fastest cycle and process times with as few constraints as possible.

For procedures, see Motion Behavior tab on page 205.

Modifying the joint influences

The joint influence controls the balance of how much the robots will use their joints. Decreasing the weight value for one axis will restrict the motion for this axis, while increasing it will promote motion on this axis relative to alternative axes.

1. On the Home tab, click the Motion Behavior tab.
2. Expand the Joint Influence group by clicking its title bar.
3. In the Select Robot box, select the robot whose joint influence you want to modify.
   The weight values for the robot axes are now displayed in the grid.
4. For each axis whose motion you want to restrict or promote, adjust the Weight value. A lower value restricts, and a higher value promotes, motions on that axis.

Modifying the TCP constraints

The joint influence controls the balance of how much the robots will use their joints. Decreasing the weight value for one axis will restrict the motion for this axis, while increasing it will promote motion on this axis relative to alternative axes.

1. On the Simulation tab, click the Motion Behavior tab.
2. Expand the TCP Constraints group by clicking its title bar.
   The directions and rotations in which you can constrain the TCP’s motion are now displayed in the grid.
3. For each pose you want to constrain, select the Enable check box and specify the constraint values (location in the TCP coordinate system). To use the values from the current TCP position, click Pick from TCP.
4. Optionally, adjust the Weight value for the constraint. A low value results in a harder constraint, while a high value allows a larger deviation.

Modifying the tool tolerance

The joint influence controls the balance of how much the robots will use their joints. Decreasing the weight value for one axis will restrict the motion for this axis, while increasing it will promote motion on this axis relative to alternative axes.

1. On the Simulation tab, click the Motion Behavior tab.
2. Expand the Tool Tolerance group by clicking its title bar.
   The directions and rotations in which you can enable tolerances are now displayed in the grid.
3. For each offset you want to set, select the Enable check box.

Continues on next page
4. In the Value column, specify the allowed deviation.

5. Optionally, adjust the Weight value for the tolerance. A low value increases the use of the tolerance, while a high value promotes motions that do not use the tolerance.

---

**Modifying the tool offset**

The tool offset sets a fixed distance between the tool and the paths.

1. On the Simulation tab, click the Motion Behavior tab.

2. Expand the Tool Offset group by clicking its title bar.
   
   The directions and rotations in which you can set offsets are now displayed in the grid.

3. For each offset you want to set, select the Enable check box.

4. In the Offset column, specify the offset distance.
3 How to program robots

3.9.5. Creating paths

Overview

When you are satisfied with the motions displayed when testing the Multimove program, the next step is to convert the temporary move instructions used by the MultiMove function to ordinary paths in RobotStudio.

Creating the paths

To create paths for the MultiMove program in RobotStudio, follow these steps:

1. On the Home tab, click Create Paths tab.
2. Expand the Settings group by clicking on its title bar.
3. Optionally, change the naming settings in the following boxes:

<table>
<thead>
<tr>
<th>Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start ID</td>
<td>Specify the first ID number for the synchronization of the instructions for the robots.</td>
</tr>
<tr>
<td>ID step index</td>
<td>Specify the increment between ID numbers.</td>
</tr>
<tr>
<td>Sync ident prefix</td>
<td>Specify a prefix for the syncident variable, which connects the sync instructions in the tasks for the tool robot and the work piece robot with each other.</td>
</tr>
<tr>
<td>Task list prefix</td>
<td>Specify a prefix for the tasklist variable, which identifies the tasks for the tool robot and the work piece robot to synchronize.</td>
</tr>
</tbody>
</table>

4. Expand the WP Robot Settings group by clicking on its title bar and then check the settings in the following boxes:

<table>
<thead>
<tr>
<th>Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP Workobject</td>
<td>Specify the workobject to which the targets generated for the workpiece robot shall belong.</td>
</tr>
<tr>
<td>WP TCP</td>
<td>Specify which tooldata the workpiece shall use when reaching its targets.</td>
</tr>
<tr>
<td>Path prefix</td>
<td>Specify a prefix for the generated paths.</td>
</tr>
<tr>
<td>Target prefix</td>
<td>Specify a prefix for the generated targets.</td>
</tr>
</tbody>
</table>

5. Expand the Generate path group by clicking on its title bar and then click Create Paths.
3.9.6. Programming external axes

Overview

This is a brief overview of the functions and commands for programming external axes in RobotStudio. For a more detailed description of external axes and how to program them, see the product manual for the external axis to use and the RAPID reference manual.

Coordinated motions

Normally, external axes are used to move the workpiece, the robot or any other mechanism. The motions of an external axis can be coordinated with those of a robot in two ways, depending on the task in which the external axis is defined.

<table>
<thead>
<tr>
<th>Task for external axis</th>
<th>Coordination method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same task as the robot’s</td>
<td>If the external axis is in the same task as the robot, the current position of active external axes is stored with each target that is created. When the robot then moves to the target, the external axis will move to the stored position as well. Modifying and optimizing the position of positioner external axes can be automated using the MultiMove function, or be performed manually for selected targets. Positions of track external axes can only be modified manually. For information about using the MultiMove function, see About programming MultiMove on page 102. For information about how to modify the position of external axes manually, see below.</td>
</tr>
<tr>
<td>Other task than the robot’s</td>
<td>If the external axis is in another task than that of the robot it shall be coordinated with, the motions of the external axis are created by MoveExt instructions, and the coordination is made by sync instructions. For positioner external axes creating or optimizing MoveExt and sync instructions can be automated way using the MultiMove function, or be performed manually by creating a path with MoveExt instructions for the positioner and then adding sync instructions to the path for the robot and the external axis. Track external axes can only be programed manually. For information about using the MultiMove function see About programming MultiMove on page 102. For information about how to use the sync instructions, see the RAPID reference manual and the MultiMove application manual.</td>
</tr>
</tbody>
</table>
3 How to program robots

3.9.6. Programming external axes

Continued

Modifying positions of external axes

When programming external axes, you often need to adjust the position of the external axis for some targets. For example, if you create a path from curves on a work piece that is attached to a positioner, the positioner will initially have the same position for all targets. By repositioning the work piece for some of the targets you might improve process time and reachability.

When targets are created in stations with a coordinated external axis, the position values of the external axis are stored in the target. With the Modify External Axis function you can reposition the external axis, thus making it possible for the robot to reach the target in new ways. For a procedure, see Modify External Axis on page 349.

To modify the external axis values for a target, the following conditions must be met:

- The external axis must be added to the system and set up correctly. For examples of how to add support for an external axis to a system, see A system with support for one robot and one track external axis and A system with support for one robot and one positioner external axis on page 155. For information about how to set up an external axis in a RobotStudio station, see Placing external axes on page 79.
- The external axis must be defined in the same task as the robot.
- The external axis must be activated.

Activation and deactivation

Activating a mechanical unit makes it controlled and monitored by the controller. Consequently, the mechanical unit must be activated before programming or running programs. If a system uses several external axes or interchangeable models with several work stations, several mechanical units might share common drive units. If this is the case, you must make sure to set the mechanical unit as active.

For more information about activating and deactivating mechanical units, see the RAPID reference manual on the instructions ActUnit and DeactUnit.

Activating and deactivating mechanical units can be done either manually, see Activate Mechanical Units on page 264, or programmatically by RAPID instructions, see below.

To activate or deactivate mechanical units programmatically

To set the mechanical units to be active programmatically by RAPID instructions, follow these steps:

1. In the Paths&Targets browser, browse down to the path in which you want to insert the activation or deactivation instruction. To insert it as the first instruction in the path, select the path node and to insert it between existing instructions, select the instruction before the intended insertion point.

2. On the Home tab, click Action Instruction to bring up a dialog box.

3. In the Instruction Templates list, select one of the ActUnit or DeactUnit instructions.

4. In the Instruction Arguments grid and the MechUnit list, select the unit to activate or deactivate.

5. Click Create. When the path is executed either through the Move along path command, or running the RAPID program, the instruction will be carried out.
3.10. Loading and saving programs and modules

Overview

RAPID programs and modules are normally stored in the RobotWare systems, as they are created. You can also save the programs to files on the PC, which makes it possible to load them to other controllers, either other virtual controllers or real IRC5 controllers.

Programs are saved from the VC

When saving a program to files on the PC from RobotStudio, it is the RAPID program stored in the system of the VC that is saved. This program is created and updated by synchronizing the station to the VC, see Synchronize to VC on page 275.

Procedures

To create or load a module or load a program, see:

• New Module on page 289.
• Load Module on page 290.
• Load Program on page 291.

To save a module or program, see:

• Save Module As on page 363.
• Save Program As on page 364.
3.11. Synchronization

Overview

To synchronize is to make sure that the RAPID program in the system running on the virtual controller corresponds to the programs in RobotStudio. You can synchronize both from RobotStudio to the virtual controller and from the virtual controller to RobotStudio.

In a RobotStudio station, robot positions and movements are defined by targets and move instructions in paths. These correspond to data declarations and RAPID instructions in the modules of the RAPID program. By synchronizing the station to the virtual controller, you create RAPID code out of the data in the station. By synchronizing the virtual controller to the station, you create paths and targets out of the RAPID program in the system running on the virtual controller.

When to synchronize the station to the VC

Synchronizing the station to the VC updates the RAPID program of the virtual controller with the latest changes in the station. This is useful to do before:

- Performing a simulation.
- Saving a program to files on the PC.
- Copying or loading RobotWare systems.

To synchronize a station to the VC, see Synchronize to VC on page 275.

When to synchronize the VC to the station

Synchronizing the VC to the station creates paths, targets and instructions that correspond to the RAPID program in the system running on the virtual controller. This is useful to do when you have:

- Started a new virtual controller which system contains existing programs.
- Loaded a program from a file.
- Text-edited the program.

To synchronize the VC to a station, see Synchronize to Station on page 274.
3.12. Using the RAPID editor

Overview

The integrated RAPID program editor is useful for editing all robot tasks other than robot motion. This section provides instructions on launching the editor and case examples illustrating several useful functions, such as keyboard shortcuts, IntelliSense, code snippets and the watch window.

Opening the program editor

To access the program editor, perform the following steps:

1. Click the Offline tab.
2. In the Program browser, select a module for editing and then click RAPID Editor.

The RAPID code of the program appears as a tabbed document window.

TIP!
The graphical layout can be consulted without closing the editor by clicking the graphics window tab.

Searching: a case example

Assume that you have programed targets and motion instructions and synchronized them to the controller. The number of targets is large, so you decide to distribute them among several modules.

You may have forgotten in which module your main procedure is found.

1. Press CTRL + F to bring up the Search and Replace dialog box.
2. In the Search Pattern box, type "PROC main". Since no modules are open, in the Search In list, select Current System, and then click Search All.

The search result is displayed in the output window.

3. Double-click the line matching your search to launch the program editor.

The entire RAPID program is then checked for errors.
3 How to program robots

3.12. Using the RAPID editor

Continued

Code Snippets

Code Snippets are pieces of code inserted in the RAPID Editor. The editor integrates these codes with the pick list. The following are the predefined code snippets in RobotStudio:

- array2x2x4.snippet
- array2x4.snippet
- array2x4x2.snippet
- array4x2.snippet
- function with return value bool.snippet
- module header.snippet
- procedure parameters.snippet
- procedure with error handler.snippet
- robtarget.snippet
- tooldata.snippet
- TRAP routine example.snippet
- wobjdata.snippet

You can create customized code snippets and add to the existing list.

**NOTE!**


Editing: a case example

Assume that you wish to create an infinite loop whereby the controller receives commands from a line PLC. The controller communicates with the PLC using digital I/O signals, but you have forgotten the exact name of the function that reads an input signal.

1. Using code snippets, create a new procedure.
2. Press **CTRL + SHIFT + SPACE** to open the pick list.
3. Double-click the I/O folder, and then double-click the **DOoutput** instruction to insert it at the insertion position.
4. Press the spacebar to display the parameter information ToolTip. As you enter parameters, the ToolTip is updated, displaying the current argument in bold. The ToolTip is closed either by concluding the instruction with a semicolon (;), or by pressing **ESC**.
3 How to program robots

3.12. Using the RAPID editor

Continued

TIP!
At any time you may press CTRL + SHIFT to complete what you have begun typing. This will either bring up a narrowed-down list of selectable parameters, or, if only one selection remains, will automatically complete your text.

Adding breakpoints: a case example
Now that you have finished editing, you may want to test your loop and add some breakpoints.

1. Place the insertion on the new statement and press F9 to set a breakpoint.
   The breakpoint, indicated by a grey circle, is stored in the editor until changes are applied, whereupon it turns red.

2. Ensure that the Ignore breakpoints button in the Program editor toolbar is not clicked, and click the Play button on the Simulation toolbar.
   The program will run and then stop at the breakpoint.

3. To run the program statement by statement, click the Step over button in the Program editor toolbar.

Applying and verifying the edits
To apply and verify your edits, follow these steps:

1. To apply your edits, click the Apply icon in the editor toolbar.
   The statement is highlighted, and breakpoint locations are adjusted, if necessary.

2. To verify the syntactic and semantic correctness of the modules, click the Check Program icon in the editor toolbar.

Executing: a case example
You might want to debug your loop or monitor a specific variable.

1. In the program editor browser, right-click the procedure you want to set as entry point, and then click Set Entry.
   The icon is marked in red.

2. In the Simulation tab, click the Play button.
   The program will run and then stop at the next breakpoint.

3. On the View menu, click Watch Window to bring up the watch window.

4. Select a variable for monitoring and drag it to the watch window.

5. Restart the loop and monitor the variable at each iteration.
3 How to program robots

3.12. Using the RAPID editor
4 How to simulate programs

4.1. Simulation Overview

About this chapter

This chapter describes how to simulate and validate robot programs. Below are short introductions to the simulation functions in RobotStudio.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play simulations</td>
<td>Simulations run entire robot programs on a virtual controller. Before you run a simulation you need to decide which paths are to be simulated. To set up a simulation, see Simulation Setup on page 257. To run a simulation, see Simulation Control on page 265.</td>
</tr>
<tr>
<td>Collision detection</td>
<td>Collision detection displays and logs collisions and near-misses for specified objects in the station. Normally used during simulation of robot programs, it can also be used when building the station. For more information, see Detecting collisions on page 119.</td>
</tr>
<tr>
<td>Event handling</td>
<td>Events can be used to connect an action to a trigger. For example, you can attach one object to another when they collide or a signal is set. For more information, see Creating an event on page 122.</td>
</tr>
<tr>
<td>I/O Simulation</td>
<td>In simulations I/O signals are normally set either by the robot program or by events. With the I/O simulator you can set signals manually, which provides a quick test of specific conditions. For more information, see Simulating I/O signals on page 123.</td>
</tr>
<tr>
<td>Simulation Monitoring</td>
<td>With the simulation monitoring functions you enhance the simulation by adding traces along the TCP movements or alerts triggered by defined speeds or motions. For more information, see Enabling simulation monitoring on page 124.</td>
</tr>
<tr>
<td>Process time measurement</td>
<td>With the process timer you measure the time for a process to complete. For more information, see Measuring process time on page 125.</td>
</tr>
</tbody>
</table>

Time handling during simulation

When simulating stations with events or several controllers, or other time managing equipment, time can be managed in two modes: either as free runtime or as time slices. RobotStudio uses time slice mode by default, but you can switch to free runtime, if required.

Free runtime

Since all controllers use the same computer resources, their synchronization might not be exactly as in the real world if they run independently of each other (called free run mode). The cycle time will be correct, but the timing for setting signals and triggering events might be inaccurate.

Time Slice

Time slices can be used to ensure that the timing for signals and other interaction between controllers is accurate. In this mode, RobotStudio synchronizes the controllers by dividing a time segment into small slices and waiting for all controllers to complete a current time slice before any controller can start anew. Thus, the controllers are synchronized, and the cycle

Continues on next page
time will be calculated correctly. The drawbacks are that the virtual FlexPendant cannot be open, and that the simulation might be somewhat slow and jerky, depending on the complexity of the simulation and the performance of the computer.

**NOTE!**

If the simulation uses events or involves several different controllers, the virtual time mode **Time Slice** shall be used to make sure that the timing between the controllers is correctly simulated.
4 How to simulate programs

4.2. Detecting collisions

Overview

With RobotStudio you can detect and log collisions between objects in the station. The basic concepts of collision detection are explained below.

Collision sets

A collision set contains two groups, Objects A and Objects B, in which you place the objects to detect any collisions between them. When any object in Objects A collides with any object in Objects B, the collision is displayed in the graphical view and logged in the output window. You can have several collision sets in the station, but each collision set can only contain two groups.

A common use of collision sets is to create one collision set for each robot in the station. For each collision set you then put the robot and its tool in one group and all objects you do not want it to collide with in the other. If a robot has several tools, or holds other objects, you can either add these to the robot’s group as well or create specific collision sets for these setups. Each collision set can be activated and deactivated separately.

Collisions and near-misses

In addition to collisions, the collision detection can also watch for near-misses, which is when an object in Objects A comes within a specified distance from an object in Objects B.

Recommendations for collision detection

In general, the following principles are recommended to facilitate collision detection:

- Simplify and defeature your models by removing everything that is not necessary for simulation purposes.
- Use as small collision sets as possible, splitting large parts and collecting in the collision sets only relevant parts.
- Enable coarse detail level when importing geometry.
- Limit the use of near-miss.
- Enable last collision detection, if the results are acceptable.

Results of creating a collision set

After you have created a collision set, see Create Collision Set on page 256, RobotStudio will check the positions of all objects and detect when any object in Objects A collides with any object in Objects B.

Activation of detection and displayal of collisions depend on how the collision detection is set up.

If the collision set is active, RobotStudio will check the positions of the objects in the groups, and indicate any collision between them according to the current color settings.

Continues on next page
4 How to simulate programs

4.2. Detecting collisions

Collision detection checks whether robots or other moving parts collide with equipment in the station. In complex stations, you can use several collision sets for detecting collisions between several groups of objects.

After collision detection has been set up, it does not need to be started, but automatically detects collisions according to the setup.

Setting when to check for collisions

To set whether to detect collisions always or only during simulation, follow these steps:

1. On the Application menu, click RobotStudio Options and under Simulation, click Collision.

2. In the Perform collision detection option, select one of the following:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>During simulation</td>
<td>Collision detection is active only during simulation (when running RAPID programs in the virtual controller).</td>
</tr>
<tr>
<td>Always</td>
<td>Collision detection is always active, even when moving objects manually or testing reachability.</td>
</tr>
</tbody>
</table>

Setting the objects for collision detection

To set the objects for collision detection, follow these steps:

1. Make sure that the objects for collision detection are placed correctly in collision sets.

2. Make sure that the collision set for the objects is activated, which is indicated by an icon in the Layout browser:

To activate or deactivate collision sets, continue with the following steps:

3. Right-click the collision set to change and then click Modify Collision set to bring up a dialog box.

4. Select or clear the Active check box and then click Apply.

Setting near-miss detection

Near-misses occur when objects in collision sets are close to colliding. Each collision set has its own near-miss settings. For setting near-miss detection, follow these steps:

1. In the Layout browser, right-click the collision set to change and then click Modify Collision set to bring up a dialog box.

2. In the Near miss box, specify the maximum distance between the objects to be considered a near-miss and then click Apply.

Continues on next page
4 How to simulate programs

4.2. Detecting collisions

Setting logging options

In addition to the graphical display of collisions, you can also log the collisions to the output window or a separate log file:

1. On the Application menu, click RobotStudio Options and under Simulation, click Collision.

2. For displaying the collision log in the output window, select the Log collisions to Output window check box.

3. For logging collisions to a separate file, select the Log collisions to file check box and enter the name and path to the log file in the box below the check box.
4.3. Creating an event

Overview

Events enhance your simulations by defining actions that are carried out when specific trigger conditions are fulfilled. You can use events to:

- Attach one object to another, for example, a work piece to a gripper when simulating material handling, see *Attaching and detaching objects on page 48*.
- Set signals, for example, when simulating signals set by equipment other than the controller, see *Simulating I/O signals on page 123*.
- Start or stop the process timer, see *Measuring process time on page 125*.

Used for creating new events, the **Create New Event Wizard** is launched from the Event manager, see *Event Manager on page 258*.

Prerequisites

Before creating the event, make sure that the station contains all signals and objects that are planned to be used as triggers or affected by the action.
4.4. Simulating I/O signals

Procedures

When simulating I/O signals you can either create events that set signal values when specified trigger conditions are fulfilled, or you can set signal values manually.

For procedures using the event manager, see Event Manager on page 258.

For procedures using the I/O simulator, see I/O Simulator on page 266.

Related information

For information about controlling I/O signals from the RAPID program, see Creating RAPID instructions for setting I/O signals on page 95.
4 How to simulate programs

4.5. Enabling simulation monitoring

4.5. Enabling simulation monitoring

Overview

The simulation monitor commands are used to visually detect critical robot movements during simulation by drawing a colored line that follows the TCP.

To enable TCP tracing

To enable TCP tracing, follow these steps:

1. On the Simulation tab, click Monitor to bring up a dialog box.
2. In the left pane, select the appropriate robot.
3. On the TCP Trace tab, select the Enable TCP Trace check box. This activates TCP tracing for the selected robot.
4. Optionally, change the length and color of the trace. For detailed information, see Monitor on page 268.

To enable simulation alerts

To enable simulation alerts, follow these steps:

1. On the Simulation menu, click Monitor to bring up a dialog box.
2. In the left pane, select the appropriate robot.
3. On the Alerts tab, select the Enable Simulation Alerts check box. This activates simulation alerts for the selected robot.
4. In the threshold value boxes, specify the threshold for the alerts. Setting the threshold to 0 is equivalent to disabling the alert. For detailed information, see Monitor on page 268.
4.6. Measuring process time

Overview

The process timer measures the time it takes for robots to run a simulation or move along a path. When the timer is activated, the clock will start as soon as a process is started. If two processes are started at the same time, the timer will not stop until the last one has stopped.

To measure the process time

To measure the process time, follow these steps:

1. On the Simulation tab, click the Clock button.
2. Play the simulation or move along path.
   
   The timer stops when the last process has stopped. If you start another process without having deactivated the Clock button, the timer will continue.
3. To reset the clock to zero, click the Reset Process Timer.
4 How to simulate programs

4.6. Measuring process time
5 Deployment and distribution

4.6. Measuring process time
5 Deployment and distribution

5.1. Copying programs

Overview

RAPID programs are normally stored in the systems that run on the virtual controllers of your station. To copy programs to systems on other controllers, save the programs to file on the PC and then load these files to the destination controllers. You can save either entire programs or specific modules.

Copying a program

To copy a program from one controller to another, follow these steps:

1. In the Offline browser, select the controller that contains the program to copy.
2. Save the program to file on the disc. For details, see Saving a program on page 364.
3. If necessary, copy the files to a location that is accessible to the other controller.
4. For instructions on loading the program to a system on a virtual controller, a FlexController or a non-running system, see the table below.

<table>
<thead>
<tr>
<th>System location</th>
<th>Do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual controller, running in RobotStudio</td>
<td>See Saving a module on page 363.</td>
</tr>
<tr>
<td>FlexController</td>
<td>Connect to the FlexController and load the program.</td>
</tr>
<tr>
<td>A non-running system stored on the PC</td>
<td>Start the system in a virtual controller, then load the program, see Adding a system on page 65 and Saving a module on page 363, respectively.</td>
</tr>
</tbody>
</table>
5.2. Pack & Go / Unpack & Work

Overview

The Pack & Go / Unpack & Work feature makes it possible to create a package (zip file) of an active station that can be unpacked on another computer. The package contains all necessary files, except media pools, but additional, option-based media pools are included. For procedures, see Pack & Go on page 174 and Unpack & Work on page 175.
5 Deployment and distribution

5.3. Screen Capture

5.3. Screen Capture

Overview

Screen capture entails two functions useful for demonstrations and training purposes:

- The screenshot function allows you to capture an image of the application, see Screenshot on page 173.
- The recorder allows you to make a recording of your work in RobotStudio, either of the entire GUI or just the graphics window, see Record Movie on page 269.

Prerequisites

Recording a movie requires that Windows Media Encoder (WME) 9 is installed on your computer.
6 Working online

5.3. Screen Capture
6.1. Connecting a PC to the service port

NOTE!
The service port shall only be used for direct connection to a PC as described in this procedure. It must not be connected to a LAN (local area network), since it has a DHCP server that automatically distributes IP addresses to all units connected to the LAN.

Contact your network administrator for more information.

NOTE!
The maximum number of connected network clients using robapi is:

- LAN: 3
- Service: 1
- FlexPendant: 1
6 Working online

6.1. Connecting a PC to the service port

The total max. number of applications using robapi running on the same PC connected to one controller has no built-in maximum however, UAS limits the number of logged on users to 50.

The total max. number of concurrently connected FTP clients is 4.

**CAUTION!**

When a cable is connected to the service port on the controller front and the service hatch or the connector cover on the single cabinet is opened, the controller will not comply with the requirements of protection class IP54.

**Ports DSQC639**

The illustration below shows the two main ports on the computer unit: the Service Port and the LAN port. Make sure the LAN (factory network) is not connected to any of the service ports!

| A | Service port on the computer unit (connected to Service port on the Control Module front through a cable) |
| B | Service port on the computer unit |
|   | LAN port on computer unit (connects to factory LAN) |
Connections to ports may be done as detailed below.

<table>
<thead>
<tr>
<th>Connection to/from:</th>
<th>Detailed in section:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connecting the Single Cabinet/Control Module to the factory LAN.</td>
<td></td>
</tr>
<tr>
<td>Connecting a PC to the Single Cabinet/Control Module service port.</td>
<td>Proceed as detailed below.</td>
</tr>
</tbody>
</table>

**Connecting a PC to the service port**

<table>
<thead>
<tr>
<th>Action</th>
<th>Illustration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Make sure the network setting on the PC to be connected is correct.</td>
<td>Refer to the system documentation for your PC, depending on the operative system you are running. The PC must be set to “Obtain an IP address automatically” or set as described in Service PC Information in the Boot Application.</td>
</tr>
<tr>
<td>2. Use the delivered category 5 Ethernet crossover boot cable with RJ45 connectors.</td>
<td>The cable is delivered in the RobotWare product box.</td>
</tr>
<tr>
<td>3. Connect the boot cable to the network port of your PC.</td>
<td>![Network Port Illustration]</td>
</tr>
<tr>
<td>4. Connect the boot cable to the service port on the controller. Connect the boot cable to the service port on the computer unit.</td>
<td>![Service Port Illustration]</td>
</tr>
</tbody>
</table>
6.2. Network settings

Overview

This topic describes the network settings for a PC connected to a controller, which is a prerequisite for working online.

You can connect the PC to the controller through an Ethernet network in the following ways:

- Local network connection
- Service port connection
- Remote network connection

Local network connection

You can connect your PC to the same Ethernet network that the controller is connected to. When the PC and the controller are connected correctly and to the same subnet, the controller will be automatically detected by RobotStudio.

The network settings for the PC depend on the network configuration. For setting up the PC, contact the network administrator.

Service port connection

When connecting to the controller's service port, you can either obtain an IP address for the PC automatically, or you can specify a fixed IP address.

If you are not sure how to set up the service port connection, contact the network administrator.

Automatic IP address

The controller's service port has a DHCP server that will automatically give your PC an IP address if it is configured for this. For detailed information see Windows help on configuring TCP/IP.

Fixed IP address

Instead of obtaining an IP address automatically, you can also specify a fixed IP address on the PC you connect to the controller.

Use the following settings for a fixed IP address:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address</td>
<td>192.168.125.2</td>
</tr>
<tr>
<td>Subnet mask</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

For detailed information about how to set up the PC network connection, see Windows help on configuring TCP/IP.

Continues on next page
NOTE!

Obtaining an IP address automatically might fail if the PC already has an IP address from another controller or Ethernet device.

To ensure that you get a correct IP address if the PC was previously connected to an Ethernet device, do one of the following:

- Restart the PC before connecting to the controller.
- Run the command `ipconfig /renew` from the command prompt after connecting the PC to the controller.

Remote network connection

To enable a connection to a controller on a remote subnet, the relevant network traffic must be allowed through any firewalls between the PC and the controller.

The firewalls must be configured to accept the following TCP/IP traffic from the PC to the controller:

- UDP port 5514 (unicast)
- TCP port 5515
- Passive FTP

All TCP and UDP connections to remote controllers are initiated by the PC, that is the controller only responds on the given source port and address.

Connecting to the controller

1. Make sure the PC is connected to the controller's service port and that the controller is running.

2. On the Application menu, point to Online and then click One Click Connect.

3. Select the controller in the Online tab

4. Click Request Write access.

<table>
<thead>
<tr>
<th>If the controller is in mode</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>You will now get Write Access if it is available.</td>
</tr>
<tr>
<td>Manual</td>
<td>A message box on the FlexPendant will allow you to grant remote Write Access to RobotStudio Online.</td>
</tr>
</tbody>
</table>
6.3. User Authorization

Overview

This section describes the controller's User Authorization System (UAS), which restricts what different users are allowed to do with the robot. This is for protecting data and functionality from unauthorized use.

The user authorization is managed by the controller, which means that the UAS settings remain for the controller regardless of which system it is running. It also means that the UAS settings apply to all tools for communicating with the controller, like RobotStudio Online or the FlexPendant. The UAS settings defines the users and groups that can access the controller, and what actions they are granted access to.

For procedures, see User Accounts on page 306.

Users

In UAS users are accounts with which persons log on to the controller. Furthermore, the users are added to groups to which access grants are given.

The users are defined in the controller by a user name and a password. For logging on to a controller, the user must type in a defined user name with a correct password.

A user can either have the state activated or deactivated in the UAS. When a user is deactivated it is not possible to log on to the controller using that account. It is the UAS administrator that activates and deactivates the users.

The Default user

All controllers have a default user named Default User, which does not have any password and cannot be removed. Other users can be added and removed by the user having the grant Manage UAS settings.

Groups

In UAS, groups are defined sets of grants for accessing the controller. To the groups you then add the users who shall have the grants defined by the group.

A good practice is to create groups that resembles the professions that work with the robots in your organization. For example, you can create groups for administrators, programmers and operators.

The Default group

All controllers have a default group named Default Group, to which all grants are given and to which the default user belongs. This group cannot be removed, but it can be changed by the user having the grant Manage UAS settings.

Continues on next page
NOTE!
There is a risk changing the group membership of the default user. If you by mistake clear the Default User check box or any Default Group grant, you will get a warning. Make sure that there is at least one user defined that has the grant Manage UAS settings. If the Default group and no other group have the grant Manage UAS settings, you may lose your ability to manage users and groups.

Grants

Grants are permissions to perform actions or access data on the controller. You use the grants by giving them to groups, to which you then add the users who shall have the grants.

Grants may be either controller grants or application grants. Depending on the actions that you will perform, you may need several grants. For procedures, see UAS Grant Viewer on page 311.

Controller grants

Controller grants are validated by the robot controller and apply to all tools and devices accessing the controller.

Application grants

Application grants are used by a specific application, for example the FlexPendant, and will only be valid using that application. Application grants can be added by additional options and used in customer applications.
6 The System Builder

6.4 System Builder Overview

Overview

This section describes how you create, build, modify and copy systems to run on virtual and real controllers. These systems may even be converted to boot media and downloaded to a real controller.

The system points out the robot models and options to use; it also stores configurations and programs for the robots. Therefore, it is good practice to use a unique system for each station even if the stations use the same basic setup. Otherwise, changes in one station may accidently overwrite data used in another station.

About virtual and real systems

The system you run on virtual controllers can either be a real system built on real RobotWare keys or a virtual system built on virtual keys.

When using real systems, the RobotWare keys define which options and robot models shall be used, thus helping you to configure the system correctly. Real systems can be run both on virtual controllers and real IRC5 controllers.

When using virtual keys, all options and robot models are available, which is useful for evaluation purposes, but requires more configuration when creating the system. Systems built on virtual keys can only be run on virtual controllers.

Prerequisites

Creating a system entails applying a predefined template to a station, reusing an existing system or letting RobotStudio propose a system based on a layout.

To create a system, the following conditions must be met:

- The RobotWare media pool must be installed on your PC.
- You must have a RobotWare key for the system, if creating a system to run on a real controller. The RobotWare key is a license key that determines which robot models to use and which RobotWare options to run on the controller. The license key is delivered with the controller.
- If you want to create a system for virtual use only, you can use a virtual key instead. Virtual keys are generated by the wizard. When using virtual keys, you select the robot models and options to use in the Modify Options section of the wizard.
- Downloading to the real controller requires a direct connection from your computer to the service or Ethernet port of the controller.
Administering systems

Systems can be administered from the System Builder dialog box in the following ways:

- View system properties, see Viewing system properties on page 141.
- Build a system, see Building a new system on page 142.
- Modify or delete a system, see Modifying a system on page 145.
- Copy a system, see Copying a system on page 149.
- Create a system from backup, see Creating a system from backup on page 150.
- Download a system to a controller, see Downloading a system to a controller on page 151.
- Create boot media, see Creating boot media on page 152.
6.4.2. Viewing system properties

Overview

All systems you create with the System Builder are stored locally on your computer. It is recommended that you store them in one or more dedicated system directories.

Viewing system properties

To view system properties and add comments, follow these steps:

1. In the System Builder dialog box, select a system from the Systems box.
   If necessary, in the System directory list, you can navigate to the folder in which your systems are stored

2. The system properties are then displayed in the System Properties box. Optionally, type a comment in the Comments box, and click Save.
6 Working online

6.4.3. Building a new system

Overview

The New Controller System Wizard, used for building a new system, is launched from the System Builder.

Starting the wizard

To start the wizard, follow these steps:
1. Click System Builder to bring up a dialog box.
2. In the Actions group, click Create New. This starts the wizard.
3. Read the information on the welcome page and click Next.

Specifying the name and location

To determine where on your computer to store the system you are creating, follow these steps:
1. In the Name box, enter a name for the system you are creating.
2. In the Path box, enter the path to the system directory in which you will store the system.
   You can also click the Browse button and browse to the system directory.
3. Click Next.

Entering the RobotWare keys

The RobotWare keys determine which RobotWare versions and parts to use in the system. Creating a system to run on either IRC5 controllers or virtual controllers requires at least two keys: one for the controller module and one for each drive module in the cabinet. The keys are delivered together with the controller.

For creating a system to run on virtual controller only (for example, in Virtual IRC5), you can use virtual keys. Virtual keys give access to all options and robot models, but limits the use of the system to virtual controllers only.

To enter the key for the controller module, follow these steps:
1. In the Controller Key box, enter the controller key. You can also click Browse and browse to the key file. If creating a system for virtual use only, select the Virtual Key check box, and the controller key will be generated by the wizard.
2. In the Media Pool box, enter the path to the media pool. You can also click Browse and browse to the folder system
3. In the RobotWare Version list, select which version of the RobotWare you want to use. Only RobotWare versions that are valid for the used key are available.
4. Click Next.

Continues on next page
6 Working online

6.4.3. Building a new system

Continued

Entering the drive keys

To enter the keys for the drive modules:

1. In the Drive Key box, enter the key for the drive module. You can also click the Browse button and browse to the key file. If you used a virtual controller key, a virtual drive key is already generated by the wizard.

2. Click the right arrow button next to the Drive Key box. The key now appears in the Added drive key list.

For real systems the drive key determines the connected robot model. For virtual systems you select the robot model in the Modify Options page. The default model is IRB140.

3. If you have a MultiMove system, repeat steps 1 and 2 for each drive key to add.

If you have a MultiMove system, make sure that the keys are numbered in the same way as their corresponding drive modules are connected to the controller module. Use the up and down arrows to rearrange the drive keys, if necessary.

4. Choose whether you want to create the system as it is now, or to continue with the wizard.

If you want to create the system as it is now, click Finish.

If you want to modify options, or add options, parameter data or additional files to the home directory, click Next.

Adding additional options

Here you can add options, such as external axes and dispense applications, that are not included in the basic system. Options require a license key and must be first imported to the media pool. To add additional options, follow these steps:

1. In the Key box, enter the option key. You can also click the Browse button and browse to the option's key file.

2. Click the Arrow button.

The option that the key unlocks is now displayed in the Added Options list.

3. Repeat steps 1 and 2 for all options you want to include.

4. Choose whether you want to create the system as it is now, or to continue with the wizard.

If you want to create the system as it is now, click Finish.

If you want to modify options, or add parameter data or additional files to the home directory, click Next.

Modifying options

Here you can set up and configure the options in your system. For virtual systems, you also select the robot models to use. To modify any options, follow these steps:

1. In the Option tree, expand the option folders to the level where you find the option you want to modify.

Only the options unlocked by the used keys are available.

2. Modify the option.
Continued

3. Repeat steps 1 and 2 for all options you want to modify.

4. Choose whether you want to create the system as it is now, or to continue with the wizard.
   If you want to create the system as it is now, click **Finish**.
   If you want to add parameter data or additional files to the home directory, click **Next**.

### Adding parameter data

Parameter data is stored in the parameter data files (.cfg files). Each parameter topic has its own parameter file. You can add only one parameter file for each topic. To add parameter data, follow these steps:

1. In the **Parameter data** box, enter the path to the folder for the parameter data files. You can also click the **Browse** button and browse to the folder.
2. In the list of parameter data files, select the file you want to include and press the **Arrow** button. Repeat for all files you want to include.
   The included parameter data files will now appear in the **Added parameter data files** list. Repeat steps 1 and 2 for each parameter data file you want to add.
3. Choose whether you want to create the system as it is now, or to continue with the wizard.
   If you want to create the system as it is now, click **Finish**.
   If you want to add additional files to the home directory, click **Next**.

### Adding files to the home directory

You can add any type of file to the system’s home directory. When the system is loaded to a controller, these files will also be loaded. To add files to the system’s home directory, follow these steps:

1. In the **Files** box, enter the path to the folder for the files you want to include. You can also click the **Browse** button and browse to the folder.
2. In the list of files, select the file to add and click the **Arrow** button. Repeat for all files you want to add.
   The added files will now appear in the **Added files** list.
3. Choose whether you want to create the system as it is now, or to continue with the wizard.
   If you want to create the system as it is now, click **Finish**.
   If you want to read a summary before you create the system, click **Next**.

### Completing the New Controller System Wizard

To complete the wizard, follow these steps:

1. Read the system summary.
2. If the system is OK, click **Finish**.
   If the system is not OK, click **Back** and make modifications or corrections.
6.4.4. Modifying a system

Overview

The Modify Controller System Wizard, used to modify existing systems, is launched from the System Builder. The wizard helps you with tasks like changing robots, adding and removing external axes and other options. A system that is running must be first shut down before modification.

Starting the wizard

To start the wizard when creating a new station:

1. If the system is currently running, on the Controller menu, point to Shutdown and then click Shutdown.
2. On the Controller menu, click System Builder to bring up a dialog box.
3. In the System directory list, enter or browse to the system directory. Select a system from the list beneath, review the system properties and add and save any comments.
4. In the Actions group, click Modify. This starts the wizard.
5. Read the information on the welcome page and click Next.

Modifying the program revision

The RobotWare versions that are available for the system are determined by the controller key. The key is essential to the system and cannot be modified.

To use another RobotWare version than the available ones, create a new system with another key.

To optionally modify the program revision, follow the appropriate step or steps:

1. To keep the current RobotWare version, select Yes and then click Next.
2. To replace the current RobotWare version, Select No, replace it.
3. In the Media pool box, enter the path to the media pool. You can also click the Browse button and browse to the folder.
4. In the New program revision box, select which version of RobotWare you want to use. Only RobotWare versions that are valid for the RobotWare key are available.
5. Click Next.
Adding or removing drive keys

The drive key corresponds to the drive modules in your controller. For MultiMove systems, you have one drive module (and one key) for each robot. The keys for your system are delivered together with the controller.

The system is created with a virtual controller key, virtual drive keys are generated by the wizard. When you have added one virtual drive key for each robot, you select which robot to use for each key on the Modify Options page.

To optionally add or remove the keys for the drive modules, follow these steps:

1. To add a key for a drive module, enter the key in the Enter Drive Key box. You can also click the Browse button and browse to the key file.

2. Click the right arrow button. The key now appears in the Added drive key list.
   If you have a MultiMove system, repeat steps 1 and 2 for each drive key to add.

3. To remove a drive module, select the corresponding key in the Added drive key list and click Remove drive key.
   If you have a MultiMove system, repeat step 3 for each drive key to remove.

4. If you have a MultiMove system, make sure that the keys are numbered in the same way as their corresponding drive modules are connected to the controller module. Use the up and down arrows to rearrange the drive keys, if necessary.

5. Choose whether you want to create the system as it is now, or to continue with the wizard.
   If you want to create the system as it is now, click Finish.
   If you want to modify options, parameter data or add files to or remove files from the home directory, click Next.

Adding or removing additional options

To optionally add or remove additional options:

1. To add an additional option, in the Enter Key box, enter the option key. You can also click the Browse button and browse to the option's key file.

2. Click the Arrow button.
   The option that the key unlocks is now displayed in the Added Options list.

3. Repeat steps 1 and 2 for all options you want to include.

4. To remove an additional option, in the Added options list, select the option you want to remove.

5. Click Remove.

6. Choose whether you want to create the system as it is now, or to continue with the wizard.
   If you want to create the system as it is now, click Finish.
   If you want to modify parameter data or add files to or remove files from the home directory, click Next.
Modifying options

To optionally modify any options, follow these steps:

1. In the Option tree, expand the option folders to the level where you find the option you want to modify.
   
   Only the options unlocked by the used keys are available.

2. Modify the option.

3. Repeat steps 1 and 2 for all options you want to modify.

4. Choose whether you want to create the system as it is now, or to continue with the wizard.
   
   If you want to create the system as it is now, click Finish.
   
   If you want to modify parameter data or add files to or remove files from the home directory, click Next.

Adding or removing parameter data

Parameter data is stored in the parameter data files (.cfg files). each parameter topic has its own parameter file. You can add only one parameter file for each topic. To add or remove parameter data, follow these steps:

1. To add parameter data, in the Parameter data box, enter the path to the folder for the parameter data files. You can also click the Browse button and browse to the folder.

2. In the list of parameter data files, select the file you want to include and press the Arrow button. Repeat for all files you want to include.
   
   The included parameter data files will now appear in the Added parameter data files list. Repeat steps 1 and 2 for each parameter data file you want to add.

3. To remove parameter data, in the Added parameter data files list, select the parameter data file to remove.

4. Click Remove.

5. Choose whether you want to create the system as it is now, or to continue with the wizard.
   
   If you want to create the system as it is now, click Finish.
   
   If you want to add to or remove files from the home directory, click Next.

Add files to or remove files from the home directory

You can add any type of file to the system’s home directory, or remove files from it. When the system is loaded to a controller, these files will also be loaded. To optionally add files to or remove files from the system’s home directory, follow these steps:

1. To add files, in the Files box, enter the path to the folder for the files you want to include.
   
   You can also click the Browse button and browse to the folder.

2. In the list of files, select the file to add and click the Arrow button. Repeat for all files you want to add.
   
   The added files will now appear in the Added files list.

3. To remove files, in the Added files list, select the file to remove.
6 Working online

6.4.4. Modifying a system

4. Click Remove.

5. Choose whether you want to create the system as it is now, or to continue with the wizard.
   If you want to create the system as it is now, click Finish.
   If you want to read a summary before you create the system, click Next.

Complete the Modify Controller System wizard

To complete the wizard, follow these steps:

1. Read the system summary.

2. If the system is OK, click Finish.
   If the system is not OK, click Back and make modifications or corrections.

Result

Modifications will take effect when the wizard is completed.
If the system has been downloaded to a controller, it must be downloaded again before the modifications will take effect on the controller.

Deleting a system

To delete a system, follow this steps:

1. From the System Builder dialog box, select the system and then click Delete.
6.4.5. Copying a system

To copy a system, follow these steps:

1. From the System Builder dialog box, select the system and then click Copy to bring up a dialog box.

2. Enter a name for the new system and a path, and then click OK.
6 Working online

6.4.6. Creating a system from backup

6.4.6. Creating a system from backup

Overview

The Create System from Backup Wizard, which creates a new system from a controller system backup, is launched from the System Builder. In addition, you can change the program revision and options.

Starting the wizard

To start the wizard, follow these steps:
1. From the System Builder dialog box, click Create from Backup. This starts the wizard.
2. Read the information on the welcome page and click Next.

Specifying the name and location

To specify the destination folder, follow these steps:
1. In the Name box, enter a name for the system you are creating.
2. In the Path box, enter the path to the system directory in which you will store the system. You can also click the Browse button and browse to the system directory.
3. Click Next.

Locating the backup

To locate a system from backup, follow these steps:
1. In the Backup folder box, enter the path to the backup folder. Alternatively, click the Browse button to browse to it. Click Next.
2. In the Media Pool box, enter the path to the media pool containing the appropriate RobotWare program. Confirm the backup information that now appears in the wizard. Click Next.
6.4.7. Downloading a system to a controller

Overview

All systems you access from the System Builder are stored on your computer. If you wish to run a system on a robot controller, you must first load it to the controller, which then requires a restart.

Load a system

To load a system to a controller, follow these steps:

1. From the System Builder dialog box, select a system and then click Download to Controller to bring up a dialog box.
2. Specify the Destination Controller for the system.

<table>
<thead>
<tr>
<th>You can select by using the...</th>
<th>if...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select controller from list option</td>
<td>the controller has been detected automatically.</td>
</tr>
<tr>
<td>Specify IP address or controller name option</td>
<td>your PC and the robot is connected to the same network. You can only use the controller name in DHCP networks.</td>
</tr>
<tr>
<td>Use service port option</td>
<td>your PC is directly connected to the controller’s service port.</td>
</tr>
</tbody>
</table>

3. Optionally, click Test Connection to confirm that the connection between the computer and the Controller is OK.
4. Click Load.
5. Answer Yes to the question Do you want to load the system.

| Yes | There will be a restart and the system is downloaded to controller. |
| No  | There will be no restart. With the next restart the system will be downloaded to controller. |
| Cancel | The downloading is canceled. |
6 Working online

6.4.8. Creating boot media

6.4.8. Creating boot media

Overview

Boot media is an entire system which the System Builder packs to a single file and commonly stores on a hard disk or USB memory. The controller then accesses the file through its Ethernet port or USB port, respectively.

Creating boot media

To create boot media, follow these steps:

1. From the System Builder dialog box, select a system and then click Boot Media.
2. In the Path box, enter the path to the folder where you want to store the boot media file. Alternatively, browse to the location.
3. Click OK.

Result

To load the boot media system to a controller, first connect it and then restart the controller with the advanced restart method X-start.
6.4.9.1. A MultiMove system with two coordinated robots

Overview

In this example we will use the System Builder to create a coordinated offline system with one IRB2400 and one IRB1600 robot to use in a new RobotStudio station.

Starting the New Controller System Wizard

To create a system like the one described above, follow these steps:
1. Click System Builder to bring up the dialog box.
2. In the dialog box, click Create New to bring up the New Controller System Wizard.
3. Read the welcome text, and click Next to continue to the next page.

Entering the name and path

1. In the Name box, enter the name of the system. The name must not contain blank spaces or non-ASCII characters.
   In this example, name the system MyMultiMove.
2. In the Path box, enter the path for the folder to save the system in, or click the Browse button to browse to the folder or create a new one.
   In this example, save the system in C:\Program Files\ABB Industrial IT\Robotics IT\RobotStudio\ABB Library\Training Systems.
3. Click Next to continue to the next page.

Entering the controller key

1. Select the Virtual key check box. A virtual controller key now appears in the Controller Key box. In this example we will use the default media pool and RobotWare version.
2. Click Next to continue to the next page.

Entering drive keys

1. Click the Right Arrow button next to the Enter Drive key box twice to create one drive key for each robot.
2. Click Next to continue to the next page.
Adding options

This system does not require any additional option keys. Click Next and continue to the next page of the wizard.

Modifying options

When creating robot systems from real robot keys, the key sets the options. But since we are using a virtual key, we have to set the options manually.

To set the options necessary for a MultiMove, follow these steps:

1. Scroll down to the RobotWare / Motion Coordination 1 group and select the MultiMove Coordinated check box.
2. Scroll down to the RobotWare / I/O control group and select the Multitasking and the Advanced RAPID check boxes.
3. Scroll down to the DriveModule1 / Drive module application group and expand the ABB Standard manipulator option. Select the IRB 2400 Type A option, manipulator variant IRB 2400L Type A.
4. Scroll down to the DriveModule2 / Drive module application group and expand the ABB Standard manipulator option. Select the IRB 1600 option, manipulator variant IRB 1600-5/1.2.
5. Click Finish and the system will be created.
6.4.9.2. A system with support for one robot and one positioner external axis

Overview
In this example we will use the System Builder to create an offline system to use in a new RobotStudio station with one IRB1600 robot and one IRBP 250D positioner external axis.

Prerequisites
When creating systems for positioner external axes, you need the media pool and the license key file for that specific positioner. In this example we will use a media pool and license key file for a demo positioner.

Paths to files and folders assume that RobotStudio and the RobotWare media pool have been installed at their default locations on Windows XP. If not, adjust the paths accordingly.

Starting the New Controller System Wizard
To create a system like the one described above, follow these steps:
1. Click System Builder to bring up a dialog box.
2. In the dialog box, click Create New to bring up the New Controller System Wizard.
3. Read the welcome text, and click Next to continue to the next page.

Entering the controller key
1. Select the Virtual key check box. A virtual controller key now appears in the Controller Key box. In this example we will use the default media pool and RobotWare version.
2. Click Next to continue to the next page.

Entering drive keys
1. Click the Right Arrow button next to the Enter Drive key box to create one drive key for the robot.
2. Click Next to continue to the next page.

Adding options
This is where we point out the key file for the positioner.
1. Next to the Enter key box, click the browse button and select the key file.

In this example, browse to and select the file extkey.kxt in the folder C:\Program Files\ABB Industrial IT\Robotics IT\MediaPool\3HEA-000-00022.01.

TIP!
In the MediaPool folder media pools for several standard positioners are installed. They are named by the positioner’s article number, with a suffix that indicates if it is configured for single-robot or MultiMove systems.
2. Click the Right Arrow button next to the Enter key box to add the key for the positioner.
3. Click Next and continue to the next page of the wizard.

Continues on next page
6 Working online

6.4.9.2. A system with support for one robot and one positioner external axis

Continued

Modifying options

When creating robot systems from real robot keys, the key sets the options. But since we are using a virtual key, we have to set the options manually. To set the options necessary for a positioner, follow these steps:

1. Scroll down to the **RobotWare / Hardware** group and select the **709-x DeviceNet** check box.
   This option is for the communication between the controller and the track external axis.

2. Scroll down to the **RobotWare / Motion coordination part 2** group and select the **Multiple Axis Positioner** check box.
   This option applies since the positioner has multiple axes.

3. Scroll down to the **RobotWare / I/O control** group and select the **Logical Cross Connection** check box.
   This option is for the communication between the virtual controller and the track external axis.

4. Scroll down to the **DriveModule1 / Drive module application** group and expand the **ABB Standard manipulator** option. Select the **IRB 1600** option.
   This option sets the robot to an IRB 1600-5/1.2.

5. Scroll down to the **DriveModule1 / Additional axes configuration** group and expand the **Add axes IRB/drive module 1600/2400/260** option and select the **R2C2 Add drive** option.
   Expand the **Drive type in position Z4** and select the **753-1 drive C in pos Z4** option.
   Expand the **Drive type in position Y4** and select the **754-1 drive C in pos Y4** option.
   Expand the **Drive type in position X4** and select the **755-1 drive C in pos X4** option.
   This option adds drive modules for the positioner axes.

6. Click **Finish** and the system will be created. When starting the system in a RobotStudio station, you have to set up the system to load a model for the positioner and to get the motions to work properly. See *Placing external axes on page 79* for more information.
6.4.9.3. Options settings for systems with positioners

Overview

This is an overview of the RobotWare options to set when creating a system for positioner external axes. Note that besides setting the RobotWare options, you must add an additional option key for the positioner.

Media pools and option keys for the positioners

If you have the media pool and option key for your positioner, you can use these files. If not, media pools for standard positioners are installed with RobotStudio. The path to these media pools in a default installation is: C:\program files\ABB Industrial IT\Robotics IT\MediaPool. In this folder a media pool for each positioner is located. These are named by the article number of the positioner, with a suffix that indicates if it is configured for a single-robot or a MultiMove system.

In the Add additional options page of the System Builder, you should add the option for the positioner by opening the mediapool folder for the positioner to add and selecting the extkey.kxt file.

Options for positioners in singe-robot systems

When adding a positioner to a single-robot system, the positioner will be added to the same task as the robot. Below, the options to set on the Modify Options page of the System Builder for such a system are listed:

- RobotWare > Hardware > 709-x DeviceNet > 709-1 Master/Slave Single
- RobotWare > Motion coordination part 2 > Multiple Axis Positioner
- RobotWare > I/O control > Logical Cross Connections
- Optionally, for using the system with ArcWare also add RobotWare > Application Arc > 633-1 Arc
- DriveModule 1 > Additional axes configuration > Add axes IRB/Drive module for your robot model > RC2C Add drive > 753-1 Drive C in pos Z4 > 754-2 Drive T in pos Y4 > 755-3 Drive U in pos X4
6 Working online

6.4.9.3. Options settings for systems with positioners

Continued

Options for positioners in MultiMove robot systems

When adding a positioner to a MultiMove robot system, the positioner shall be added to a task of its own (thus you also have to add a drive key for the positioner). Below, the options to set on the Modify Options page of the System Builder for such a system are listed:

- **RobotWare > Hardware > 709-x DeviceNet > 709-1 Master/Slave Single**
- **RobotWare > Motion coordinated part 1 > 604-1 MultiMove Coordinated**
  Optionally, expand the MultiMove Coordinated option and select process options for the robots.
- **RobotWare > Motion coordination part 2 > Multiple Axis Positioner**
- **RobotWare > I/O control > Logical Cross Connections**

- Optionally, for using the system with ArcWare, add **RobotWare > Application Arc > 633-1 Arc**

- **DriveModule 1 > Additional axes configuration > Add axes IRB/Drive module for your robot model > RC2C Add drive > 753-1 Drive C in pos Z4 > 754-2 Drive T in pos Y4 > 755-3 Drive U in pos X4**. For the other drive modules, no additional axes should be configured.
6.5. Handle I/O

Overview

The I/O system handles input and output signals to and from the controller. Below are the parts of the system described, as well as common types of signals.

The I/O system window is used to view and set previously configured signals, and to activate and deactivate I/O units.

The I/O system

The I/O system of a controller consists of I/O buses, I/O units and I/O signals. The I/O buses are the controller's connections for I/O units (for instance I/O boards) and the I/O units contain channels for the actual signals.

The I/O buses and units are displayed in the robot view, as child nodes under each controller and the I/O signals are displayed in the I/O window.

I/O signals

I/O signals are used to communicate between the controller and external equipment, or to change variables within a robot program.

Input signals

Input signals notify something for the controller, for instance a feeder belt can set an input signal when it has positioned a work piece. The input signal can then be programmed to start a specific part of the robot program.

Output signals

The controller uses output signals to notify that a specified condition has been fulfilled. For instance, after the robot has finished its sequence, an output signal can be set. This signal can then be programmed to start a feeder belt, update a counter or trigger any other action.

Simulated signals

A simulated signal is a signal that is manually given a specific value that overrides the actual signal. Thus simulated signals might be useful for testing robot programs without activating or running equipment.

Virtual signals

Virtual signals are signals that are not configured to belong to a physical I/O unit. Instead, they reside inside the controller's memory. A common use for virtual signals is to set variables and store changes in a robot program.

Procedures

For using the I/O system window, see Inputs / Outputs on page 282.
For adding a signal, see Add signals on page 292.
6.6. Configure systems

Configuring system parameters

System parameters may be configured as follows:

- To view topics, types, instances and parameters
- To edit the parameters of an instance
- To copy and paste instances
- To add and delete instances
- To load and save complete configuration files to and from controllers

When working with configurations, the following tools, see Configuration editor on page 297, are useful:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Configuration Editor</td>
<td>With the Configuration Editor you work with the types and instances of a specific topic.</td>
</tr>
<tr>
<td>The Instance Editor</td>
<td>With the Instance Editor, you specify the values of the parameters in the instances of system parameter types.</td>
</tr>
</tbody>
</table>

**NOTE!**

To edit system parameters, you must have write access to the controller.

Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System parameters</strong></td>
<td>The sum of all parameters that configure the system, these are divided into topics and types.</td>
</tr>
<tr>
<td><strong>Topic</strong></td>
<td>A collection of parameters relating to a specific area, and the highest level in the system parameter structure. Examples are Controller, Communication and Motion.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>A set of parameters for a specific configuration task. A type can be seen as a pattern describing the construction and properties for the parameters included in the task. For instance, the type <strong>Motion System</strong> defines which parameters shall be used for configuring a motion system.</td>
</tr>
<tr>
<td><strong>Instance</strong></td>
<td>An actualization of a type, an instance is a specific set of parameters with unique values created from a type pattern. In the Configuration Editor, each row in the Instance list is an instance of the type selected in the Type list.</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td>A property to set when configuring the robot system.</td>
</tr>
<tr>
<td><strong>Configuration file</strong></td>
<td>Contains all public parameters of a specific topic.</td>
</tr>
</tbody>
</table>

Viewing configurations

1. To view the topics of a controller, from the Offline/Online tab, expand the Configuration node for the controller.
   
   All topics in are now displayed as child nodes to the Configuration node.

2. To view the types and instances of a topic, double-click the topic node for the topic to view.

CONTINUES ON NEXT PAGE
The Configuration Editor is now opened, listing all types of the topic in the **Type name** list. In the **Instance** list, each instance of the type selected in the Type name list is displayed as row. The parameter values of the instances are displayed in the columns of the instance list.

3. To view detailed parameter information for an instance, double-click the instance. The instance editor now displays the current value, restrictions and limits of each parameter in the instance.

### Editing parameters

You can either edit the parameters of one single instance, or you can edit several instances at one time. Editing several instances at one time is useful when you want to change the same parameter in several instances, like when moving signals from one I/O unit to another.

1. In the **Offline/Online** tab, expand the **Controller** and the **Configuration** node and double-click the topic that contains the parameters to edit.

   This opens the Configuration Editor.

2. In the **Type name** list of the Configuration Editor, select the type that the parameter to edit belongs to.

   The instances of the type is now displayed in the Instance list of the Configuration Editor.

3. In the **Instance** list, select the instances to edit and press the Enter Key. To select several instances at once, hold down the SHIFT or CTRL key while selecting.

   The Instance Editor is now displayed.

4. In the Parameter list of the Instance Editor, select the parameter to edit and change the value of the parameter in the **Value** box.

   When editing several instances at one time, the parameter values you specify will be applied to all instances. For parameters that you do not specify any new value, each instance will keep its existing value for that parameter.

5. Click **OK** to apply the changes to the configuration database of the controller.

   For many parameters, the changes will not take affect until the controller is restarted. If your changes require a restart, you will be notified of this.

   You have now updated the controller's system parameters. If the changes require a restart of the controller, the changes will not take affect until you do this. If you are going to make several changes, you can wait with the restart until all changes are done.

### Adding instances

With the Configuration Editor, you can select a type and create a new instance of it. For example, adding a new instance of the type Signal creates a new signal in the system.

1. In the **Offline/Online** tab, expand the **Controller** and the **Configuration** node and double-click the topic that contains the type of which you want to add an instance.

   This opens the Configuration Editor.

2. In the **Type name** list of the Configuration Editor, select the type of which you want to add an instance.
3. On the **Controller** menu, point to **Configuration** and click **Add type** (the word type is replaced by the type you selected previously).

You can also right-click anywhere in the configuration editor and then select **Add type** from the shortcut menu.

A new instance with default values is added and displayed in the **Instance Editor** window.

4. If required, edit the values.

5. Click **OK** to save the new instance.

The values in the new instance are now validated. If the values are valid, the instance is saved. Otherwise, you will be notified of which parameter values to correct.

For many instances, the changes will not take affect until the controller is restarted. If your changes require a restart you will be notified of this.

You have now updated the controller's system parameters. If the changes require a restart of the controller, the changes will not take affect until you do this. If you are going to make several changes, you can wait with the restart until all changes are done.

**Copying an instance**

1. In the **Offline/Online** tab, expand the **Controller** and the **Configuration** node and double-click the topic that contains the instance to copy.

   This opens the Configuration Editor.

2. In the **Type name** list of the Configuration Editor, select the type of which you want to copy an instance.

3. In the **Instance** list, select one or several instances to copy.

   If you select several instances and they don't have the same value for all parameters, these parameters will have no default values in the new instances.

4. On the **Controller** menu, point to **Configuration** and click **Copy Type** (the word type is replaced by the type you selected previously).

   You can also right-click the instance to copy and then select **Copy Type** from the shortcut menu.

   A new instance with the same values as the one you copied is added and displayed in the **Instance Editor** window.

5. Change the name of the instance. If required, also edit the other values.

6. Click **OK** to save the new instance.

   The values in the new instance are now validated. If the values are valid, the instance is saved. Otherwise, you will be notified of which parameter values to correct.

   For many instances, the changes will not take affect until the controller is restarted. If your changes require a restart you will be notified of this.

   You have now updated the controller's system parameters. If the changes require a restart of the controller, the changes will not take affect until you do this. If you are going to make several changes, you can wait with the restart until all changes are done.

(Continues on next page)
Deleting an instance

1. In the Offline/Online tab, expand the Controller and the Configuration node and double-click the topic that contains the type of which you want to delete an instance. This opens the Configuration Editor.

2. In the Type name list of the Configuration Editor, select the type of which you want to delete an instance.

3. In the Instance list, select the instance to delete.

4. On the Controller menu, point to Configuration and then click Delete type (the word type is replaced by the type you selected previously).
   You can also right-click the instance to delete and then select Delete type from the shortcut menu.

5. A message box is displayed, asking if you want to delete or keep the instance. Click Yes to confirm that you want to delete it.

   For many instances, the changes will not take affect until the controller is restarted. If your changes require a restart you will be notified of this.

   You have now updated the controller's system parameters. If the changes require a restart of the controller, the changes will not take affect until you do this. If you are going to make several changes, you can wait with the restart until all changes are done.

Save one configuration file

The system parameters of a configuration topic can be saved to a configuration file, stored on the PC or any of its network drives.

The configuration files can then be loaded into a controller. They are thereby useful as backups, or for transferring configurations from one controller to another.

1. In the Offline/Online tab, expand the Configuration node and select the topic to save to a file.

2. On the Controller menu, point to Configuration and select Save System Parameters.
   You can also right-click the topic and then select Save System Parameters from the shortcut menu.

3. In the Save As dialog box, browse for the folder to save the file in.

4. Click Save.

Saving several configuration files

1. In the Offline/Online tab, select the Configuration node.

2. On the Controller menu, point to Configuration and click Save System Parameters.
   You can also right-click the configuration node and then click Save System Parameters.

3. In the Save System Parameters dialog box, select the topics to save to files. Then click Save.

4. In the Browse for Folder dialog box, browse for the folder to save the files in, and then click OK.

   The selected topics will now be saved as configuration files with default names in the specified folder.

Continued on next page
6 Working online

6.6. Configure systems

Continued

Loading a configuration file

A configuration file contains the system parameters of a configuration topic. They are thereby useful as backups, or for transferring configurations from one controller to another.

When loading a configuration file to a controller, it must be of the same major version as the controller. For instance, you cannot load configuration files from an S4 system to an IRC 5 controller.

1. In the Offline/Online tab, select the Configuration node.

2. On the Controller menu, point to Configuration and select Load Parameters.

   You can also right-click the configuration node and then select Load Parameters from the context menu.

   This opens the Select mode dialog box.

3. In the Select mode dialog box, select how you want to combine the parameters in the configuration file to load with the existing parameters:

<table>
<thead>
<tr>
<th>If you want to</th>
<th>then</th>
</tr>
</thead>
<tbody>
<tr>
<td>replace the entire configuration of the topic with the one in the configuration file.</td>
<td>select Delete existing parameters before loading</td>
</tr>
<tr>
<td>add new parameters from the configuration file to the topic, without modifying the existing ones.</td>
<td>click Load parameters if no duplicates</td>
</tr>
<tr>
<td>add new parameters from the configuration file to the topic and update the existing ones with values from the configuration file. Parameters that only exist in the controller and not in the configuration file will not be changed at all.</td>
<td>click Load parameters and replace duplicates</td>
</tr>
</tbody>
</table>

4. Click Open and browse to the configuration file to load. Then click Open again.

5. In the information box, click OK to confirm that you want to load the parameters from the configuration file.

6. When the loading of the configuration file is finished, close the Select mode dialog box.

   If a restart of the controller is necessary for the new parameters to take affect, you will be notified of this.
6.7. Handle events

Overview

An event is a message that notifies you that something has happened to the robot system, be it merely a change in operation mode or a severe error that calls for your immediate attention. If the event requires any action from you, this is stated in the event.

Events are displayed in the event logs of the FlexPendant and RobotStudio.

The event log keeps you informed of system status, allowing you to:

• view controller events.
• filter events.
• sort events.
• get detailed information about an event.
• save event log files on your PC.
• clear event records.

Event Log list

The event log list consists of all events matching your filter settings, with the following information for each event:

<table>
<thead>
<tr>
<th>Type</th>
<th>The event type is an indication of the severity of the event.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>The event code is a number that identifies the event message.</td>
</tr>
<tr>
<td>Title</td>
<td>The event title is a short event description.</td>
</tr>
<tr>
<td>Category</td>
<td>The event category is an indication of the source of the event.</td>
</tr>
<tr>
<td>Seq. Number</td>
<td>The sequential number indicates the chronological order of the event.</td>
</tr>
<tr>
<td>Date and Time</td>
<td>Date and time is when the event occurred.</td>
</tr>
</tbody>
</table>

When you select an event in the list, detailed information will appear to the right.

Event type

The event type is an indication of the severity of the event.

There are three types of events:

<table>
<thead>
<tr>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>A normal system event, such as starting and stopping programs, change of operating mode, motors on/off and so on. Information messages never require any action from you, but can be useful for error tracking, statistics collecting or monitoring user triggered event routines.</td>
</tr>
<tr>
<td>Warning</td>
<td>An event that you need to be aware of, but not so severe that the process or RAPID program needs to be stopped. Warnings, however, often indicate underlying problems that sooner or later must be solved. Warnings must sometimes be acknowledged.</td>
</tr>
<tr>
<td>Error</td>
<td>An event that prevents the robot system from proceeding. The running process or RAPID program cannot continue, but is stopped. All errors must be acknowledged. Most errors also require some immediate action from you in order to solve the problem.</td>
</tr>
</tbody>
</table>

Continues on next page
6 Working online

6.7. Handle events

Continued

NOTE!
This information is also indicated by color: blue for information, yellow for warning and red for an error which needs to be corrected in order to proceed

Event code
The event code is a number that identifies the event message. Together with the event date and time each event has a unique identity.

Event title
The event title is a short description of the event.

Event category
The category is an indication of the source of the event.

<table>
<thead>
<tr>
<th>Category</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common</td>
<td>All recent events.</td>
</tr>
<tr>
<td>Operational</td>
<td>Events related to changes in operation or operating mode.</td>
</tr>
<tr>
<td>System</td>
<td>Events related to the current system.</td>
</tr>
<tr>
<td>Hardware</td>
<td>Events related to controller hardware.</td>
</tr>
<tr>
<td>Program</td>
<td>Events related to the running process applications and RAPID programs.</td>
</tr>
<tr>
<td>Motion</td>
<td>Events related to the movement of robots or other mechanical units.</td>
</tr>
<tr>
<td>I/O &amp; Communication</td>
<td>Events related to input and output signals, serial or network communication and process buses.</td>
</tr>
<tr>
<td>User</td>
<td>Custom messages that have been programed into RAPID programs.</td>
</tr>
<tr>
<td>Internal</td>
<td>Internal low-level controller errors for ABB service personnel.</td>
</tr>
<tr>
<td>Process</td>
<td>Events related to Industrial Processes options., such as Spot, Arc and Dispense.</td>
</tr>
<tr>
<td>Cfg</td>
<td>Errors in a configuration file.</td>
</tr>
</tbody>
</table>

Depending on how the system is configured, additional categories may exist.

Sequential number
The sequential number indicates the chronological order of the event; the higher the number the more recently the event occurred.

Date and time
Date and time indicate exactly when the event occurred. Along with the event code, this timestamp guarantees that each event has a unique identity.

Event description
When you select an event in the list, detailed information about it will be displayed to the right. This includes a description, and when appropriate also consequences, causes and suggested actions to solve the problem.
Overview

The Event Log automatically logs all controller events once it is started. By default, events are displayed in the chronological order specified by Seq Numbers.

**NOTE!**

Any modifications to the list you see will never affect the event log of the controller. What you see is just a copy.

Managing events

1. In the Robot View Explorer, select a system.
2. Double-click the **Events** node.

<table>
<thead>
<tr>
<th>To sort events</th>
<th>Click the header for the column you want to sort by. To switch between ascending and descending sorting, click the header once again.</th>
</tr>
</thead>
<tbody>
<tr>
<td>To filter events</td>
<td>In the <strong>Category</strong> list select the event category you want displayed.</td>
</tr>
<tr>
<td>To clear the event log</td>
<td>Click <strong>Clear</strong>. This will not affect the event log of the robot controller. It might still be impossible, however, to retrieve all events from a cleared record once again, as the oldest ones may have been erased from the controller hard disk due to lack of space. It is therefore recommended to save the record to a log file before clearing.</td>
</tr>
<tr>
<td>To save all events to a single log file on the computer</td>
<td>Check the <strong>Log to file</strong> check box. If it remains checked, the log file will be updated with new events as they occur.</td>
</tr>
<tr>
<td>To save events of one or several categories to files on the computer</td>
<td>Click <strong>Save</strong> and then make your category choice. Specify the location for the log file(s) in the <strong>Browse For Folder</strong> dialog and then click <strong>OK</strong>. If you select <strong>All</strong> when selecting categories, a log file for each event category will be created.</td>
</tr>
</tbody>
</table>

Retrieving controller events

To clear the list and retrieve all existing events from the robot controller:

1. Optionally, save the existing Event Log record.
2. Select whether you want the list to be updated when new events occur, or if you are only interested in viewing events that have already occurred.

<table>
<thead>
<tr>
<th>To ...</th>
<th>...then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>get automatic updates when new events occur</td>
<td>check the <strong>Auto Update</strong> check box. (Checked by default.)</td>
</tr>
<tr>
<td>say no to automatic updates when new events occur</td>
<td>clear the <strong>Auto Update</strong> check box.</td>
</tr>
</tbody>
</table>

3. Click **Get** to clear the current list, fetch and display all events that are currently stored in the controller log files.
6.8. Handle devices

Overview

The Device Browser displays the properties and trends of the various hardware and software devices in a robot controller. It is launched from the Properties command on the Online tab. For procedures, see Device Browser on page 316.
6.9. RAPID Watch Online

Overview

The RAPID Watch Online window displays the variable status during the program execution. It has the following tabs:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Displays variable name</td>
</tr>
<tr>
<td>Value</td>
<td>Displays variable value</td>
</tr>
<tr>
<td>Type</td>
<td>Displays type of datatype</td>
</tr>
<tr>
<td>Source</td>
<td>Displays system name</td>
</tr>
</tbody>
</table>

**NOTE!**

When running a program in continuous execution mode, the contents of RAPID Watch Online window do not update until program execution stops.
The Application menu is accessed from the RobotStudio button in the upper left corner of the GUI.
7 The Application Menu

7.2. New Station

Creating a station with a template system

1. Click New Station to bring up a dialog box. The Template System icon is already selected by default.
2. In the Select Template System list, either select an appropriate template or click Browse and browse to one.
3. In the System group, enter a name and location, and then click OK.

Creating a station with an existing system

1. Click New Station to bring up a dialog box.
2. Click the Existing system icon to bring up the Existing System dialog box.
3. In the Select System Pool list, select a folder.
4. In the Systems Found list, select a system.
5. Click OK.

Creating an empty station

1. Click New Station to bring up a dialog box.
2. Click the Empty Station icon. The station will be created and visible in the graphics window.
7 The Application Menu

7.3. Screenshot

Prerequisites

For optimal results, first configure the options, see Options: General: Screenshot on page 179.

Capturing the screen

1. Click Screenshot.

If the image is saved to file, this is reported in the output window.
7 The Application Menu

7.4. Pack & Go

7.4. Pack & Go

Packing a station

1. Click Pack & Go to open the Pack & Go Wizard.
2. On the Welcome to the Pack & Go Wizard page, click Next.
3. On the Destination page, specify the destination directory of the package by either entering the path in the box or browsing to it. Click Next.
4. On the Libraries page, select one of the three options. Click Next.
5. On the Systems page, select the check box to include backups of all robot systems. Optionally, select the check box to include a media pool for additional options. Click Next.
6. On the Ready to pack page, review the information and then click Finish.
7. On the Pack & Go succeeded page, review the results and then click Close.
7.5. Unpack & Work

Unpacking a station

1. Click Unpack & Work to open the Unpack & Work Wizard.
2. On the Welcome to the Unpack & Work Wizard page, click Next.
3. On the Select package page, specify the package to be unpacked by either entering the path in the box or browsing to it. Specify as well the destination directory of the files. Click Next.
4. On the Controller Systems page, specify the path to the media pool and the RobotWare version. Optionally, select the check box to automatically restore backup. Click Next.
5. On the Ready to unpack page, review the information and then click Finish.
6. On the Unpack & Work succeeded page, review the results and then click Close.
7 The Application Menu

7.6. Station Viewer

7.6. Station Viewer

Overview

The Station Viewer can playback a station in 3D on computers that do not have RobotStudio installed. It packages the station file together with files needed to view the station in 3D. It can also play recorded simulations.

Prerequisites

- Works on Windows XP SP2 and Windows Vista
- .NET Framework 2.0 must be installed on the playback computer

Creating and loading a Station Viewer

1. To create a Station Viewer, on the Application menu, click Collaborate.
2. Select Save Station as Viewer to bring up a dialog box.
3. Specify a file name and save as .exe file.
   NOTE! Select the option Show comments on startup and add text in the box to view the comment when the Station Viewer is started.
   NOTE! Select the option Include last executed simulation to include a recording of the simulation. By default this option is disabled.
4. To load a Station Viewer, double-click the package (.exe) file on the target computer.
   The results are displayed in the Output window and the embedded station file is automatically loaded and presented in a 3D view.
   NOTE! You can open any station created in RobotStudio 5.12 with the Station Viewer. You can pan, rotate, and navigate in the graphical viewer.

Configuring user settings of a Station Viewer

To configure the user settings of a Station Viewer, on the Application menu, click RobotStudio Options.

Command Buttons

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply</td>
<td>Click this button to save all options in the current page.</td>
</tr>
<tr>
<td>Reset</td>
<td>Click this button to reset to the settings you had before this session all values that you have changed on the current page.</td>
</tr>
<tr>
<td>Default</td>
<td>Click this button to reset to their default values all settings on the current page.</td>
</tr>
</tbody>
</table>

Options: General: Appearance

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select application language</td>
<td>Select the language to be used.</td>
</tr>
<tr>
<td></td>
<td>NOTE! The default language is the same as that of the target user's operating system if available, otherwise it is English.</td>
</tr>
<tr>
<td>Select color theme</td>
<td>Select the color to be used.</td>
</tr>
</tbody>
</table>

Continues on next page
When you run a simulation, the movements and visibility of objects are recorded. This recording is optionally included in the Station Viewer.

Simulation control buttons are enabled when the Station Viewer contains a recorded simulation.

Following are the Simulation control buttons:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Play</strong></td>
<td>Starts or resumes simulation playback</td>
</tr>
<tr>
<td><strong>Stop</strong></td>
<td>Stops simulation playback</td>
</tr>
<tr>
<td><strong>Reset</strong></td>
<td>Resets all objects to their initial state and process time display to zero</td>
</tr>
<tr>
<td><strong>Run mode</strong></td>
<td>Select to run the simulation once or continuously</td>
</tr>
<tr>
<td><strong>Process time</strong></td>
<td>Displays the current simulation time</td>
</tr>
</tbody>
</table>

**NOTE!** If the simulation executes a VSTA macro, it is not reflected when playing back the simulation in the Station Viewer.

---

**Options: General: Graphics**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Select API to use for 3D graphics</strong></td>
<td>Select the low-level API to be used for rendering 3D graphics. The default value is Direct3D, but OpenGL might be faster or stabler. <strong>NOTE!</strong> No need to restart when changing this option.</td>
</tr>
<tr>
<td><strong>Background color</strong></td>
<td>Select the color from the color theme, or from the color stored in the stations.</td>
</tr>
</tbody>
</table>
## 7.7. RobotStudio Options

### Common buttons

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Apply</strong></td>
<td>Click this button to save all options in the current page.</td>
</tr>
<tr>
<td><strong>Reset</strong></td>
<td>Click this button to reset to the settings you had before this session all values that you have changed on the current page.</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>Click this button to reset to their default values all settings on the current page.</td>
</tr>
</tbody>
</table>

### Options: General: Appearance

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Select application language</strong></td>
<td>Select the language to be used.</td>
</tr>
<tr>
<td><strong>Select color theme</strong></td>
<td>Select the color to be used.</td>
</tr>
<tr>
<td><strong>Display Position Edit boxes with Red/Green/Blue background</strong></td>
<td>Select the check box if you want to display the position boxes in the modify dialog boxes with colored background. Default value: selected.</td>
</tr>
</tbody>
</table>

### Options: General: Licensing

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>View installed license keys</strong></td>
<td>Click to view the license keys per feature, version, type, expiration date and status.</td>
</tr>
<tr>
<td><strong>Request a license</strong></td>
<td>Click to request a license.</td>
</tr>
<tr>
<td><strong>Install a license</strong></td>
<td>Click to install a license.</td>
</tr>
</tbody>
</table>

### Options: General: Units

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quantity</strong></td>
<td>Select the quantity for which you want to change the units.</td>
</tr>
<tr>
<td><strong>Unit</strong></td>
<td>Select the unit for the quantity.</td>
</tr>
<tr>
<td><strong>Display decimals</strong></td>
<td>Enter the number of decimals that you want to be displayed.</td>
</tr>
<tr>
<td><strong>Edit decimals</strong></td>
<td>Enter the number of decimals that you want when modifying.</td>
</tr>
</tbody>
</table>

### Options: General: Advanced

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable Windows Error Reporting</strong></td>
<td>Select this check box to send error data directly to a Microsoft server to be downloaded and analyzed by ABB support. Available only on Windows XP or later.</td>
</tr>
<tr>
<td><strong>Number of undo/redo steps</strong></td>
<td>The number of operations that can be undone or redone. Lowering this value can decrease memory usage.</td>
</tr>
<tr>
<td><strong>Show acknowledge dialog box when deleting objects</strong></td>
<td>Warns when deleting objects.</td>
</tr>
<tr>
<td><strong>Show acknowledge dialog box when deleting targets and corresponding move instructions</strong></td>
<td>Warns when deleting targets and move instructions.</td>
</tr>
<tr>
<td><strong>Warn about running Virtual Controller processes on startup</strong></td>
<td>Warns of orphaned VC processes.</td>
</tr>
</tbody>
</table>
## 7 The Application Menu

### 7.7. RobotStudio Options

**Options: General: Files & Folders**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Project Folder</td>
<td>Enter the path to your project folder. This will be the folder displayed in the open and save dialog boxes in RobotStudio.</td>
</tr>
<tr>
<td>...</td>
<td>To browse for your project folder, click the browse button.</td>
</tr>
<tr>
<td>Automatically create document subfolders</td>
<td>Select this check box to enable the creation of individual subfolders for document types.</td>
</tr>
<tr>
<td>Enable Autosave</td>
<td>Select the check box to automatically save the station with defined intervals. Default value: cleared</td>
</tr>
<tr>
<td>Interval</td>
<td>Specify the interval between the savings when using Autosave in this box.</td>
</tr>
</tbody>
</table>

**Options: General: Screenshot**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entire application window</td>
<td>Select this option to capture the entire application.</td>
</tr>
<tr>
<td>Active document window</td>
<td>Select this option to capture the active document window, typically the graphics window.</td>
</tr>
<tr>
<td>Copy to clipboard</td>
<td>Select this check box to save the captured image to the system clipboard.</td>
</tr>
<tr>
<td>Save to file</td>
<td>Select this check box to save the captured image to file.</td>
</tr>
<tr>
<td>Location</td>
<td>Specify the location of the image file. The default location is the &quot;My Pictures&quot; system folder.</td>
</tr>
<tr>
<td>...</td>
<td>Browse for the location.</td>
</tr>
<tr>
<td>File name</td>
<td>Specify the name of the image file. The default name is &quot;RobotStudio&quot; to which is added a date.</td>
</tr>
<tr>
<td>The file suffix list</td>
<td>Select the desired file format. The default format is JPG.</td>
</tr>
</tbody>
</table>

**Options: General: Screen Recorder**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Framerate</td>
<td>Specify the framerate in frames per second.</td>
</tr>
<tr>
<td>Same as window</td>
<td>Select this option to use the same resolution as in the graphics window.</td>
</tr>
<tr>
<td>Limit resolution</td>
<td>Select this option to scale down the resolution.</td>
</tr>
<tr>
<td>Maximum width</td>
<td>Specify the maximum width in pixels.</td>
</tr>
<tr>
<td>Maximum height</td>
<td>Specify the maximum height in pixels.</td>
</tr>
<tr>
<td>Output file format</td>
<td>Select the output file format. The default format is AVI.</td>
</tr>
<tr>
<td>Video compression</td>
<td>Select the video compression format.</td>
</tr>
<tr>
<td></td>
<td>NOTE! DivX format is not supported.</td>
</tr>
<tr>
<td>Start recording after</td>
<td>Select this option to start recording after the specified time.</td>
</tr>
<tr>
<td>Stop recording after</td>
<td>Select this option to stop recording after the specified time.</td>
</tr>
</tbody>
</table>

**Options: Robotics: Editor**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrap long lines</td>
<td>Select this check box if you want to wrap long lines.</td>
</tr>
<tr>
<td>Text styles</td>
<td>Specify the appearance of the various text classes.</td>
</tr>
</tbody>
</table>

Continues on next page
7 The Application Menu

7.7. RobotStudio Options

Continued

<table>
<thead>
<tr>
<th>Options:Robotics:Rapid</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show dialog when warning for globally defined workobjects</td>
<td>Select this check box if you want RobotStudio to display a warning when there are workobjects with the same name that have been declared as in other tasks. Default value: selected.</td>
</tr>
<tr>
<td>Show synchronize dialog box after loading program /module</td>
<td>Select this check box if you want the synchronize dialog box to be displayed when you have loaded a program or a module. Default value: selected.</td>
</tr>
<tr>
<td>Show notification that default data is used</td>
<td>Select this check box if you want to be notified that wobj0 and/or tool0 is active and will be used in the current action. Default value: selected.</td>
</tr>
<tr>
<td>Show synchronization notification</td>
<td>Default value: selected.</td>
</tr>
<tr>
<td>Set as active when creating tooldata</td>
<td>Select this check box if you want newly created tooldata to be set as active. Default value: selected.</td>
</tr>
<tr>
<td>Set as active when creating workobjects</td>
<td>Select this check box if you want newly created workobjects to be set as active. Default value: selected.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options:Robotics:Synchronization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use default synchronization locations</td>
<td>Converting data, such as target to Workobject, shall use the default behavior for synchronization locations. Default value: selected.</td>
</tr>
<tr>
<td>Show default synchronization locations notification</td>
<td>Notifies of the behavior above. Default value: selected.</td>
</tr>
<tr>
<td>Declaration default locations</td>
<td>Specify the locations for corresponding objects when synchronizing to the VC.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options:Robotics:Mechanism</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approach Vector</td>
<td>Select the approach vector. Default value: Z.</td>
</tr>
<tr>
<td>Travel Vector</td>
<td>Select the travel vector. Default value: X.</td>
</tr>
<tr>
<td>Enable configuration check for jump to target/ move instruction</td>
<td>Select this check box if you want to enable the configuration check configurations when jumping to target or move instructions. When selected and a target does not have a validated configuration assigned, you will be asked to set one. When cleared, the configuration closest to the current one is used. Default value: selected.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options:Robotics:Virtual FlexPendant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always on top</td>
<td>Select this check box if you want to have the virtual FlexPendant always on top. Default value: selected.</td>
</tr>
<tr>
<td>Enable transparency</td>
<td>Select this check box if you want parts of the virtual FlexPendant to be transparent. Default value: selected.</td>
</tr>
<tr>
<td>Show virtual Operator Window</td>
<td>Select this check box to enable the Operator window feature. Default value: Disabled.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options:Online:Authentication</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recent Users</td>
<td>Lists the recent users.</td>
</tr>
</tbody>
</table>
7 The Application Menu

7.7. RobotStudio Options

<table>
<thead>
<tr>
<th>Remove/Remove All</th>
<th>Click these buttons to remove one or all recent users, respectively.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Automatic Logoff</td>
<td>Select the check box if you want to log off automatically.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Determines the length of the session before being automatically logged off.</td>
</tr>
</tbody>
</table>

Options:Graphics:Renderer

| Select API to use for 3D graphics | Select the low-level API to be used for rendering 3D graphics. The default value is OpenGL, but Direct3D might be faster or stabler. |

Options:Graphics:Appearance

<table>
<thead>
<tr>
<th>Background color</th>
<th>Click the colored rectangle to change the background color.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>Select the check box if you want to display the background color in gradient style. Default value: cleared.</td>
</tr>
<tr>
<td>Show floor</td>
<td>Select the check box if you want the floor (at z=0) to be displayed. Change the floor color by clicking the colored rectangle. Default values: selected.</td>
</tr>
<tr>
<td>Color</td>
<td>Click the colored rectangle to change the color of the floor.</td>
</tr>
<tr>
<td>Transparent</td>
<td>Select the check box if you want the floor to be transparent. Default values: selected.</td>
</tr>
<tr>
<td>Show UCS Grid</td>
<td>Select the check box if you want the UCS grid to be displayed. Default value: selected.</td>
</tr>
<tr>
<td>Grid Space X</td>
<td>Change the UCS grid space in the X coordinate direction by entering the requested value in the box. Default value: 1000 mm (or equivalent in other units).</td>
</tr>
<tr>
<td>Grid Space Y</td>
<td>Change the UCS grid space in the Y coordinate direction by entering the requested value in the box. Default value: 1000 mm (or equivalent in other units).</td>
</tr>
<tr>
<td>Show coordinate system</td>
<td>Select the check box if you want the coordinate systems to be displayed. Default value: selected.</td>
</tr>
</tbody>
</table>

Options:Graphics:Performance

<table>
<thead>
<tr>
<th>Detail level</th>
<th>Select if the detail level is to be Auto, Fine, Medium or Coarse. Default value: Auto.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cull back-facing triangles</td>
<td>Select the check box if you want to ignore the back-facing triangles. Default value: selected. Culling back-facing triangles improves the graphics performance but may give unexpected display if surfaces in models are not faced correctly.</td>
</tr>
<tr>
<td>Enable twosided lighting</td>
<td>Select the check box if you want to enable two-sided lighting. Default value: cleared.</td>
</tr>
<tr>
<td>Cull objects smaller than</td>
<td>Select the size in pixels under which objects will be disregarded. Default value: 2 pixels.</td>
</tr>
<tr>
<td>Store model data on graphics card (recommended)</td>
<td>Select the check box if you want to store the model data on a graphics card. Default value: cleared. Storing model data on a graphics card improves graphics performance, but causes stability problems for some hardware combinations.</td>
</tr>
</tbody>
</table>
The settings you make here are generic for all objects in RobotStudio. With the Graphic Appearance dialog box you can, however, override some of these settings for single objects.

Options: Graphics: Behavior

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation sensitivity</td>
<td>Select the navigation sensitivity when using the mouse movements or navigation buttons by clicking the bar and dragging it into position. Default value: 1.</td>
</tr>
<tr>
<td>Selection radius (pixels)</td>
<td>Change the selection radius (that is, how close the mouse cursor click must be to an item to be selected) by entering the requested pixel value in the box. Default value: 5.</td>
</tr>
<tr>
<td>Selection highlight</td>
<td>Set if the selected object shall be distinguished in the Graphics window by a color, by an outline or not at all. Default value: color.</td>
</tr>
<tr>
<td>Highlight color</td>
<td>Click the colored rectangle to change the highlight color.</td>
</tr>
<tr>
<td>Activate selection preview</td>
<td>Select the check box to enable temporarily highlighting of items that may be selected when the mouse cursor passes over them. Default value: selected.</td>
</tr>
<tr>
<td>Show local coordinate system for selected objects</td>
<td>Select the check box to show the local coordinate system for the selected objects. Default value: selected.</td>
</tr>
</tbody>
</table>

Options: Graphics: Geometry

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Tolerance</td>
<td>Enter the maximum deviation of surface normals in the Fine, Medium or Coarse boxes. Default values (in deg): Fine: 10, Medium: 22.5, Coarse: 45 (or equivalent in other units).</td>
</tr>
<tr>
<td>Surface Tolerance</td>
<td>Enter the maximum spatial deviation of surfaces in the Fine, Medium or Coarse boxes. Default values (in deg): Fine: 2, Medium: 10, Coarse: 50 (or equivalent in other units).</td>
</tr>
<tr>
<td>Curve Tolerance</td>
<td>Enter the maximum spatial deviation of curves in the Fine, Medium or Coarse boxes. Default values (in deg): Fine: 0.2, Medium: 1, Coarse: 5 (or equivalent in other units).</td>
</tr>
</tbody>
</table>

Options: Simulation: Collision

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform collision detection</td>
<td>Select if collision detection is to be performed during simulation or always. Default value: During simulation.</td>
</tr>
<tr>
<td>Stop simulation at collision</td>
<td>Select this check box if you want the simulation to stop at a collision. Default value: cleared.</td>
</tr>
<tr>
<td>Log collisions to Output window</td>
<td>Select this check box if you want the collisions to be logged to the output window. Default value: selected.</td>
</tr>
<tr>
<td>Log collisions to file</td>
<td>Select this check box if you want to log the collisions to a file. Browse for the file to log in by clicking the browse button. Default value: cleared.</td>
</tr>
<tr>
<td>Enable fast collision detection</td>
<td>Select this check box to enhance the performance by detecting collisions between geometrical bounding boxes instead of geometrical triangles. This might result in falsely reported collisions, since the triangles are the true geometry and the bounding boxes always are larger. All true collisions will, however, be reported. The larger the object, the greater the number of false collisions that are likely to be detected.</td>
</tr>
<tr>
<td>View</td>
<td>Click this button to open the log file specified in the file box in Notepad.</td>
</tr>
<tr>
<td>Clear</td>
<td>Click this button to delete the log file specified in the file box.</td>
</tr>
</tbody>
</table>
Options: Simulation: Virtual Time

<table>
<thead>
<tr>
<th>Virtual Time mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Time mode - Free run</td>
<td>This option makes RobotStudio always use the free run mode.</td>
</tr>
<tr>
<td>Virtual Time mode - Time Slice</td>
<td>This option makes RobotStudio always use the time slice mode.</td>
</tr>
</tbody>
</table>

Options: Simulation: Accuracy

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation speed</td>
<td>Sets the simulation speed relative to real time.</td>
</tr>
<tr>
<td>Simulation timestep</td>
<td>Specifies the simulation timestep.</td>
</tr>
</tbody>
</table>
7 The Application Menu

7.7. RobotStudio Options
8 The Home Tab

8.1. Overview

The Home tab contains the controls required for building stations, creating systems, programming paths and placing items.
8.2. ABB Library

About this button

With this button, you can choose robots, positioners and tracks from their respective galleries.
8.3. Import Library

About this button

With this button, you can import equipment, geometries, positioners, robots, tools and training objects to your station libraries.

Click the lower split button to display galleries of tools, conveyors, cabinets and other equipment.
8.4. Robot System

About this button

With the Robot System button, you can either create a system from layout or template, choose an existing system, or select a system from a robot gallery and setup a conveyor tracking mechanism.

Creating a system from layout

1. Click From Layout to bring up the first page of the wizard.
2. In the Name box, enter the name of the system.
3. In the Location box, enter the path to the folder where the system will be stored. Alternatively, click Browse and browse to the folder.
4. In the Media Pool box, enter the path to the media pool. Alternatively, click Browse and browse to the folder.
5. In the RobotWare Version list, select the version of RobotWare you want to use.
6. Click Next.
7. In the Mechanisms box, select the mechanisms that you want to include in the system.
8. Click Next.

The wizard now proposes a mapping of the mechanisms to a specific motion task, in accordance with the following rules:

- Only one TCP robot is allowed per task.
- Up to six motion tasks may be added, but only four TCP robots can be used, and they must be assigned to the first four tasks.
- The number of tasks may not exceed the number of mechanisms.
- If the system contains one TCP robot and one external axis, they will be assigned to the same task. It is, however, possible to add a new task and assign the external axis to it.
- If the system contains more than one TCP robot, any external axes will be assigned to a separate task. It is, however, possible to move them to other tasks.
- The number of external axes in a task is limited by the number of available drive modules in the cabinet (one for large robots, two for medium, three for small).

Note! If only one mechanism was selected in the previous page, this page will not be shown.

Tasks can be added and removed using the respective buttons; mechanisms can be moved up or down using the respective arrows. To map the mechanisms to tasks, follow this step:

9. Optionally, make any edits in the mapping, and then click Next.

10. Verify the summary and then click Finish.

If the system contains more than one robot, the number of tasks and the baseframe positions of the mechanism should be verified in the System Configuration window.

Continues on next page
Adding a template system

1. Click From Template to bring up a dialog box.
2. In the Select Template System list, either select an appropriate template or click Browse and browse to one.
3. In the Libraries group, select whether to import libraries or to use the existing station libraries.
4. In the System group, enter a name and location, and then click OK.

Adding an existing system

1. Click Existing to bring up a dialog box.
2. In the Select System Pool list, select a folder.
3. In the Systems Found list, select a system.
4. In the Libraries group, select whether to import libraries or to use the existing station libraries.
5. Click OK.

Selecting a system from a robot gallery

1. Click Quick System to bring up a gallery, and then click the appropriate robot.

Setting up a conveyor

1. Click Setup.
2. In the Part Sequence tab, select Part from Available Parts.
   The right arrow button is enabled.
3. Click right arrow button to move the Part to Parts moved by Conveyor list.
4. Click up and down arrow buttons to move the selected part in Parts moved by Conveyor list.
5. In the Part Tracking tab, select Part from Parts moved by Conveyor list.
6. Select CNV1 from the Mechanical Unit list.
7. Select a workobject from the Workobject list.
8. Click Add. The workobject appears in the list.
   NOTE! If the same workpiece is tracked by more than one robot, add a pair of workobject for each robot that tracks the workpiece. This procedure has to be repeated for each workpiece that should be tracked.
9. Click OK.
10. Activate the Conveyor Mechanical Unit (CNV1). See Activate Mechanical Units on page 264.

Continues on next page
Removing objects from conveyor

1. Click Setup.
   The Conveyor Setup dialog box appears.

2. In the Part Sequence tab, select Part from the Parts moved by Conveyor list
   The left arrow button is enabled.

3. Click left arrow to remove the part from the Parts moved by Conveyor list to Available Parts list.
8.5. Import Geometry

Importing a geometry

1. Click Import Geometry.
2. Browse to the library where the geometry is located.
   For predefined geometries, click the ABB Library icon to the left in the dialog box.
3. Select the requested geometry and Detail Levels. Click Open.
   If you want the geometry to move with another object, attach it to the requested object, see Attach to on page 323.
   To modify the detail level for import of geometries, see RobotStudio Options on page 178.
8 The Home Tab

8.6 Frame

8.6.1. Frame

Creating a frame

1. Click Frame.
2. In the dialog box, specify the positions for the frame.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Position</td>
<td>Click in one of these boxes, and then click the frame position in the graphics window to transfer the values to the Frame Position boxes.</td>
</tr>
<tr>
<td>Frame Orientation</td>
<td>Specify the coordinates for the frame orientation.</td>
</tr>
<tr>
<td>Set as UCS</td>
<td>Select this check box to set the created frame as the user coordinate system.</td>
</tr>
</tbody>
</table>
8.6.2. Frame from Three Points

Creating a frame from three points

1. Click **Frame from Three points** to bring up a dialog box.

2. Decide how you want to specify the frame:

<table>
<thead>
<tr>
<th>To specify the frame using</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>X, Y and Z coordinates, a point on the X axis and a point in the X-Y plane</td>
<td><strong>Position</strong></td>
</tr>
<tr>
<td>two points on the X axis and one point on the Y axis</td>
<td><strong>Three Point</strong></td>
</tr>
</tbody>
</table>

3. If you select **Position**:
   - Enter the **Position** for the object.
   - Enter the **Point on X axis** for the object.
   - Enter the **Point on X-Y plane** for the object.
   - Click **Create**.

4. If you select **Three Point**:
   - Enter the **First Point on X axis** for the object. This is the point closest to the frame’s origin.
   - Enter the **Second Point on X axis** for the object. This is the point further away in the positive X direction.
   - Enter the **Point on Y axis** for the object.
   - Click **Create**.

The Create Frame From Three Points dialog box

<table>
<thead>
<tr>
<th>Position</th>
<th>Select this option if you want to create the frame by using a position and two points.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Position</td>
<td>Click in one of these boxes, and then click the frame position in the graphics window to transfer the values to the Frame Position boxes.</td>
</tr>
<tr>
<td>Point on X axis</td>
<td>Click in one of these boxes, and then click the point position in the graphics window to transfer the values to the Point on X axis boxes.</td>
</tr>
<tr>
<td>Point on X-Y plane</td>
<td>Click in one of these boxes, and then click the point position in the graphics window to transfer the values to the Point on X-Y plane boxes.</td>
</tr>
<tr>
<td>Three Point</td>
<td>Select this option if you want to create the frame by using three points.</td>
</tr>
<tr>
<td>First Point on X axis</td>
<td>Click in one of these boxes, and then click the point position in the graphics window to transfer the values to the First Point on X axis boxes.</td>
</tr>
<tr>
<td>Second Point on X axis</td>
<td>Click in one of these boxes, and then click the point position in the graphics window to transfer the values to the Second Point on X axis boxes.</td>
</tr>
</tbody>
</table>
8 The Home Tab

8.6.2. Frame from Three Points

Continued

<table>
<thead>
<tr>
<th><strong>Point on Y axis</strong></th>
<th>Click in one of these boxes, and then click the point position in the graphics window to transfer the values to the <strong>Point on Y axis</strong> boxes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set as UCS</strong></td>
<td>Select this check box to set the created frame as the user coordinate system.</td>
</tr>
</tbody>
</table>
8.7. Workobject

Creating a workobject

1. Click Workobject to bring up a dialog box.
2. In the Misc Data group, enter the values for the new workobject.
3. In the User Frame group, do one of the following:
   - Set the position of the user frame by entering values for the Position x, y, z and the Rotation rx, ry, rz for the workobject by clicking in the Values box.
   - Select the user frame by using the Frame by points dialog box.
4. In the Object Frame group you can reposition the object frame relative to the user frame by doing any of the following:
   - Set the position of the object frame by selecting values for Position x, y, z by clicking in the Values box.
   - For the Rotation rx, ry, rz, select RPY (Euler XXY) or Quaternion, and enter the rotation values in the Values dialog box.
   - Select the object frame by using the Frame by points dialog box.
5. In the Sync Properties group, enter the values for the new workobject.
6. Click Create. The workobject will be created and displayed under the Targets node under the robot node in the Paths&Targets browser.

The Create Workobject dialog box

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify the name of the workobject.</td>
</tr>
<tr>
<td>Robot holds workobject</td>
<td>Select whether the workobject is to be held by the robot. If you select True, the robot will hold the workobject. The tool can then either be stationary or held by another robot.</td>
</tr>
<tr>
<td>Moved by mechanical unit</td>
<td>Select the mechanical unit that moves the workobject. This option is applicable only if Programmed is set to False.</td>
</tr>
<tr>
<td>Programmed</td>
<td>Select True if the workobject is to use a fixed coordinate system, and False if a movable (that is, external axes) will be used.</td>
</tr>
<tr>
<td>Position x, y, z</td>
<td>Click in one of these boxes, and then click the position in the graphics window to transfer the values to the Position boxes.</td>
</tr>
<tr>
<td>Rotation rx, ry, rz</td>
<td>Specify the rotation of the workobject in the UCS.</td>
</tr>
<tr>
<td>Frame by points</td>
<td>Specify the frame position of the user frame.</td>
</tr>
<tr>
<td>Position x, y, z</td>
<td>Click in one of these boxes, and then click the position in the graphics window to transfer the values to the Position boxes.</td>
</tr>
<tr>
<td>Rotation rx, ry, rz</td>
<td>Specify the rotation of the workobject.</td>
</tr>
<tr>
<td>Frame by points</td>
<td>Specify the frame position of the object frame.</td>
</tr>
<tr>
<td>Storage type</td>
<td>Select PERS or TASK PERS. Select the Storage Type TASK PERS if you intend to use the workobject in multimove mode.</td>
</tr>
<tr>
<td>Module</td>
<td>Select the module in which to declare the workobject.</td>
</tr>
</tbody>
</table>
8 The Home Tab

8.8. Tooldata

Creating tooldata

1. In the Layout browser, make sure the robot in which to create the tooldata is set as the active task.
2. Click Tooldata to bring up a dialog box.
3. In the Misc Data group:
   • Enter the Name of the tool.
   • Select whether the tool is to be held by the robot in the Robot holds tool list.
4. In the Tool Frame group:
   • Define the Position x, y, z of the tool.
   • Enter the Rotation rx, ry, rz of the tool.
5. In the Load Data group:
   • Enter the Weight of the tool.
   • Enter the Center of gravity of the tool.
   • Enter the Inertia of the tool.
6. In the Sync Properties group:
   • In the Storage type list, select PERS or TASK PERS. Select TASK PERS if you intend to use the tooldata in MultiMove mode.
   • In the Module list, select the module in which to declare the tooldata.
7. Click Create. The tooldata appears as a coordinate system in the graphics window.
8 The Home Tab

8.9.1. Teach Target

8.9 Target

8.9.1. Teach Target

Teaching a target

To teach a target, follow these steps:

1. In the Layout browser, select the workobject and tool for which you want to teach the target.

2. Jog the robot to the preferred position.
   
   **NOTE:** To jog a robot linearly, its VC must be running.

3. Click Teach Target.

4. A new target will be created in the browser, under the active workobject node. In the graphics window a coordinate system will be created at the TCP position. The configuration of the robot at the target will be saved.
8.9.2. Create Target

Creating a target

1. In the Layout browser, select the workobject in which you want to create the target.
2. Click Create Target to bring up a dialog box.
3. Select the Reference coordinate system you want to use to position the target:

<table>
<thead>
<tr>
<th>If you want to position the target</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>absolute in the world coordinate system of the station</td>
<td>World</td>
</tr>
<tr>
<td>relative to the position of the active workobject</td>
<td>Work Object</td>
</tr>
<tr>
<td>in a user-defined coordinate system</td>
<td>UCS</td>
</tr>
</tbody>
</table>

4. In the Points box, click Add New and then click the desired position in the graphics window to set the position of the target. You can also enter the values in the Coordinates boxes and click Add.
5. Enter the Orientation for the target. A preliminary cross will be shown in the graphics window at the selected position. Adjust the position, if necessary. To create the target, click Create.
6. If you want to change the workobject for which the target is to be created, expand the Create Target dialog box by clicking the More button. In the WorkObject list, select the workobject in which you want to create the target.
7. If you want to change the target name from the default name, expand the Create Target dialog box by clicking the More button and entering the new name in the Target name box.
8. Click Create. The target will appear in the browser and in the graphics window.

NOTE: The created target will not get any configuration for the robot axes. To add the configuration values to the target, use either ModPos or the Configurations dialog box.

NOTE: If using external axes, the position of all activated external axes will be stored in the target.

The Create Target dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>Click in one of these boxes, and then click the position in the graphics window to transfer the values to the Position boxes.</td>
</tr>
<tr>
<td>Orientation</td>
<td>Specify the orientation of the target.</td>
</tr>
<tr>
<td>Add</td>
<td>Click this button to add a point and its coordinates to the Points list.</td>
</tr>
<tr>
<td>Modify</td>
<td>Click this button to modify an already defined point, after you have selected it in the Points list and entered new values.</td>
</tr>
<tr>
<td>Points</td>
<td>The target points. To add more points, click Add New, click the desired point in the graphics window, and then click Add.</td>
</tr>
<tr>
<td>More/Less</td>
<td>Click this button to expand or collapse parts of the create target dialog box.</td>
</tr>
</tbody>
</table>
8.9.2. Create Target

*Continued*

<table>
<thead>
<tr>
<th><strong>Target name</strong></th>
<th>Here you can change the name of the target you are creating. It is visible only when the create target dialog box is expanded.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Workobject</strong></td>
<td>Here you can change the workobject in which the target is to be created. It is visible only when the create target dialog box is expanded.</td>
</tr>
</tbody>
</table>
8.9.3. Create Jointtarget

Creating a jointtarget

1. Click Create Jointtarget to bring up a dialog box.
2. If you want to change the default name of the jointtarget, enter the new name in the Name box.
3. In the Axes Values group, do as follows:
   - For the Robot axes, click the Values box and then click the down arrow. The Joint Values dialog box will be displayed. Enter the joint values in the boxes and click Accept.
   - For the Joint axes, click the Values box and then click the down arrow. The Joint Values dialog box will be displayed. Enter the joint values in the boxes and click Accept.
4. Click Create. The jointtarget will appear in the browser and in the graphics window.

The Create Jointtarget dialog box

<table>
<thead>
<tr>
<th>Name</th>
<th>Specify the name of the jointtarget.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot axes</td>
<td>Click the Values list, enter the values in the Joint values dialog box and click Accept.</td>
</tr>
<tr>
<td>External axes</td>
<td>Click the Values list, enter the values in the Joint values dialog box and click Accept.</td>
</tr>
<tr>
<td>Storage Type</td>
<td>Select the Storage Type TASK PERS if you intend to use the jointtarget in multimove mode.</td>
</tr>
<tr>
<td>Module</td>
<td>Select the module in which you want to declare the jointtarget.</td>
</tr>
</tbody>
</table>
8.10. Empty Path

Creating an empty path

1. In the Paths&Targets browser, select the folder in which you want to create the path.
2. Click Empty Path.
3. To set the correct motion properties for the targets, select the active process in the Change Active Process box in the Elements toolbar.
4. NOTE: If the active template is set to MoveAbsJoint:
   • A target that is dragged into a path will be converted into a jointtarget (recognized by a different icon on in the browser).
   • Jointtargets and their instructions can only use wobj0 and tool0.
   • One target can not be used as different types, for example, MoveJoint, but must be deleted and re-created.
   • When the target has been synchronized with the virtual controller, the jointtarget values will be calculated and inserted in the RAPID program.
8.11. Path from Curve

Prerequisites

To create a path from a curve, the curve must first have been created in the station.

Creating a path from a curve

1. In the Layout browser, select the task in which you want to create the path.
2. Click Path From Curve to bring up a dialog box.
3. In the Select curves from graphics box, click Add New and then click the requested curve in the graphics window.
4. Select Create on curve or Reference surface:

<table>
<thead>
<tr>
<th>To</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>ensure that the path is generated on the curve.</td>
<td>Create on curve</td>
</tr>
<tr>
<td>If cleared, the path may be generated on the reference surface, if one is used.</td>
<td></td>
</tr>
<tr>
<td>use a reference surface.</td>
<td>Reference Surface</td>
</tr>
<tr>
<td>The initial orientation of the target will be normal to the surface, and the parameters of the target will relate to this orientation.</td>
<td></td>
</tr>
</tbody>
</table>
5. Select the workobject in which targets are to be created from the Insert targets in workobject list.
6. Set the Target Parameters:

<table>
<thead>
<tr>
<th>To</th>
<th>Enter values in</th>
</tr>
</thead>
<tbody>
<tr>
<td>specify the rotation around the X axis.</td>
<td>Approach</td>
</tr>
<tr>
<td>specify the rotation around the Y axis.</td>
<td>Travel</td>
</tr>
<tr>
<td>specify the rotation around the Z axis.</td>
<td>Spin</td>
</tr>
<tr>
<td>create a start target at the specified distance from the first target on the curve, in the approach direction.</td>
<td>Approach</td>
</tr>
<tr>
<td>create a departure target at the specified distance from the last target on the curve, in the approach direction.</td>
<td>Depart</td>
</tr>
<tr>
<td>start the path at a different location along the curve than the start point.</td>
<td>Offset Start</td>
</tr>
<tr>
<td>end the path at a different location along the curve than the end point.</td>
<td>Offset End</td>
</tr>
<tr>
<td>create an offset which will be the same at every target along the path.</td>
<td>Local Target Offset</td>
</tr>
</tbody>
</table>

Continues on next page
7. Set the **Approximation Parameters**:

<table>
<thead>
<tr>
<th>To</th>
<th>Select or enter values in</th>
</tr>
</thead>
<tbody>
<tr>
<td>specify the maximum allowed deviation between the chord of the curve and the path. A low tolerance will generate more targets than a high tolerance.</td>
<td>Max chord dev</td>
</tr>
<tr>
<td>select whether a linear motion shall represent the curve or if circular motion should be used in the path.</td>
<td>Line/circular</td>
</tr>
<tr>
<td>specify the minimum distance between the targets. This box is applicable only if a linear motion has been selected above.</td>
<td>Min dist.</td>
</tr>
<tr>
<td>select the maximum radius for circular motions in the path. Larger circular motions will be replaced by a linear motion. This box is applicable only if a circular motion has been selected above.</td>
<td>Max rad.</td>
</tr>
</tbody>
</table>

8. Click **Create**. The path and its targets appear in the browser and in the graphics window.

**NOTE!**

The targets created do not have any robot axis configurations set, which is required for use in programs with configuration monitoring. For information about setting robot axes configurations, see *Setting robot axis configuration for paths on page 87*.

---

The Create Path From Curve dialog box

| Select curves from graphics | Select the curve that you want to create a path from by clicking **Add New** and then clicking the selected curve in the graphics window. |
| Create on curve             | This check box ensures that the path is generated on the curve. If cleared, the path may be generated on the reference surface if one is used. |
| Reference Surface           | This check box is selected when a reference surface is to be used. The initial orientation of the target will be normal to the surface, and the parameters of the target will relate to this orientation. |
| Insert targets in workobject| Select the workobject in which you want to create the path from the list. |
| **Target Parameters**       |                                                                 |
| Approach                    | Specifies the rotation angle around the X axis. |
| Travel                      | Specifies the rotation angle around the Y axis. |
| Spin                        | Specifies the rotation angle around the Z axis. |
| Approach                    | Creates a start target at the specified distance from the first target on the curve, in the approach direction. |
| Depart                      | Creates a departure target at the specified distance from the last target on the curve, in the approach direction. |
| Offset Start                | If you want a different starting point than the beginning of the curve, enter the offset from the start point in the **Offset Start** box. |
| Offset End                  | If you want a different end point than the end of the curve, enter the offset from the end point in the **Offset End** box. |
8 The Home Tab

8.11. Path from Curve

Continued

<table>
<thead>
<tr>
<th><strong>Local Target Offset</strong></th>
<th>Enter the local target offset to generate the same offset at each target.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max Chord Dev</strong></td>
<td>Specifies the maximum allowed deviation between the chord of the curve and the path. A low tolerance will generate a more accurate path with more targets than a high tolerance.</td>
</tr>
<tr>
<td><strong>Linear/circular</strong></td>
<td>Selects whether linear segments shall represent the curve or if circular segments should be used in the path.</td>
</tr>
<tr>
<td><strong>Min Distance</strong></td>
<td>Specifies the minimum distance between the targets. This box is only applicable for linear paths.</td>
</tr>
<tr>
<td><strong>Max Rad</strong></td>
<td>Specifies the maximum radius for circular movements in the path. Larger circular segments will be replaced by linear segments. This box is applicable only for circular paths.</td>
</tr>
</tbody>
</table>
8.12. MultiMove

Overview

For browsing between the pages of the MultiMove window, click the tabs in the navigation pane. By default the tabs are arranged in an order that corresponds to the typical workflow:

Setup tab

<table>
<thead>
<tr>
<th>System Config</th>
<th>Select System</th>
<th>Here you select the system that contains the robots to program.</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Each robot in the system is presented in its own row in this grid. In the columns you make the settings as described below.</td>
<td></td>
</tr>
<tr>
<td>Enable</td>
<td>Select this check box to use the robot in the MultiMove program.</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Specify if the robot holds the tool or the work piece.</td>
<td></td>
</tr>
<tr>
<td>Robot</td>
<td>Displays the name of the robot.</td>
<td></td>
</tr>
</tbody>
</table>

Path Config

<table>
<thead>
<tr>
<th>Update</th>
<th>Click this button to update the paths in the grid if any of the paths have been changed. The button turns red if a change has been detected and an update is necessary.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paths</td>
<td>Each path in the station is presented in its own row in this grid. In the columns you make the settings as described below.</td>
</tr>
<tr>
<td>Enable</td>
<td>Select this check box for the paths to use for the program.</td>
</tr>
<tr>
<td>Order</td>
<td>Displays the order in which the paths will be executed. To change the order, use the lists in the path column for rearranging the rows in which the paths appear.</td>
</tr>
<tr>
<td>Path</td>
<td>Sets the path to be executed here.</td>
</tr>
</tbody>
</table>

Start Position

<table>
<thead>
<tr>
<th>Select Robot that other shall jump to</th>
<th>When creating a new start position, select a robot that the other will try to reach here.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply</td>
<td>Jumps the other robots to the new start position.</td>
</tr>
</tbody>
</table>

Motion Behavior tab

This is used for specifying constraints and rules for how the robots shall move relative to each other. The default setting is no particular constraints, which results in the fewest joint movements. However, changing the motion behavior might be useful for:

- Locking the orientation or position of the tool.
- Optimizing cycle time or reachability by allowing tolerances.
- Avoiding collisions or singularity by restricting joint motions.
Both Joint Influence and TCP Constraints restrict the robot’s motions. Changes in these settings might result in lower performance or situations where it is impossible to find proper solutions. The weight values for Joint Weights and TCP Constraints set how much the setting for each joint or TCP direction shall affect the robots relative to each other. It is the difference between the weight values that matters, not the absolute values. If contradicting behaviors have been set, the one with the lowest weight value will win.

Tool Tolerance, instead of restricting, enables more motions. Therefore, tolerances may improve cycle and process times and enhance the reachability of the robots. Tolerances, too, have weight value; here is set how much the robots shall use the tolerance. A low value indicates that the tolerance will be used a lot, while a high value means that the robots will try to avoid using the tolerance.

The joint influence controls the balance of how much the robots will use their joints. Decreasing the weight value for one axis will restrict the motion for this axis, while increasing it will promote motion on this axis relative to alternative axes.

The TCP constraints control the position and orientation of the tool. Enabling a TCP constraint will decrease the motion of the tool and increase the motion of the work piece. The tool tolerances control the allowed deviation between the tool and the work piece. By default, tolerances are not enabled, which means that no deviation is allowed. Enabling a tolerance, if applicable, might improve motion performance. For example, if the tool is symmetric around its Z axis, you can enable the Rz tolerance without affecting the accuracy of the generated paths.

The tool offset sets a fixed distance between the tool and the paths.

<table>
<thead>
<tr>
<th>Joint Influence</th>
<th>Select Robot</th>
<th>Select the robot’s joints to constrain in this box.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joints for Robot</td>
<td>Displays the robot’s joints and their constraint weights. Each joint is presented in its own row.</td>
<td></td>
</tr>
<tr>
<td>Axis</td>
<td>Displays which axis the constraint affects.</td>
<td></td>
</tr>
<tr>
<td>Influence</td>
<td>Specify how much the motion for the axis is constrained. 0 means a locked axis, while 100 means no constraint relative to default constraint values.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TCP Constraints</th>
<th>Active TCP</th>
<th>This grid displays the position and rotations of the TCP together with their constraint weights.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable</td>
<td>Select this check box to activate the constraint for this TCP pose.</td>
<td></td>
</tr>
<tr>
<td>Pose</td>
<td>Displays the TCP pose that is affected by the constraint.</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Specify the pose value to constrain at. Either type the value, or click the Pick from TCP button to use the values of the current TCP position.</td>
<td></td>
</tr>
<tr>
<td>Influence</td>
<td>Specify how much the motion for the TCP value is constrained. 0 means a locked TCP at this pose, while 100 means no constraint relative to default constraint values.</td>
<td></td>
</tr>
</tbody>
</table>

Continues on next page
8.12. MultiMove

Create Paths tab

This tab is used for creating RobotStudio paths for the MultiMove robots. The created paths will accord with the motions displayed during the most recently played test simulation.

With the settings group you set up the MultiMove properties that connect the tasks for the tool robot and work piece robot to each other.

With the WP robot settings group you set up the properties for the task that will be generated for the work piece robot.

The generate path group contains the button that creates the paths.

<table>
<thead>
<tr>
<th>Tool Tolerance</th>
<th>Enable</th>
<th>Select this check box to activate the tolerance for this tool pose.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pose</td>
<td>Displays the tool pose that is affected by the constraint.</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>Specify the pose value to apply the tolerance around.</td>
</tr>
<tr>
<td></td>
<td>Influence</td>
<td>Specify the size of the tolerance. 0 means no deviation is allowed, while 100 means all deviations are allowed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tool Offset</th>
<th>Enable</th>
<th>Select this check box to activate the offset for this tool pose.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pose</td>
<td>Displays the tool pose that is affected by the offset setting.</td>
</tr>
<tr>
<td></td>
<td>Offset</td>
<td>Specify the value of the offset here.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Settings</th>
<th>Start ID</th>
<th>Specify the first ID number for the synchronization of the instructions for the robots.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ID step index</td>
<td>Specify the gap between the succeeding ID numbers.</td>
</tr>
<tr>
<td></td>
<td>Sync ident prefix</td>
<td>Specify a prefix for the syncident variable, which connects the sync instructions in the tasks for the tool robot and the work piece robot with each other.</td>
</tr>
<tr>
<td></td>
<td>Task list prefix</td>
<td>Specify a prefix for the tasklist variable, which identifies the tasks for the tool robot and work piece robot to synchronize.</td>
</tr>
<tr>
<td></td>
<td>Path Prefix</td>
<td>Specify a prefix for the generated paths.</td>
</tr>
<tr>
<td>WP robot settings</td>
<td>Target Prefix</td>
<td>Specify a prefix for the generated targets.</td>
</tr>
<tr>
<td></td>
<td>WP Workobject</td>
<td>Specify the work object to which the targets generated for the work piece robot shall belong.</td>
</tr>
<tr>
<td></td>
<td>WP TCP</td>
<td>Specify which tool data the work piece robot shall use when reaching its targets.</td>
</tr>
</tbody>
</table>
8 The Home Tab

8.12. MultiMove

Continued

| Generate Paths | Create paths | Clicking this button generates paths in RobotStudio for the latest tested motions according to the settings specified. |

**MultiTeach tab**

With this tab you teach complete synchronized move instructions for the robots in the MultiMove program.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Path Prefix</th>
<th>Specify a prefix for the paths to create.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Target Prefix</td>
<td>Specify a prefix for the generated targets here.</td>
</tr>
<tr>
<td></td>
<td>Start ID</td>
<td>Specify the first ID number for the synchronization of the instructions for the robots.</td>
</tr>
<tr>
<td></td>
<td>ID step index</td>
<td>Specify the gap between the succeeding ID numbers.</td>
</tr>
<tr>
<td>Sync ident prefix</td>
<td>Specify a prefix for the syncident variable, which connects the sync instructions in the tasks for the tool robot and the work piece robot with each other.</td>
<td></td>
</tr>
<tr>
<td>Task list prefix</td>
<td>Specify a prefix for the tasklist variable, which identifies the tasks for the tool robot and work piece robot to synchronize.</td>
<td></td>
</tr>
<tr>
<td>Select type of Sync instruction</td>
<td>Select the type of synchronization to use. <strong>Coordinated</strong> implies that all move instructions are synchronized for the robots. <strong>Semicoordinated</strong> implies that the robots work independently at some times and wait for each other at other (like when repositioning the work piece). For detailed information about the coordination types, see the Application manual - Multimove.</td>
<td></td>
</tr>
<tr>
<td>Setup</td>
<td>Select the robots for teaching targets. This grid also displays the workobjects and tools that will be used for the targets.</td>
<td></td>
</tr>
<tr>
<td>Teach</td>
<td>MultiTeach Information</td>
<td>Displays a hierarchal tree which contains the created move instructions. The tree is organized the same way as the tree in the Layout browser.</td>
</tr>
<tr>
<td></td>
<td>MultiTeach</td>
<td>Creates move instructions for the robots selected in the settings to their current positions. The created move instructions are immediately inserted at their correct places in the MultiTeach Information tree.</td>
</tr>
</tbody>
</table>

Continues on next page
8 The Home Tab

8.12. MultiMove

Continued

Test tab

RobotStudio’s MultiMove window has a page with commands for testing multimove programs. Its default placement is at the bottom of the MultiMove window.

The status group displays the status of the simulation, that is, whether the current settings have been tested or if errors have occurred.

In addition to the status group, information from the virtual controller is also displayed in RobotStudio’s Output window during simulation.

<table>
<thead>
<tr>
<th>Play</th>
<th>&lt;&lt; &lt; &gt; &gt;</th>
<th>Jumps the robots, respectively, to the previous and next targets in paths. The double arrow buttons jump several targets at once, while the single arrow buttons jump one target for each click.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play</td>
<td></td>
<td>Click this button to move the robots along the paths. Play also has a list box in which you can activate the following commands:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Save current position</strong>: Saves the current start position. Since the calculated motions are based on the robot start position saving the start position is useful when testing alternative solutions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Restore saved position</strong>: Moves the robots back to the saved start positions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Restore last closed loop position</strong>: Moves the robots back to the list used start position.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Close loop</strong>: Finds a suitable start position based on the robots’ current positions and prepares the calculation of movements.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Calculate</strong>: Calculates and executes the movements.</td>
</tr>
<tr>
<td>Simulation speed</td>
<td>Sets the speed at which the simulation is performed.</td>
<td></td>
</tr>
<tr>
<td>Settings</td>
<td>Stop at end</td>
<td>Select this check box to make the simulation stop after running the paths one time. If cleared, the simulation will continue playing over and over until it is stopped manually.</td>
</tr>
</tbody>
</table>

Continues on next page
### The MultiMove configure system wizard

The MultiMove configure system wizard guides you through configuring robots and workobjects for MultiMove system. If the workobjects are not configured correctly when the MultiMove functions are started, you will be asked whether to run the wizard. You can also start it manually from the Tools page of MultiMove.

The wizard contains four pages, the information pane at the bottom indicates the current page.

<table>
<thead>
<tr>
<th>Workpiece robot</th>
<th>The workpiece robot page contains a list in which you select the robot that holds the workpiece. Only one robot can be set up as workpiece robot. If your station has several robots that hold the workpiece, set up one of them as workpiece robot and the other as tools robots, and create paths for these robots in which they only hold the workpiece.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool robots</td>
<td>The tool robots page contains a list in which you select the robots that operate on the work piece. All robots selected as tools robots will be coordinated to the workpiece robot. Any robot of the system that is selected neither as workpiece robot or tool robot will not be coordinated.</td>
</tr>
<tr>
<td>Workobjects</td>
<td>The Workobjects page contains a box for each tool robot in which you specify the workobject in which the targets for the MultiMove paths shall be created. The wizard will attach this workobject to the workpiece robot, for enabling MultiMove. Either type in a name for a new Workobject to create in the box, or select the Use existing WorkObject check box and then select one from the list.</td>
</tr>
<tr>
<td>Result</td>
<td>The Result page displays a summary of the station configuration. Either click Done to finish or Previous to go back and change the setup.</td>
</tr>
</tbody>
</table>
NOTE!

The wizard will not help you configure the RobotWare system correctly. If the correct options are not used you will not be able to synchronize generated MultiMove paths to the virtual controller, even if you can use the MultiMove functions in RobotStudio.

The Analyze path tool

This tool checks whether existing paths are coordinated correctly for MultiMove.

The analyzer opens in a window of its own and contains three pages. The information pane at the bottom indicates the current page.

<table>
<thead>
<tr>
<th>Select Paths</th>
<th>Enable</th>
<th>Select to include the task in the analysis.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Displays the name of the task.</td>
<td></td>
</tr>
<tr>
<td>Path</td>
<td>Select the path to analyze for the current task.</td>
<td></td>
</tr>
<tr>
<td>Analyze</td>
<td>Analyze</td>
<td>Click this button to start the analysis.</td>
</tr>
</tbody>
</table>

Report

- xx0600002648: OK. The paths are compatible in the specified aspect.
- xx0600002649: Information. The paths are not fully compatible in the specified aspect, but the robot program may still be executable.
- xx0600002650: Error. The paths are not compatible in the specified aspect, and the robot program is not executable.

The Recalculate ID tool

This is one of the tools for working manually with MultiMove programs. It sets new sync ID arguments to the move instructions in a MultiMove path. By using the tool with the same start ID and ID step index on all paths that shall be synchronized, you can be sure that the IDs match if all paths contain the same number of move instructions.

<table>
<thead>
<tr>
<th>Title</th>
<th>Displays the name of the path to recalculate IDs for.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start ID</td>
<td>Sets the number of the first ID in the path.</td>
</tr>
<tr>
<td>ID Step index</td>
<td>Sets the size of the step to increase the ID number for each move instruction.</td>
</tr>
<tr>
<td>Only update instructions that have ID defined</td>
<td>Select this check box to recalculate only those IDs for instructions that have existing IDs. Clear this check box to also create IDs for instructions that have no IDs (for example, if you have added new move instructions that shall be coordinated).</td>
</tr>
<tr>
<td>The Only update instructions between SyncMove On/Off check box.</td>
<td>Select this check box to affect only those move instructions that belong to already synchronized parts of the path. Clear this check box to update IDs for instructions in all parts of the path.</td>
</tr>
</tbody>
</table>
The Convert path to MultiMove path tool

This is one of the tools for working manually with MultiMove programs. It adds sync ID arguments to all move instructions in the path and, optionally, SyncMoveOn/Off instructions, thus preparing an ordinary path for MultiMove use.

You use the tool on one path a time, so for creating a MultiMove program, you convert one path for each robot and then create a tasklist and syncidents which you add to the Sync instructions.

<table>
<thead>
<tr>
<th>Title</th>
<th>Displays the name of the path to recalculate IDs for.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start ID</td>
<td>Sets the number of the first ID in the path.</td>
</tr>
<tr>
<td>ID Step index</td>
<td>Sets the size of the step to increase the ID number for each move instruction.</td>
</tr>
<tr>
<td>Add SyncMove On/Off before and after</td>
<td>Select this check box to add instructions that start and stop the synchronization.</td>
</tr>
</tbody>
</table>

The Create Tasklist tool

This is one of the tools for working manually with MultiMove programs. It creates a variable of the RAPID data type tasks, which identifies the tasks that will be synchronized. In each SyncMoveOn or WaitSyncTask instruction, you then specify which tasklist to use.

<table>
<thead>
<tr>
<th>Tasklist name</th>
<th>Specifies the name of the tasklist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks that will be included</td>
<td>Select the check box for each task to include in the list.</td>
</tr>
</tbody>
</table>

The Create Syncident tool

This is one of the tools for working manually with MultiMove programs. It creates a variable of the RAPID data type SyncIdent, which identifies the sync instructions that shall be synchronized.

<table>
<thead>
<tr>
<th>Syncident name</th>
<th>Specifies the name of the SyncIdent variable to create.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks that Syncident will be created in</td>
<td>Select the check box for each task to use this Syncident in.</td>
</tr>
</tbody>
</table>
8.13. Teach Instruction

Teaching a move instruction

1. In the Layout browser, make sure the settings for active robot, workobject, tool, motion type and path are appropriate for the move instruction to create.

2. Jog the robot to the desired location. If jogging the robot using the freehand mode, you can also use snap modes for snapping its TCP to objects in the station.

3. Click Teach Instruction. A move instruction is now created last in the path.
8 The Home Tab

8.14. Move Instruction

8.14. Move Instruction

Creating a move instruction and a corresponding target

To create a move instruction, follow these steps:
1. Click **Move Instruction** to bring up a dialog box.
2. Select the **Reference coordinate system** for the move instruction.
3. Enter the **Position** to reach for the move instruction by clicking **Add New** in the **Coordinates** box and then click the required to-points in the graphics window. You can also enter the values in the **Coordinates boxes** and click **Add**.
4. Enter the **Orientation** for the move instruction.
5. By clicking the **More/Less** button, you can expand or collapse the **Create Move Instruction** dialog box. When the dialog box has been expanded, you can change the **Target name** and the **Work object** to which the target (with the move instruction) will belong.
6. Click **Create** to create the move instruction. The move instruction will appear under the path node as a reference to the target.

The Create Move Instruction dialog box for jointtarget movements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Here you can change the name of the target you create when creating the move instruction.</td>
</tr>
<tr>
<td>Robot axes</td>
<td>Specify the joint values for the robot. Select the box and click the list to set the values.</td>
</tr>
<tr>
<td>External axes box</td>
<td>Specify the joint values for external axes, if any exist in the station. Select the box and click the list to set the values.</td>
</tr>
<tr>
<td>Storage Type</td>
<td>Click this button to expand or collapse parts of the create move instruction dialog box.</td>
</tr>
<tr>
<td>Module</td>
<td>Specify the module in which the jointtarget shall be declared.</td>
</tr>
</tbody>
</table>
8.15. Action Instruction

Creating an action instruction

1. In the Paths&Targets browser, select where to insert the action instruction.

<table>
<thead>
<tr>
<th>To insert the action instruction</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>at the beginning of a path</td>
<td>the path</td>
</tr>
<tr>
<td>after another instruction</td>
<td>the proceeding instruction</td>
</tr>
</tbody>
</table>

2. Right click Path and select Insert Action Instruction.
   The Create Action Instruction dialog box appears.

3. From the Instruction Templates list, select the action instruction to create.

4. Optionally, modify instruction arguments in the Instruction Arguments grid. For detailed information about the arguments for each instruction, see the Rapid reference manual.

5. Click Create.
8.16. Instruction Template Manager

**Importing a template**
1. Click **Import** to bring up the **Open File** dialog box.
2. Select the file to import, and click **OK**.

**Exporting a template**
1. Select an exportable node in the tree view and click **Export** to bring up the **Save File** dialog box.
2. Click **OK**.

**Validating the templates**
1. Select a node in the tree view and click **Validate**.
   Any invalidity will be reflected by the icons and ToolTips of the respective node and reported in the Output window.

**About the Instruction Template Manager**

The Instruction Template Manager is used to add support for instructions other than the default set that comes with the RobotStudio.

For example, a robot controller system with the RobotWare Dispense option has specialized move instructions related to gluing like DispL and DispC. You can manually define the instruction templates for these using the Instruction Template Manager. The instruction templates are exported to XML format and reused later.

RobotStudio has pre-defined XML files that are imported and used for robot controller systems with the appropriate RobotWare options. These XML files has both the Move and Action instructions.

The instruction template supports the following Robotware options:

- Cap (Continuous Application Process)
- Disp (Dispense)
- Trigg (Fixed Position Events)
- Spot Pneumatic
- Spot Servo
- Spot Servo Equalizing
- Paint

*Continues on next page*
NOTE! It is recommended to use RobotStudio ArcWelding PowerPac while using RobotWare Arc.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Buttons for importing, exporting and validating.</td>
</tr>
<tr>
<td>2</td>
<td>The instruction template tree. This hierarchical tree set organizes the templates. Templates are always the lowest level nodes. For details about specific nodes in the tree, see item 5 and below.</td>
</tr>
<tr>
<td>3</td>
<td>Brief description for editing and creating instruction templates.</td>
</tr>
<tr>
<td>4</td>
<td>The Instruction grid. All arguments and settings for the object selected in the tree are displayed here. Only white boxes are editable. Red values indicate that the values are invalid.</td>
</tr>
<tr>
<td>5</td>
<td>The Instruction templates top node. Here you can see to which task the templates belong.</td>
</tr>
<tr>
<td>6</td>
<td>The Action instructions node contains everything related to action instruction templates.</td>
</tr>
<tr>
<td>7</td>
<td>An Action instruction description node, here represented by the Set DO instruction, defines the arguments that can be set for the action instruction templates of that kind. You can create action instruction descriptions for all action instructions known by the system running on the virtual controller.</td>
</tr>
<tr>
<td>8</td>
<td>An Action instruction template node, here represented by Default, contains instances of the action instruction descriptions, with defined values for the arguments.</td>
</tr>
<tr>
<td>9</td>
<td>The Move instructions node contains everything related to move instruction templates.</td>
</tr>
</tbody>
</table>
### 8 The Home Tab

**8.16. Instruction Template Manager**

Continued

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>The Move instruction descriptions node contains all move instructions descriptions for the task. If the description for an instruction is not present in the list, right-click this node to add it. You can create move instruction descriptions for all move instructions known by the system running on the virtual controller.</td>
</tr>
<tr>
<td>11</td>
<td>A Move instruction description node, here represented by the MoveAbsJ node, defines the arguments that can be set for the move instruction templates of that kind. Unlike action instructions, instruction templates related to a certain move instruction descriptions are not stored in child nodes under the description, due to a more complex hierarchy.</td>
</tr>
<tr>
<td>12</td>
<td>The Process definitions node, which gathers all process definitions, contains sets of process templates which in turn contain instruction templates optimized for specific processes.</td>
</tr>
<tr>
<td>13</td>
<td>A Process definition node, here represented by the generic Move process, contains sets of process templates which in turn contain instruction templates optimized for specific processes.</td>
</tr>
<tr>
<td>14</td>
<td>A Process template node, here represented by the generic Default process, contains sets of move instruction templates with argument values optimized for specific processes. A process template can hold one move instruction template for each move instruction type defined by a move instruction description.</td>
</tr>
<tr>
<td>15</td>
<td>A Move instruction template node, here represented by MoveJ, contains instances of move instruction descriptions with argument values customized for specific processes.</td>
</tr>
</tbody>
</table>
8.17 The Freehand Group

8.17.1. Move

Moving an item

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

Rotating an item

1. In the Layout browser, select the item you want to rotate.
2. Click Rotate.
3. In the graphics window, click one of the rotational rings and drag the item into position.
   If you press the ALT key while rotating, the item will snap 10 degrees at a time.
8.17.3. Jog Joint

Jogging the joints of a robot

1. In the **Layout** browser, select the robot you want to move.
2. Click **Jog Joint**.
3. Click the joint you want to move and drag it to the preferred position.

   If you press the **ALT** key when jogging the joints of the robot, the robot will move 10 degrees at a time. If you press the **f** key, the robot will move 0.1 degree at a time.
8.17.4. Jog Linear

Jogging the TCP of a robot

1. In the Layout browser, select the robot you want to move.
2. In the Freehand group, click Jog Linear. A coordinate system will be displayed at the TCP of the robot.
3. Click the axis you want to move and drag the TCP to the preferred position.

If you press the f key while jogging the robot linearly, the robot will move with a smaller step size.
8.17.5. MultiRobot Jog

Jogging robots in multirobot mode

1. In the Freehand group, click MultiRobot Jog. Select the robots to be jogged from the list of available robots.

2. Select the jogging mode, jog one of the robots and the other ones will follow the movement.
8 The Home Tab

8.18. Viewpoint

8.18. Viewpoint

Overview
A Viewpoint stores the location and direction of a virtual camera in the 3D environment. It stores points of interest in a station that are used to create camera movements during simulation.

Creating Viewpoint
You can create a viewpoint in a station in two ways:
1. In the Home tab, click View and select Create Viewpoint.
2. In the Layout browser, right-click the station and select Create Viewpoint.
A viewpoint is created and displayed (as an eye icon) in the layout browser to the left.
NOTE! The position and direction of the Viewpoint can also be visualized as an arrow in the 3D graphics.
NOTE! By default, the newly created viewpoints are not visible and cannot be selected by clicking on the graphics.

Viewpoint functions
In the Layout browser, right-click Viewpoint to perform these functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move to Viewpoint</td>
<td>Moves the active 3D view to the location stored in the viewpoint.</td>
</tr>
<tr>
<td>Update Viewpoint</td>
<td>Modifies the viewpoint to the current location and direction of the active 3D view.</td>
</tr>
<tr>
<td>Visible</td>
<td>Toggles the visibility of the viewpoint 3D representation.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the viewpoint.</td>
</tr>
<tr>
<td>Rename</td>
<td>Renames the viewpoint.</td>
</tr>
</tbody>
</table>

Move to Viewpoint
You can also move an active 3D view to the location stored in the Viewpoint using the Event manager.
2. Add an event. See Event Manager on page 258.
   The Create new event dialog box appears.
3. Select Simulation under Activation and Simulation time under Event trig type. Click Next.
4. Set the activation time. Click Next.
5. Select Move to Viewpoint from Set Actio type. Click Next.
6. Select the viewpoint from Select Viewpoint and set the transition time.
7. Click Finish.
   NOTE! Move to viewpoint function is also executed when replaying the simulation in a Station Viewer.
9 The Modeling Tab

9.1. Overview

The Modeling tab contains the controls for creating and grouping components, creating bodies, measurements and CAD operations.
9.2. Component Group

Creating a component group

1. Click **Component Group**. The **Group** node will be displayed in the **Layout** browser.
2. Click the objects to add to the group. Drag them to the **Group** node.
9.3. Empty Part

Creating an empty part

1. Click Empty Part. The Part node will be displayed in the Layout browser.
9 The Modeling Tab

9.4. Solid

Creating a solid

1. Click **Solid** and then click the type of solid you want to create to bring up a dialog box.
2. Enter requested values in in the dialog box and click **Create**. For detailed information about the specific dialog box for the curve to create, see below:

The Create Box dialog box

![Create Box Diagram]

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corner Point (A)</td>
<td>Click in one of these boxes, and then click the corner point in the graphics window to transfer the values to the Corner Point boxes, or type the position. The corner point will be the local origin of the box.</td>
</tr>
<tr>
<td>Orientation</td>
<td>If the object shall be rotated relative to the reference coordinate system, specify the rotation.</td>
</tr>
<tr>
<td>Length (B)</td>
<td>Specify the box dimension along its X axis.</td>
</tr>
<tr>
<td>Width (C)</td>
<td>Specify the box dimension along its Y axis.</td>
</tr>
<tr>
<td>Height (D)</td>
<td>Specify the box dimension along its Z axis.</td>
</tr>
</tbody>
</table>

The Create Box from 3-Points dialog box

![Create Box from 3-Points Diagram]

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corner Point (A)</td>
<td>This point will be the local origin of the box. Either type the position, or click in one of the boxes and then select the point in the graphics window.</td>
</tr>
</tbody>
</table>
The Modeling Tab

9.4. Solid

Continued

<table>
<thead>
<tr>
<th>Point on diagonal of XY-plane (B)es</th>
<th>This point is the corner, diagonal to the local origin. It sets the X and Y directions of the local coordinate system, as well as the dimension of the box along these axes. Either type the position, or click in one of the boxes and then select the point in the graphics window.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indication Point Z-axis (C)</td>
<td>This point is the corner above the local origin. It sets the Z direction of the local coordinate system, as well as the dimension of the box along the Z axis. Either type the position, or click in one of the boxes and then select the point in the graphics window.</td>
</tr>
</tbody>
</table>

The Create Cone dialog box

![Create Cone diagram](image)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Center Point (A)</td>
<td>Click in one of these boxes, and then click the center point in the graphics window to transfer the values to the Base Center Point boxes, or type the position. The center point will be the local origin of the cone.</td>
</tr>
<tr>
<td>Orientation</td>
<td>If the object shall be rotated relative to the reference coordinate system, specify the rotation.</td>
</tr>
<tr>
<td>Radius (B)</td>
<td>Specify the radius of the cone.</td>
</tr>
<tr>
<td>Diameter</td>
<td>Specify the diameter of the cone.</td>
</tr>
<tr>
<td>Height (C)</td>
<td>Specify the height of the cone.</td>
</tr>
</tbody>
</table>

The Create Cylinder dialog box

![Create Cylinder diagram](image)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
</table>
9.4. Solid

Continued

<table>
<thead>
<tr>
<th>Base Center Point (A)</th>
<th>Click in one of these boxes, and then click the center point in the graphics window to transfer the values to the Base Center Point boxes, or type the position. The center point will be the local origin of the cylinder.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orientation</td>
<td>If the object shall be rotated relative to the reference coordinate system, specify the rotation.</td>
</tr>
<tr>
<td>Radius (B)</td>
<td>Specify the radius of the cylinder.</td>
</tr>
<tr>
<td>Diameter</td>
<td>Specify the diameter of the cylinder.</td>
</tr>
<tr>
<td>Height (C)</td>
<td>Specify the height of the cylinder.</td>
</tr>
</tbody>
</table>

The Create Pyramid dialog box

Reference: Select the Reference coordinate system to which all positions or points will be related.

<table>
<thead>
<tr>
<th>Base Center Point (A)</th>
<th>Click in one of these boxes, and then click the center point in the graphics window to transfer the values to the Base Center Point boxes, or type the position. The center point will be the local origin of the pyramid.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orientation</td>
<td>If the object shall be rotated relative to the reference coordinate system, specify the rotation.</td>
</tr>
<tr>
<td>Center to Corner Point (B)</td>
<td>Either type the position, or click in the box and then select the point in the graphics window.</td>
</tr>
<tr>
<td>Height (C)</td>
<td>Specify the height of the pyramid.</td>
</tr>
<tr>
<td>Number of Sides</td>
<td>Specify the number of sides of the pyramid. The maximum number of sides is 50.</td>
</tr>
</tbody>
</table>

The Create Sphere dialog box

Reference: Select the Reference coordinate system to which all positions or points will be related.
9 The Modeling Tab

9.4. Solid

Continued

<table>
<thead>
<tr>
<th>Center Point (A)</th>
<th>Click in one of these boxes, and then click the center point in the graphics window to transfer the values to the Center Point boxes, or type the position. The center point will be the local origin of the sphere.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius (B)</td>
<td>Specify the radius of the sphere.</td>
</tr>
<tr>
<td>Diameter</td>
<td>Specify the diameter of the sphere.</td>
</tr>
</tbody>
</table>
9.5. Surface

Creating a surface

1. Click **Surface** and then click the type of solid you want to create to bring up a dialog box.
2. Enter requested values in the dialog box and click **Create**. For detailed information about the specific dialog box for the curve to create, see below:

The Create Surface Circle dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the <strong>Reference</strong> coordinate system, to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center Point (A)</td>
<td>Click in one of these boxes, and then click the center point in the graphics window to transfer the values to the <strong>Center Point</strong> boxes, or type the position. The center point will be the local origin of the circle.</td>
</tr>
<tr>
<td>Orientation</td>
<td>If the object shall be rotated relative to the reference coordinate system, specify the rotation.</td>
</tr>
<tr>
<td>Radius (B)</td>
<td>Specify the radius of the circle.</td>
</tr>
<tr>
<td>Diameter</td>
<td>Specify the diameter of the circle.</td>
</tr>
</tbody>
</table>

The Create Rectangle dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the <strong>Reference</strong> coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Point (A)</td>
<td>Click in one of these boxes, and then click the center point in the graphics window to transfer the values to the <strong>Start Point</strong> boxes, or type the position. The start point will be the local origin of the rectangle.</td>
</tr>
<tr>
<td>Orientation</td>
<td>If the object shall be rotated relative the reference coordinate system, specify the rotation.</td>
</tr>
</tbody>
</table>

Continues on next page
9 The Modeling Tab

9.5. Surface

The Create Surface Polygon dialog box

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Length (B)</strong></td>
<td>Specify the length of the rectangle.</td>
</tr>
<tr>
<td><strong>Width (C)</strong></td>
<td>Specify the width of the rectangle</td>
</tr>
</tbody>
</table>

The Create Surface from Curve dialog box

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Select Curve from graphics</strong></td>
<td>Select a curve by clicking it in the graphics window.</td>
</tr>
</tbody>
</table>
Creating a curve

1. Click Curve and then click the curve you want to create to bring up a dialog box.
2. Enter requested values in in the dialog box and click Create. For detailed information about the specific dialog box for the curve to create, see below:

The Create Line dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start point (A)</td>
<td>Click in one of these boxes, and then click the start point in the graphics window to transfer the values to the Start Point boxes.</td>
</tr>
<tr>
<td>End Point (B)</td>
<td>Click in one of these boxes, and then click the end point in the graphics window to transfer the values to the End Point boxes.</td>
</tr>
</tbody>
</table>

The Create Circle dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center point (A)</td>
<td>Click in one of these boxes, and then click the center point in the graphics window to transfer the values to the Center Point boxes.</td>
</tr>
<tr>
<td>Orientation</td>
<td>Specify the orientation coordinates for the circle.</td>
</tr>
<tr>
<td>Radius (A-B)</td>
<td>Specify the radius of the circle.</td>
</tr>
<tr>
<td>Diameter</td>
<td>Alternatively, specify the diameter.</td>
</tr>
</tbody>
</table>
9 The Modeling Tab

9.6. Curve

Continued

The Create Three Points Circle dialog box

Reference | Select the Reference coordinate system to which all positions or points will be related.
--- | ---
First Point (A) | Click in one of these boxes, and then click the first point in the graphics window to transfer the values to the First Point boxes.
Second Point (B) | Click in one of these boxes, and then click the second point in the graphics window to transfer the values to the Second Point boxes.
Third Point (C) | Click in one of these boxes, and then click the third point in the graphics window to transfer the values to the Third Point boxes.

The Create Arc dialog box

Reference | Select the Reference coordinate system to which all positions or points will be related.
--- | ---
Start Point (A) | Click in one of these boxes, and then click the start point in the graphics window to transfer the values to the Start Point boxes.
Mid Point (B) | Click in one of these boxes, and then click the second point in the graphics window to transfer the values to the Mid Point boxes.
End Point (C) | Click in one of these boxes, and then click the end point in the graphics window to transfer the values to the End Point boxes.

Continues on next page
The Create Elliptical Arc dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center Point (A)</td>
<td>Click in one of these boxes, and then click the center point in the graphics window to transfer the values to the Center Point boxes.</td>
</tr>
<tr>
<td>Major Axis End Point (B)</td>
<td>Click in one of these boxes, and then click the end point for the major axis of the ellipse in the graphics window to transfer the values to the Major Axis End Point boxes.</td>
</tr>
<tr>
<td>Minor Axis End Point (C)</td>
<td>Click in one of these boxes, and then click the end point for the minor axis of the ellipse in the graphics window to transfer the values to the Minor Axis End Point boxes.</td>
</tr>
<tr>
<td>Start Angle (α)</td>
<td>Specify the start angle for the arc, measured from the major axis.</td>
</tr>
<tr>
<td>End Angle (β)</td>
<td>Specify the end angle for the arc, measured from the major axis.</td>
</tr>
</tbody>
</table>

The Create Ellipse dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center Point (A)</td>
<td>Click in one of the Center Point boxes, and then click the center point in the graphics window to transfer the values to the Center Point boxes.</td>
</tr>
<tr>
<td>Major Axis End Point (B)</td>
<td>Click in one of these boxes, and then click the end point for the major axis of the ellipse in the graphics window to transfer the values to the Major Axis End Point boxes.</td>
</tr>
<tr>
<td>Minor Radius (C)</td>
<td>Specify the length of the minor axis of the ellipse. The minor radius will be created perpendicular to the major axis.</td>
</tr>
</tbody>
</table>
The Modeling Tab

9.6. Curve

Continued

The Create Rectangle dialog box

Reference

Select the Reference coordinate system to which all positions or points will be related.

Start Point (A)

Click in one of these boxes, and then click the start point in the graphics window to transfer the values to the Start Point boxes. The rectangle will be created in the positive coordinate directions.

Orientation

Specify the orientation coordinates for the rectangle.

Length (B)

Specify the length of the rectangle along the x axis.

Width (C)

Specify the width of the rectangle along the y axis.

The Create Polygon dialog box

Reference

Select the Reference coordinate system to which all positions or points will be related.

Center Point (A)

Click in one of these boxes, and then click the center point in the graphics window to transfer the values to the Center Point boxes.

First Vertex Point (B)

Click in one of these boxes, and then click the first vertex point in the graphics window to transfer the values to the First Vertex Point boxes. The distance between the center point and the first vertex point will be used for all vertex points.

Vertices

Specify the number of points to be used when creating the polygon. The maximum number of vertices is 50.

Continues on next page
The Create Polyline dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Coordinates</td>
<td>Specify each node of the polyline here, one at a time, by either typing the values, or by clicking in one of these boxes, and then selecting the point in the graphics window to transfer its coordinates.</td>
</tr>
<tr>
<td>Add</td>
<td>Click this button to add a point and its coordinates to the list.</td>
</tr>
<tr>
<td>Modify</td>
<td>Click this button to modify an already defined point, after you have selected it in the list and entered new values.</td>
</tr>
<tr>
<td>List</td>
<td>The nodes of the polyline. To add more nodes, click Add New, click the desired point in the graphics window, and then click Add.</td>
</tr>
</tbody>
</table>

The Create Spline dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the Reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Coordinates</td>
<td>Specify each node of the spline here, one at a time, by either typing the values, or by clicking in one of these boxes, and then selecting the point in the graphics window to transfer its coordinates.</td>
</tr>
<tr>
<td>Add</td>
<td>Click this button to add a point and its coordinates to the list.</td>
</tr>
<tr>
<td>Modify</td>
<td>Click this button to modify an already defined point, after you have selected it in the list and entered new values.</td>
</tr>
<tr>
<td>List</td>
<td>The nodes of the spline. To add more nodes, click Add New, click the desired point in the graphics window, and then click Add.</td>
</tr>
</tbody>
</table>
9.7. Border

Creating a curve

1. Click \textit{Border} and then click the border you want to create to bring up a dialog box.
2. Enter requested values in the dialog box and click \textit{Create}. For detailed information about the specific dialog box for the border to create, see below:

\pagebreak

The Create Border Between Bodies dialog box

To use the create border between bodies command, the station must contain at least two objects.

\begin{table}
\begin{tabular}{|l|l|}
\hline
\textbf{First Body} & Click in this box and then select the first body in the graphics window. \\
\hline
\textbf{Second Body} & Click in this box and then select the second body in the graphics window. \\
\hline
\end{tabular}
\end{table}

The Create Border Around Surface dialog box

To use the create border around surface command, the station must contain at least one object with a graphical representation.

\begin{table}
\begin{tabular}{|l|l|}
\hline
\textbf{Select Surface} & Click in this box and then select a surface in the graphics window. \\
\hline
\end{tabular}
\end{table}

\pagebreak

Continues on next page
9 The Modeling Tab

9.7. Border

Continued

The Create Border From Points dialog box

To use the create border from points command, the station must contain at least one object.

<table>
<thead>
<tr>
<th>Selected Object</th>
<th>Click in this box and then select an object in the graphics window.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Coordinates</td>
<td>Specify the points that define the border here, one at a time, by either typing the values, or by clicking in one of these boxes, and then selecting the point in the graphics window to transfer its coordinates.</td>
</tr>
<tr>
<td>Add</td>
<td>Click this button to add a point and its coordinates to the list.</td>
</tr>
<tr>
<td>Modify</td>
<td>Click this button to modify an already defined point, after you have selected it in the list and entered new values.</td>
</tr>
<tr>
<td>List</td>
<td>The points that define the borders. To add more points, click <strong>Add New</strong>, click the desired point in the graphics window, and then click <strong>Add</strong>.</td>
</tr>
</tbody>
</table>
9.8. Intersect

The Intersect dialog box

<table>
<thead>
<tr>
<th>Keep Original</th>
<th>Select this check box to keep the original bodies when creating the new body.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intersect... (A)</td>
<td>Select the body from which you want to make an intersection (A) by clicking it in the graphics window.</td>
</tr>
<tr>
<td>...and (B)</td>
<td>Select the body with which you want to make an intersection (B) by clicking it in the graphics window. A new body will be created based on the common area between the selected bodies A and B.</td>
</tr>
</tbody>
</table>
9.9. Subtract

The Subtract dialog box

<table>
<thead>
<tr>
<th>Keep Original</th>
<th>Select this check box to keep the original bodies when creating the new body.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtract... (A)</td>
<td>Select the body from which you want to subtract (A) by clicking it in the graphics window.</td>
</tr>
<tr>
<td>...with (B)</td>
<td>Select the body you want to subtract (B) by clicking it in the graphics window. A new body will be created based on the area of body A subtracted with the common volume between body A and B.</td>
</tr>
</tbody>
</table>
9.10. Union

The Union dialog box

<table>
<thead>
<tr>
<th>Keep Original</th>
<th>Select this check box to keep the original bodies when creating the new body.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union... (A)</td>
<td>Select the body that you want to unify (A) by clicking it in the graphics window.</td>
</tr>
<tr>
<td>...and (B)</td>
<td>Select the body that you want to unify (B) by clicking it in the graphics window. A new body will be created based on the areas of the two selected bodies A and B.</td>
</tr>
</tbody>
</table>
9 The Modeling Tab

9.11. Extrude Surface or Curve

Extruding a surface or curve

1. From the selection level toolbar, select Surface or Curve, as appropriate.
2. In the graphics window, select the surface or curve you wish to extrude. Click Extrude Surface or Extrude Curve, as appropriate. The Extrude Surface or Curve dialog box opens below the Modeling browser.
3. For extrusion along a vector, fill in the values. For extrusion along a curve, select the Extrude Along Curve option. Click the Curve box, and select the curve in the Graphics window.
4. If you wish the form to appear as a surface model, clear the Make Solid check box.
5. Click Create.

The Extrude Surface or Curve dialog box

<table>
<thead>
<tr>
<th>Surface or Curve</th>
<th>Denotes the surface or curve to be extruded. To select the surface or curve, first click in the box, then select the surface or curve in the graphics window.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extrude Along Vector</td>
<td>Enables extrusion along a specified vector.</td>
</tr>
<tr>
<td>From Point (mm)</td>
<td>The start point of the vector.</td>
</tr>
<tr>
<td>To Point (mm)</td>
<td>The end point of the vector.</td>
</tr>
<tr>
<td>Extrude Along Curve</td>
<td>Enables extrusion along a specified curve.</td>
</tr>
<tr>
<td>Curve</td>
<td>Denotes the curve used as a sweep path.</td>
</tr>
</tbody>
</table>

To select the curve, first click the box, then the curve in the graphics window.

Make Solid              | Select this check box to convert the extruded form into a solid.                                                                                                                         |
9.12. Line from Normal

Creating a line from normal

1. Click Surface Selection.
2. Click Line to Normal to bring up a dialog box.
3. Click on a face to select it in the Select Face box.
4. In the Length box, specify a length for the line.
5. Optionally, select the Invert Normal check box to invert the direction of the line.
6. Click Create.
9.13. The Measure Group

TIP!
Make sure to select the appropriate snap mode and selection levels before making your measurements.

Measuring distances or angles

1. Click the type of measurement you want to use:

<table>
<thead>
<tr>
<th>To measure the</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>distance between two points you select in the graphics window.</td>
<td>Point to point</td>
</tr>
<tr>
<td>angle defined by three points you select in the graphics window. The first point to select is the converging point, thereafter you shall select one point on each line.</td>
<td>Angle</td>
</tr>
<tr>
<td>diameter, with the circle defined by three points you select from the graphics window.</td>
<td>Diameter</td>
</tr>
<tr>
<td>closest distance between two objects you select in the graphics window.</td>
<td>Minimum distance</td>
</tr>
</tbody>
</table>

The mouse pointer will turn into a ruler when you have activated any of the measurement functions.

2. In the graphics window, select the points or objects to measure between. Information about the measuring points is displayed in the Output window.

The results will be displayed in the Measurements tab in the Output window when all points have been selected.

3. Optionally, repeat step 3 for making a new measurement of the same type.

TIP!
You can also activate and deactivate the measurement functions from the measurement toolbar.
9.14. Create Mechanism

Create a new mechanism

1. Click Create Mechanism.
   The Mechanism Modeler opens in create mode.
2. In the Mechanism Model Name box, enter a mechanism name.
3. From the Mechanism Type list, select a mechanism type.
4. In the tree structure, right-click Links, and then click Add Link to bring up the Create Link dialog box.
   A suggested name appears in the Link Name box.
5. In the Selected Part list, select a part (which will be highlighted in the graphics window) and click the arrow button to add the part to the Parts list box.
   The Selected Part list then automatically selects the next part, if any more are available. Add these, as required.
6. Select a part in the Parts list box, enter any values in the Selected Parts group boxes, and then click Apply to Part.
   Repeat for each part, as required.
7. Click OK.
8. In the tree structure, right-click Joints, and then click Add Joint to bring up the Create Joint dialog box.
   A suggested name appears in the Joint Name box.
9. Complete the Create Joint dialog box, and then click OK.
10. In the tree structure, right-click Frame/Tool Data, and then click Add Frame/Tool to bring up the Create Frame/Tool dialog box.
    A suggested name appears in the Frame/Tool Data name box.
11. Complete the Create Frame/Tool dialog box, and then click OK.
    The validity criteria for the Frame/Tool node are as follows:
12. In the tree structure, right-click Calibration, and then click Add Calibration to bring up the Create Calibration dialog box.
13. Complete the Create Calibration dialog box, and then click OK.
14. In the tree structure, right-click Dependency, and then click Add Dependency to bring up the Create Dependency dialog box.
15. Complete the Create Dependency dialog box, and then click OK.
16. If all nodes are valid, compile the mechanism, see Compiling a mechanism on page 248.

Create Conveyor mechanism

1. Click Create Mechanism.
   The Mechanism Modeler opens in create mode.
2. In the Mechanism Model Name dialog box, enter a mechanism name.
3. From the Mechanism Type list, select Conveyor.
4. From the Selected Part list, select Part.
5. In the **Position of Calibration frame** list, enter the base frame values relative to the local origin of the selected graphic component.

6. In the **Conveyor length** box, enter the length of the conveyor. The **Compile Mechanism** button is enabled.

7. In the **Attachment Points** box, set the **Pitch** and **Count** value.

8. Click **Add** to create new attachment points.

9. Click **Compile Mechanism**, to compile the mechanism. See *Compiling a mechanism on page 248*.

10. In the **Layout** browser, right-click the conveyor mechanism and select **Save As Library**. Close the station.

11. Build a new system. See *Building a new system on page 142*.

   **NOTE!** On the **Modify Options** page of the **System Builder**, scroll down to the **Motion coordination part 3** group and select **606-1 Conveyor Tracking** check box.

12. Create new station using this new system. See *Robot System on page 188*.

   **NOTE!** After starting the system, when asked to select the library for the conveyor mechanism browse and select the already saved library.

### Compiling a mechanism

When compiling, a new mechanism, created in the create mode of the Mechanism Modeler, is added to the station with the default name "Mechanism_" followed by an index number.

When compiling, an existing editable mechanism, modified in the modify mode of the Mechanism Modeler, is saved without any poses, joint mapping or transition times.

To compile a mechanism, follow these steps:

1. To compile a new or edited mechanism, click **Compile Mechanism**.
   
   The mechanism is inserted into the active station. The link parts are cloned with new names, but the corresponding links will update their part references. When the Mechanism modeler is closed, these cloned parts will be removed.

2. The Mechanism Modeler now switches to modify mode. To complete the mechanism, see below.

### Completing or modifying a mechanism

To complete the modeling of a mechanism, follow these steps:

1. If the values in the **Joint Mapping** group are correct, click **Set**.

2. Configure the Poses grid. To add a pose, click **Add** and then complete the **Create Pose** dialog box. Click **Apply**, followed by **OK**.
   
   To add a pose, click **Add** and then complete the **Create Pose** dialog box. Click **Apply**, followed by **OK**.

   To edit a pose, select it in the grid, click **Edit**, and then complete the **Modify Pose** dialog box. Click **OK**.

   To remove a pose, select it in the grid and then click **Remove**.

3. Click **Edit Transition Times** to edit transition times.

4. Click **Close**.
### The Create Mechanism dialog box

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanism Model Name</td>
<td>Specifies the model name of the mechanism.</td>
</tr>
<tr>
<td>Mechanism Type</td>
<td>Specifies the mechanism type.</td>
</tr>
<tr>
<td>Tree structure</td>
<td>The components of the mechanism in a tree structure. The tree structure will not be visible unless the mechanism is editable. Each node (link, joint, frame, calibration and dependency) can be edited in its own dialog box, see below.</td>
</tr>
<tr>
<td>Compile Mechanism</td>
<td>Click this button to compile the mechanism. This button will not be visible unless the mechanism is editable and the mechanism model name is valid.</td>
</tr>
</tbody>
</table>

### The Create Conveyor Mechanism dialog box

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanism Model Name</td>
<td>Specifies the model name of the conveyor mechanism.</td>
</tr>
<tr>
<td>Mechanism Type</td>
<td>Specifies the different mechanism types.</td>
</tr>
<tr>
<td>Selected Part</td>
<td>Specifies the part to be selected for the conveyor.</td>
</tr>
<tr>
<td>Position of Calibration frame</td>
<td>Specifies the baseframe value relative to the local origin of the selected graphic component.</td>
</tr>
<tr>
<td>Conveyor Length</td>
<td>Specifies the length of the conveyor.</td>
</tr>
<tr>
<td>Attachment Points</td>
<td>Specifies the conveyor position to attach the workpieces.</td>
</tr>
<tr>
<td>Compile Mechanism</td>
<td>Click this button to compile the mechanism. This button will not be visible unless the mechanism is editable and the mechanism model name is valid.</td>
</tr>
</tbody>
</table>

### The Create/Modify Link dialog box

A link is a moving component of a mechanism. Selecting a link node will highlight it in the graphics window.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link Name</td>
<td>Specifies the name of the link.</td>
</tr>
<tr>
<td>Selected Part</td>
<td>Specifies the parts to add to the Part list box.</td>
</tr>
<tr>
<td>Set as BaseLink</td>
<td>The BaseLink is where the kinematical chain begins. This must be the parent of the first joint. A mechanism may have only one BaseLink.</td>
</tr>
<tr>
<td>Remove Part</td>
<td>Click this button to remove a selected part from the Part list box.</td>
</tr>
<tr>
<td>Selected Part</td>
<td>This group manipulates the transform of the selected part.</td>
</tr>
<tr>
<td>Part Position</td>
<td>Specify the position of the part.</td>
</tr>
<tr>
<td>Part Orientation</td>
<td>Specify the orientation of the part.</td>
</tr>
<tr>
<td>Apply to Part</td>
<td>Click this button to apply the settings to the part.</td>
</tr>
</tbody>
</table>
The Create/Modify Joint dialog box

A joint is the axis where two links move in relation to each other, rotationally or prismatically. Selecting a joint node will display a yellow-green line in the graphics window.

<table>
<thead>
<tr>
<th>Joint Name</th>
<th>Specifies the name of the joint.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint Type</td>
<td>Specifies the joint type. The default option is Rotational. Changing the Joint Type clears the Joint Limits below.</td>
</tr>
<tr>
<td>Parent Link</td>
<td>Specifies the parent link, usually the first joint of the mechanism.</td>
</tr>
<tr>
<td>Child Link</td>
<td>Specifies the child link. To be valid, the parent and child links may not be identical to each other, and the pair must be unique.</td>
</tr>
<tr>
<td>Active</td>
<td>Select this check box to make the joint active. An active joint is one that a user can move, while an inactive joint is a slave to an active joint.</td>
</tr>
<tr>
<td>Joint Axis</td>
<td>This group specifies the axis around or along which the child moves.</td>
</tr>
<tr>
<td>First Position</td>
<td>Specify the start point of the axis vector.</td>
</tr>
<tr>
<td>Second Position</td>
<td>Specify the end point of the axis vector.</td>
</tr>
<tr>
<td>Jog Axis</td>
<td>Demonstrates how the child link moves along its axis.</td>
</tr>
<tr>
<td>Limit Type</td>
<td>Specifies the limits in each direction to which a joint may move. The options are Constant, Variable and No.</td>
</tr>
<tr>
<td>Joint Limits</td>
<td>This group is visible in Constant or Variable mode.</td>
</tr>
<tr>
<td>Min Limit</td>
<td>Specifies the minimum joint limit.</td>
</tr>
<tr>
<td>Max Limit</td>
<td>Specifies the maximum joint limit.</td>
</tr>
<tr>
<td>Variable Limits</td>
<td>In Variable mode, variable limit points may be added as an advanced way of delimiting the area of movement.</td>
</tr>
</tbody>
</table>

The Modify Frame/Tool Data dialog box

A frame/tool data node determines the link and location of a frame.

| Frame/Tool Data name | Specifies the name of the frame or tool data. |
| Belongs to Link      | Specifies the link to which the frame or tool belongs. |
9 The Modeling Tab

9.14. Create Mechanism

The Create Calibration dialog box

A calibration contains transforms for calibrating the joints. Two calibrations cannot share the same joint.

<table>
<thead>
<tr>
<th>Calibration belongs to Joint</th>
<th>Specifies the joint to be calibrated.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>Specify the position of the transform.</td>
</tr>
<tr>
<td>Orientation</td>
<td>Specify the orientation of the transform.</td>
</tr>
</tbody>
</table>

The Create Dependency dialog box

A dependency is a relationship between two joints, by either a factor or a complex formula.

<table>
<thead>
<tr>
<th>Joint</th>
<th>Specifies the joint whose motion will be controlled by other joints.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use LeadJoint and factor</td>
<td>Select this option to specify a lead joint and factor.</td>
</tr>
<tr>
<td>LeadJoint</td>
<td>Specifies the lead joint.</td>
</tr>
<tr>
<td>Factor</td>
<td>This list holds a double which denotes the extent to which the lead joint will control the main joint.</td>
</tr>
<tr>
<td>Use Formula</td>
<td>Select this option to enter a formula in the box.</td>
</tr>
</tbody>
</table>

The Modify Mechanism dialog box

The Modify Mechanism dialog box contains the objects found in the Create mechanism dialog box, as well as the following:

<table>
<thead>
<tr>
<th>Joint Mapping</th>
<th>These boxes handle the joint mapping of the mechanism. When editing, the mechanism must be disconnected from its library. The values must be integers from 1 – 6 in ascending order.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td>Click this button to set the joint mapping.</td>
</tr>
<tr>
<td>Poses</td>
<td>Displays the poses and their joint values. Selecting a pose will move the mechanism to it in the graphics window.</td>
</tr>
<tr>
<td>Add</td>
<td>Click this button to bring up the Create Pose dialog box for adding a pose.</td>
</tr>
<tr>
<td>Edit</td>
<td>Click this button to bring up the Modify Pose dialog box for editing a selected pose.</td>
</tr>
<tr>
<td>Remove</td>
<td>Click this button to remove the selected pose.</td>
</tr>
</tbody>
</table>

Continued on next page
9 The Modeling Tab

9.14. Create Mechanism

Continued

**Set Transition Times**
Click this button to edit the transition times.

### The Create/Modify Pose dialog box

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pose Name</td>
<td>Specifies the name of the pose. If the pose is a SyncPose, this box is not editable. The names &quot;HomePosition&quot; and &quot;SyncPosition&quot; are disallowed.</td>
</tr>
<tr>
<td>Home Pose</td>
<td>Select this box to specify the home pose of the mechanism. If selected, the non-editable pose name will be &quot;HomePose&quot;.</td>
</tr>
<tr>
<td>Launch Joint Jog Tool</td>
<td>Click this button to bring up the joint jog tool.</td>
</tr>
<tr>
<td>Use Current</td>
<td>Click this button to set the current joint values in the Joint Values group.</td>
</tr>
<tr>
<td>Reset Values</td>
<td>Click this button to reset the joint values in the Joint Values group to what they were when the dialog box was opened.</td>
</tr>
<tr>
<td>Joint Values</td>
<td>Specify the joint values of the pose.</td>
</tr>
</tbody>
</table>

### The Set Transition Times dialog box

The **Set Transition Times** dialog box is designed like a distance table in a road atlas. The default values are zero.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Pose</td>
<td>Specifies the start of the transition for the named pose.</td>
</tr>
<tr>
<td>To Pose</td>
<td>Specifies the end of the transition for the named pose.</td>
</tr>
</tbody>
</table>
9.15. Create Tool

Creating a tool

You can create a robot hold tool by using the Create Tool Wizard. The wizard allows you to easily create a tool from an existing part or by using a dummy part to represent a tool. To create a tool complete with tooldata, follow these steps:

1. Click Create Tool.
2. In the Tool Name box, enter a tool name and choose one of the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Existing</td>
<td>Select one of the existing parts from the list. The selected part will represent the tool graphics. The selected part must be a single part. Parts with attachments cannot be selected.</td>
</tr>
<tr>
<td>Use Dummy</td>
<td>A cone will be created to represent the tool.</td>
</tr>
</tbody>
</table>

3. Continue entering the Mass of the tool, the Center of Gravity and the Moment of Inertia Ix, Iy, Iz, if these values are known. If you do not know the correct values, the tool can still be used for programming motions, but this data must be corrected before running the program on real robots or measuring cycle times.

Tip! If the tool is built from materials with a similar density, you can find the center of gravity by clicking the tool model using the Center of gravity snap mode.

4. Click Next to go the next page of the wizard.
5. In the TCP Name box, enter a name for the Tool Center Point (TCP). The default name is the same as the name of the tool. If creating several TCPs for one tool, each TCP must have a unique name.
6. Enter the position of the TCP relative to the world coordinate system, which represents the tool mounting point, by any of the methods below:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read values from existing target or frame</td>
<td>Click in the Values from Target/Frame box, then select the frame either in the graphics window or the Paths&amp;Targets browser.</td>
</tr>
<tr>
<td>Enter position and orientation manually.</td>
<td>In the Position and Orientation boxes, type the values. NOTE: If Use Dummy Part is selected, the position value can not be 0,0,0. At least one coordinate has to be &gt; 0 in order for a cone to be created.</td>
</tr>
</tbody>
</table>

7. Click the arrow right button to transfer the values to the TCP(s): box.
   If the tool shall have several TCPs, repeat steps 5 to 7 for each TCP.
8. Click Done. The tool will be created and appear in the Objects browser and in the graphics window.

Creating tooldata for an existing geometry

To create tooldata for an existing geometry, follow these steps:

1. Make sure the robot in which to create the tooldata is selected.
2. Click Create Tool and select Use Existing and the imported tool from the list.

Continues on next page
3. Enter the requested data in the boxes in the Create Tool Wizard.

4. Attach the tool by dragging it to the robot.

What to do next

To make the tool ready to use, do one of the following:

- To make the robot hold the tool, attach the tool to the robot.
- In the graphics window, check the position and orientation of the TCP. If it is incorrect, modify the values in the tool frame part of the tooldata.
- To simplify future usage of the created tool, save it as a library. On the File menu, click Save As Library. Browse to the folder where you want to store the tool component, enter a name for the tool component and click Save.
10 The Simulation Tab

10.1. Overview

The Simulation tab

The Simulation tab contains the controls for setting up, configuring, controlling, monitoring and recording simulations.
10.2. Create Collision Set

Overview

A collision set contains two groups, *Objects A* and *Objects B*, in which you place the objects to detect any collisions between them. When any object in *Objects A* collides with any object in *Objects B*, the collision is displayed in the graphical view and logged in the output window. You can have several collision sets in the station, but each collision set can only contain two groups.

Creating a collision set

1. Click **Create Collision Set** to create a collision set in the **Layout** browser.
2. Expand the collision set and then drag one of the objects to the **ObjectsA** node to check for collisions.
   
   If you have several objects you want to check for collisions with objects in the **ObjectsB** node, for example, the tool and the robot, drag all of them to the **ObjectsA** node.
3. Drag the objects to the **ObjectsB** node to check for collisions.
   
   If you have several objects you want to check for collisions with objects in the **ObjectsA** node, for example, the work piece and the fixture, drag all of them to the **ObjectsB** node.
10.3. Simulation Setup

Prerequisites

To set up a simulation, the following conditions must be met:

- At least one path must have been created in the station.
- The paths to be simulated must have been synchronized to the virtual controller.

Setting up a simulation

1. Click Simulation Setup to bring up a dialog box.
2. In the Select Active Tasks box, select the tasks to be active in the simulation.
3. If a single controller has been selected, select either the Continuous or Single Cycle option.
4. From the Available Procedures list, transfer the procedures to be active in the simulation to the Main Sequence lists by selecting them and clicking the left arrow button between the lists. (This creates a procedure call in the main procedure).
5. To start the simulation from another procedure than the actual Main procedure (that is, to set a temporary Main procedure instead of overwriting the current one), click Select entry point and then specify the procedure to use as Main.
6. Click Apply to set the simulation. If you click OK, the simulation will be set and the dialog box will be closed.

NOTE!

If the simulation uses events or involves several different controllers, the virtual time mode Time Slice shall be used to make sure that the timing between the controllers is simulated correctly. For more information about virtual time modes, see Simulation Overview on page 117.

The Setup Simulation dialog box

<table>
<thead>
<tr>
<th>Select Active Tasks</th>
<th>Select the tasks to be active in the simulation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Run Mode options</td>
<td>Select the run mode when a single controller is selected. The options are Continuous and Single Cycle.</td>
</tr>
<tr>
<td>Main Sequence</td>
<td>Displays the procedures that will be run when you play the simulation.</td>
</tr>
<tr>
<td>&lt;-</td>
<td>Click the arrow left button to transfer the selected procedure to the Main Sequence box.</td>
</tr>
<tr>
<td>X</td>
<td>Click this button to remove the selected procedures or sequences from the Main sequence box.</td>
</tr>
<tr>
<td>arrow up</td>
<td>Click the arrow up button to move the sequence up in the list in the Main Sequence box or in the Available Procedures box.</td>
</tr>
<tr>
<td>arrow down</td>
<td>Click the arrow down button to move the sequence down in the list in the Main Sequence box or in the Available Procedures box.</td>
</tr>
<tr>
<td>Available Procedures</td>
<td>Displays all procedures available in the controller.</td>
</tr>
<tr>
<td>Select entry point</td>
<td>Click this button to set up several simulations at the same time. In the dialog box, enter a new name for the main procedure, and select the procedures to be included in the new simulation. Click OK.</td>
</tr>
</tbody>
</table>
10.4. Event Manager

Creating an event

1. Click Event Manager.
2. Click Add to open the New Event Wizard.
3. Complete the New Event wizard to create the event.

Event manager main parts

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Task pane. Here you create new events, or copy or delete existing events selected in the Event grid.</td>
</tr>
<tr>
<td>2</td>
<td>The Event grid. Displays all events in the station. Here you select events to edit, copy or delete.</td>
</tr>
<tr>
<td>3</td>
<td>The Trigger editor. Here you edit the properties of the events trigger. The upper part of the trigger editor is the same for all triggers, and the lower part adapts to the selected trigger type.</td>
</tr>
<tr>
<td>4</td>
<td>The Action editor. Here you edit the properties of the events action. The upper part of the action editor is the same for all actions, and the lower part adapts to the selected action type.</td>
</tr>
</tbody>
</table>

The task pane parts

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Starts the Create New Event Wizard.</td>
</tr>
</tbody>
</table>
The Simulation Tab

10.4. Event Manager

The event grid columns

In the event grid, each row is an event and the columns in the grid display their properties:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part</td>
<td>Description</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the event selected in the Event grid.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies the event selected in the Event grid.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refreshes the Event manager.</td>
</tr>
<tr>
<td>Activation</td>
<td>Displays whether the event is active or not.</td>
</tr>
<tr>
<td>On</td>
<td>The action is always carried out when the trigger event occurs.</td>
</tr>
<tr>
<td>Simulation</td>
<td>The action is only carried out if the trigger event occurs when running a simulation.</td>
</tr>
<tr>
<td>Off</td>
<td>The action is not carried out when the trigger event occurs.</td>
</tr>
<tr>
<td>Trigger Type</td>
<td>Displays the type of condition that triggers the action.</td>
</tr>
<tr>
<td>I/O Signal Trigger</td>
<td>Changes a digital I/O signal.</td>
</tr>
<tr>
<td>I/O Connection</td>
<td>Simulates the behavior of a Programmable Logic Controller (PLC).</td>
</tr>
<tr>
<td>Collision Trigger</td>
<td>Starts or ends a collision or near-miss between objects in a collision set.</td>
</tr>
<tr>
<td>Trigger System</td>
<td>When the trigger type is I/O Signal Trigger, this column displays to which system the signal used as trigger belongs.</td>
</tr>
<tr>
<td>Trigger Name</td>
<td>The name of the signal or collision set used as trigger.</td>
</tr>
<tr>
<td>Trigger Parameter</td>
<td>Displays the condition of the event under which triggering occurs.</td>
</tr>
<tr>
<td>0</td>
<td>The I/O signal used as trigger switches to false.</td>
</tr>
<tr>
<td>1</td>
<td>The I/O signal used as trigger switches to true.</td>
</tr>
<tr>
<td>Started</td>
<td>A collision starts within the collision set used as trigger.</td>
</tr>
<tr>
<td>Ended</td>
<td>A collision ends within the collision set used as trigger.</td>
</tr>
<tr>
<td>Near miss started</td>
<td>A near-miss starts within the collision set used as trigger.</td>
</tr>
<tr>
<td>Near miss ended</td>
<td>A near-miss ends within the collision set used as trigger.</td>
</tr>
</tbody>
</table>

Continued on next page
10.4. Event Manager

Continued

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Type</td>
<td>Displays the action type that occurs in conjunction with the trigger&lt;br&gt;&lt;br&gt;<strong>I/O Signal Action</strong> = Changes the value of a digital input or output signal.&lt;br&gt;&lt;br&gt;<strong>Attach Object</strong> = Attaches an object to another.&lt;br&gt;&lt;br&gt;<strong>Detach Object</strong> = Detaches an object from another.&lt;br&gt;&lt;br&gt;<strong>Turn On/Off Simulation Monitor</strong> = Toggles the simulation monitor of a specific mechanism.&lt;br&gt;&lt;br&gt;<strong>Turn On/Off Timer</strong> = Toggles the process timer.&lt;br&gt;&lt;br&gt;<strong>Move Mechanism to Pose</strong> = Moves the selected mechanism to a predefined pose and thereafter sends a station signal. Activates or deactivates the process timer.&lt;br&gt;&lt;br&gt;<strong>Move Graphical Object</strong> = Moves a graphical object to a new position and orientation.&lt;br&gt;&lt;br&gt;<strong>Show/Hide Graphical Object</strong> = Shows or hides the graphical object.&lt;br&gt;&lt;br&gt;<strong>Do Nothing</strong> = No action occurs.&lt;br&gt;&lt;br&gt;<strong>Multiple</strong> = The event triggers multiple actions, either all at once or one at a time, each time the trigger is activated. Each action can be viewed in the action editor.</td>
</tr>
<tr>
<td>Action System</td>
<td>When the action type is Change I/O, this column displays the system to which the signal to change belongs.&lt;br&gt;&lt;br&gt;A dash (-) signifies a virtual signal.</td>
</tr>
<tr>
<td>Action Name</td>
<td>Displays the name of the signal to change, when the action type is Change I/O.</td>
</tr>
<tr>
<td>Action Parameter</td>
<td>Displays the condition after the action has occurred.&lt;br&gt;&lt;br&gt;0 = The I/O signal will be set to false.&lt;br&gt;&lt;br&gt;1 = The I/O signal will be set to true.&lt;br&gt;&lt;br&gt;On = Turns the process timer on.&lt;br&gt;&lt;br&gt;Off = Turns the process timer off.&lt;br&gt;&lt;br&gt;<strong>Object1 -&gt; Object2</strong> = Displays the object to which another will be attached when the action type is Attach object.&lt;br&gt;&lt;br&gt;<strong>Object1 &lt;-&gt; Object2</strong> = Displays the object from which another will be detached when the action type is Detach object.&lt;br&gt;&lt;br&gt;<strong>Ended</strong> = A collision ends within the collision set used as trigger.&lt;br&gt;&lt;br&gt;<strong>Near miss started</strong> = A near-miss starts within the collision set used as trigger.&lt;br&gt;&lt;br&gt;<strong>Near miss ended</strong> = A near-miss ends within the collision set used as trigger.&lt;br&gt;&lt;br&gt;<strong>Multiple</strong> = Signifies multiple actions.</td>
</tr>
<tr>
<td>Time</td>
<td>Displays the time when the event trigger was executed.</td>
</tr>
</tbody>
</table>
The trigger editor parts

In the trigger editor you set the properties of the trigger. The upper part of the editor is common for all types of triggers, and the lower part adapts to the trigger type at hand.

Parts common to triggers

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation</td>
<td>Sets whether the event is active or not.</td>
</tr>
<tr>
<td></td>
<td><strong>On</strong> = The action is always carried out when the trigger event occurs.</td>
</tr>
<tr>
<td></td>
<td><strong>Simulation</strong> = The action is only carried out if the trigger event occurs when running a simulation.</td>
</tr>
<tr>
<td></td>
<td><strong>Off</strong> = The action is not carried out when the trigger event occurs.</td>
</tr>
<tr>
<td>Comment</td>
<td>Text box for comments and notes about the event.</td>
</tr>
</tbody>
</table>

Parts specific to I/O signal triggers

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Controller</td>
<td>Select the system to which the I/O to use as a trigger belongs.</td>
</tr>
<tr>
<td>Signals</td>
<td>Displays all signals that can be used as triggers.</td>
</tr>
<tr>
<td>Trigger Condition</td>
<td>For digital signals, sets whether the event shall trigger when the signals are set as true or false. For analog signals, which are only available for station signals, the event shall trigger under any of the following conditions: Greater than, Greater/Equal, Less than, Less/Equal, Equal to, Not equal to.</td>
</tr>
</tbody>
</table>

Parts specific to I/O connection triggers

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Opens a dialog box for adding an activator signal to the Activator Signals pane.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes a selected activator signal.</td>
</tr>
<tr>
<td>Add =&gt;</td>
<td>Opens a dialog box for adding an operator symbol to the Connections pane.</td>
</tr>
<tr>
<td>Remove</td>
<td>Removes a selected operator symbol.</td>
</tr>
<tr>
<td>Delay (s)</td>
<td>Specifies the delay in seconds.</td>
</tr>
</tbody>
</table>

Parts specific to Collision triggers

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collision Type</td>
<td>Set the kind of collision to use as trigger.</td>
</tr>
<tr>
<td></td>
<td><strong>Started</strong> = Triggers when a collision starts.</td>
</tr>
<tr>
<td></td>
<td><strong>Ended</strong> = Triggers when a collision ends.</td>
</tr>
<tr>
<td></td>
<td><strong>Near miss started</strong> = Triggers when a near-miss starts.</td>
</tr>
<tr>
<td></td>
<td><strong>Near miss ended</strong> = Triggers when a near-miss ends.</td>
</tr>
<tr>
<td>Collision set</td>
<td>Select the collision set to use as trigger.</td>
</tr>
</tbody>
</table>
The action editor parts

In the action editor you set the properties of the actions for the event. The upper part of the editor is common to all types of actions, and the lower part adjusts to the selected action.

Parts common to all actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Action</td>
<td>Adds a new action that occurs when the triggering condition fulfills. You can add several different actions that either are performed at once or one at a time each time the event triggers. The following types of actions are available: Change I/O = Changes the value of a digital input or output signal. Attach object = Attaches an object to another. Detach object = Detaches and object from another. Turn On/Off Timer = Activates or deactivates the process timer. Do Nothing = No action occurs (might be useful for manipulating sequences of actions).</td>
</tr>
<tr>
<td>Remove Action</td>
<td>Removes the action selected in the Added Actions list.</td>
</tr>
<tr>
<td>Cyclic</td>
<td>When selected, the actions are performed one at a time each time the trigger occurs. When all actions in the list have been performed, the event will restart with the first action in the list. When cleared, all actions are performed at once every time the trigger occurs.</td>
</tr>
<tr>
<td>Added Actions</td>
<td>Lists all actions of the event, in the order they will be executed.</td>
</tr>
<tr>
<td>Arrow</td>
<td>Rearranges the order in which the actions are executed.</td>
</tr>
</tbody>
</table>

Parts specific to I/O Actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Controller</td>
<td>Displays all systems of the station. Select the system to which the I/O to change belongs.</td>
</tr>
<tr>
<td>Signals</td>
<td>Displays all signals that can be set.</td>
</tr>
<tr>
<td>Action</td>
<td>Sets whether the event shall set the signals to true or false. If the action is connected to an I/O Connection, this group will not be available.</td>
</tr>
</tbody>
</table>

Parts specific to Attach actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach object</td>
<td>Select an object in the station to attach.</td>
</tr>
<tr>
<td>Attach to</td>
<td>Select the object in the station to attach to.</td>
</tr>
<tr>
<td>Update position</td>
<td>Update position = Moves the local origin of the attached object to the attachment point of the other object when making the attachment. For mechanisms, the attachment point is the TCP or the flange; for other objects, it is the local origin.</td>
</tr>
<tr>
<td>Keep position</td>
<td>Keep position = Keeps the current position of the object to attach when making the attachment.</td>
</tr>
<tr>
<td>Flange index</td>
<td>If the mechanism you attach the object to has several flanges (attachments points), select the one to use.</td>
</tr>
<tr>
<td>Offset Position</td>
<td>Optionally, specify an offset between the objects when making the attachment.</td>
</tr>
</tbody>
</table>

Continues on next page
### 10 The Simulation Tab

#### 10.4. Event Manager

### Table: Event Manager Actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset Orientation</td>
<td>Optionally, specify an offset between the objects when making the attachment</td>
</tr>
</tbody>
</table>

### Parts specific to Detach actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detach object</td>
<td>Select an object in the station to detach.</td>
</tr>
<tr>
<td>Detach from</td>
<td>Select the object in the station to detach from.</td>
</tr>
</tbody>
</table>

### Parts specific to Turn On/Off Simulation Monitor actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanism</td>
<td>Selects the mechanism.</td>
</tr>
<tr>
<td>Turn Simulation Monitor On/Off</td>
<td>Sets whether the action shall start or stop the simulation monitor function.</td>
</tr>
</tbody>
</table>

### Parts specific to Turn On/Off Timer actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn On/Off Timer</td>
<td>Sets whether the action shall start or stop the process timer.</td>
</tr>
</tbody>
</table>

### Parts specific to Move Mechanism to Pose actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanism</td>
<td>Selects the mechanism.</td>
</tr>
<tr>
<td>Pose</td>
<td>Selects between SyncPose and HomePose.</td>
</tr>
<tr>
<td>Station signal to set when Pose reached</td>
<td>Lists the station signals that are sent after the mechanism reaches its pose.</td>
</tr>
<tr>
<td>Add Digital</td>
<td>Click this button to add a digital signal to the grid.</td>
</tr>
<tr>
<td>Remove</td>
<td>Click this button to remove a digital signal from the grid.</td>
</tr>
</tbody>
</table>

### Parts specific to Move Graphical Object actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical Object to Move</td>
<td>Select a graphical object in the station to move.</td>
</tr>
<tr>
<td>New Position</td>
<td>Sets the new position of the object.</td>
</tr>
<tr>
<td>New Orientation</td>
<td>Sets the new orientation of the object.</td>
</tr>
</tbody>
</table>

### Parts specific to Show/Hide Graphical Object actions

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical Object</td>
<td>Select a graphical object in the station.</td>
</tr>
<tr>
<td>Show/Hide</td>
<td>Sets whether the object is shown or hidden.</td>
</tr>
</tbody>
</table>

### Parts specific for a Call VSTA Macro action

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Macros</td>
<td>Displays all available macros in the station.</td>
</tr>
</tbody>
</table>
10 The Simulation Tab

10.5. Activate Mechanical Units

10.5. Activate Mechanical Units

To activate or deactivate mechanical units manually

1. Click Activate Mechanical Units to bring up a dialog box.

2. In the Activate Mechanical Units dialog box, select the check boxes for the mechanical units to set as active. When activating a mechanical unit that shares a common drive unit, the other mechanical unit sharing that drive unit will be deactivated automatically.
10.6. Simulation Control

Running a simulation

1. Optionally, click **Enable Process Timer**.
2. In the **Simulation Control** group, click the **Play** button.
   The output window will now display messages from the virtual controller.
3. To pause the simulation, click the **Pause** button.
4. To stop the simulation, click the **Stop** button.
5. To reset the simulation to its initial state, click the **Reset** button.
10.7. I/O Simulator

Setting I/O signals using the I/O Simulator

1. Click I/O Simulator. This opens the I/O simulator.
2. If the station contains several systems, select the appropriate one in the Select System list.
3. In the Filter list and I/O Range list, make selections that display the signals to set.
   Depending on the filter used, you might also set a filter specification.
4. To change the value of a digital I/O signal, click it.
   To change the value of an analog signal, type the new value in the value box.

The I/O Simulator window

With RobotStudio’s I/O simulator you view and manually set existing signals, groups and cross-connections during program execution, thus making it possible to simulate or manipulate the signals.

The I/O simulator displays the signals for one system at a time in groups of 16 signals. For handling large sets of signals, you can filter which signals to display and also create custom lists with favorite signals for quick access.

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Select System. Select the system whose signals you want to view.</td>
</tr>
<tr>
<td>2.</td>
<td>Filter type. Select the type of filter to use.</td>
</tr>
<tr>
<td>3.</td>
<td>Filter Specification. Select the filter for limiting the signal display. For example, if Board is set as filter type, then you select the board whose signals you want to view.</td>
</tr>
</tbody>
</table>
The Simulation Tab

10.7. I/O Simulator

Continued

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td>Inputs. Displays all input signals that pass the applied filter. If more than 16 signals pass, only 16 signals at a time are displayed. Then use the I/O range list to select the signals to view.</td>
</tr>
<tr>
<td>5.</td>
<td>Outputs. Displays all output signals that pass the applied filter. If more than 16 signals pass, only 16 signals at a time are displayed. Then use the I/O range list to select the signals to view.</td>
</tr>
<tr>
<td>6.</td>
<td>Edit Lists. Click this button to create or edit lists of favorite signals.</td>
</tr>
<tr>
<td>7.</td>
<td>I/O Range. When more than 16 signals pass the filter, use this list to select the range of signals to display.</td>
</tr>
</tbody>
</table>

Types of signal filters

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board</td>
<td>Displays all signals on a specific board. To select a board, use the Filter Specification list.</td>
</tr>
<tr>
<td>Group</td>
<td>Displays all signals that belong to a specific group. To select a group, use the Filter Specification list.</td>
</tr>
<tr>
<td>User List</td>
<td>Displays all signals in a favorite list. To select a list, use the Filter Specification list.</td>
</tr>
<tr>
<td>Digital Inputs</td>
<td>Displays all digital input signals of the system.</td>
</tr>
<tr>
<td>Digital Outputs</td>
<td>Displays all digital output signals of the system.</td>
</tr>
<tr>
<td>Analog Inputs</td>
<td>Displays all analog input signals of the system.</td>
</tr>
<tr>
<td>Analog Outputs</td>
<td>Displays all analog output signals of the system.</td>
</tr>
</tbody>
</table>

Signal icons

<table>
<thead>
<tr>
<th>Signal Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Signal Icon 1" /></td>
<td>Digital signal with value 1.</td>
</tr>
<tr>
<td><img src="image2" alt="Signal Icon 2" /></td>
<td>Digital signal with value 0.</td>
</tr>
<tr>
<td><img src="image3" alt="Signal Icon 3" /></td>
<td>The cross in the upper right corner indicates that the signals are a cross-connection.</td>
</tr>
<tr>
<td><img src="image4" alt="Signal Icon 4" /></td>
<td>The -1 in the upper right corner indicates that the signal is inverted.</td>
</tr>
<tr>
<td><img src="image5" alt="Signal Icon 5" /></td>
<td>Value box for groups or analog signals.</td>
</tr>
</tbody>
</table>
## 10.8. Monitor

### The TCP Trace tab

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable TCP Trace</td>
<td>Select this check box to activate tracing of the TCP path for the selected robot.</td>
</tr>
<tr>
<td>Trace length</td>
<td>Specify the maximum length of the trace in millimeters.</td>
</tr>
<tr>
<td>Trace Color</td>
<td>Displays the color of the trace when no alerts are activated. To change the color of the trace, click the colored box.</td>
</tr>
<tr>
<td>Alert color</td>
<td>Displays the color of the trace when any of the alerts defined on the Alerts tab exceeds a threshold value. To change the color of the trace, click the colored box.</td>
</tr>
<tr>
<td>Clear Trace</td>
<td>Click this button to remove the current trace from the graphics window.</td>
</tr>
</tbody>
</table>

### The Alerts tab

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Simulation Alerts</td>
<td>Select this check box to activate simulation alerts for the selected robot.</td>
</tr>
<tr>
<td>Log Alerts to Output Window</td>
<td>Select this check box to see a warning message when a threshold value is exceeded. If TCP trace is not enabled, this is the only display of the alert.</td>
</tr>
<tr>
<td>TCP Speed (m/s)</td>
<td>Specify the threshold value for TCP speed alerts.</td>
</tr>
<tr>
<td>TCP Acceleration (m/s2)</td>
<td>Specify the threshold value for TCP acceleration alerts.</td>
</tr>
<tr>
<td>Wrist Singularity (deg)</td>
<td>Specify how close joint five can be to zero rotation before alerting.</td>
</tr>
<tr>
<td>Joint Limits (deg)</td>
<td>Specify how close each joint can be to its limits before alerting.</td>
</tr>
</tbody>
</table>
10.9. Record Movie

Prerequisites

For optimal results, first configure the options, see Options: General: Screen Recorder on page 179.

Recording the screen

1. In the Record Movie group, click Record application to capture the entire application window, or Record graphics to capture just the graphics window.
2. When you are done, click Stop Recording. A dialog box appears in which you may choose to save the recording or discard it.
3. Click View Recording to playback the latest capture.

Recording the simulation

1. In the Record Movie group, click Record Simulation to record the next simulation to a video clip.
2. When you are done, click Stop Recording.
   The simulation is saved in a default location which is displayed in the output window.
3. Click View Recording to playback the recording.
   The recording of simulation starts when you click Play in the Simulation tab.

NOTE!

Record Simulation gives better output quality than Record application or Record graphics.
10 The Simulation Tab

10.10.1. Conveyor Tracking

10.10 Conveyor Tracking Mechanism

10.10.1. Conveyor Tracking

Overview

Conveyor tracking is the function where the robot follows a workobject mounted on a moving conveyor.

This section describes how to create a conveyor, add and remove objects to and from the conveyor, create targets during tracking, and simulate conveyor.

For more information, see the Application manual - Conveyor tracking.

Conveyor tracking mechanism

This procedure describes the workflow for making a conveyor tracking system work in RobotStudio.

1. Create a conveyor mechanism. See Create Conveyor mechanism on page 247.
2. Setup the conveyor. See Setting up a conveyor on page 189 and Encoder Unit on page 296.
   
   NOTE! To setup a conveyor tracking station with two robots working on the same conveyor. See Setting up a conveyor tracking station with two robots working on the same conveyor on page 56.

3. Jog the conveyor as well as the robot and teach some targets. See Mechanism Joint Jog on page 340.
4. Simulate the conveyor. See Conveyor Simulation on page 271.
5. Remove objects from conveyor. See Removing objects from conveyor on page 190.
10.10.2. Conveyor Simulation

Running a conveyor simulation

1. Create Action Instructions. See Action Instruction on page 215.
   
   **NOTE!** Create the following five action instructions along with Move instructions:
   
   ConfI\Off, ActUnit CNV1, WaitWObj Workobject_1, DropWObj Workobject_1 and DeactUnit CNV1.
   
   The following program is an example showing how the sequence of instructions appears:
   
   ```
   ConfI\Off;
   MoveJ p0, vmax, fine, tool1;
   ActUnit CNV1;
   WaitWObj wobjcnvl;
   MoveL p10, v1000, z1, tool\Obj:wobjcnvl;
   MoveL p20, v1000, z1, tool\Obj:wobjcnvl;
   MoveL p30, v500, z20, tool\Obj:wobjcnvl;
   MoveL p40, v500, fine, tool1;
   DropWObj wobjcnvl;
   MoveL p0, v500, fine;
   DeactUnit CNV1;
   ```
   
   **NOTE!** If an error occurs while executing the program, the controller reaches the Guard state. In this state, RobotStudio cannot execute the program during the next simulation. To recover from this state, open the Control Panel and switch to Manual Mode and then to Auto Mode.
   
   For more information, see Application manual - Conveyor Tracking.
   
2. Synchronize to VC. See Synchronize to VC on page 275.

3. Set up the Simulation. See Simulation Setup on page 257.

4. Click Simulation.

   The Conveyor Simulation dialog appears.

5. In the Conveyor Speed box, set the speed during simulation.

   **NOTE!** To move the conveyor in the backward direction, select the Reverse check box.

6. Click Apply.

7. Click Play to run the simulation.

   **NOTE!** The conveyor speed and direction can be changed while running the simulation.

   **NOTE!**

   To jump the conveyor back to the start position, click Reset. This button remains enabled as long as the station has at least one conveyor.
10 The Simulation Tab

10.10.2. Conveyor Simulation
11 The Offline Tab

11.1. Overview

The Offline tab

The Offline tab contains the controls for synchronization, configuration and tasks assigned to the VC.
11.2. Synchronize to Station

Synchronizing to the station

1. Click **Synchronize to Station** to bring up a dialog box.
2. Select the paths to be synchronized to the station from the list.
3. Click **OK**.
11.3. Synchronize to VC

Synchronizing to the virtual controller

1. In the **Offline** browser and the Active task list, select the task you want to synchronize.
2. Click **Synchronize to VC** to bring up a dialog box.
3. Select the check boxes for the elements to synchronize.

<table>
<thead>
<tr>
<th>Element</th>
<th>In this column, all elements that can be synchronized, such as data, paths and targets, are displayed. To include the element in the synchronization, select the check box in front of the element name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module</td>
<td>In this column, the module in the RAPID program to which the element will be synchronized is displayed. To change the module for an element, right-click the current one and select, or type a new in the list.</td>
</tr>
<tr>
<td>Storage class</td>
<td>In this column, the storage class of each element is displayed. For detailed information about storage classes for the different elements, see the <em>RAPID Reference Manual</em>. To change the storage class for an element, right-click the current one and select, or type a new in the list. The column is visible only in advanced mode.</td>
</tr>
<tr>
<td>Inline</td>
<td>In this column, you can specify whether the target for the move instruction shall be declared in the data section (<em>False</em>) or written directly in the instruction (<em>True</em>). The column is visible only in advanced mode.</td>
</tr>
<tr>
<td>Select path targets</td>
<td>Click this button to select in the Elements column all targets in the path. This button is available only in advanced mode after a path is selected.</td>
</tr>
<tr>
<td>Simple/Advanced</td>
<td>Click this button to toggle between the simple and advanced display modes. In simple mode, the tooldata, workobjects, storage classes and inline columns will not be displayed.</td>
</tr>
</tbody>
</table>

4. Optionally, click **Advanced** and review or change the settings for the synchronization.
5. Click **OK**.
11 The Offline Tab

11.4 Backup

11.4.1. Backing up a system

Overview

When backing up a system you copy all the data needed to restore the system to its current state:

- Information about software and options installed on the system.
- System's home directory and all its content.
- All robot programs and modules in the system.
- All configuration and calibration data of the system.

Prerequisites

To backup a system you must:

- Write access to the controller
- Logged on to the controller with appropriate grants. For more information, see User Authorization on page 137.

Creating a Backup

To create a backup, follow these steps:

1. In the Offline browser, select the system you want to backup.
2. Click Backup and select Create Backup.
   The Create Backup dialog box appears.
3. Enter a new backup name and specify a location for the backup, or keep the default ones.
4. Click Backup.
   The progress of the backup appears in the Output window.

Result

When the back up is complete you will have a folder with the name of the back up in the specified location. This folder contains a set of subfolders which all together comprises the backup:

<table>
<thead>
<tr>
<th>Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backinfo</td>
<td>Contains information necessary for re-creating the system's software and options from the mediapool.</td>
</tr>
<tr>
<td>Home</td>
<td>Contains a copy of the system's home directory content.</td>
</tr>
<tr>
<td>Rapid</td>
<td>Contains one subfolder for each task in the system's program memory. Each of these task folders contains separate folders for program modules and system modules.</td>
</tr>
<tr>
<td>Syspar</td>
<td>Contains the system's configuration files.</td>
</tr>
</tbody>
</table>

CAUTION!

If the content of the Backup folder is changed, it is not possible to restore the system from backup.
11.4.2. Restoring a system from backup

Overview

When restoring a system from backup, the current system gets the same content as when the backup was performed. Restoring a system replaces the following contents in the current system with the content from the backup:

- All RAPID programs and modules in the system.
- All configuration and calibration data of the system.

**NOTE!** The system's home directory and all its content are copied from the backup to the current system.

Prerequisites

To restore a system you must:

- Write access to the controller.
- Logged on to the controller with appropriate grants. For more information, see *User Authorization on page 137*.

Restoring a system

To restore a system, follow these steps:

1. In the **Offline** browser, select the system you want to restore.
2. Click **Backup** and select **Restore Backup**.
   The Restore from Backup dialog box appears.
3. In the **Restore from Backup** dialog box, select which backup to use for restoring the system.
4. Click **Restore**.
   The progress of the restore appears in the Output window.
5. When the restore is complete, restart the controller to load the restored system. See *Restarting a VC on page 66*.

**NOTE!**
Before proceeding, make sure that the system from the backup is compatible with the controller you are restoring.

System ID and Template ID

If the system from the backup does not originate from the controller you are restoring, you will get a message about the mismatch.

The following table describes the possible cases:

<table>
<thead>
<tr>
<th>Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System ID and Template ID are identical</td>
<td>When both System ID and Template ID are identical between current system and backup system, there is no difference between the systems.</td>
</tr>
</tbody>
</table>
### 11.4.2. Restoring a system from backup

*Continued*

**Case** | **Description**
--- | ---
Mismatch in System ID | If there is a mismatch in System ID there could be a minor difference between current system and backup system, as for example different languages. RobotStudio warns regarding mismatch. **NOTE!** Before you restore, make sure that the backup system is compatible with the current system.

Mismatch in both System ID and Template ID | There can be mismatch in both System ID and Template ID between current system and backup system, as for example different robot types. RobotStudio warns that this is not a recommended action. **NOTE!** You must have a correct backup to the system that you restore.
11.5. Events

The Event Log tab

1. Category
   This list contains the different event categories. The default category *Common* includes all categories.

2. Auto Update
   This checkbox is checked by default, meaning that new events will appear in the list when they occur. Clearing the checkbox disables this automatic update. Checking it again, however, will fetch and display the events missed while it was cleared.

3. Clear
   This button clears the current event record. This does not affect the event log of the controller, which can be retrieved again by clicking the Get button.

4. Get
   This button retrieves and displays all events currently stored in the controller.

5. Save
   This button saves the event records of the selected event categories to log files on the computer.

6. Log to file
   This checkbox enables all events currently shown by the Event Log to be saved to a log file on the computer. If it remains checked the log file will be updated with new events as they occur.

7. Event Information
   This box displays information about the event selected in the event list.

8. Event Record
   The event record is shown as a list of events of a selected category. The severity of the event is indicated by color: blue for information, yellow for warning and red for an error which needs to be corrected in order to proceed.
### 11.6. RAPID editor

#### Buttons on the RAPID editor toolbar

The table below describes the buttons on the **RAPID editor** toolbar:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Start" /></td>
<td><strong>Start</strong>&lt;br&gt;Starts the execution of all normal RAPID tasks in the system.</td>
</tr>
<tr>
<td><img src="image" alt="Stop" /></td>
<td><strong>Stop</strong>&lt;br&gt;Stops the execution of all normal RAPID tasks in the system.</td>
</tr>
<tr>
<td><img src="image" alt="Step over" /></td>
<td><strong>Step over</strong>&lt;br&gt;Starts and executes one statement in all normal tasks in the system.</td>
</tr>
<tr>
<td><img src="image" alt="Step in" /></td>
<td><strong>Step in</strong>&lt;br&gt;Starts and executes into a routine, while stopping at the beginning of the routine.</td>
</tr>
<tr>
<td><img src="image" alt="Step out" /></td>
<td><strong>Step out</strong>&lt;br&gt;Executes all remaining statements of the current routine, and stops after the call to the current routine.</td>
</tr>
<tr>
<td><img src="image" alt="Ignore breakpoints" /></td>
<td><strong>Ignore breakpoints</strong>&lt;br&gt; Ignores all breakpoints during simulation.</td>
</tr>
<tr>
<td><img src="image" alt="Toggle breakpoint" /></td>
<td><strong>Toggle breakpoint</strong>&lt;br&gt; Toggles a breakpoint at the cursor.</td>
</tr>
<tr>
<td><img src="image" alt="Apply changes" /></td>
<td><strong>Apply changes</strong>&lt;br&gt; Applies the changes to the module in the program editor to the system.</td>
</tr>
<tr>
<td><img src="image" alt="Check program" /></td>
<td><strong>Check program</strong>&lt;br&gt; Verifies the syntactic and semantic correctness of all modules in the system.</td>
</tr>
<tr>
<td><img src="image" alt="Print" /></td>
<td><strong>Print</strong>&lt;br&gt; Prints the contents of the program editor.</td>
</tr>
</tbody>
</table>
11 The Offline Tab

11.6. RAPID editor

About RAPID Editor

The RAPID editor enables you to view and edit programs loaded into the controller program memory. With the RAPID editor you edit the RAPID code of the program's modules. Each module you open will appear in a program editor window of its own, where you type in the code. For more information, see Using the RAPID editor on page 113.

RAPID Editor functions

RAPID editor has the following functions:

- Cut, copy, paste and drag and drop - supports clipboard handling of rich text.
- Undo and redo - supports undo and redo operations.
- Go To Line - navigates to a specific line within the Editor using a keyboard shortcut.
- Selection modes - possibility to select text by character, row and column.
- Line numbers - appears in the left margin of the editor.
- Keyboard shortcuts - see Keyboard Shortcuts on page 49.

Programming functions

The RAPID Editor has the following RAPID specific functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax coloring</td>
<td>Writes each part of the code with a specific color, based on its syntactic function in RAPID. Helps to quickly recognize words and misspellings. The automatic coloring of the text is based on the syntax of the RAPID language.</td>
</tr>
<tr>
<td>Parameter Info</td>
<td>Shows available parameters in a tool tip for the RAPID instructions you type. Helps to type in a procedure or function call in the editor by showing all the optional and required arguments that must be specified.</td>
</tr>
<tr>
<td>Pick list</td>
<td>Helps to navigate through a hierarchy of small code snippets which can be inserted at the insertion position.</td>
</tr>
<tr>
<td>Routine list</td>
<td>Is a combo box containing all routines in the RAPID module of the editor.</td>
</tr>
<tr>
<td>Complete word</td>
<td>Automatically completes the keyword by predicting the keyword or identifier that you want to type in without you actually typing.</td>
</tr>
</tbody>
</table>
11.7. Inputs / Outputs

The I/O system window

From the I/O system you can view and set input and output signals.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
<th>Logical State</th>
<th>Unit</th>
<th>Bus</th>
</tr>
</thead>
</table>

1. The Name column
   This column shows the name of the signal. The name is set by the I/O unit's configuration and cannot be changed from the I/O system.

2. The Type column
   This column shows which type of signal it is, by using any of the abbreviations described below. The signal type is set by the I/O unit's configuration and cannot be changed from the I/O system.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DI</td>
<td>Digital input signal</td>
</tr>
<tr>
<td>DO</td>
<td>Digital output signal</td>
</tr>
<tr>
<td>AI</td>
<td>Analog input signal</td>
</tr>
<tr>
<td>AO</td>
<td>Analog output signal</td>
</tr>
<tr>
<td>GI</td>
<td>Group of signals, working as one input signal</td>
</tr>
<tr>
<td>GO</td>
<td>Group of signals, working as one output signal</td>
</tr>
</tbody>
</table>

3. The Value column
   This column shows the value of the signal. The value can be changed by double-clicking the signal row.

4. The Logical State column
   This column shows whether the signal is simulated or not. When a signal is simulated, you specify a value that overrides the actual signal. Changing the logical state by turning the simulation on or off can be done from the I/O system.

5. The Unit column
   This column shows to which I/O unit the signal belongs. This is set by the I/O unit's configuration and cannot be changed from the I/O system.

6. The Bus column
   This column shows to which I/O bus the signal belongs. This is set by the I/O bus' configuration and cannot be changed from the I/O system.
11.8. System Builder

Functions of the System Builder

For procedures using the various functions of the System Builder, see System Builder Overview on page 139.
11 The Offline Tab

11.9. Rapid Tasks

Prerequisites

You must have write access to the controller to be able to open the Task Window. You must also be logged on to the controller as a user who has either of the grants Full access or Execute program.

Different states

The following table shows different states when actions are not possible in the Task Window:

<table>
<thead>
<tr>
<th>If...</th>
<th>RobotStudio gives a message that informs the user that...</th>
</tr>
</thead>
<tbody>
<tr>
<td>the user does not have the grant Execute program or Full access</td>
<td>the operation is not possible.</td>
</tr>
<tr>
<td>the user changes from manual mode to automatic mode, or vice versa, the user loses the write access and the motors are in off state</td>
<td>the Start operation is not possible.</td>
</tr>
</tbody>
</table>

The safety system of the controller

It is not possible to override the controller’s safety system, that is, you cannot stop a background task (Static and SemiStatic) that has the TrustLevel set to another value than NoSafety.

For detailed information about the different TrustLevel values, see Technical reference manual - system parameters.

The Task window

The Task window is divided in two parts, one left part with functionality (1) to operate the task(s) and one right part (2) with the task list and information about the tasks.

Continues on next page
If the prerequisites are met, you can operate the task, such as start and stop the task, move the program pointer to main and set the run mode. This part will be opened as default when you open the Task Window, but you can hide this part.

The following table describes the different buttons in the functionality part:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>The <strong>show/hide</strong> button. Shows/hides the functionality part.</td>
</tr>
</tbody>
</table>
| 4 | The **start** button. Starts the selected tasks.  
**Note!** You can start Normal tasks, but you can only start a Static or SemiStatic task if the **TrustLevel** is set to NoSafety. |
|   | **Danger!**  
When starting a task, the manipulator axes may move very quickly and sometimes in unexpected ways! Make sure no personnel is near the manipulator arm! |
| 5 | The **stop** button. Stops the selected tasks.  
**Note!** You can stop Normal tasks, but you can only stop a Static or SemiStatic task if the **TrustLevel** is set to NoSafety. |
| 6 | The **Move PP to Main** button. Moves the program pointer to Main. Applies to all selected tasks. |
| 7 | The **Task names** check box. Selected check box indicates that the task will be operated. |
| 8 | The **Select** buttons. Select All or None of the tasks. |
| 9 | The **Run Mode** buttons. Set run mode to Continuous or Single cycle.  
The **Run Mode** indicates the mode of the controller. |

| Task Name | The task name. |
| Task in Foreground | Which task that is in the foreground, for example main.  
Gives indirectly the priority. |
| Type | Normal/Static/SemiStatic:  
• Normal: The task reacts on START/STOP requests. The task is stopped when an emergency stop occurs. The TrustLevel for the Normal task corresponds to the TrustLevel NoSafety.  
• Static: At warm start of the controller, the task restarts at the current position.  
• SemiStatic: The task restarts from the beginning at all warm starts of the controller.  
Default value is SemiStatic.  
**Note!**  
If the type of task is Static or SemiStatic, the following is valid only for the Task Window: If the **TrustLevel** is set to NoSafety, the task can be stopped by the stop button on the Task Window. If the **TrustLevel** is set to SysFail, SysHalt or SysStop, the task **cannot** be stopped. |
| Mech. Unit | If the task has robot specific instructions, this specifies which robots they are valid for. |
| Run Mode | Continuous or single cycle. |
| Program Name | The name of the program in the specific task. |
| Module Name | The current module name. |
| Routine Name | The current routine name. |

Continued on next page
### 11 The Offline Tab

#### 11.9. Rapid Tasks

*Continued*

<table>
<thead>
<tr>
<th>State</th>
<th>Ready/Running/Stopped:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Ready: The program has no PP (program pointer). To get a PP to the program, click the <strong>Move PP to main</strong> button.</td>
</tr>
<tr>
<td></td>
<td>• Running: The program is running.</td>
</tr>
<tr>
<td></td>
<td>• Stopped: The program has stopped.</td>
</tr>
</tbody>
</table>

You can operate the task list as following:

- All columns can be adjusted in the standard windows way - drag or double-click between the column headers.
- If all the columns do not fit into the window, a horizontal scroll bar will occur.
- If all the tasks do not fit into the window, a vertical scroll bar will occur.
- You can sort the task information in all columns by clicking the column heading.
11.10. Restart

Restarting a VC

1. In the Layout browser, select the controller to restart.

2. Click Restart and then click either Warmstart to activate the changes and restart the VC or I-start to restart the VC with the current system and default settings.
11.11. Shutdown

Shutting down a VC

1. In the Offline browser, select the controller to shut down.
2. Click Shutdown and then click either Shutdown to close the connection to the VC or Terminate Virtual Controllers to end all VC processes.
11.12. New Module

Creating a module

1. Click **New Module** to bring up a dialog box.
2. Select the module type and permission and click **Create**.
11.13. Load Module

Loading a module

1. Click **Load Module** to bring up a dialog.
2. Browse to the location of the module to be loaded to your station and click **Open**.
11.14. Load Program

**Loading a program**

1. Click **Load Program** to bring up a dialog box.
2. Browse to the location of the program to be loaded to your station and click **Open**.
## 11.15. Add signals

### Parts

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Name Expression</td>
<td>Defines the name for one or more signals.</td>
</tr>
<tr>
<td>Signal Type</td>
<td>Defines the type of signal.</td>
</tr>
<tr>
<td>Assigned Unit</td>
<td>Defines the I/O unit to which the signal belongs.</td>
</tr>
<tr>
<td>Identification Label</td>
<td>Optionally, offers filtering and sorting based on this category.</td>
</tr>
<tr>
<td>Unit Mapping Expression</td>
<td>Defines the bits in the I/O memory map of the assigned unit to which the signal is mapped.</td>
</tr>
<tr>
<td>Signal Category</td>
<td>Optionally, offer filtering and sorting based on this category.</td>
</tr>
<tr>
<td>Invent Physical Value</td>
<td>Applies an inversion between the physical value of the signal and its logical representation in the system.</td>
</tr>
</tbody>
</table>

**NOTE!**

You must have write access to the controller to be able to open the add signal window.
11.16. Set Task Frames

Modifying Task frame

1. In the Offline browser, click Set Task Frame.
   The Modify Task Frames dialog box appears.
2. Set the reference to World, UCS, or Local.
3. Edit the position and orientation of task frames in the Task Frames coordinate box.
   This updates the position of all frames relative to the system.
4. Click Apply.
11.17. System Configuration

Overview

The system configuration window contains functions for making and viewing advanced system configurations, such as changing controller and baseframe positions, calibrating and setting up external axes.

The left side of the system configuration window contains a hierarchical tree with which you browse to different aspects of the system. The right side contains a property sheet for the aspect selected in the tree. Below are short descriptions of the property sheets for each aspect node of the system configuration.

CAUTION!

Changing the system configurations may result in corrupted systems or unexpected robot behaviors. Be sure to understand the effects of the changes before proceeding.

The System node

The system node contains a box with information about the system and a button for loading new parameters (configuration files) to the system.

The task node

The task node has no property page.

The mechanism folder node

The property page of this node contains controls for mapping and setting axis and joints. It is from this page you set up external axes.

The mechanism library node

The property page of this node contains controls for changing the baseframe of the robot or mechanism. Here, too, you specify whether the baseframe is moved by another mechanism (coordinated motion), like a track external axis.

Updating the baseframe position

1. Move the mechanical unit (robot or external axis) to its new location using the ordinary tools for moving and placing objects.
2. In the Offline browser, select the controller for the mechanical unit.
3. Click System Configuration to bring up a dialog box.
   NOTE! The System configuration dialog box can also be opened by:
   • In the Paths&Targets browser, right-click a station.
   • Select Configuration and click System Configuration.
4. Select the node for the mechanical unit in the hierarchical tree. The baseframe property sheet for the robot is now displayed.

Continues on next page
5. Select the baseframe position values to use after restarting the robot.

<table>
<thead>
<tr>
<th>Select</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller values</td>
<td>Reset all changes to the baseframe made since the last time the</td>
</tr>
<tr>
<td></td>
<td>system was started.</td>
</tr>
<tr>
<td>Stored station values</td>
<td>Reset all changes made to the baseframe since the last time the</td>
</tr>
<tr>
<td></td>
<td>station was saved.</td>
</tr>
<tr>
<td></td>
<td>Optionally, you can enter new values in the baseframe</td>
</tr>
<tr>
<td></td>
<td>coordinate boxes (relative to the controller world coordinate</td>
</tr>
<tr>
<td></td>
<td>system).</td>
</tr>
<tr>
<td>Use current station values</td>
<td>Read and use the current location of the baseframe.</td>
</tr>
<tr>
<td></td>
<td>Optionally, you can enter new values in the baseframe</td>
</tr>
<tr>
<td></td>
<td>coordinate boxes (relative to the controller world coordinate</td>
</tr>
<tr>
<td></td>
<td>system).</td>
</tr>
</tbody>
</table>

6. Click **OK**.
11 The Offline Tab

11.18. Encoder Unit

11.18. Encoder Unit

Configuring a Conveyor Encoder Unit

1. Click Encoder Unit.
   
   The Configure Conveyor Encoder Unit dialog box appears.

   **NOTE!** The Configure Conveyor Encoder Unit dialog box can also be opened by:

   • In the Paths&Targets browser, right-click a station.
   • Select Configuration and click Encoder Unit.

2. Select CNV1 from the Mechanical Unit list.

3. In the Parameters box, enter the values for Maximum Distance, Minimum Distance, Queue Tracking Distance and Start Window Width.

   **NOTE!**

   If any of the parameter values are changed, the controller must be restarted.

4. Click OK.

5. Click Yes to restart the controller.
11.19. Configuration editor

**Configuration editor**

1. The **Type name** list displays all available configuration types for the selected topic.
   The list of types is static. This means you cannot add, delete or rename types.

2. The **Instance** list displays all system parameters of the type selected in the **Type name** list. Each row in the list is an instance of the system parameter type. The columns show each specific parameter and its value for each instance of the parameter type.

**Instance editor**

1. The **Parameter** list displays the parameters and their value for the open instance.
2. The **Value** box displays the type of parameter and the value of the parameter.
3. The **Restriction** box displays the restrictions for the parameter. The restrictions have to be fulfilled to update the controller database.
4. The **Limits** box displays the limits for the parameter.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xx</td>
<td>Editable parameter</td>
</tr>
</tbody>
</table>
11 The Offline Tab

11.19. Configuration editor

Continued

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Icon" /></td>
<td>Parameter that is not applicable for this instance and therefore not editable.</td>
</tr>
<tr>
<td><img src="image2" alt="Icon" /></td>
<td>Parameter that is read-only.</td>
</tr>
</tbody>
</table>
11.20. Load Parameters

Prerequisite

You must have write access to the controller.

Loading parameters

1. In the Offline/Online browser, select the configuration node.

2. Click Load Parameters to bring up a dialog box.

3. In the dialog box, select how you want to combine the parameters in the configuration file to load with the existing parameters:

<table>
<thead>
<tr>
<th>If you want to...</th>
<th>then select...</th>
</tr>
</thead>
<tbody>
<tr>
<td>replace the entire configuration of the topic with the one in the configuration file.</td>
<td>Delete existing parameters before loading</td>
</tr>
<tr>
<td>add new parameters from the configuration file to the topic, without modifying the existing ones.</td>
<td>Load parameters if no duplicates</td>
</tr>
<tr>
<td>add new parameters from the configuration file to the topic and update the existing ones with values from the configuration file. Parameters that only exist in the controller and not in the configuration file will not be changed at all.</td>
<td>Load parameters and replace duplicates</td>
</tr>
</tbody>
</table>

4. Click Open and browse to the configuration file to load. Then click Open again.

5. In the information box, click OK to confirm that you want to load the parameters from the configuration file.

6. When the loading of the configuration file is finished, close the Select mode dialog box.
   If a restart of the controller is necessary for the new parameters to take affect, you will be notified of this.
11 The Offline Tab

11.21. Save System Parameters

Overview

The system parameters of a configuration topic can be saved to a configuration file, stored on the PC or any of its network drives.

The configuration files can then be loaded into a controller. They are thereby useful as backups, or for transferring configurations from one controller to another.

File-naming conventions

The configuration files should be named with a name that relates to their corresponding topics. When saving configuration files, the correct name for each file will be suggested by default.

Saving one configuration file

1. In the Offline/Online browser, expand the Configuration node and select the topic to save to a file.
2. Click Save System Parameters.
3. In the Save As dialog box, browse for the folder to save the file in.
4. Click Save.

Saving several configuration files

1. In the Robot View Explorer, select the Configuration node.
2. Click Save System Parameters.
3. In the Save System Parameters dialog box, select the topics to save to files. Then click Save.
4. In the Browse for Folder dialog box, browse for the folder to save the files in. Then click OK.

The selected topics will now be saved as configuration files with default names in the specified folder.
12 The Online Tab

12.1. Overview

The Online tab

The Online tab contains controls for managing the real controller.
12.2. Add Controller

Adding a controller

1. Click Add Controller to bring up a dialog box in which all available controllers are listed.
2. If the controller is not found in the list, type its IP address in the IP Address box, and then click Refresh.
3. Select the controller in the list and click OK.
12.3. Request Write Access

Overview

You can access the controllers in your Robot View with either read only access or Write access. You need Write access for editing programs and configurations or in any other way change data on the controller.

Prerequisites for Write access

You can get Write access to any controller as long as the prerequisites are fulfilled.

<table>
<thead>
<tr>
<th>When the Controller is in Mode:</th>
<th>This has to be fulfilled:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>The Write access must not be taken by any other user.</td>
</tr>
<tr>
<td>Manual</td>
<td>The remote Write access must be granted on the FlexPendant. For safety reasons, a FlexPendant user can also recall this remote Write access in manual mode.</td>
</tr>
</tbody>
</table>

If the prerequisites are not fulfilled you will be denied, or lose, the Write access. This means that if you have Write access in auto mode and the controller is switched over to manual mode you will lose the Write access without any warning. This is because the FlexPendant unit by default has the Write access in manual mode, for safety reasons. The same will happen if the remote Write access in manual mode is recalled from the FlexPendant unit.

Result

The Controller Status window will be updated when the request for Write access is granted. If the Write access is denied, a message is displayed.
12.4. Release Write Access

Overview

You can access the controllers in your Robot View with either read only access or write access. Since several users can be logged on to one controller but only one can have write access, you can release the write access when you do not need it anymore.

Result

The Controller Status window will be updated when your access right has changed from read/write to read only.
12.5. Import Options

Importing system options

1. Click **Import System Options** to bring up a dialog box.
2. In the **Option source** box, enter the path to the folder where the options to import are located. You can also click the Browse button and browse to the folder.
3. In the **Media pool destination** box, enter the path to the media pool you want to store the options in. You can also click the Browse button and browse to the media pool folder.
4. Select the options to import and click **Import**.
   
   To select several options at once, do one of the following:

<table>
<thead>
<tr>
<th>To select</th>
<th>then hold down</th>
</tr>
</thead>
<tbody>
<tr>
<td>several adjacent options</td>
<td>the SHIFT key and select the first and the last option.</td>
</tr>
<tr>
<td>several non-adjacent options</td>
<td>the CTRL key and select each option.</td>
</tr>
</tbody>
</table>

5. Click **OK**.

Removing system options

1. Click **Import System Options** to bring up a dialog box.
2. In the **Media pool destination** list, enter the path to the media pool from which you want to delete the options. You can also click the Browse button and browse to the media pool folder.
3. Select the options to delete and click **Remove**.
   
   To select several options at once, do one of the following:

<table>
<thead>
<tr>
<th>To select</th>
<th>then hold down</th>
</tr>
</thead>
<tbody>
<tr>
<td>several adjacent options</td>
<td>the SHIFT key and select the first and the last option.</td>
</tr>
<tr>
<td>several non-adjacent options</td>
<td>the CTRL key and select each option.</td>
</tr>
</tbody>
</table>

4. Click **OK**.
12.6. User Accounts

Overview

All the procedures below require the following steps to be taken before managing the details:

1. In the Online browser, select the controller to which you want to manage a user or group.
2. From the Online tab, request write access to the controller.
3. On the Online tab, click User Accounts.

Users tab

With the Users tab you set which users will be able to log on to the controller and which groups the users shall belong to.

Users tab Parts

1. The Add button. Opens a dialog box for adding new users.
2. The Edit button. Opens a dialog box for changing the user’s log on name and password.
3. The Delete button. Deletes the selected user account from the controller.
4. The Users on this Controller list. Shows the user accounts defined on this controller. The list has two columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>The name of the user account</td>
</tr>
<tr>
<td>Status</td>
<td>Shows if the account is activated or deactivated. When deactivated, it is not possible to log on using that account.</td>
</tr>
</tbody>
</table>

Continues on next page
5. The **Activated/Deactivated** item box. Changes the status of the user account.

6. The **User’s groups/User’s grants** list.

   The **User’s groups** list shows which group(s) the user is a member of. For changing the membership of a group, select or clear the checkbox in front of the group name.

   The **User’s grants** list shows the available grants for the selected User’s group(s). When selecting a grant from the User’s grants list, a description of the selected grant appears.

---

**Adding a user**

1. On the **Users** tab, click **Add** to bring up a dialog box.

2. In the **User Name** box, enter the user name. Use only characters from the ISO Latin -1 character set and no more than 16 characters.

3. In the **Password** box, enter the user’s password. The password you type in will not be visible. Use only characters from the ISO Latin -1 character set and no more than 16 characters.

4. In the **Retype Password** box, enter the user's password again.

5. Click **OK** to add the new user and close the dialog box.

6. Click **OK**.

---

**Deleting a user**

1. On the **Users** tab, select the user to delete from the **Users on this controller** list and click **Delete**.

2. To the question *Are you sure you want to remove this user?*, answer **Yes**.

3. Click **OK**.

---

Continues on next page
12.6. User Accounts

Setting up group membership

1. On the Users tab, select the user from the Users on this controller list.
2. In the User's groups list, select the groups the user shall be a member of.
3. Click OK.

Changing a name or password

1. On the Users tab, select the user to edit from the Users on this controller list and click Edit user.
   This opens the Edit dialog box.
2. To change the user name, enter the new name in the User Name box. Use only characters from the ISO Latin -1 character set and no more then 16 characters.
3. To change the password, enter the new password in the Password box, then retype the password in the Retype Password box. Use only characters from the ISO Latin -1 character set and no more then 16 characters.
4. Click OK to save the changes to the user and close the dialog box.
5. Click OK.

Activating or deactivating a user

1. On the Users tab, select the user from the Users on this controller list and click the status text (Activated or Deactivated). An item box appears and you can change the status.
   The user's new state is now displayed in the status column of the Users on this controller list.
2. Click OK.

Continues on next page
Exporting a user list

1. On the Users tab, select the user from the Users for this Controller list and click Export.
   This opens a Save as dialog box, in which you specify the name and location for the file
   with the user list.

Importing a user list

1. On the Users tab, select the user from the Users for this Controller list and click Import.
   This opens an Open file dialog box, in which you browse to the file with the list to import.
   When you have selected the file, the ImportOptionsForm dialog appears.

<table>
<thead>
<tr>
<th>Select ...</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete existing users and groups before importing</td>
<td>Earlier groups and users will be deleted.</td>
</tr>
<tr>
<td>Advanced options</td>
<td>A new dialog appears. Import users but don’t replace duplicates means that you will not replace the existing users.</td>
</tr>
<tr>
<td></td>
<td>Import users and replace duplicates means that you will replace the existing users.</td>
</tr>
<tr>
<td></td>
<td>Import groups but don’t replace duplicates means that you will not replace the existing groups.</td>
</tr>
<tr>
<td></td>
<td>Import groups and replace duplicates means that you will replace the existing groups.</td>
</tr>
</tbody>
</table>

Adding a group

1. On the Groups tab, click Add.
   This opens the Add new group dialog box.

2. In the Group Name box, enter the name of the group. Use only characters from the ISO Latin -1 character set and no more than 16 characters.

3. Click OK to add the new group and close the dialog box.

4. Click OK.

Renaming a group

1. On the Groups tab, select the group to rename from the Groups on this controller list and click Rename.
   This opens the Rename Group dialog box.

2. In the Group Name box, enter the name of the group. Use only characters from the ISO Latin -1 character set and no more than 16 characters.

3. Click OK to rename the group and close the dialog box.
4. You will now be asked if you want the users who belong to this group to continue belonging to a group with the old name as well as the new one.

<table>
<thead>
<tr>
<th>Click</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Let the users of the group be members to groups with both the old and the new name. However, the old group will no longer be defined in the controller's UAS since it is replaced by the new group. This option might be useful if you plan to recreate the old group, or copy the user's settings to another controller who has the old group defined.</td>
</tr>
<tr>
<td>No</td>
<td>Delete the user's memberships to the old group. This is to just replace the old group name with the new one.</td>
</tr>
<tr>
<td>Cancel</td>
<td>To cancel the change and keep the old group name, with its user's memberships.</td>
</tr>
</tbody>
</table>

5. Click OK.

Deleting a group

1. On the Groups tab, select the group to delete from the Groups on this controller list and click Delete.

2. You will now be asked if you want the users who belong to this group to continue belonging to it though it is not valid.

<table>
<thead>
<tr>
<th>Click</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Let the users of the group remain members to it even if it is no longer defined in the controller's UAS. This option might be useful if you plan to recreate the group, or copy the user's settings to another controller who has the group defined.</td>
</tr>
<tr>
<td>No</td>
<td>Delete the user's memberships to the group.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Cancel the change and keep the group, with its user's memberships.</td>
</tr>
</tbody>
</table>

3. Click OK.

Giving grants to a group

1. On the Groups tab, select the group from the Groups on this controller list.

2. In the Controller grants/Application grants list, select the grants to give to the group.

3. Click OK.
12.7. UAS Grant Viewer

The Grants Overview dialog box

Parts

You can sort the information in all three columns by clicking the column heading.

1. The **Grant** column shows the name of the grant key.

2. The **Valid** column shows from which RobotWare versions the grant is valid.

   Systems with different RobotWare versions may use different grants. When you install a system on a controller, the grants for that system are added to the controller’s grant list. Grants are never removed from the grant list but may be declared invalid if they are not applicable for a system. All this is handled automatically during installation of systems.

3. The **Groups owning this Grant** column shows which groups on this controller that have the grant.

Examples of common actions to perform

<table>
<thead>
<tr>
<th>Action</th>
<th>Necessary grants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename the controller.</td>
<td>Modify controller properties</td>
</tr>
<tr>
<td><em>(A restart of the controller is necessary.)</em></td>
<td>Remote warm start</td>
</tr>
<tr>
<td>Change system parameters and load configuration files.</td>
<td>Modify configuration</td>
</tr>
<tr>
<td></td>
<td>Remote warm start</td>
</tr>
<tr>
<td>Install a new system.</td>
<td>Administration of installed system</td>
</tr>
<tr>
<td>Perform a backup.</td>
<td>Backup and save</td>
</tr>
<tr>
<td><em>(A restart of the controller is necessary.)</em></td>
<td>Remote warm start</td>
</tr>
</tbody>
</table>

Continues on next page
## 12 The Online Tab

### 12.7. UAS Grant Viewer

Continued

<table>
<thead>
<tr>
<th>Action</th>
<th>Necessary grants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore a backup.</td>
<td>Restore a backup</td>
</tr>
<tr>
<td>(A restart of the controller is necessary.)</td>
<td>Remote warm start</td>
</tr>
<tr>
<td>Load/delete modules.</td>
<td>Load program</td>
</tr>
<tr>
<td>Create new modules.</td>
<td>Load program</td>
</tr>
<tr>
<td>Edit code in RAPID modules.</td>
<td>Edit RAPID code</td>
</tr>
<tr>
<td>Save modules and programs to disk.</td>
<td>Backup and save</td>
</tr>
<tr>
<td>Start program execution from Task Window.</td>
<td>Execute program</td>
</tr>
<tr>
<td>Create a new I/O signal, that is, add a new instance of the type Signal. (A restart of the controller is necessary.)</td>
<td>Modify configuration</td>
</tr>
<tr>
<td>Set the value of an I/O signal.</td>
<td>Remote warm start</td>
</tr>
<tr>
<td>Access to controller disks from File Management Window.</td>
<td>I/O write access</td>
</tr>
</tbody>
</table>

### Controller grants

<table>
<thead>
<tr>
<th>Action</th>
<th>Necessary grants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full access</td>
<td>This grant includes all controller grants, also new grants added in future RobotWare versions. The grant does not include any application grants.</td>
</tr>
<tr>
<td>Manage UAS settings</td>
<td>Gives access to read and write the UAS configuration, that is to read, add, remove and modify UAS users and groups.</td>
</tr>
<tr>
<td>Execute program</td>
<td>Gives access to perform the following:</td>
</tr>
<tr>
<td></td>
<td>• Start/step program (stop is always allowed)</td>
</tr>
<tr>
<td></td>
<td>• Move PP to Main</td>
</tr>
<tr>
<td></td>
<td>• Execute service routines</td>
</tr>
<tr>
<td>Perform ModPos and HotEdit</td>
<td>Gives access to perform the following:</td>
</tr>
<tr>
<td></td>
<td>• Modify or teach positions in RAPID code (ModPos)</td>
</tr>
<tr>
<td></td>
<td>• During execution modify positions in RAPID code as single points or as a path (HotEdit)</td>
</tr>
<tr>
<td></td>
<td>• Restore ModPos/HotEdit positions to original</td>
</tr>
<tr>
<td></td>
<td>• Modify current value of any RAPID variable</td>
</tr>
<tr>
<td>Modify current value</td>
<td>Gives access to modify current value of any RAPID variable. This grant is a subset of the grant Perform ModPos and HotEdit. (Available from RobotWare 5.07.03.)</td>
</tr>
<tr>
<td>I/O write access</td>
<td>Gives access to perform the following:</td>
</tr>
<tr>
<td></td>
<td>• Set I/O signal value</td>
</tr>
<tr>
<td></td>
<td>• Set signal as simulated and remove simulation</td>
</tr>
<tr>
<td></td>
<td>• Set I/O unit and bus as enabled/disabled</td>
</tr>
<tr>
<td>Backup and save</td>
<td>Gives access to perform a backup and to save modules, programs and configuration files.</td>
</tr>
<tr>
<td>Restore a backup</td>
<td>Gives access to restore backup and perform B-start.</td>
</tr>
<tr>
<td>Modify configuration</td>
<td>Gives access to modify the configuration database, that is to load configuration files, change system parameter values and add/delete instances.</td>
</tr>
<tr>
<td>Load program</td>
<td>Gives access to load/delete modules and programs.</td>
</tr>
<tr>
<td>Remote warm start</td>
<td>Gives access to perform warm start and shutdown (S-start) from a remote location. No grant is required to perform warm start via a local device, as for example the FlexPendant.</td>
</tr>
</tbody>
</table>
### Edit RAPID code
Gives access to perform the following:
- Modify code in existing RAPID modules
- Frame calibration (tool, workobj)
- Commit ModPos/HotEdit positions to current values
- Rename program

### Program debug
Gives access to perform the following:
- Move PP to routine
- Move PP to cursor
- HoldToRun
- Activate/deactivate RAPID tasks
- Request write access from the FlexPendant
- Acknowledge Auto mode without restoring simulated I/O signals or deactivated tasks
- Enable/disable non-motion execution

### Decrease production speed
Gives access to decrease speed from 100% in Auto mode. This grant is not required if speed is already below 100%, or controller is in Manual mode.

### Calibration
Gives access to perform the following:
- Fine calibrate mechanical unit
- Calibrate base frame
- Update/clear SMB data

**Note!** Frame calibration (tool, workobj) requires the grant *Edit RAPID code*. Manual offset of mechanical unit calibration data and loading new calibration data from file require the grant *Modify configuration*.

### Administration of installed systems
Gives access to perform the following:
- Install new system
- P-start
- I-start
- X-start
- C-start
- Select System
- Install system from device

This grant gives full FTP access, that is, the grant gives the same rights as *Read access to controller disks and Write access to controller disks*.

### Read access to controller disks
Gives external read access to controller disks. This grant is only valid for explicit disk access, for example with an FTP client or the File Manager of RoboStudio. It is possible, for example, to load a program from /hd0a without this grant.

### Write access to controller disks
Gives external write access to controller disks. This grant is only valid for explicit disk access, for example with an FTP client or the File Manager of RoboStudio. It is possible, for example, to save a program to the controller disk or perform a backup without this grant.

### Modify controller properties
Gives access to set controller name, controller ID and system clock.

### Delete log
Gives access to delete messages in the controller Event Log.

### Revolution counter update
Gives access to update the revolution counter.

*Continues on next page*
12 The Online Tab

12.7. UAS Grant Viewer

Continued

### Application grants

<table>
<thead>
<tr>
<th>Access to the ABB menu on FlexPendant</th>
<th>Value <strong>true</strong> gives access to the ABB menu on the FlexPendant. This is the default value if a user does not have the grant. Value <strong>false</strong> means that the user cannot access the ABB menu when the controller is in Auto mode. The grant has no effect in Manual mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log off FlexPendant user when switching to Auto mode</td>
<td>A user having this grant is automatically logged off from the FlexPendant when switching from Manual mode to Auto mode.</td>
</tr>
</tbody>
</table>
12.8. FlexPendant Viewer

Overview
FlexPendant Viewer is an add-in to RobotStudio that retrieves and displays a screenshot from the FlexPendant. The screenshot is generated automatically at the moment of the request.

Prerequisites
To install FlexPendant Viewer, the system must have the option PC interface installed.
The controller you want to retrieve screen shots from must be added to your robot view.
A FlexPendant must be connected to the controller. If no FlexPendant is currently connected (option Hot plug is installed and the jumper plug is used) then no screen shot can be retrieved.

Using FlexPendant Viewer
1. Make sure you are connected to the controller.
2. Click FlexPendant Viewer.
   A screen shot will be displayed in the workspace.
3. To reload the screen shot, click Reload in the workspace.
4. To set an automatic reload period for the screen shot, click on the menu Tools, point to FlexPendant Viewer and click Configure.
   Set the desired reload period and select the check-box Activated. Then click OK.

Results on the controller
The screenshot will automatically be saved as a file on the controller. When a new request is sent, a new screenshot is generated and saved, overwriting the previous file.
No message will be displayed on the FlexPendant.
12.9. Device Browser

Updating the tree view

1. To update the tree view, press F5.

Displaying the properties of a device

1. Select a device in the tree view to view its properties or values in the right-hand pane.

Displaying a trend

1. Select a device in the tree view and then double-click any property in the right-hand pane that has a numerical value to bring up a trend view.

   The trend view collects data at a rate of one sample per second.

Hiding, stopping, starting or clearing a trend

1. Right-click anywhere in the trend view and then click the appropriate command.
13 The Add-Ins Tab

13.1. Overview

The Add-Ins tab

The Add-Ins tab contains the control for PowerPacs and the VSTA.
13.2. Visual Studio Tools for Applications

Building an add-in

2. Create a new project or open an existing one. Compose or edit as required.
3. To build the add-in, from the Build menu, click Build.
   The add-in will be loaded in RobotStudio and appear as a user add-in in the Add-In browser.
4. Save the project to your hard disk if you want the add-in to remain with the user.
   Otherwise, open a station, right-click in the Add-In browser and then click Add to Station.
14 The Context Menus
14 The Context Menus

14.1. Add to Path

14.1. Add to Path

Creating a move instruction based on an existing target

1. Select the target for which to create the move instruction.
2. From the Home menu, in the Path Programming group, select the type of move instruction to create.
3. Click Add to Path.

The move instruction will appear under the path node as a reference to the original target.
14.2. Align Frame Orientation

The Align Frame Orientation dialog box

<table>
<thead>
<tr>
<th>Reference</th>
<th>Specify the frame or target for which you want to align the selected objects here.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Align Axis</td>
<td>The axis you specify here will be aligned as on the reference target/frame for all selected objects.</td>
</tr>
<tr>
<td>Lock Axis</td>
<td>The axis you specify here will not be changed on the selected objects by the align function, but will keep its orientation.</td>
</tr>
</tbody>
</table>
14.3. Align Target Orientation

Aligning target orientation

1. Select the targets whose orientation you wish to change.
2. Click Align Target Orientation to bring up a dialog box.
3. In the Reference box, specify the target whose orientation you want to use as reference, by first click in the box and then selecting the target either from the graphics view or the Layout browser.
4. In the Align Axis box, select the axis whose orientation you want to copy from the reference target to the selected ones.
5. In the Lock Axis box, select the axis to rotate the target around. The orientation of this axis will not be changed on the targets. For example, if the Z axis of all targets are orientated normally to the surface of the work piece and you want to keep it this way, you should lock the Z axis.
6. Click Apply.

TIP!

You can change the Align and Lock axis and click Apply again to reorientate the targets until you deselect them.
14.4. Attach to

Attaching an object

1. In the Layout browser, right-click the child object, click Attach to and click the parent object in the list.

Attaching an object by drag and drop

1. In the Layout browser, drag the child object to the parent object.
2. In the displayed message, click the corresponding button:

<table>
<thead>
<tr>
<th>To</th>
<th>Click</th>
</tr>
</thead>
<tbody>
<tr>
<td>attach the child object and move it to the attachment point</td>
<td>Yes</td>
</tr>
<tr>
<td>attach the child object and keep its position</td>
<td>No</td>
</tr>
<tr>
<td>not perform the attachment</td>
<td>Cancel</td>
</tr>
</tbody>
</table>
14.5. Auto Configuration

Setting a robot axis configuration for all targets in a path

1. In the Paths & Targets browser, select a path and then click AutoConfiguration. The robot now steps through each target in the path and sets the configurations.

2. If the first target in the path has no configuration assigned, the one closest to the robot’s current configuration will be used. If the first target has a configuration assigned, the assigned one will be used. Thus, the result of the autoconfiguration may vary depending on the first target’s configuration.
14.6. Check Reachability

Checking the reachability

1. In the Paths&Targets browser, select the task that contains the targets or move instructions to check.

2. Click Reachability to open a dialog box. Make sure that the Check reachability for targets, paths, move instructions check box is selected.
   Optionally, select or clear the Check reachability for targets, paths, move instructions check box for showing and hiding the reachability display.

3. In the Layout browser, select the objects whose reachability you want to check. If paths are selected, the reachability will be checked for all move instructions in each path.

4. In the Reachability dialog box, click Add.
   The frames for the selected objects will now change color based on their reachability status.

<table>
<thead>
<tr>
<th>Color</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>The object can be reached.</td>
</tr>
<tr>
<td>Yellow</td>
<td>The object can be reached at its current position, but not at its current orientation</td>
</tr>
<tr>
<td>Red</td>
<td>The object cannot be reached at its current position.</td>
</tr>
</tbody>
</table>
14.7. Configurations

Manually setting a robot axis configuration for single targets

1. In the Paths & Targets browser, select a target and then click Configurations to bring up a dialog box.

2. If more than one configuration solution exist, examine them by clicking them, one at a time.
   The position of the robot with the selected configuration will be displayed in the graphics window, and the joint values for the configuration will be displayed in the joint values list below the configurations list.
   In most cases, selecting a configuration similar to the previous one is the best choice.

3. Select the configuration to use and click Apply.
14.8. Control Panel

The Control Panel dialog box

<table>
<thead>
<tr>
<th><strong>Operation Mode</strong></th>
<th>This group contains the three operational modes of the controller represented by option buttons.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Auto</strong></td>
<td>This option corresponds to the Auto mode on the FlexPendant. Moving between the <strong>Auto</strong> and <strong>Manual Full Speed</strong> options must proceed via the <strong>Manual</strong> option.</td>
</tr>
<tr>
<td><strong>Manual</strong></td>
<td>This option corresponds to the Manual mode on the FlexPendant.</td>
</tr>
<tr>
<td><strong>Manual Full Speed</strong></td>
<td>This option corresponds to the Manual 100% mode on the FlexPendant. Moving between the <strong>Auto</strong> and <strong>Manual Full Speed</strong> options must proceed via the <strong>Manual</strong> option.</td>
</tr>
<tr>
<td><strong>Push Motors</strong></td>
<td>Click this button to turn on the motors.</td>
</tr>
<tr>
<td><strong>Enable Device</strong></td>
<td>In a manual mode, click this button to simulate holding the enabling device to turn on the motors.</td>
</tr>
<tr>
<td><strong>Release Device</strong></td>
<td>In a manual mode, click this button to turn off the motors.</td>
</tr>
<tr>
<td><strong>Reset Emergency Stop</strong></td>
<td>If the controller enters the emergency stop state, click this button to reset the state.</td>
</tr>
</tbody>
</table>
14.9. Convert Frame to Workobject

Converting a frame to a workobject

1. In the Layout browser, select a frame.
3. Optionally, rename or edit the workobject in any way.
14.10. Convert to Move Circular

Prerequisites

At least two targets, the via-point target and the end point target, must have been created. A path containing at least the via-point target and the end point target, in correct order, must have been created.

Converting to Move Circular

1. In the Paths&Targets browser, expand the path node that contains the move instruction to be converted.

2. Select the move instruction that contains the via-point of the circular motion together with the succeeding move instruction, which will serve as the end point. You can select several instructions by holding down the SHIFT key while clicking the instructions.

3. Click Convert to Move Circular. The two selected move instructions will be converted to a circular move instruction, which includes the via-point and the end point.

TIP!

To convert two move instructions to a circular motion, you can also select and right-click both move instructions at once and then click Convert to Circular.
14.11. Copy / Apply Orientation

**Copying and applying an orientation**

1. In the browser, select the object or target from which to copy the orientation.
2. On the **Modify** menu, click **Copy Orientation**.
3. In the browser, select the object or target to which to apply the orientation.
4. On the **Modify** menu, click **Apply Orientation**. This can be performed on several targets or a group of selected targets.
14.12. Detach

Detaching an object

1. In the Layout browser, right-click the attached object (child) and then click Detach. The child will be detached from the parent and return to its position before the attachment.
14 The Context Menus

14.13. Execute Move Instruction

14.13. Execute Move Instruction

Prerequisites

The move instruction must exist.
A virtual controller must be running for the robot with the move instruction.

Executing a move instruction

1. In the Paths & Targets browser, browse to the motion instruction to execute through the Controller, Tasks and Paths nodes.

2. Click Execute move instruction. The TCP of the active robot will move from the current location to the motion instruction according to the programmed motion properties. If the target for the motion instruction does not have a stored configuration, the robot will use the configuration nearest the current one.

**Overview**

With the graphic appearance dialog box you set the graphic properties for an individual object. The settings made here override the generic settings made in the options dialog box. The dialog box contains one view group where you select the part of the object to affect, and three tabs with settings.

**Graphic Appearance: The View group**

<table>
<thead>
<tr>
<th>Selection</th>
<th>Set the parts of the object whose appearance you wish to change. When <strong>Body</strong> or <strong>Surface/Curve</strong> is selected, you select the object to work with in the preview display. <strong>NOTE!</strong> To change the appearance of a part: 1. On the <strong>Modify</strong> menu, click <strong>Graphic Appearance</strong>. A color box with the following options appears - Metals, Light colors, medium colors, Dark colors. 2. Select any of the options to change the appearance.</th>
</tr>
</thead>
</table>

**Graphic Appearance: The Rendering tab**

<table>
<thead>
<tr>
<th>Backface culling</th>
<th>Select this check box to view only the front sides (the direction of the positive normal) of the faces in the model. This improves the graphics performance and indicates the directions of the faces in the model. Clear this check box to also display the back sides of the faces in the model. This ensures that the model is displayed correctly even if the faces are in the wrong direction. If a model with faces in the wrong direction is used for programming, you might get unpredicted results.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twosided lightning</td>
<td>Select this check box to light the object from two sides. Clear this check box to light the object from one side.</td>
</tr>
<tr>
<td>Flat shade</td>
<td>Select this check box to change the appearance of shades.</td>
</tr>
<tr>
<td>Flip normals</td>
<td>Click this button to change the direction of all faces on the model.</td>
</tr>
<tr>
<td>Line width</td>
<td>Specify the width of lines in the object here.</td>
</tr>
<tr>
<td>The detail level options</td>
<td>Select the detail level of the model here. Only the levels selected in the <strong>Options</strong> dialog box when the object was created or imported are available.</td>
</tr>
</tbody>
</table>

**Graphic Appearance: The Colors tab**

| Simple Color | Click this color box to select another color for the object. |
| Transparency | Use this slider to control the transparency of the object. |
| The color boxes | Set the color of the object for different light situations here. |
| Shininess | Specify the reflectiveness of the object here. |
### 14 The Context Menus


Continued

---

#### Graphic Appearance: The Textures tab

| Effect                  | Specifies the type of graphic effect to be used on the selected part.  
|-------------------------|---------------------------------------------------------------|
|                         | • Basic : Simple texturing/environment mapping or just a solid color.  
|                         | • Anisotropic: Brushed metal effect simulating an anisotropic light model  
|                         | • Bump Mapping: Texture that specifies the bumpiness of the surface  
|                         | • Parallex Mapping: Improved version of bump mapping  
|                         | **NOTE!** Anisotropic, Bump Mapping and Parallex Mapping require DirectX9-class hardware to be correctly displayed in the 3D window.  
| Base texture            | Specifies the basic structure of the selected part. It is a standard 24-bit image displayed on a 3D surface.  
| Environment map         | Provides a highly reflective appearance to the surface.  
| Normal/Bump Map         | Specifies a texture that defines the bumpiness of the surface. **NOTE!** This option is enabled only during Bump mapping and Parallex mapping effect.  
| Blend with material     | Select this check box to blend the texture with the colors and properties specified on the Material tab.  
| Swap u/v                | Click this button to swap the horizontal and vertical directions of the texture.  
| Modify                  | Select along which directions the commands listed below shall be applied. u is the horizontal axis of the texture. v is the vertical axis of the texture.  
| Normalize               | Click this button to set the ratio between the dimensions of the object and the texture to 1.  
| Flip                    | Click this button to invert the coordinates along the selected axes. This is the same as mirroring around the other axis.  
| Stretch                 | Click this button to stretch the texture along the selected axes.  
| Shrink                  | Click this button to shrink the texture along the selected axes.  
| Shift <                 | Click this button to move the texture along the selected axes.  
| Shift >                 | Click this button to move the texture along the selected axes.  
---
14.15. Interpolate Path

Reorienting targets in a path by interpolation

1. In the Layout browser or the graphics window, select the path with the targets to reorient.
2. Click Interpolate Path. to bring up a dialog box.
3. With the Interpolate type options, select whether to use Linear or Absolute interpolation.
   Linear interpolation distributes the difference in orientation evenly, based on the targets positions along the length of the path. By contrast, absolute interpolation distributes the difference in orientation evenly, based on the targets’ sequence in the path.
4. If using the Select Start/End option, select the start and end targets for the interpolation in the Start target and End target boxes, respectively.
5. Optionally, with the Lock Axis options, select an axis to lock.
6. Click Apply.
14.16. Invert

**Inverting the direction of a face**

1. Right-click in the *Modeling* browser, point to *Filter* and make sure that both *Show Bodies* and *Show Faces* are selected.

2. In the *Modeling* browser, expand the node for the object and browse down to and select the face which direction you want to invert.

3. In the *Modeling* browser, expand the node for the object and browse down to and select the face whose direction you want to invert.

4. Click *Invert*. If the option backface culling is activated, the face will now shift from visible to not visible, or the other way around, depending on from which direction you view the face. If backface culling is deactivated, there will be no visible indication that the direction of the face has been inverted.
14.17. Jump to Target

Jumping to a target

1. In the Paths&Targets browser, browse to the target to jump to through the Controller, Tasks and WorkObjects nodes.

2. Click Jump to target.
   If the target has a valid configuration for the robot axes stored, the active TCP of the robot will immediately be positioned at the target. If no valid configuration is stored, the Select Robot configuration dialog box is displayed.

3. In the Select Robot Configuration dialog box, select a suitable configuration solution and click Apply. The selected configuration is now stored with the target.

NOTE!
You can deactivate the configuration check when jumping to targets. The robot will then use the configuration solution closest to the current one when reaching the target. For more information, see RobotStudio Options on page 178.
14 The Context Menus

14.18. Linked Geometry

14.18. Linked Geometry

Overview
The Linked Geometry feature allows you to load geometry from a shared repository. If the source file is updated, then the station will be updated with a single click.

Adding Link
You can add a link to a geometry in two ways:
1. In the Home tab, click Import Geometry to open a dialog box.
   Select the option Link to Geometry.
2. In the Layout browser, right-click an existing part in the station and select Add Link.
   A dialog box opens where you can select the CAD file to be linked.

Editing Link
To edit an existing link:
1. In the Layout browser, right-click an existing part in the station.
2. Select the option Link to Geometry and click Edit Link.

Deleting Link
To delete an existing link:
1. In the Layout browser, right-click an existing part in the station.
2. Select the option Link to Geometry and click Delete Link.

Updating Linked Geometry
To update a linked geometry:
1. In the Layout browser, right-click an existing part in the station, component group or the station.
2. Select the option Link to Geometry and click Update Linked Geometry.
   The update result is displayed in the output window.
   NOTE! When you select a component group or a station, all linked geometries within the group or station is updated.
   NOTE! If the timestamp on the file is newer than the timestamp stored in the station, all corresponding parts will be updated from the source location.
14.19. The Library Group

Modifying a library component

1. In the Layout browser, select the library you wish to modify.
2. Click Disconnect Library.
3. Select the library and then make any modifications to it.
4. Select the modified library, and then click Save As Library.
14.20. Mechanism Joint Jog

Jogging the joints of a robot

1. In the Layout browser, select the robot.
2. Click Mechanism Joint Jog to bring up a dialog box.
3. Each row in the Jog Joints dialog box represents a joint of the robot. Jog the joints either by clicking and dragging the bar at each row, or by using the arrows to the right of each row.

Set the length of each step in the Step box.

The Mechanism Joint Jog dialog box

<table>
<thead>
<tr>
<th>Joint</th>
<th>Move the joints of the objects by dragging the slider on the row corresponding to each joint. Alternatively, click the buttons to the right of the row, or type a value.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cfg</td>
<td>The current configuration value.</td>
</tr>
<tr>
<td>TCP</td>
<td>The current position of the TCP.</td>
</tr>
<tr>
<td>Step</td>
<td>Specify the length of the joint movements for each click of the buttons to the right of each joint row.</td>
</tr>
<tr>
<td>External Axis</td>
<td>If the robot uses external axes, you can select an axis to jog from this list. The external axes must belong to the same task as the object you are jogging for occurring in this list. If no external axes are present in the same task, this list is not available.</td>
</tr>
<tr>
<td>Lock TCP</td>
<td>Select this check box to reposition the robot according to the jogging of the external axis.</td>
</tr>
<tr>
<td></td>
<td>For track-external axes the robot will reposition so that the TCP is locked relative to the world coordinate system.</td>
</tr>
<tr>
<td></td>
<td>For positioner-external axes the robot will reposition so that the position of the robot’s TCP is locked relative to the attachment point of the positioner. The robot will move with the positioner the same way as when using multi-robot jog.</td>
</tr>
<tr>
<td></td>
<td>If no external axes are present in the same task, this check box is not available.</td>
</tr>
<tr>
<td>External axes joint</td>
<td>Move the joint of the external axes by dragging the slider on the row corresponding to each joint. Alternatively, click the buttons to the right of the row, or type a value.</td>
</tr>
<tr>
<td></td>
<td>If no external axes are present in the same task, this check box is not available.</td>
</tr>
</tbody>
</table>

Jogging a conveyor

1. Create Empty Path. See Empty Path on page 201.
2. In the Layout browser, select the conveyor.
3. Right-click Conveyor Mechanism and select Mechanical Joint Jog.

The Joint Jog dialog box appears.
4. Jog the conveyor by moving the slider and click Teach Instruction.

A move instruction is added to the path.

Continues on next page
NOTE!
When you jog the conveyor mechanism, objects on the conveyor are also moved.

- If you jog the conveyor mechanism beyond the maximum distance, the workobject will be dropped.
- If you jog the conveyor mechanism beyond the zero position, the workobject that belongs to the first part is attached to the conveyor attachment point.

If the workobject is dropped in Teach Mode, you can jog the conveyor backwards to connect it again.
14.21. Mechanism Linear Jog

Jogging the TCP of a robot by using the Linear Jog dialog box

1. In the Layout browser, select the robot.
2. Click Mechanism Linear Jog to bring up a dialog box.
3. Each row in the Linear Jog dialog box represents a direction or rotation for the TCP. Jog the TCP along the preferred direction or rotation, either by clicking and dragging the bar at each row, or by using the arrows to the right of each row.
4. From the Reference list, you can select the coordinate system that you want to jog the robot relative to.
5. In the Step box, specify the step movement per deg/rad.
14.22. Mirror Path

The Mirror Path dialog box

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate</td>
<td>Select this option to keep the existing path when mirroring.</td>
</tr>
<tr>
<td>Replace</td>
<td>Select this option to remove the existing path after mirroring.</td>
</tr>
<tr>
<td>X-Y, X-Z and Y-Z</td>
<td>Select the plane to mirror the path around with these options. The plane is defined by the selected axes and position of the reference coordinate system selected below.</td>
</tr>
<tr>
<td>Reference</td>
<td>Select the frame or coordinate system to define the mirror plane in.</td>
</tr>
<tr>
<td></td>
<td>To use another frame than any of the predefined ones, select Select Frame from the list and specify the frame in the box below.</td>
</tr>
<tr>
<td>Select Frame</td>
<td>If Select Frame is used as Reference frame, specify the frame to use here by first clicking in the box and then selecting the frame from the graphics window or the Layout browser.</td>
</tr>
<tr>
<td>Flip axis X/YZ</td>
<td>Select one of these options to mirror the orientation of the targets. When any of these are selected, the robot will approach the targets in a mirrored way.</td>
</tr>
<tr>
<td></td>
<td>The axis you select will change the most for achieving the mirrored orientation, while the other one will be kept as near to its current direction as possible.</td>
</tr>
<tr>
<td></td>
<td>The axis that is set to the robot’s approach vector cannot be selected.</td>
</tr>
<tr>
<td>Keep orientation</td>
<td>Select this option to keep the orientation of the targets. When selected, the robot will go to the mirrored position, but approach the target from the same direction as for the original target.</td>
</tr>
<tr>
<td>Mirror Robot Configura-tion</td>
<td>Select this option to also mirror the robot axis configuration for the targets. Selecting this option will mirror the robot's motions completely. For using this option, the following conditions must be met:</td>
</tr>
<tr>
<td></td>
<td>• The Reference frame must be set to Baseframe.</td>
</tr>
<tr>
<td></td>
<td>• The Mirror plane must be set to X-Z.</td>
</tr>
<tr>
<td></td>
<td>• The tool of each move instruction must have its TCP in the X-Z plane of tool0.</td>
</tr>
<tr>
<td></td>
<td>• All targets in the path must have robot axis configuration set.</td>
</tr>
<tr>
<td></td>
<td>• The virtual controller must be running.</td>
</tr>
<tr>
<td>More / Less</td>
<td>Click this button to show or hide the commands for naming and location of generated targets and paths.</td>
</tr>
<tr>
<td>New path name</td>
<td>Specify the name of the path that will be generated by the mirroring here.</td>
</tr>
<tr>
<td>Target prefix</td>
<td>Specify a prefix for the targets that will be generated by the mirroring here.</td>
</tr>
<tr>
<td>Receiving robot</td>
<td>Specify the robot task in which the new targets and path shall be created.</td>
</tr>
<tr>
<td>Receiving work object</td>
<td>Specify the work object in which the new targets shall be created.</td>
</tr>
</tbody>
</table>
14 The Context Menus

14.23. Modify Curve

14.23. Modify Curve

Extending a curve with a straight line in the direction of the curve’s tangent

1. Click **Modify curve** to bring up a dialog box.
2. Select **Extend** as modify method and then click the curve segment, close to the appropriate vertex.
   Note that when you rest the pointer over the curve the endpoint closest to the pointer is highlighted. This is the endpoint that will be selected for extension when you click the curve.
3. In the **Distance from endpoint** box, enter the length of the extension. In the graphics window a yellow line displays a preview of the extension.
4. Click **Apply**.

Joining curves

1. Click **Modify curve** to bring up a dialog box.
2. Select **Join** as modify method and then click the curves to join in the graphics window. The curves may be either intersecting or adjacent to be joined.
   The **Selected curves** list displays the curves that will be joined. To remove a curve from the list, select the list entry and press the DEL key.
3. In the **Tolerance** list, enter a value in millimeters. Adjacent curves whose end points lie within the tolerance will be valid for the operation.
4. Click **Apply**.

Projecting curves on a surface

1. Click **Modify curve** to bring up a dialog box.
2. Select **Project** as modify method and then click the curves to project in the graphics window.
   Note that when you rest the pointer over the curve, the projection direction is displayed. The project direction is always the negative Z direction of the User Coordinate System. To change the projection direction, create a new frame with the desired orientation and set it as user coordinate system.
   The **Selected curves** list displays the curves that will be projected. To remove a curve from the list, select the list entry and press the DEL key.
3. Click in the **Target bodies** list and then click the bodies to project on in the graphics window. The bodies must be in the projection direction and be big enough to cover the projected curves.
   To remove a bodies from the list, select the list entry and press the DEL key.
4. Click **Apply**. A new curve will now be created in a new part, wrapped around the surface of the selected bodies.

Continues on next page
Reversing curves

1. Click **Modify curve** to bring up a dialog box.

2. Select **Reverse** as modify method and then click the curves to reverse in the graphics window.

   Note that when resting the pointer over a curve, the current direction of the curve is displayed by yellow arrows.

   The **Selected curves** list displays the curves that will be reversed. To remove a curve from the list, select the list entry and press the DEL key.

3. Click **Apply**. The curves will now be reversed.

Splitting a curve

1. Click **Modify curve** to bring up a dialog box.

2. Select **Split** as modify method and then click the curve at the point to split at. Only open curves can be split.

   Note that when you rest the pointer over the curve, point of the split is highlighted. This point is affected by the current snap mode setting.

3. Click **Apply**. The curve will now be split to two separate curves in the same part.

Trimming a curve

1. Click **Modify curve** to bring up a dialog box.

2. Select **Trim** as modify method and then click the curve segment to trim.

   Note that when you rest the pointer over the curve, the closest vertices are highlighted. The segment between these points will be trimmed.

3. Click **Apply**. The selected part of the curve will now be removed.

**Generic content of the Modify Curve dialog box**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extend</strong></td>
<td>Extend a curve at any vertex with a straight line in the direction of the curve's tangent.</td>
</tr>
<tr>
<td><strong>Join</strong></td>
<td>Join two or more curves into one. The original curves will be deleted when joining curves.</td>
</tr>
<tr>
<td><strong>Project</strong></td>
<td>Project a curve onto a surface or a body, creating a new curve on the target part.</td>
</tr>
<tr>
<td><strong>Reverse</strong></td>
<td>Reverse the direction of curves.</td>
</tr>
<tr>
<td><strong>Split</strong></td>
<td>Split a curve in two bodies. Only open curves can be split.</td>
</tr>
<tr>
<td><strong>Trim</strong></td>
<td>Cut a segment of a curve between intersection or end points.</td>
</tr>
</tbody>
</table>
## Extend-specific information

<table>
<thead>
<tr>
<th><strong>Selected curve</strong></th>
<th>Display the name of the curve to extend. Select the curve by clicking it in the graphics window. Note that when you rest the pointer over the curve the endpoint closest to the pointer is highlighted. This is the endpoint that will be selected for extension when you click the curve.</th>
</tr>
</thead>
</table>

[Image: selected_curve.png]

<table>
<thead>
<tr>
<th><strong>Endpoint to extend</strong></th>
<th>Display the position of the endpoint to extend. To change the endpoint, select the curve again, but click it closer to the other endpoint.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Distance from endpoint</strong></th>
<th>Type the length of the extension here. You can also click a point in the graphics window to set the length. A yellow line indicates the length of the extension.</th>
</tr>
</thead>
</table>

[Image: endpoint_distance.png]

## Join-specific information

<table>
<thead>
<tr>
<th><strong>Selected curves</strong></th>
<th>Display the names of the curves to join. Select the curves by clicking them in the graphics window. To remove a curve from the list, select the list entry and press the DEL key. The curves may be either intersecting or adjacent to be joined.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Tolerance</strong></th>
<th>Specifies the distance within which the end points of adjacent curves may lie to be joined.</th>
</tr>
</thead>
</table>

[Image: join_curves.png]
14 The Context Menus

14.23. Modify Curve

**Project-specific information**

| Selected curve | Display the name of the curves to project. Select the curves by clicking them in the graphics window. Note that when you rest the pointer over the curve the projection direction is displayed. The project direction is always the negative Z direction of the User Coordinate System. To change the projection direction, create a new frame with the desired orientation and set it as user coordinate system. |
| Target bodies | Display the name of the bodies to project the curve on. Select the bodies by first clicking in the box and then clicking the bodies in the graphics window. |

**Reverse-specific information**

| Selected curves | Display the names of the curves to reverse. Select the curves by clicking them in the graphics window. Note that when resting the pointer over a curve the current direction of the curve is displayed by yellow arrows. |

Continued
### 14 The Context Menus

14.23. Modify Curve

*Continued*

#### Split-specific information

<table>
<thead>
<tr>
<th><strong>Selected curve</strong></th>
<th>Display the name of the curve to split. Select the curve by clicking it in the graphics window. Note that when you rest the pointer over the curve, the point where the split will be is highlighted. This point is affected by the current snap mode setting.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Point on curve</strong></th>
<th>These boxes show the position of the split point. To change the split point, select the curve again, but click another part of it.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

#### Trim-specific information

<table>
<thead>
<tr>
<th><strong>Selected curve</strong></th>
<th>Display the name of the curve to trim. Select the curve by clicking it in the graphics window. Note that when you rest the pointer over the curve the closest vertexes are highlighted. It is the segment between those points that will be cut.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>First section endpoint</strong></th>
<th>Display the position of the first trimming point.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Second section endpoint</strong></th>
<th>Display the position of the second trimming point.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

Trim only works on single curves with intersection points. If you want to trim a curve that intersects with another curve, first join the two curves.
14.24. Modify External Axis

Modifying external axis positions in targets

1. Select the targets (one or several) you want to modify, either in the Layout browser or in the graphics window. If you select several targets, the values you specify will be applied to all selected targets.

2. Click **Modify External Axis** to bring up a dialog box.

3. Edit the values of the axis by performing any of the following:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type a new position value for an axis</td>
<td>In the <strong>Eax</strong> column, select the value of the external axis you want to edit, and enter the new value.</td>
</tr>
<tr>
<td>Jog the axis to the new position</td>
<td>Use the arrow buttons to the left of the Joint Values column for jogging the axis. Then click the right arrow button between the Joint values column and the Eax column to transfer the current joint value to the Eax value.</td>
</tr>
</tbody>
</table>

4. Click **Apply**.

The Modify External Axis dialog box

<table>
<thead>
<tr>
<th>&lt;</th>
<th>Jog the joint of the external axis corresponding to each row by clicking the &lt; button.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>Jog the joint of the external axis corresponding to each row by clicking the &gt; button.</td>
</tr>
<tr>
<td>value box</td>
<td>Enter the axis value for the corresponding joint of the external axis in the value box.</td>
</tr>
<tr>
<td>&lt;-</td>
<td>With the arrow left button, you transfer the value from the Eax box to the corresponding value box.</td>
</tr>
<tr>
<td>-&gt;</td>
<td>With the arrow right button, you transfer the value from the value box to the corresponding Eax box.</td>
</tr>
<tr>
<td><strong>Eax</strong></td>
<td>Specify the value of the corresponding joint of the external axis.</td>
</tr>
</tbody>
</table>
14.25. Modify Instruction

Modifying an instruction

1. In the Paths & Targets browser, select the instruction you want to modify. If you want to apply the same properties to several instructions, press the CTRL key and select them.

2. Click Modify Instruction to bring up a dialog box.

3. For move instructions, select joint or linear motion in the Motion type list.

4. In the Instruction Argument group, modify the values for the instruction.

   For details about each argument, see the selected instruction in the RAPID Reference Manual. For an overview of the arguments for move instructions, see below.

5. When you have finished modifying, click Apply.

Arguments for move instructions

The table below is an overview of common arguments for move instructions. For detailed information about the arguments, see the selected instruction in the RAPID Reference Manual.

<table>
<thead>
<tr>
<th>To set the</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>subsequent instructions to be executed at once.</td>
<td>|Conc</td>
</tr>
<tr>
<td>destination target for the instruction target.</td>
<td>ToPoint</td>
</tr>
<tr>
<td>speed for the tool center point, the tool reorientation and external axes.</td>
<td>Speed</td>
</tr>
<tr>
<td>velocity of the TCP in mm/s directly in the instruction (it will be substituted for the corresponding speed data).</td>
<td>|V</td>
</tr>
<tr>
<td>total time in seconds during which the robot moves (it will be substituted for the corresponding speed data).</td>
<td>|T</td>
</tr>
<tr>
<td>size of the generated corner path.</td>
<td>Zone</td>
</tr>
<tr>
<td>position accuracy of the robot TCP directly in the instruction (the length of the path will be substituted for the corresponding zone specified in the zone data).</td>
<td>|Z</td>
</tr>
<tr>
<td>tool used for the movement (the TCP of this tool will be positioned at the destination target).</td>
<td>|Tool</td>
</tr>
<tr>
<td>workobject to which the robot position in the instruction belongs.</td>
<td>|Wobj</td>
</tr>
</tbody>
</table>
14.26. Modify Mechanism

The Modify Mechanism dialog box

See *The Modify Mechanism dialog box on page 251*. 
14 The Context Menus

14.27. Modify Tooldata

14.27. Modify Tooldata

Modifying tooldata

1. In the Layout browser, select the tooldata you want to modify.
2. Click Modify Tooldata to bring up a dialog box.
3. In the Misc Data group:
   • Modify the Name of the tool.
   • Select if the tool is to be held by the robot in the Robot holds tool list.
4. In the Tool Frame group:
   • Modify the Position x, y, z of the tool.
   • Modify the Rotation rx, ry, rz of the tool.
5. In the Load Data group:
   • Enter a new Weight for the tool.
   • Modify the Center of gravity for the tool.
   • Modify the Inertia for the tool.
6. In the Sync Properties group:
   • In the Storage type list, select PERS or TASK PERS. Select TASK PERS if you intend to use the tooldata in MultiMove mode.
   • In the Module list, modify the module in which to declare the tooldata.
7. Click Apply.
14.28. Modify Workobject

Modifying a workobject

1. In the Layout browser, select the workobject you want to modify.
2. Click Modify Workobject to bring up a dialog box.
3. In the Misc Data group, modify the values for the workobject:
   - Enter a Name for the workobject.
   - In the Robot holds workobject list, select True or False. If you select True, the robot will move the work piece instead of the tool.
   - In the Moved by mechanical unit list, select the mechanical unit with which the robot movements are coordinated. This setting is only valid when Programmed has been set to False.
   - In the Programmed list, select True or False. True means that the workobject will use a fixed coordinate system, and False that a movable coordinate system (that is, coordinated external axes) will be used.
4. In the User Frame group, do one of the following:
   - Modify the user frame by entering values for the Position x, y, z and the Rotation rx, ry, rz for the workobject. Click in one of these boxes, and then click the position in the graphics window to transfer the values.
   - Modify the user frame by using the Frame by points dialog box, see Frame from Three Points on page 193.
5. In the Object Frame group, do one of the following:
   - Modify the object frame by selecting values for Position x, y, z and Rotation rx, ry, rz for the workobject.
   - Modify the object frame by using the Frame by points dialog box.
6. In the Sync Properties group, modify the values for the workobject:
   - In the Storage type list, select PERS or TASK PERS. Select TASK PERS if you intend to use the workobject in MultiMove mode.
   - In the Module list, select the module in which to declare the workobject.
7. Click Apply.

NOTE!

If you change the position of a workobject that is used in a program, you have to synchronize the affected paths to the virtual controller; otherwise, the program will not be updated.
14.29. Move Along Path

Prerequisites

At least one path must have been created in the station.

A virtual controller must be running for the robot to move along the path.

Moving along a path

1. In the Paths & Targets browser, select the path to move along.
2. Click Move along path. In the graphics window, the robot will move along the path.
14.30. Move to Pose

Prerequisites

At least one joint position must be defined.

Only one mechanism may be selected at a time.

Moving to a pose

1. In the Layout browser, select one mechanism to move.
2. Click Move to Pose and then click one of the available poses. In the graphics window, the mechanism will move to the pose.
14.31. Place

Placing an item

1. Select the item you want move.
2. Click **Place** and then click one of the commands to bring up a dialog box.

<table>
<thead>
<tr>
<th>If you want to move the item</th>
<th>Choose</th>
</tr>
</thead>
<tbody>
<tr>
<td>from one position to another without affecting the orientation of the object.</td>
<td>One Point</td>
</tr>
<tr>
<td>Select the axes to be affected.</td>
<td></td>
</tr>
<tr>
<td>according to the relationship between a start and a finish line.</td>
<td>Two points</td>
</tr>
<tr>
<td>The object will move to match the first point, then it will rotate to match the second point.</td>
<td></td>
</tr>
<tr>
<td>according to the relationship between a start plane and a finish plane.</td>
<td>Three Points</td>
</tr>
<tr>
<td>The object will move to match the first point, then it will rotate to match the third point.</td>
<td></td>
</tr>
<tr>
<td>from one position to a target or frame position and simultaneously change the orientation of the object according to the frame orientation.</td>
<td>Align Object by Frame</td>
</tr>
<tr>
<td>The position of the object changes according the orientation of the to-point coordinate system.</td>
<td></td>
</tr>
</tbody>
</table>

3. Set the reference coordinate system you want to use.
4. Click the points in the graphics window to transfer values to the from-point boxes to the to-point boxes. For detailed information, see the tables below.
5. Click **Apply**.

**The Place Object by One Point dialog box**

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Point - From</td>
<td>Click in one of these boxes, and then click the primary point in the graphics window to transfer the values to the <strong>Primary Point - From</strong> boxes.</td>
</tr>
<tr>
<td>Primary Point - To</td>
<td>Click in one of these boxes, and then click the primary point in the graphics window to transfer the values to the <strong>Primary Point - To</strong> boxes.</td>
</tr>
<tr>
<td>Translate along these axes</td>
<td>Select whether the translation is to be performed along the X, Y or Z axis, or several of the axes.</td>
</tr>
</tbody>
</table>

**The Place Object by Two Points dialog box**

<table>
<thead>
<tr>
<th>Reference</th>
<th>Select the reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Point - From</td>
<td>Click in one of these boxes, and then click the primary point in the graphics window to transfer the values to the <strong>Primary Point - From</strong> boxes.</td>
</tr>
</tbody>
</table>

Continues on next page
### The Place an Object by Three Points dialog box

<table>
<thead>
<tr>
<th><strong>Reference</strong></th>
<th>Select the reference coordinate system to which all positions or points will be related.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Point - From</strong></td>
<td>Click in one of these boxes, and then click the primary point in the graphics window to transfer the values to the Primary Point - To boxes.</td>
</tr>
<tr>
<td><strong>Primary Point - To</strong></td>
<td>Click in one of these boxes, and then click the primary point in the graphics window to transfer the values to the Primary Point - To boxes.</td>
</tr>
<tr>
<td><strong>Point on X-Axis - From</strong></td>
<td>Click in one of these boxes, and then click the point on the x axis in the graphics window to transfer the values to the Point on X-Axis - From boxes.</td>
</tr>
<tr>
<td><strong>Point on X-Axis - To</strong></td>
<td>Click in one of these boxes, and then click the point on the x axis in the graphics window to transfer the values to the Point on X-Axis - To boxes.</td>
</tr>
<tr>
<td><strong>Point on Y-Axis - From</strong></td>
<td>Click in one of these boxes, and then click the point on the y axis in the graphics window to transfer the values to the Point on Y-Axis - From boxes.</td>
</tr>
<tr>
<td><strong>Point on Y-Axis - To</strong></td>
<td>Click in one of these boxes, and then click the point on the y axis in the graphics window to transfer the values to the Point on Y-Axis - To boxes.</td>
</tr>
<tr>
<td><strong>Translate along these axes</strong></td>
<td>Select whether the translation is to be performed along the X, Y or Z axis, or several of the axes.</td>
</tr>
</tbody>
</table>

### The Place Object with Frame dialog box

| **Select Frame** | Specify the name of the frame with which you want to place the object. |
14 The Context Menus

14.32. Remove Unused Targets

14.32. Remove Unused Targets

Removing unused targets

1. In the Paths&Targets browser, select either the Controller node or the Task node from which you wish to remove the unused targets, and then click Remove Unused Targets.

2. To the question “Are you sure you want to remove unused targets?”, answer Yes. All targets that not are used by any move instructions are now removed.
14.33. Rename Targets

Renaming targets

1. In the Paths & Targets browser, select the targets to rename.
   To rename all targets in one or several paths, select the paths that contain the targets.
2. Click Rename targets to bring up a dialog box.
3. In the Target Prefix box, enter a text string to precede the target numbers.
4. Optionally, in the Increment box and the Start with box, change the numbering series for the target names.
5. Optionally, in the Target Suffix box, enter a text string to follow the target numbers.
6. Click Apply.
## 14.34. Reverse Path

### The commands

<table>
<thead>
<tr>
<th>Simple</th>
<th>Here you reverse only the target sequence. The new path will keep the move instruction for each path segment and just reverse the programed positions.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="xx0500002041" alt="Path" /></td>
</tr>
<tr>
<td></td>
<td>Note that move instructions are not changed, just the targets. Even the MoveAbsJ instruction to the joint target is preserved, but placed last.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advanced</th>
<th>Both target sequence and move instructions are reversed in a way that corresponds to recording the robot movements and playing the movie backwards. For example, if the robot used a linear motion to move from a target, it will use a linear motion to move to the target after the reversal.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="xx0500002042" alt="Path" /></td>
</tr>
<tr>
<td></td>
<td>Note that the move instructions have changed together with the targets. For example, in the original path, a joint motion was used to reach target 20 and a linear motion to leave it. After the reversal there is a linear motion to the target and joint motion from it. Also, note that the jointtarget has been converted into an ordinary target; otherwise, it would not be possible to program a linear motion to that position.</td>
</tr>
</tbody>
</table>
14.35. Rotate

Rotating an item

1. Select the item you want to rotate.
2. Click **Rotate** to bring up a dialog box.
3. Select the reference coordinate system you want to use:

<table>
<thead>
<tr>
<th>If you want to move the item</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>absolute in the coordinate system of the station</td>
<td>World</td>
</tr>
<tr>
<td>relative to the coordinate system of its parent</td>
<td>Parent</td>
</tr>
<tr>
<td>relative to its own coordinate system</td>
<td>Local</td>
</tr>
<tr>
<td>relative to the user-defined system</td>
<td>UCS</td>
</tr>
<tr>
<td>relative to an axis defined by two points</td>
<td>User defined axis</td>
</tr>
</tbody>
</table>

4. Specify the rotation of the item in the **Rotate around x, y, z** by first clicking in one of the boxes, and then click the center position in the graphics window to transfer the values.
5. If you have selected the coordinate system **User defined axis**, specify the **Axis start point x, y, z** and the **Axis end point x, y, z**.
6. Specify the **Rotation** of the item and the axis around which the rotation is to occur.
7. Click **Apply**.
14.36. Rotate Path

Rotating a path:

1. In the **Layout** browser or the graphics window, select the paths to rotate.
2. Click **Rotate path** to bring up a dialog box.
3. In the **Reference frame** list, select the frame to rotate the paths around.

<table>
<thead>
<tr>
<th>Select</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>World</td>
<td>rotate around the station’s world coordinate system</td>
</tr>
<tr>
<td>Baseframe</td>
<td>rotate around the robot’s baseframe</td>
</tr>
<tr>
<td>UCS</td>
<td>rotate around a frame or target that previously has been set to User Coordinate System.</td>
</tr>
<tr>
<td>Select Frame</td>
<td>rotate around an existing target or frame other than the listed ones. When using <strong>Select Frame</strong>, specify the frame to rotate around further down.</td>
</tr>
</tbody>
</table>

4. If **Selected frame** was selected in the **Reference frame** list, specify a frame or target in the text box by clicking in the box and then selecting the frame in the graphics window.
5. With the **Rotation axis** options, select the axis of the frame to rotate around.
6. In the **Rotation angle** box, enter the rotation.
7. Click **Apply**.
14.37. Save Module As

Saving a module

1. Click **Save Module As** to bring up a dialog box.
2. Browse to the location of the module to be loaded to your station and click **Save**.
14 The Context Menus

14.38. Save Program As

14.38. Save Program As

Saving a program

1. Click Save Program As to bring up a dialog box.
2. Browse to the location where you want to save your program, and click Save.
14.39. Set Local Origin

Setting the origin of the local coordinate system

1. If the object you want modify is a library component, first disconnect it from the library.
2. In the Layout browser or the graphics window, select the part to modify.
3. Click Set Local Origin to bring up a dialog box.
4. In the Set Local Origin dialog box, select the reference coordinate system you want to use:

<table>
<thead>
<tr>
<th>If you want to move</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>relative to the part’s current local coordinate system</td>
<td>Local</td>
</tr>
<tr>
<td>relative to the coordinate system of its parent</td>
<td>Parent</td>
</tr>
<tr>
<td>absolute in the coordinate system of the station</td>
<td>World</td>
</tr>
<tr>
<td>relative to a user-defined coordinate system</td>
<td>UCS</td>
</tr>
</tbody>
</table>

5. In the Position X, Y, Z boxes, either type the new position or, select it by first clicking in one of the value boxes and then clicking the point in the graphics window.
6. Type the Orientation.
7. Click Apply.
14.40. Set Normal to Surface

Setting the target orientation normal to a surface

1. In the Paths & Targets browser, select the target to modify.
2. Click Set Normal To Surface to bring up a dialog box.
3. On the Selection Level toolbar, set the selection level.
   • To align the target to a specific surface, set the selection level to surface.
   • To align the target to a specific point at the surface, set the selection level to part.
4. In the graphics window, click the reference surface. This will transfer the name of the part or surface to the Surface box.
5. In the Approach Direction, click the button for the axis to be used as the approach direction.
6. To set the distance between the surface and the target in the approach direction, specify an Offset value.
7. Click Apply.
14.41. Set Position

Positioning an item

1. Select the item you want to move.
2. Click Set Position to bring up a dialog box.
3. In the Set Position dialog box, select the reference coordinate system you want to use:

<table>
<thead>
<tr>
<th>If you want to move the item</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>relative to its own coordinate system</td>
<td>Local</td>
</tr>
<tr>
<td>relative to the coordinate system of its parent</td>
<td>Parent</td>
</tr>
<tr>
<td>absolute in the coordinate system of the station</td>
<td>World</td>
</tr>
<tr>
<td>relative to a user-defined coordinate system</td>
<td>UCS</td>
</tr>
</tbody>
</table>

4. In the Position X, Y, Z boxes, either type the new position, or select it by first clicking in one of the value boxes and then clicking the point in the graphics window.
5. Specify the Orientation for the item.
6. Click Apply.
14.42. Tool Compensation

Offsetting a path to compensate for tool radius

1. In the Paths&Targets browser or the graphics window, select the path.
2. Click Tool Compensation to bring up a dialog box.
3. In the Distance box, enter the size of the compensation (normally, the tool radius).
4. Using the Direction options, select whether the new path shall be on the left or the right side of the current path.
5. Click Apply.
14.43. Translate Path

Translating a path

1. In the Paths&Targets browser or the graphics window, select the paths to translate.

2. Click Translate path to bring up a dialog box.

3. In the Reference frame list, select the coordinate system to use as reference for moving the paths.

<table>
<thead>
<tr>
<th>Select</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>World</td>
<td>move relative to the origin of the world coordinate system</td>
</tr>
<tr>
<td>Base Frame</td>
<td>move relative to the origin of the robot's baseframe</td>
</tr>
<tr>
<td>UCS</td>
<td>move relative to the origin of a frame or target that previously has been set to User Coordinate System.</td>
</tr>
<tr>
<td>Select Frame</td>
<td>move relative to the origin of an existing target or frame other then the listed ones. When using Select Frame, specify the frame to use further down.</td>
</tr>
<tr>
<td>Point to Point</td>
<td>move the path from one point to another without specifying any coordinate system.</td>
</tr>
</tbody>
</table>

4. If Select frame was selected in the Reference frame list, specify a frame or target in the text box by clicking in the box and then selecting the frame from the graphics window.

5. In the Translation vector box, specify the distance to move the path along the X, Y and Z axes of the reference frame.

   Translation vector is applicable only if a reference frame is used. If Point to Point is used as reference, specify the start and end points for the translation, instead. To do this, click in one of the boxes for the point to specify and then select the point in the graphics window, or type the coordinates of the point.

6. Click Apply.
14.44. View Robot at Target

Viewing a robot at a target

1. Click View Robot at Target.
2. Select a target, either in the Paths&Targets browser or in the graphics window.
3. The robot will be shown at each selected target whenever a target is selected. By stepping through the targets in the browser, it will be easy to see how the position of the robot changes.
4. To turn the function off, click the command again.
14.45. View Tool at Target

Viewing a tool at a target

1. Click View Tool at Target and select the tool you want to view at the target.

2. Select a target, either in the Paths&Targets browser or in the graphics window. You can also multiselect targets to show several copies of the tool.

   A copy of the tool will be shown at the selected target. By stepping through the targets in the browser, it is easy to see how the tool orientation changes.

3. To turn the function off, click the command and clear the check box.
14 The Context Menus

14.45. View Tool at Target
A
action instruction about 21
alerts activate 124
B
base coordinate system 22
browser Layout 34
Modeling 37
Paths & Targets 35
C
CAD file
troubleshoot and optimize 71
CAD formats convert 70
collision detection 119
sets 119
configuration robot axis 24
collision monitoring about 25
confJ about 25
ConfL about 25
controller system create 139
controller world coordinate system 22
cordination systems 22
cycle time
measure 125
D
data declaration 19
E
director 113
element select 47
Ethernet 134
event create 122
external axis program 109
F
frame converting to workobject 83
creating by points 83
frames 22
function 19
G
gemetry
troubleshoot and optimize 71
Grants, about 138
Grants, give to groups 310

Group, about 137
Group, add 309
Group, add user 308
Group, remove 310
Group, rename 309
Groups, give grants 310
I
I/O
set 123
import 68
instruction about 21
item select 47
J
jog mechanism 84
robot 84
several mechanisms 84
jointtarget creating 85
K
keyboard shortcuts 49
L
library troubleshoot and optimize 71
local coordinate system set 77
local origin set 77
M
MediaPool 18
module 19
Move instruction teach 213
move instruction about 21
MoveJ teach 213
MoveL teach 213
MultiMove
programming workflow 102
O
object
set local origin 77
troubleshoot and optimize 71
orientations 91
align target 92
copy and apply 93
target normal to surface 91
unordered 91
output window 41
Index

P
pack, unpack 129
part
  set local origin 77
Password, change for user 308
path 87
  about 21
  compensating 88
  creating 87
  creating from curve 87
  reversing 87
  rotating 88
  setting axis configuration 87
  translating 88
PC, connecting 132
positioner.program 109
procedure 19
process time
  measure 125
program 113
  copy 128
  programming
    overview 82
R
RAPID
  concepts 19
  copy program 128
RAPID instructions 94
reachability
  test 100
remote subnet 136
robot
  programming overview 82
RobotWare 17
  license key 17
RobotWare system 17
  routine 19
S
safety 14
service port 132
signals
  set 123
simulate
  alerts 124
  create event 122
  measure process time 125
  set signals 123
  TCP trace 124
simulation 117, 119
station
  build workflow 54
  pan 46
  rotate 46
  zoom 46
station world coordinate system 22
synchronization 112
  station to VC 112
  VC to station 112
system
  copy 149
  create 139
  create from backup 150
  create with positioner 155
  modify 145
RobotWare 17
System, create 139
T
target 85
  about 21
  creating 85
  modifying 85
  modifying with ModPos 85
  removing unused 86
  renaming 86
  teaching 85
TCP 22
TCP trace
  activate 124
Tool Center Point coordinate system 22
tooldata 76
tools 76
track
  program 109
  trap 19
U
UCS 23
unpack 175
user coordinate system 23
User, about 137
User, add 307
User, add to group 308
User, change password 308
User, change user name 308
User, enable and disable 308
User, remove 307
V
VSTA 29
W
WorkObject 23
workobject
  creating 83
  modifying 83
workobjects 83
world coordinate system 22