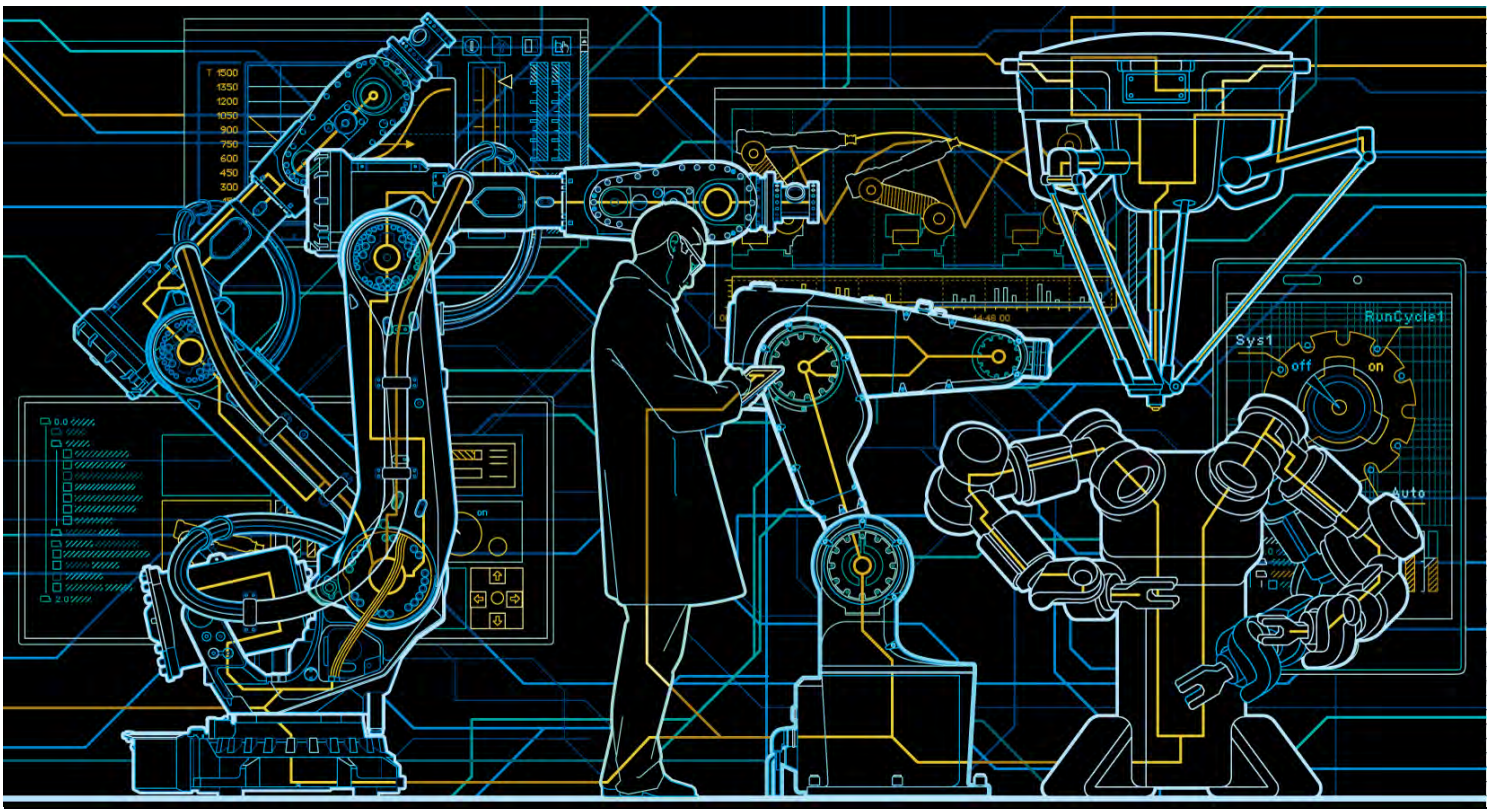


Operating manual
Machining PowerPac - CAM Converter Functionality

Power and productivity
for a better world™

Trace back information:
Workspace Main version a154
Checked in 2016-03-27
Skribenta version 4.6.284

Operating manual

Machining PowerPac - CAM Converter Functionality

RobotStudio 6.03

Document ID: 3HAC050262-001

Revision: B

ABB Engineering (Shanghai) Ltd.
No.4528, KangXin Highway, PuDong New District,
SHANGHAI 201319
CHINA

Table of contents

Overview of the manual

About this manual

This manual describes how to use Machining PowerPac - CAM Converter to convert CNC G-code to RAPID machining programs.

Usage

This manual should be used when working with Machining PowerPac - CAM Converter.

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:

- Industrial robots and their terminology
- RAPID programming language
- RobotStudio

Organization of chapters

The manual is organized in the following chapters:

Chapter	Content
1 Introduction	Describes terms and concepts of Machining PowerPac - CAM Converter.
2 Installation	Describes how to install Machining PowerPac - CAM Converter.
3 Navigating Machining PowerPac - CAM Converter	Describes the graphical user interface of Machining PowerPac - CAM Converter.
4 Workflow for Machining PowerPac - CAM Converter	Describes how to work with Machining PowerPac - CAM Converter.
5 Reference information	Describes parse rule and export rule for converting CNC code to RAPID program; provides information of supported features.

References

Reference	Document ID
Operating manual - RobotStudio	*3HAC032104-001*
Operating manual - Machining PowerPac - Machining Functionality	*3HAC054781-001*
Technical reference manual - RAPID overview	*3HAC050947-001*

Revision

Revision	Description
-	First edition

Continues on next page

Continued

Revision	Description
A	Released with RobotStudio 6.01 • The title of this manual is changed from *CAM Converter PowerPac* to *Machining PowerPac - CAM Converter Functionality*. • Hardware requirements in *Required hardware on page 18* are updated. • Path editor function in *Editing a path on page 71* is updated. • Some figures are updated.
B	Released with RobotStudio 6.03 • Application limitation of Machining PowerPac - CAM Converter is added. See *Application limitation on page 10*.

1 Introduction

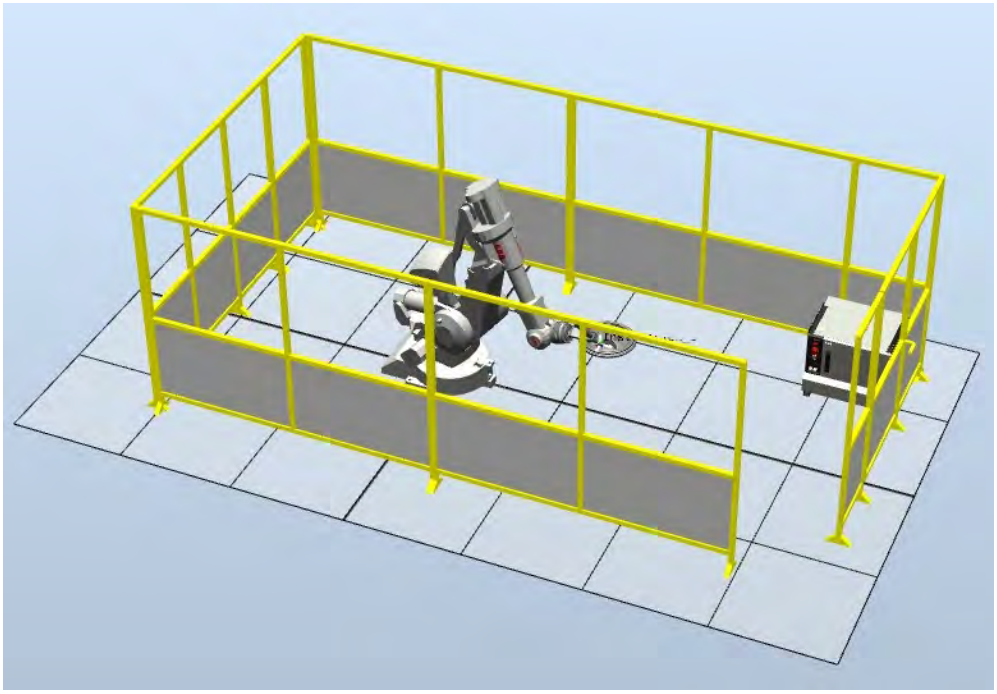
1.1 Introduction to Machining PowerPac - CAM Converter

Overview

Machining PowerPac - CAM Converter is a process specific add-in to RobotStudio. This software provides a new solution to solve difficulties when converting complex G-code to RAPID programs.

For details on robot points, RobotStudio coordinate systems, and camera adjustment, see the RobotStudio and RobotWare manuals.

The following picture shows a typical robotic machining cell, which uses a robot as the machining tool manipulator.



xx1500000456

Key features

- Machining PowerPac - CAM Converter enables users to complete the converting process without leaving its User Interface (UI) or switching to RobotStudio.
- Through offline simulation, users can verify and modify robot paths before exporting them to RAPID.
- With Work Envelope, users can view the robot orientation space, and adjust the work space to fully cover the path needed for running a specific task.
- The UI design follows Computer Aided Manufacturing (CAM) style and matches CAM engineer's background. It is easy to use with minimum requirements.

Continues on next page

- Convert CNC G-code to RAPID program: Parse the CNC code and convert it to RAPID program based on the convert rule. The parsing rule and convert rule can be customized.

Prerequisites

To use Machining PowerPac - CAM Converter, we recommend that you have a basic knowledge of:

- RobotStudio
- RAPID processing

Application limitation

Machining PowerPac - CAM Converter supports only ABB 6-axis robots, except for IRB 6640ID, IRB 6640 LeanID, and 6-axis painting robots IRB 52, IRB 5400, and IRB 580.

1.2 Definitions and Abbreviations

1.2.1 Abbreviations

RS	RobotStudio
TCP	The tool center point coordinate system
CAM	Computer Aided Manufacturing
RW	RobotWare
UI	User Interface

1.2.2 Typical frames/coordination systems

Overview

This section provides an introduction to the coordinate systems used mostly for offline programming. In RobotStudio, you can either use the coordinate systems (that are explained below) or the user-defined coordinated systems for co-relating elements and objects.

Frame/Coordinate systems

A frame/coordinate system defines a plane or space by axes from a fixed point called the origin. Robot targets and positions are located by measurements along the axes of coordinate systems.

A robot uses several coordinate systems, each suitable for specific types of jogging or programming.

- The base coordinate system is located at the base of the robot. It is the easiest one for just moving the robot from one position to another.
- The work object coordinate system is related to the work piece and is often the best one for programming the robot.
- The tool coordinate system defines the position of the tool the robot uses when reaching the programmed targets.
- The world coordinate system that defines the robot cell, all other coordinate systems are related to the world coordinate system, either directly or indirectly. It is useful for jogging, general movements and for handling stations and cells with several robots or robots moved by external axes.
- The user coordinate system is useful for representing equipment that holds other coordinate systems, like work objects.

The coordinate systems are co-related hierarchically. The origin of each coordinate system is defined as a position in one of its ancestries. The following are the descriptions of the commonly used coordinate systems.

Tool center point (TCP)

The position of the robot and its movements are always related to the tool center point. This point is normally defined as being the center point of the tool or somewhere on the tool, for example at the cutting head tip of a laser cutting gun, at the center of a gripper, or at the end of a grading tool.

The tool center point coordinate system is also called TCP. Different TCPs can be defined for one robot, but only one may be active at any time. All robots have one predefined TCP at the robot tool mounting point, called **tool0**.

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

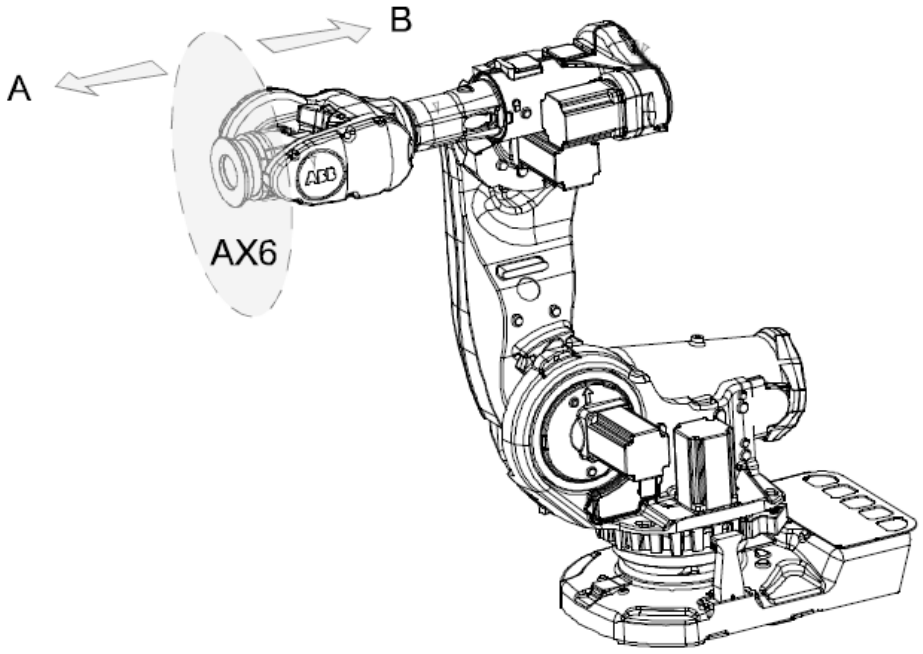
When a position is recorded, it is the position of the TCP that is recorded. This is also the point that moves along a given path at a given velocity.

Tool

A tool is an object that can be mounted directly or indirectly on the robot turning disk or fitted in a fixed position within the robot working range.

Continues on next page

A fixture or a cutting gun is not a tool. All tools must be defined with a TCP. Each tool that can be used by the robot must be measured and its data stored to achieve accurate positioning of the tool center point.



A	Tool side
B	Robot side

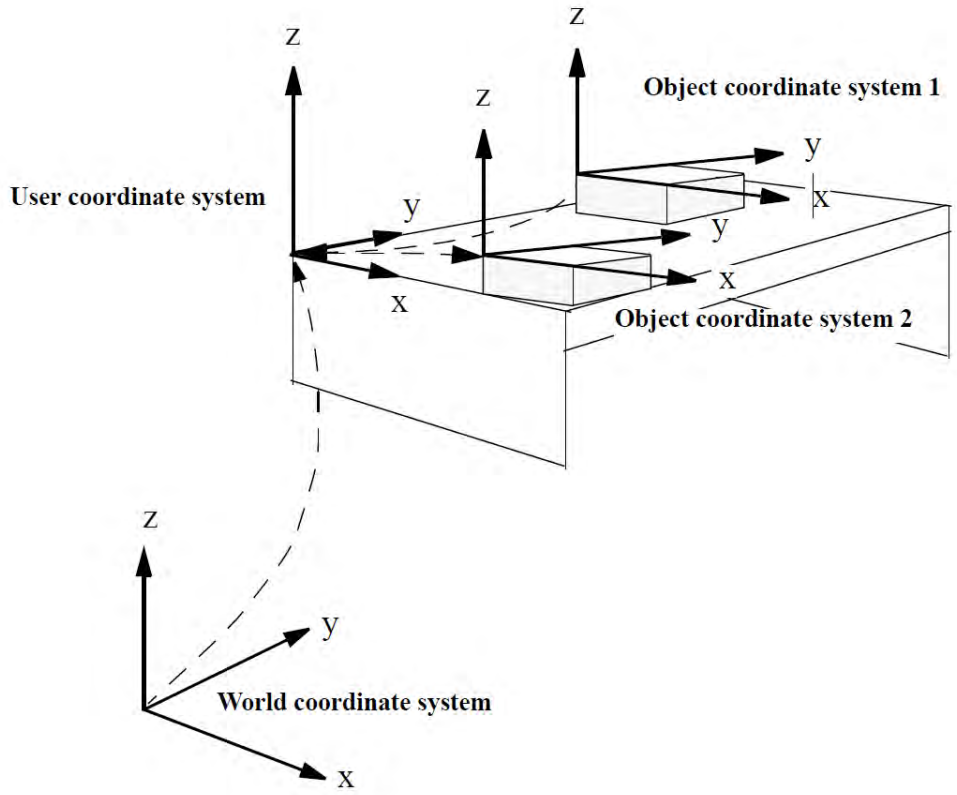
Work object

A coordinate system referenced to an work object is called an work object coordinate system. This coordinate system is also very suited to off-line programming since the positions specified can usually be taken directly from a drawing of the work object. The work object coordinate system can also be used when jogging the robot.

Continues on next page

1.2.2 Typical frames/coordination systems
Continued

The work object coordinate system is defined based on the user coordinate system. It must be defined in two frames, the user frame (related to the world frame) and the object frame (related to the user frame).



The programmed positions are always defined relative to a work object coordinate system.

If a fixture is moved/turned, this can be compensated for by moving/turning the user coordinate system. Neither the programmed positions nor the defined work object coordinate systems need to be changed. If the work object is moved/turned, this can be compensated for by moving/turning the work object coordinate system.

If the user coordinate system is movable, that is, coordinated additional axes are used, then the object coordinate system moves with the user coordinate system. This makes it possible to move the robot in relation to the object even when the workbench is being manipulated.

World Frame/World Coordinate system (WCS)

The RobotStudio 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 (when using RobotStudio).

Base Frame (BF)/Base coordinate system (BCS)

The base coordinate system is called the Base Frame (BF). Each robot in the station, both in RobotStudio and the real world, has a base coordinate system which is always located at the base of the robot.

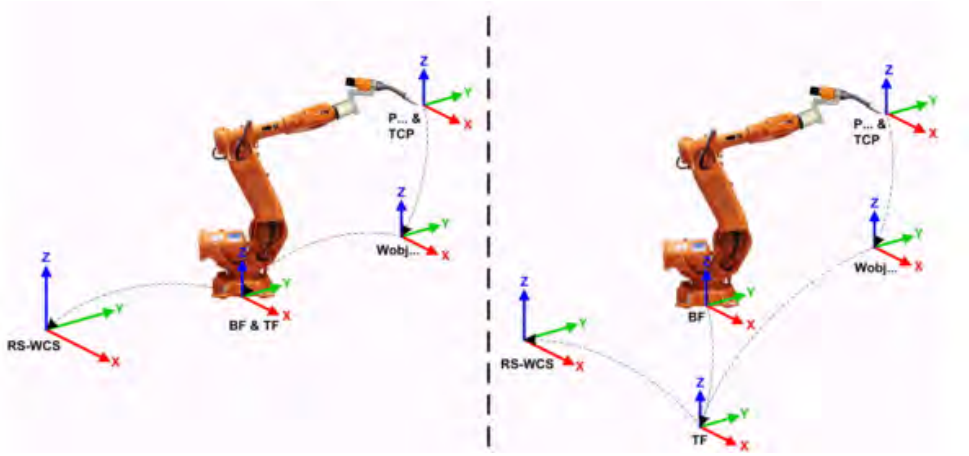
Continues on next page

Task Frame (TF)

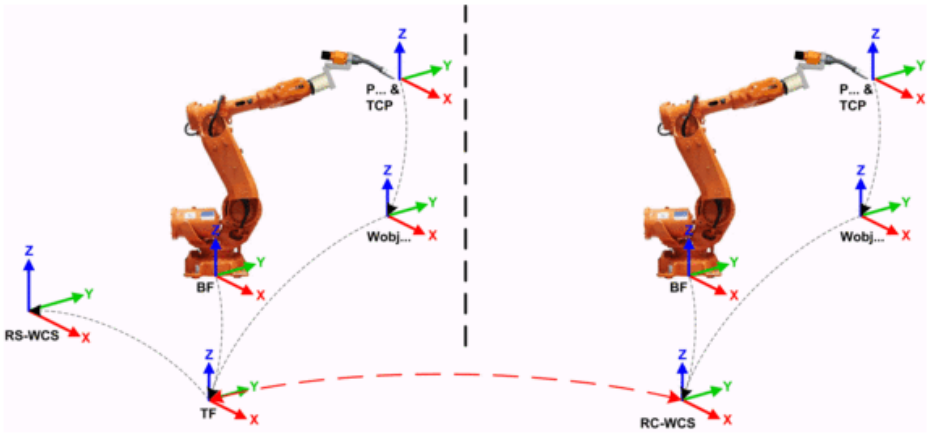
The Task Frame represents the origin of the robot controller world coordinate system in RobotStudio.

The following picture illustrates the difference between the base frame and the task frame.

In the picture to the left, the task frame is located at the same position as the robot base frame. In the picture to the right, the task frame is moved to another position.



The task frame in RobotStudio is mapped to the robot controller coordinate system in the real world, for example, on the work shop floor.



User Frame/User Coordinate System

The user coordinate system is used to get different coordinate systems for different fixtures or working surfaces. A fixture, however, may include several work objects that are to be processed or handled by the robot. It often helps to define a coordinate system for each object to make it easier to adjust the program if the object is moved or if a new object, the same as the previous one, is to be programmed at a different location.

Reference Coordinate System

In Machining PowerPac, each target is combined with a local reference coordinate system, which is named Reference Coordinate System (RCS).

Continues on next page

1.2.2 Typical frames/coordination systems
Continued

RCS is primarily used as a helper frame for making it easier to create and manipulate targets.

The default RCS is defined

- X axis direction is along the path direction.
- Z axis direction is along the normal vector of the surface.
- Y axis direction follows the right hand rule.

Object Frame

The object frame is based on the user frame. Users can define the object frame by two means:

- Position. Assign the origin of the object frame, a point on X axis, a point on XY platform to define the frame.
- Three-point. Assign the first point on X axis, the second point on X axis and a point on Y axis to define the frame.

2 Installation

2.1 Downloading and license

Getting the software

The latest version of Machining PowerPac with the CAM Converter functionality can be downloaded at:

http://new.abb.com/products/robotics/robotstudio/downloads.

The downloaded software will give you 30 days free use of the Add-In.

2.2 Prerequisites and system requirements

Prerequisites

To install Machining PowerPac with the CAM Converter functionality, you must have the following items:

- RobotStudio and RobotWare installed on your computer.
- Machining PowerPac installation package.
- A license certificate.
- A Windows account with administrator's privileges.

System Requirements

To work with Machining PowerPac - CAM Converter, the following is required:

Required hardware

- CPU: 2.0 GHz Intel Pentium 4 or faster processor.
- Memory: 1 GB RAM or more (More is recommended).
- Available disk space: 5+ GB on the system disk, 250+ MB on the installation disk.
- Screen resolution: 1920 x 1080 pixels (Recommended).
- Colors: 256 or higher.
- DPI: 120 dpi.
- Mouse: Three-button mouse.

Software requirements

- Microsoft Windows 7.
- RobotStudio 6.03.

2.3 Installing

Overview

To install Machining PowerPac with the CAM Converter functionality, RobotStudio and RobotWare must be installed on your computer first.

Installing Machining PowerPac with the CAM Converter functionality

Use this procedure to install the Machining PowerPac with the CAM Converter functionality:

	Action	Note
1	Browse to Machining PowerPac installation package and double-click **ABB Machining PowerPac 6.03.exe**. The installation starts.	
2	Read the License Agreement and accept the terms.	
3	Click **Install.**	
4	When the installation is finished, click **Finish** to close the installation wizard.	

Installing a license

The procedure of installing a license is the same as that of RobotStudio. See the procedure for installing RobotStudio license for reference.

2.4 Getting started

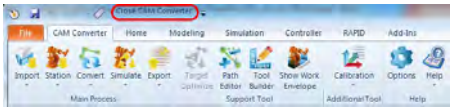
Starting Machining PowerPac - CAM Converter

Use this procedure to start a CAM Converter session.

	Action	Illustration/Note
1	Open **RobotStudio**.	
2	Create a new station or load an existing one. For more information on how to manage a station, see *Operating manual - RobotStudio* .	**Note** Machining PowerPac - CAM Converter can not be started with an empty station.
3	On the **Add-Ins** tab on the RobotStudio ribbon, click **CAM Converter** from the **PowerPacs** group.	
4	The CAM Converter user interface will be displayed.	

Closing Machining PowerPac - CAM Converter

Use these methods to close Machining PowerPac with the CAM Converter functionality.

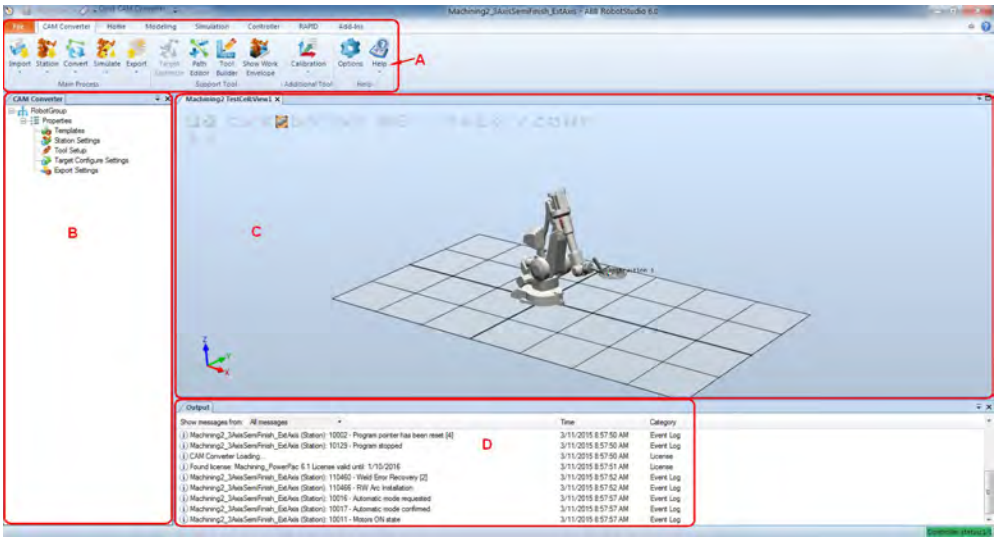
	Action	Illustration/Note
1	Click the **Close CAM Converter** button on the RobotStudio title bar.	
2	The Machining PowerPac - CAM Converter should now be closed.	
3	Alternatively, you can click the **CAM Converter** button in the CAM Converter ribbon tab.	

3 Navigating Machining PowerPac - CAM Converter

3.1 Overview

The graphical user interface

The GUI of Machining PowerPac - CAM Converter contains four main parts as shown in the following picture.



xx1400000044

	Item	Description
A	CAM Converter ribbon tab	The ribbon tab contains general tools for a converting process.
B	CAM Converter Post Tree	The **CAM Converter Post Tree** organizes the robot group in a tree structure. The robot group is then divided into nodes and enables interaction with the components. With RobotGroup being the head node, it contains program groups and operations and offers interaction on different levels.
C	3D graphics window	The graphics window is coordinated with these panes: A path highlighted in the browser, is highlighted with the same color in the graphics window. A simulation appearing in the graphics window is represented in the path view by a robot cursor stepping through the path in the path view. The graphics window is an important source to input geometry targets. By selecting part models in the window, you can create or modify a target in the geometry space.
D	Output window	The **Output** window is used to display system announcements and alerts.

Continues on next page

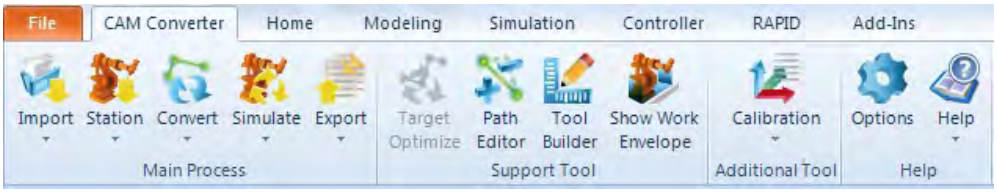
Note

In addition, various windows will emerge when certain tools are used.

3.2 CAM Converter ribbon tab

Overview

Below is the CAM Converter ribbon tab. For guidelines on how to execute a certain function, see *Workflow for Machining PowerPac - CAM Converter on page 39*.



xx1400000049

Note

Any action, including configuration, simulation, and optimization, when started from the CAM Converter ribbon tab, is on a global basis. To work on a local basis, interact with the sub nodes in the CAM Converter Post Tree. For details on the Post Tree, see CAM Converter Post Tree.

Main Process group

Group	Button	Description
Import 	Import CNC Code	For importing a model into the CAM Converter with CNC code. Supported file types include 3 axis, 5 axis and APT code with these file extensions: .nc, .cls, and .apt.
	Import CAD Model	For importing a CAD model into the CAM Converter. Only .sat file can be imported this way.
Station 	Design Layout	For editing the design layout for the existing station in the 3D graphics window.
	Edit Station 	For editing the station by setting properties for Robot, Tooling, WorkObject, and ExternalAxis.

Continues on next page

3.2 CAM Converter ribbon tab
Continued

Group	Button	Description
Convert 	Tool Setup 	For configuring mapping from CAM tool to Robot tool, adjusting robot tool and tool radius compensation distance.
	Target Configure Settings 	For adjusting robot arm, setting axis properties, rotary type, and index value.
	Convert 	For converting CNC G-code to robot paths.
Simulate 	Simulate	For simulating chosen data to test layout and robot program.
	Edit Collision Set	For editing collision sets and selecting objects to test for collision.
Export 	Export Template 	For setting an export program template by selecting an xml style sheet.
	Export Settings 	For adjusting export settings and setting target folder.
	Export RAPID	Export ABB RAPID robot program

Support Tool group

Button	Description
Target Optimize	For automatically checking and resolving robot path process errors.
Path Editor 	For editing parameters on the robot path.
Tool Builder 	For managing the tool library; creating, deleting, and changing cutter.

Continues on next page

Button	Description
Show Work Envelope	For viewing the robot orientation space and adjusting the work space.

Additional group

Group	Button	Description
Calibration 	Calibration List	Lists calibration tasks.
	Robot Hold Tool	For calibrating fixed work objects.
	Robot Hold WorkObject	For calibrating robot-hold work objects.
	Export 	Exports calibration tasks.

Help group

Group	Button	Description
Options	-	For setting general properties based on user preferences.
Help	Contents	For opening the help file.
	About	For displaying the version and other useful information.

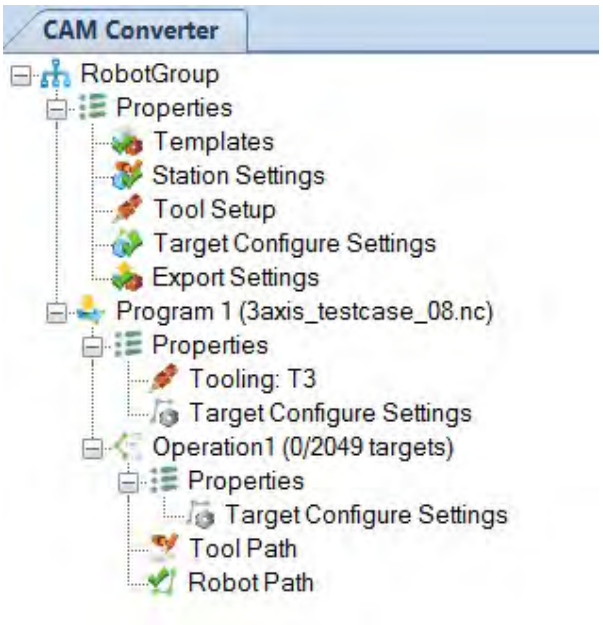
3.3 CAM Converter Post Tree

Overview

The **CAM Converter Post Tree** provides an overview of the different program groups, tools, settings and properties. Using Post Tree, you can perform any action you can with the CAM Converter ribbon tab.

The Post Tree is always visible on the left side of the screen. It changes depending on amount of program groups. The main RobotGroup is divided into program groups and later into operations.

The **RobotGroup** is the main branch of the tree along with properties that are passed down to all branches. On the same level as **Properties** under RobotGroup, there are program groups with belonging tools, operations, tool paths, robot paths, and settings in Properties and Operation level. When a change is applied on the RobotGroup, it is applied to all the sub-branches as well. But a change on a sub branch will not have any impact on other branches of the same level nor the main branch.



xx1400000050

Continues on next page

3.3.1 CAM Converter Post Tree status

Post Tree status icons

The CAM Converter Post Tree icons may vary depending on its current status. Below is a table of the different icon status.

Icon	Status	Description
	None	On Settings node: There are no current settings. Input needed or the related function cannot work. On Robot Path node: The conversion hasn't been done.
	Wrong	On Settings node: The settings are incorrect and need update. On Robot Path node: Conversion of some or all nodes failed.
	Out of Time	On Robot Path node: Some settings have been modified and the robot path is out of time. Re-convert it.
	Right	On Settings node: Settings are fine. The settings can be default. On Robot Path node: The robot path is fine.
	Locked	On Robot Path node: Robot path locked and cannot be modified until unlocked.

3.3.2 CAM Converter Post Tree interaction

Overview

> **Note**
>
> **RobotGroup** is the head branch of the Post Tree, with **Program Groups** being sub-nodes, containing operations.

If RobotPath under any **Operation** is locked (by right-clicking that **RobotPath** and then left-click **Lock**), it will ignore any command given by superior nodes.

Post Tree task table

When right-click the nodes, there will be context menu displayed. This table describes these tasks in the context menus.

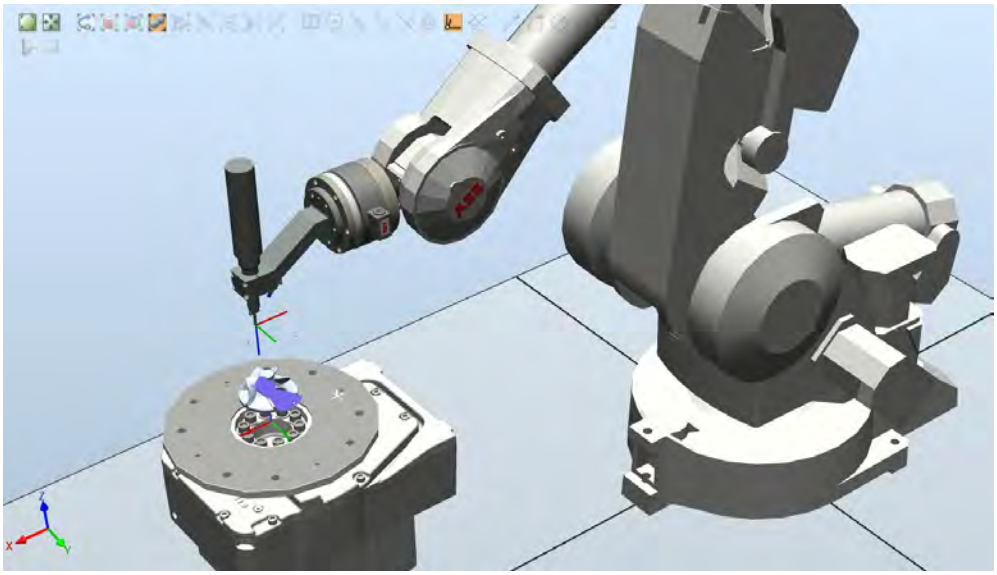
Node	Task	Description
RobotGroup	Show Tool Path	Displays tool path of the entire tree in the 3D graphics window.
	Delete All Programs	Delete all pragrams in Robot group.
	Clear All Previews	Cleare all previews in Robot group.
	Import CNC Code	Opens **Import CNC code** window.
	Import CAD Model	Opens **Import CAD model** window.
	Edit Collison Set	Opens **Collison Objects selections** window and edit collison set.
	Export RAPID	Exports Robot path into RAPID File under the path which system identified.
RobotGroup Properties	Templates	Opens **Files** window for displaying CNC rule library and export rule library.
	Station Settings	Opens **Read existing station** window for editing station properties.
	Tool Setup	Opens **Tool Setup** window for edit tooldata.
	Target Configure Settings	Opens **TargetConfig** window for robot configuration, axis settings, interpolation method, and index settings.
	Export Settings	Opens **Export Settings** window for configuring export settings.
Program Group	Show Tool Path	Displays tool path of the program group in the 3D graphics window.
	Edit Tool Path	Edit tool path in **Instructions** window.
	Simulate	Opens Simulate window.
	Convert	Opens **Convert** window for converting points.
	Delete	Delete Program Group.
Program Group Properties	Tool Setup	Opens **Tool Setup** window for mapping from CAM tool to Robot tool and adjusting radius compensation radius.
	Target Configure Setting	Opens **TargetConfig** window for program group axis settings.

Continues on next page

Node	Task	Description
Program Group Operation	Show Tool Path	Displays tool path of the Operation in the 3D graphics window.
	Edit Tool Path	Edit tool path in **Instructions** window.
	Delete	Delete Program Group operation.

Post Tree actions

In the Post Tree, click Tool Path, the tool path will be displayed in the 3D graphics window as shown in the picture below.

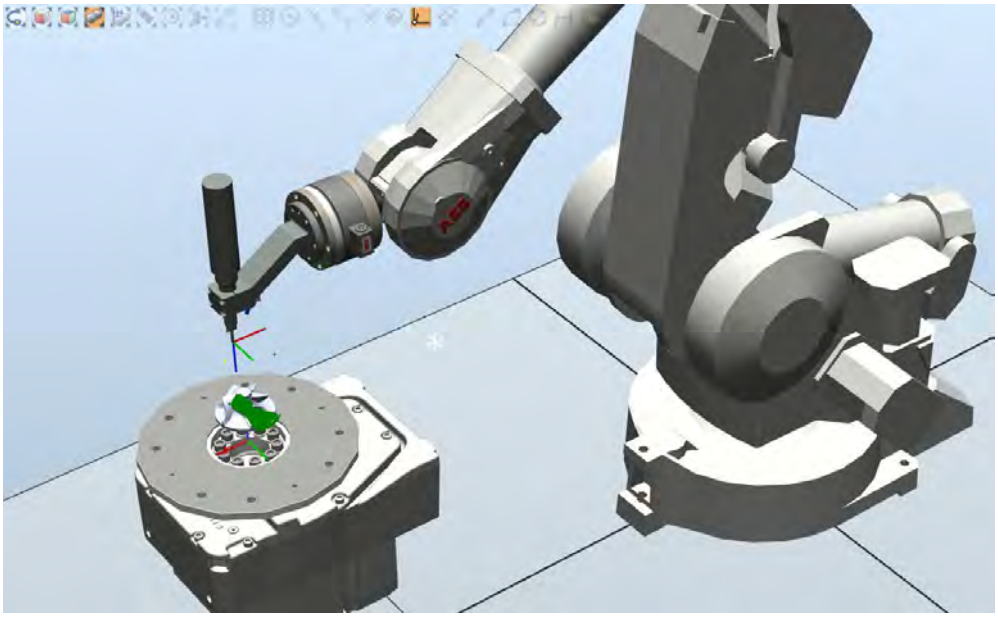


xx1400000123

Continues on next page

3.3.2 CAM Converter Post Tree interaction
Continued

In the Post Tree, select a programe and choose convert, then Click Robot Path after conversion, the robot path will be displayed in the 3D graphics window as shown in the picture below.

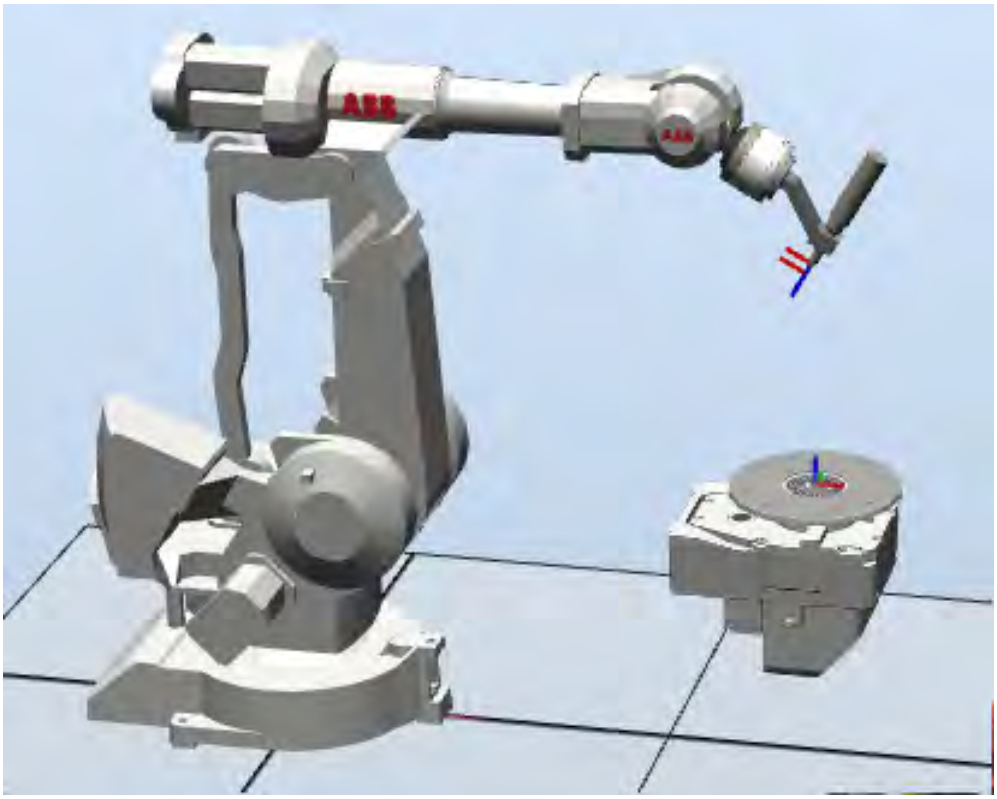


xx1400000124

3.4 3D graphics window

3D view interface

In this window, you can choose to view settings, control graphics view, create new views, view/hide the selected targets, frames, paths, parts, and mechanisms. For detailed information, see *Operating manual - RobotStudio*.



xx1400000052

3.5 CAM Converter Options

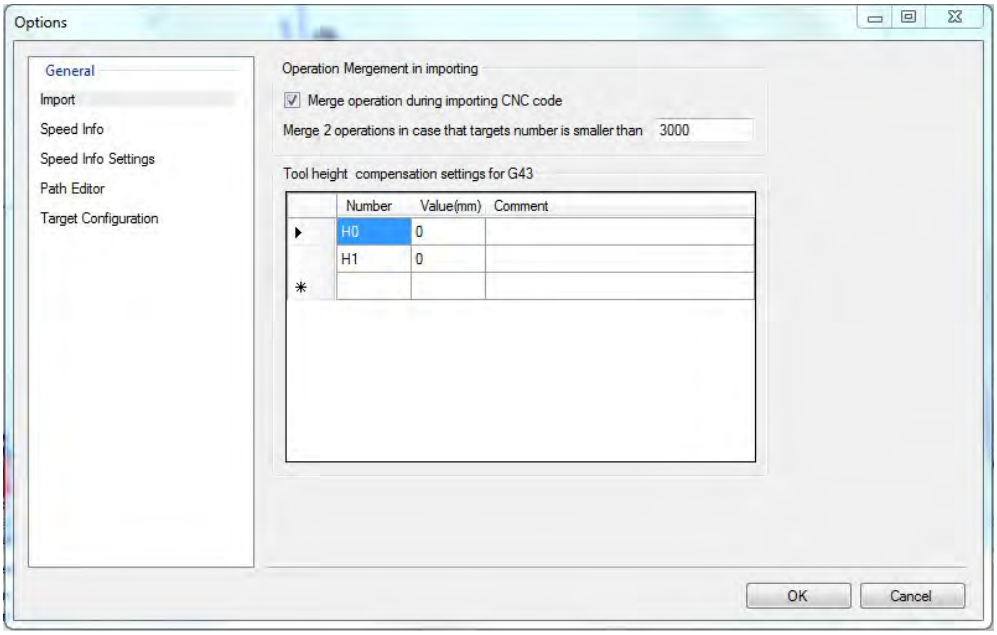
Overview

The CAM Converter Options provides default and recommended preferences.

Continues on next page

3.5.1 Import

Import interface



xx1400002317

Operation Mergement in importing

When CNC code files are imported, the parser creates new operation meeting the rapid move instruction or tool change.

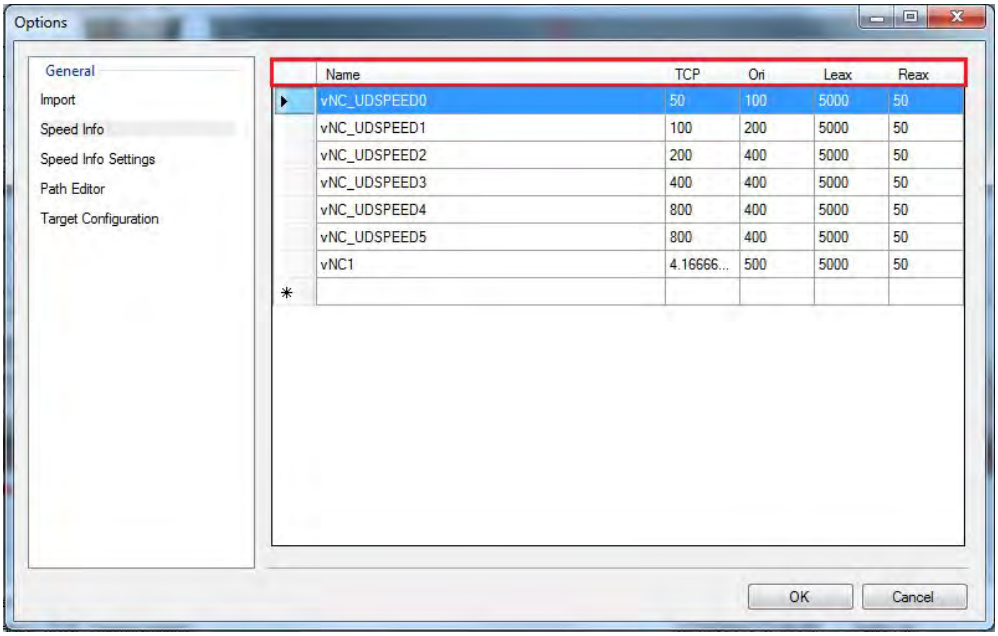
If you check the checkbox named **Merge operation during importing CNC code** and fill out a positive integer in text box, the new operation will not be created until the target number of the current operation is greater than the specified value.

Tool height compensation table

When CNC code parser encounters G43/G44 in parsing G-code files, it will refer to the tool height compensation table for z-axis compensation value.

3.5.2 Speed Info

Speed info settings



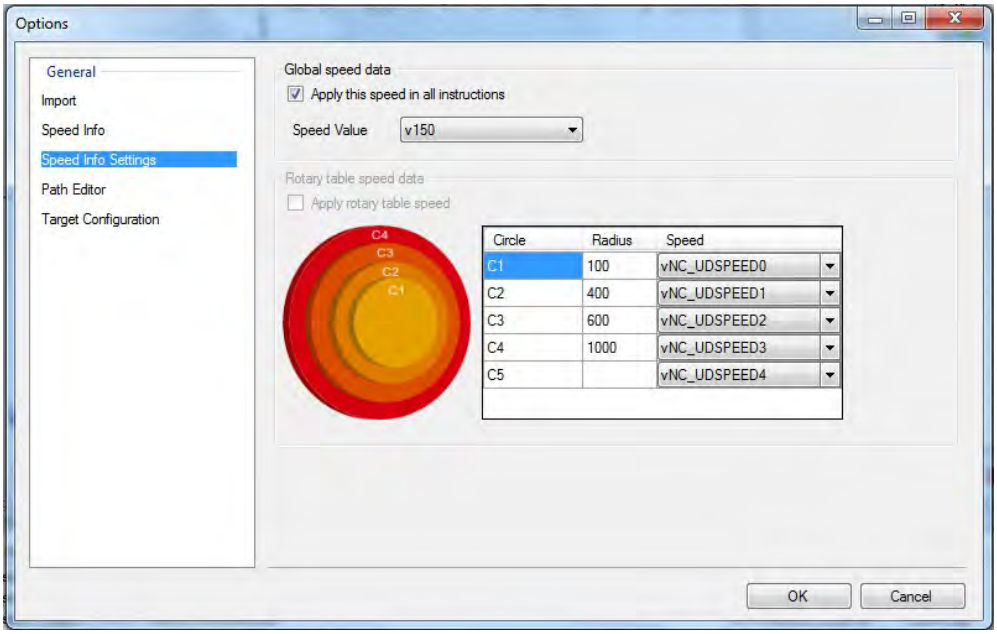
xx1400002318

There is a customized speed info table. You can add the speed info here and use it in **Speed Info Settings**.

Item	Description
Name	Speed info name.
TCP	The speed of the tool center point (TCP) in mm/s.
Ori	The reorientation speed of the TCP expressed in degrees/s.
Leax	The speed of linear external axes in mm/s.
Reax	The speed of rotating external axes in degrees/s.

3.5.3 Speed Info Settings

Speed Info Settings interface



xx1400002319

Global speed data

When **Apply this speed in all instructions** is checked, all the robot instruction in current project will use this speed info.

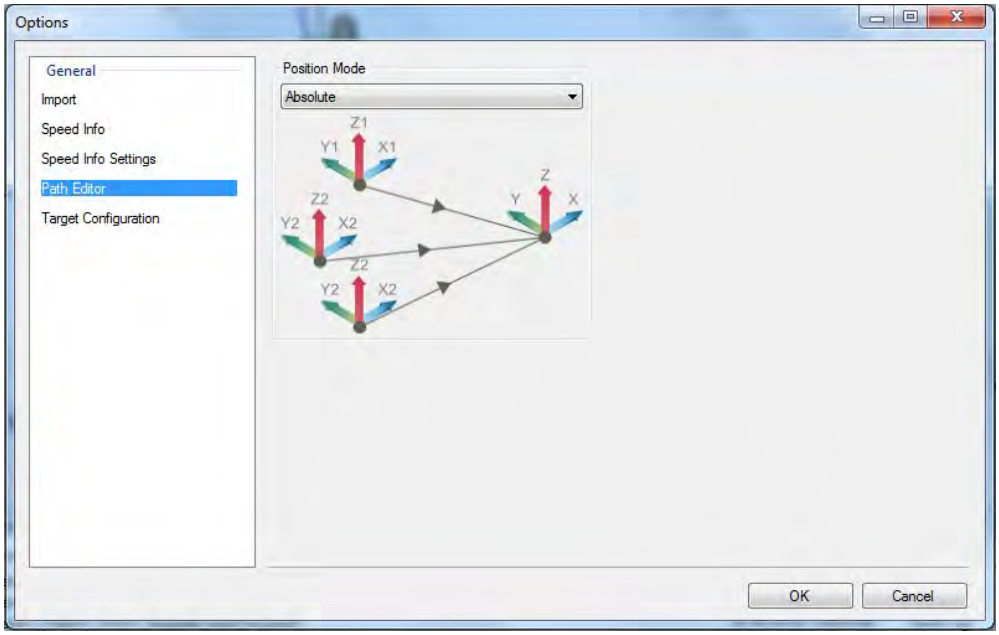
Rotary table speed data

According to the feature of rotary table, with same angular velocity, the position with bigger radius has bigger linear velocity. So we split the motion space into five region around the rotary axis: C1(inner circle), C2, C3, C4, C5(outside).

C1 using minimum speed, and C5 using maximum speed.

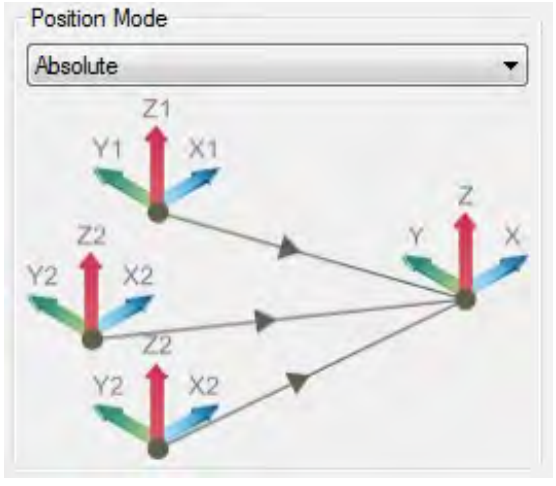
3.5.4 Path Editor

Path Editor

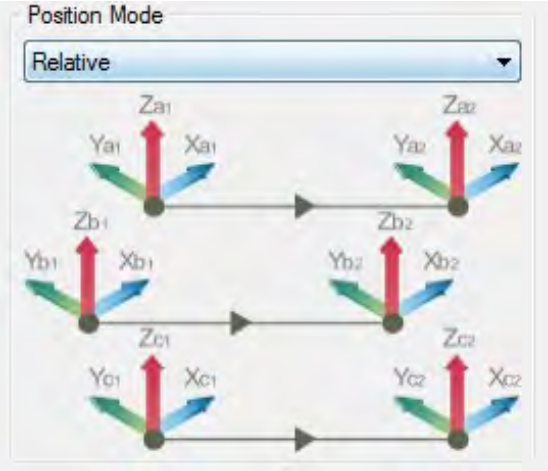


xx1400002320

When using the set position tool to change targets position and orientation, and the “World” coordinate system is selected, user can modify **Position Mode** for different effect.

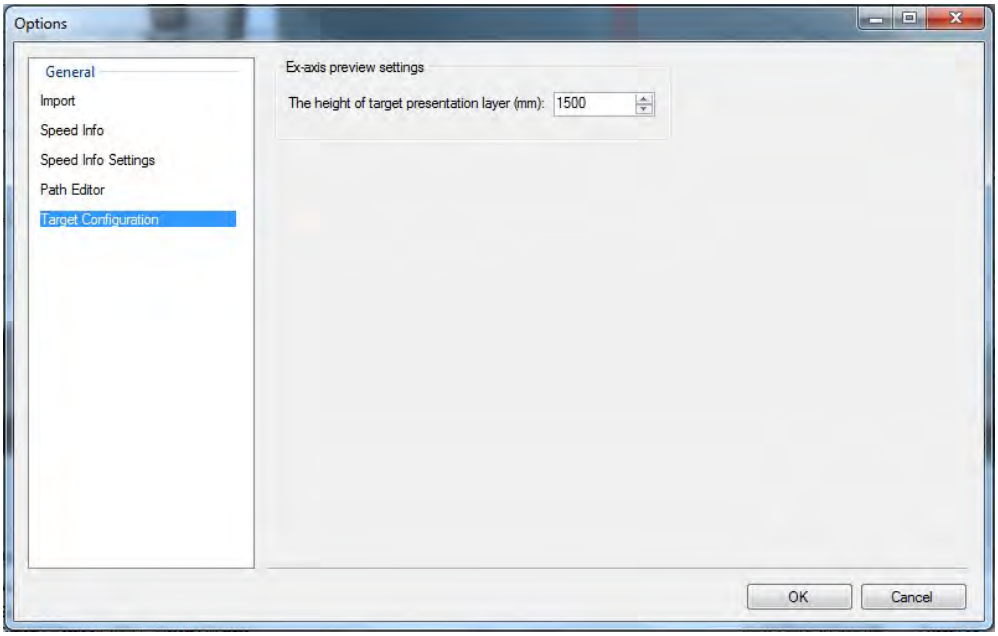
	Position mode	Illustration/Note
1	Absolute mode	 Position and orientation are absolute to world coordinate system. The selected targets will be changed to the same position and orientation.

Continues on next page

	Position mode	Illustration/Note
2	Relative mode	 The selected targets will be changed corresponding position offset or orientation offset relative to the world coordinate system.

3.5.5 Target Configuration

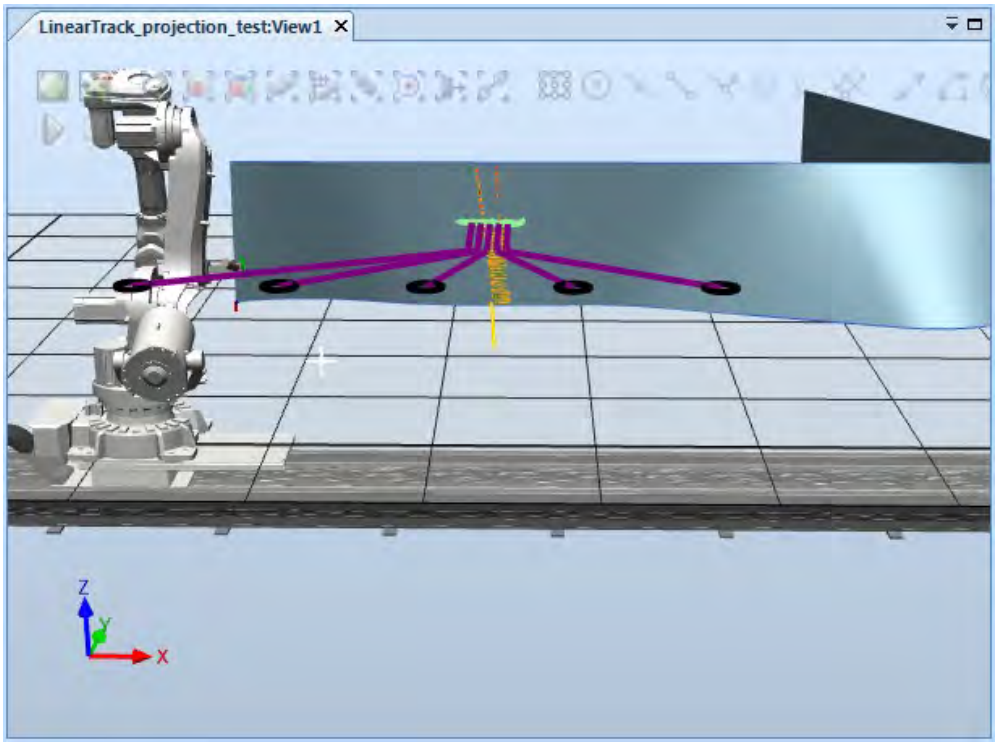
Target Configuration interface



xx1400002323

Ex-axis preview settings

The height of target presentation layer specifies the position of the linear track preview.



xx1400002326

4 Workflow for Machining PowerPac - CAM Converter

4.1 About the workflow

Overview

The following is a recommended workflow for working with Machining PowerPac - CAM Converter. After you complete the workflow, you can perform these tasks in any order.

Tools on the CAM Converter ribbon tab will be described in each section.

Note

The CAM Converter station (contains at least one robot system) should be set up in RobotStudio environment before starting Machining PowerPac - CAM Converter, as the Machining PowerPac - CAM Converter can only convert CNC machining paths to RAPID paths and cannot build stations.

Note

Refer to *Operating manual - RobotStudio* for detailed information on how to setup a station in RobotStudio.

Workflow of Machining PowerPac - CAM Converter

No.	Task	Description
1	Managing tools	Managing tools.
2	Managing converter station	Managing the converter station.
3	Importing	Importing 3 axis CNC code, 5 axis CNC code, APT Code, and CAD models.
4	Converting tool path to robot path	Converting tool path to robot path.
5	Simulating robot path	Simulating robot path and testing for collision.
6	Editing the generated robot path	Editing the generated robot path manually.
7	Exporting RAPID file	Exporting RAPID file.

4.2 Importing

Overview

Both CNC Codes and CAD models can be imported into the Machining PowerPac - CAM Converter. Supported CNC Code file extensions include .nc, .cls, and .apt. In addition, supported formats include 3 axis G-code, APT code and 5 axis G-code. While the 3 axis G-code and APT code share the same UI, 5 axis G code uses a separate import interface.

Note

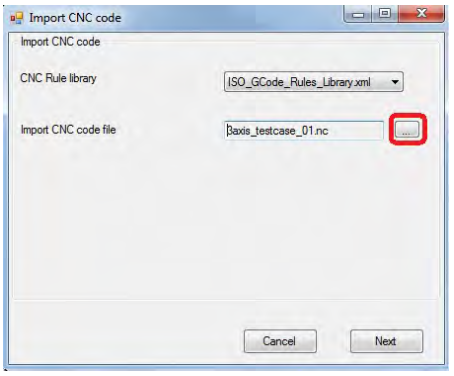
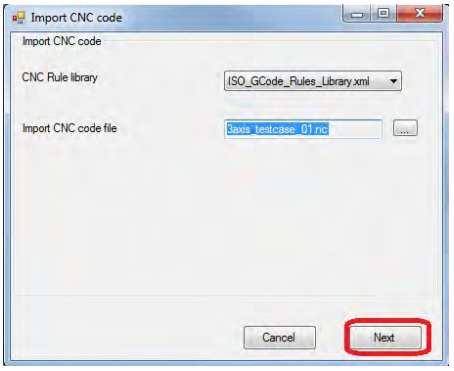
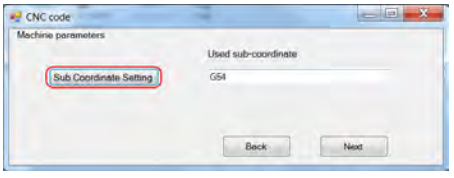
When parsing the CNC file, for an APT file, the parser logic is hard-coded, but for G-code files, the parser logic is based on the selected CNC code template.

Continues on next page

4.2.1 Importing CNC code - 3 axis and APT

Importing 3 axis

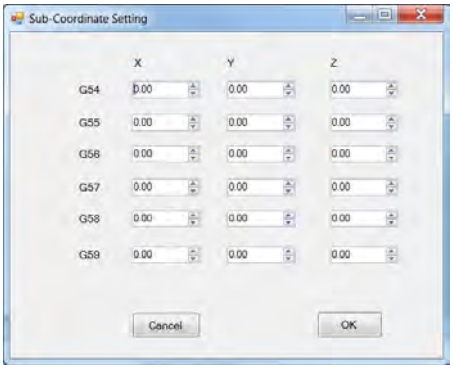
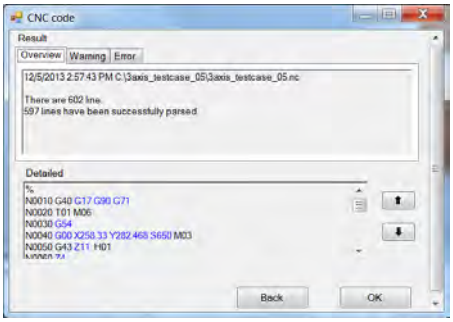
Use this procedure to import CNC code.

	Action	Illustration/Note
1	To import CNC code, click **Import** and select **Import CNC Code** in the CAM Converter ribbon tab. Alternatively, right click RobotGroup in the CAM Converter **Post Tree** and click **Import CNC Code**.	
2	A new window will appear with options to browse for the target CNC code file. Locate the **.nc** file and check it against a CNC rule library if necessary.	
3	Click **Next** to proceed or cancel import by closing this window.	
4	If the importing CNC code includes G54-G59 sub-coordinates, an extra window will appear. Click **Sub Coordinate Settings** button to edit the settings.	

Continues on next page

4.2.1 Importing CNC code - 3 axis and APT

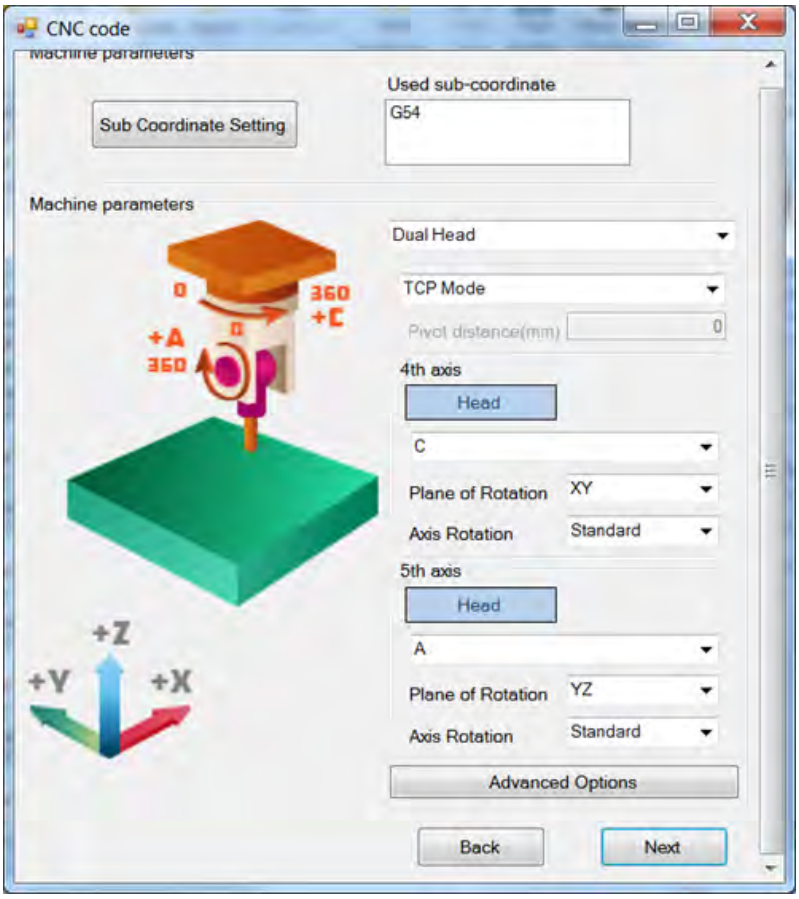
Continued

	Action	Illustration/Note
5	Edit the sub coordinate settings and click **OK** to exit this window.	
6	Click **Next** in the CNC code window to begin importing.	
7	Check result in the new **Result** window for warnings and errors. Press **OK** to confirm the import. Details of the parsed lines can be viewed in the **Detailed** textbox.	
8	The imported program group will appear in the Post Tree.	

4.2.2 Importing Process - 5 axis

Importing 5 axis

The procedure of importing 5 axis G-code is the same as importing normal CNC code except the import interface has more options. See *Importing CNC code - 3 axis and APT on page 41* for detailed information.



xx1400000128

Item	Description
Sub Coordinate Setting	Display the **Sub-Coordinate Setting** window when clicked.
Used sub-coordinate	Choose sub-coordinates from a template.
Dual Head drop-down list	Choose between dual head, dual table, or head table.
Mode drop-down list	Choose whether you want to enable TCP Mode. For non-TCP Mode, you need to enter Pivot distance in mm.
Axis 4	Select A, B or C axis. Axis 5 will change accordingly.
Axis 4 Plane of Rotation	Select plane of rotation around XY, ZX, YZ, or other planes.
Axis 4 Rotation	Select standard or reverse axis rotation.
Axis 5	Select A, B or C axis. Axis 4 will change accordingly.
Axis 5 Plane of Rotation	Select plane of rotation around XY, ZX, YZ, or other planes.

Continues on next page

4.2.2 Importing Process - 5 axis

Continued

Item	Description
Axis 5 Rotation	Select standard or reverse axis rotation.
Advanced Options	Display **Advanced Options** for 5-axis CNC G-code menu.

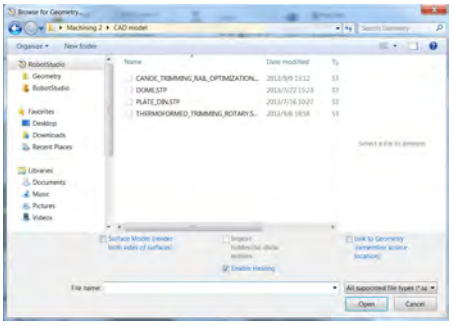
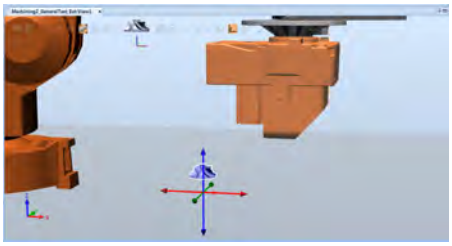
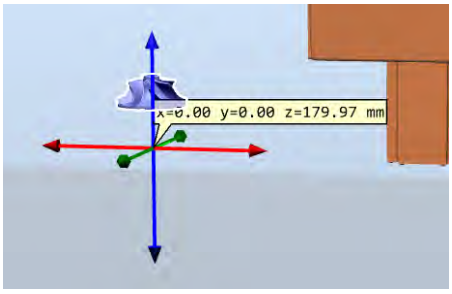
Use this procedure to import 5 axis CNC G-code.

	Action	Note
1	Import the 5 axis CNC G-code as importing the 3 axis CNC G-code. A new **CNC code import window** will appear.	See *Importing CNC code - 3 axis and APT on page 41*.
2	Select **Sub Coordinate Setting** and edit the parameters. Confirm with **OK**.	
3	Select type of CNC machine (double head, head table, or double table) and whether you want **TCP mode** in the drop-down list.	**Note** If NO TCP Mode is activated, enter **Pivot distance** in mm.
4	Select the **4th axis** from the drop-down list. Select **Plane of Rotation** and **Axis Rotation** for 4th axis.	
5	Select the **5th axis** from the drop-down list. Select **Plane of Rotation** and **Axis Rotation** for 5th axis.	
6	Click **Advanced Options**, select axis configuration for the 4th and 5th axis.	
7	Click **Next**.	
8	After confirming, click **OK** to import.	
9	The imported program group will appear in the Post Tree.	

4.2.3 Importing CAD model

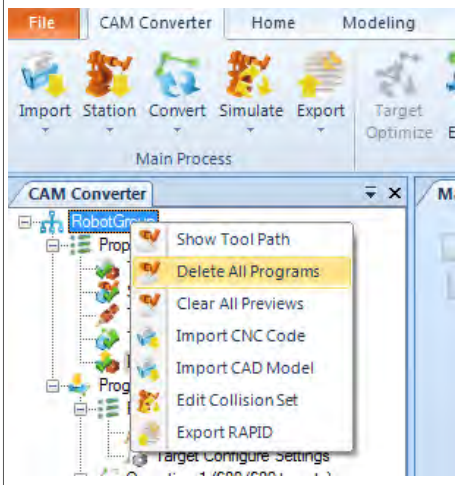
Procedure

Use this procedure to import a CAD model:

	Action	Note
1	Select the **Import CAD Model** in the **CAM Converter** ribbon tab or right click on the **RobotGroup** in the Post Tree and then select **Import CAD Model** to start the import.	
2	Locate your CAD file and click **Open**. The CAD Model will be imported into the CAM Converter. **Tip** Supported CAD Model file types: • ACIS files • IGES files • STEP files • VDAFS files • Pro ENGINEER files • Inventor files • Catia V4 files • Catia V5 files • VRML files • STL files • COLLADA files • Obj files • 3DS files • RSGFX files	
3	The imported CAD model is automatically selected upon import, and ready for further adjustment. The default position changed to (0, 0, 0).	
4	Drag the CAD model to change its position and orientation in the CAD Model window.	

4.2.4 Deleting imported data

Deleting method

	Action	Illustration/Note
1	After importing, the program is shown in the Post Tree. To remove programs from the Post Tree, right click the **RobotGroup**. On the pop-up menu, click **Delete All Programs** to delete all imported data.	

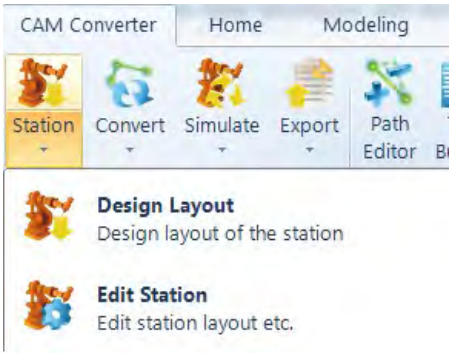
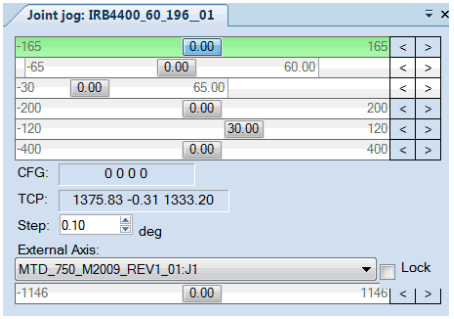
4.3 Managing station

4.3.1 Design layout

Overview

Design Layout is used for adding an object frame in addition to the normal user frame and becomes available only after import.

Procedure

	Action	Illustration/Note
1	Select **Design Layout** from the **Station** category in the ribbon tab to open the desgin layout window.	
2	Two new windows will display: **Joint jog** and **Design Layout** window.	
3	For detailed infomation of Joint jog, refer to *Operating manual - RobotStudio*.	
4	The design layout window will be described below. See *Design layout window on page 48*.	

Continues on next page

4.3.1 Design layout
Continued

Design layout window

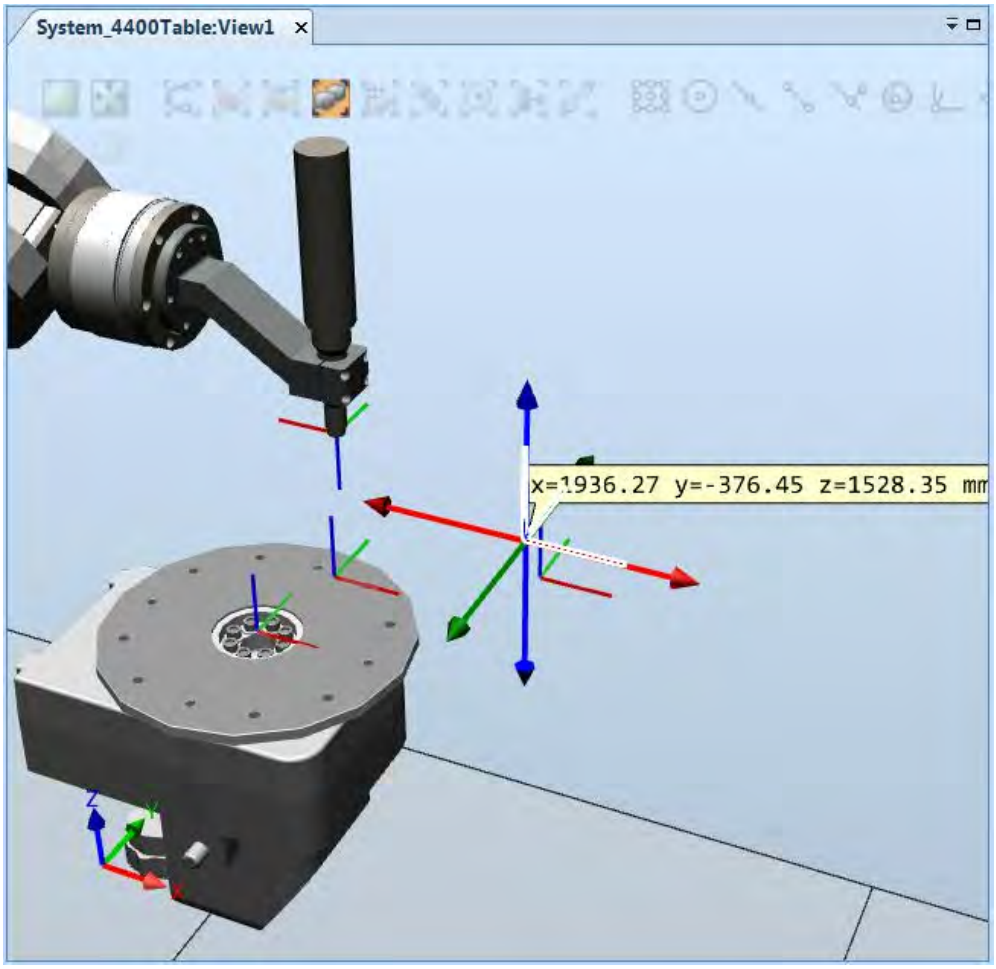
xx1400000769

No.	Item	Description
1	User Frame Alignment	Coordinate for aligning the user frame. Unit: mm.
2	User Frame Orientation	Orientation for aligning the user frame. Unit: deg.
3	ObjectFrame Alignment	Coordinate for aligning the object frame. Unit: mm.
4	ObjectFrame Orientation	Orientation for aligning the object frame. Unit: deg.

Continues on next page

Dragging the user frame in 3D graphics window

User frame can also be dragged directly using the controls in the 3D graphics window.



xx1400000770

Note

The object frame can only be set with the parameters in the **Design Layout** window.

Note

There might be delay of several seconds when dragging the coordinate.

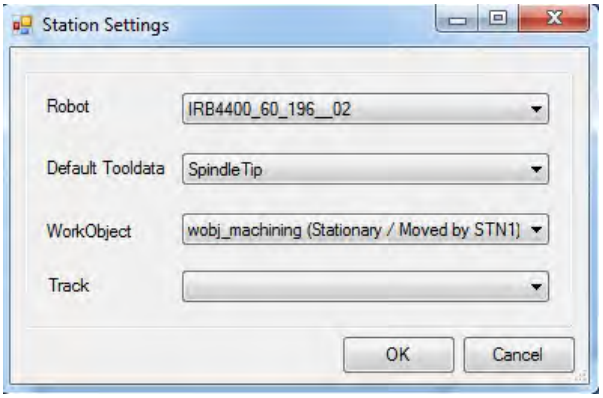
4.3.2 Edit station

Editing an existing station

Use this procedure to edit an existing station.

	Action	Note
1	Select **Edit Station** in the **Station** group from the **CAM Converter** ribbon tab.	
2	A **Station Settings** window will appear.	See *Read existing station window on page 50* for information of this window.
3	Select the options from the four drop-down lists and click **OK** to confirm your choice.	

Read existing station window



xx1400000129

Item	Description
Robot	Select Robot from the drop-down list for the existing station.
Default Tooldata	Select machining tool for the existing station.
WorkObject	Select WorkObject for the existing station.
Track	Show the Linear track used in the existing station.

4.4 Converting tool path to robot path

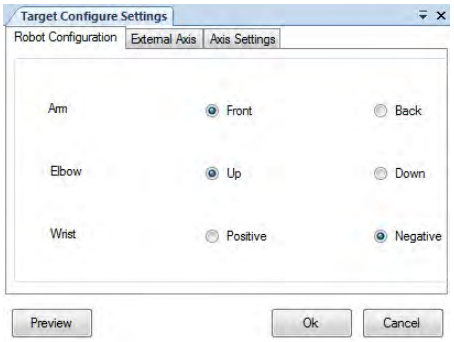
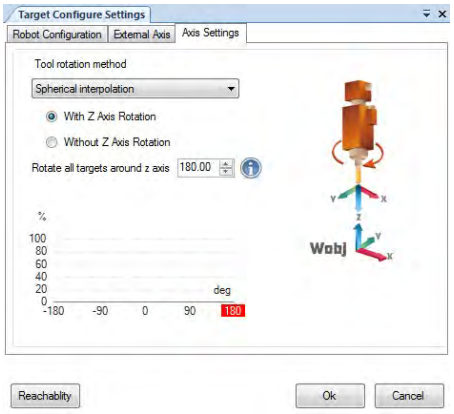
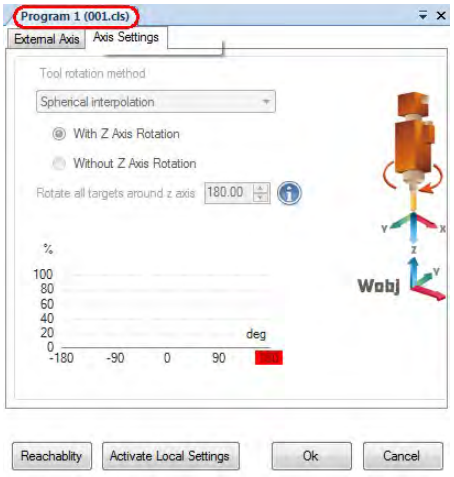
Overview

This section describes guidelines on converting tool path to robot path along with setting parameters for the conversion.

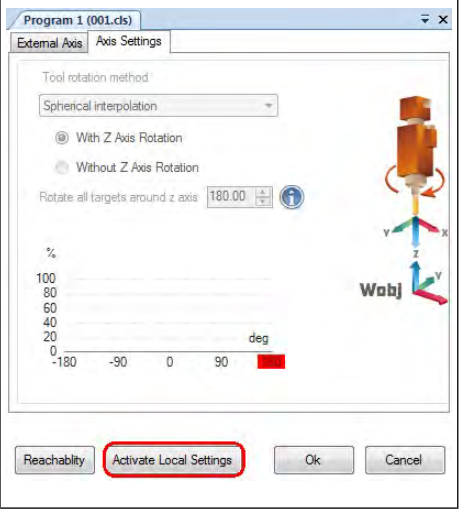
Continues on next page

4.4.1 Configuring a target

Start Target Configure Settings

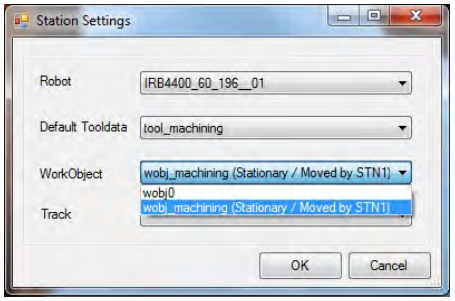
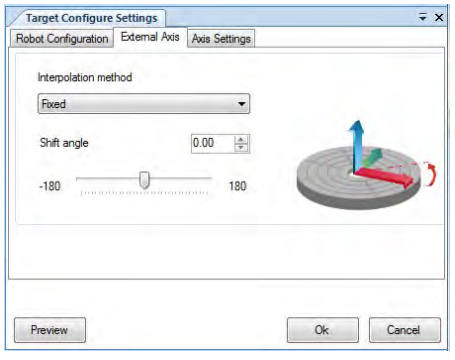
	Action	Illustration/Note
1	Select **Target Configure Setting** in the **Convert** group to start target configuration.	
2	Set the configuration for the robot in the **Robot Configuration** tab: • **Arm**: Sets arm position to Front or Behind. • **Elbow**: Sets elbow position to Up or Down. • **Wrist**: Sets wrist position to Positive or Negative.	
3	Set axis settings in **Axis Settings** tab.	
4	Alternatively, you can access target configure from the Post Tree by right clicking any Target Configure Settings under properties and clicking **Edit**. The local target configure settings has different window title that indicates the specific program group and operation.	

Continues on next page

	Action	Illustration/Note
5	Note: In order to configure a local Target Configure Setting, you will need to first activate it by clicking Active Local Setting.	

Settings with external axis

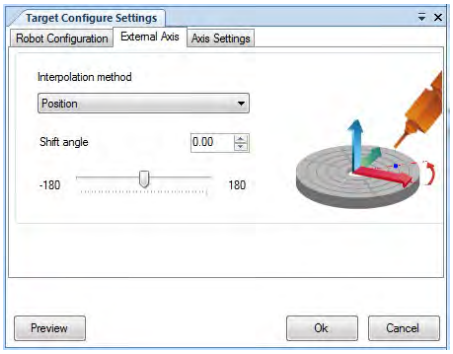
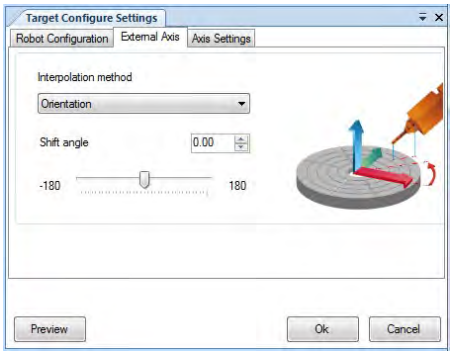
Positioner

	Action	Illustration/Note
1	Use **Station** -> **Edit station**, then select the **WorkObject** that moved by positioner.	
2	Select the **External Axis** on the second tab. Set the **Interpolation method**. The first option is **Fixed**. When Fixed is chosen as the interpolation method, the external axis spins alongside the robot arm. **Shift angle**: Set Shift angle.	

Continues on next page

4.4.1 Configuring a target

Continued

	Action	Illustration/Note
3	The second option is **Position**. When Position is chosen as the Interpolation method, the robot arm is set to only move along the set red axis which can be previewed with the **Preview** button.	
4	The third option is **Orientation**. Then set the **Shift angle**. When Orientation is chosen as the Interpolation method, the robot arm can freely move along any parallel axis of the fixed red axis. This setting can be previewed with the **Preview** button.	
5	Confirm your choice by clicking **OK**.	

Linear Track

Linear track component make the robot move along with the linear track and handle the large scale work object. It interpolates the whole movement into robot and linear track and ensure they can work together. There are three types of motion modes for linear track:

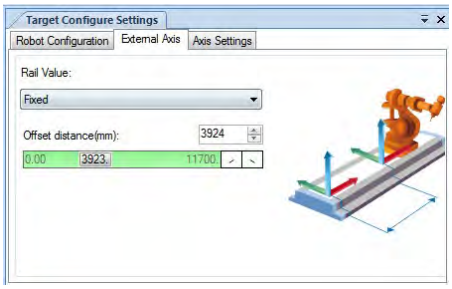
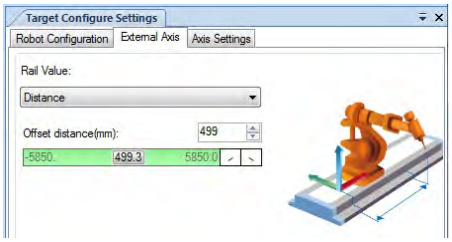
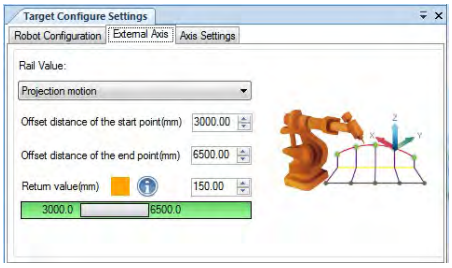
- Fixed

  The robot base is in fixed position on the linear track.
- Distance

  The distance between the robot base and robot target in linear track direction is fixed.
- Projection

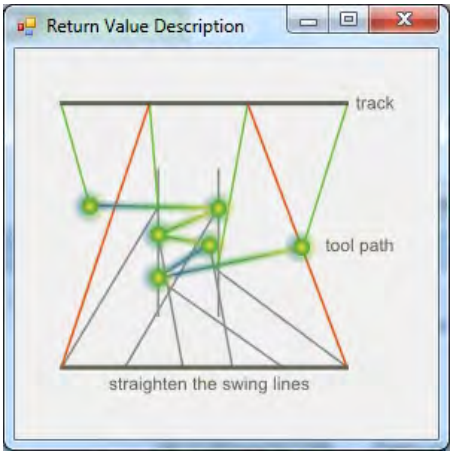
  When the robot path swings forward and backward, users can use this strategy to prevent the swing motion of robot on linear track.

Continues on next page

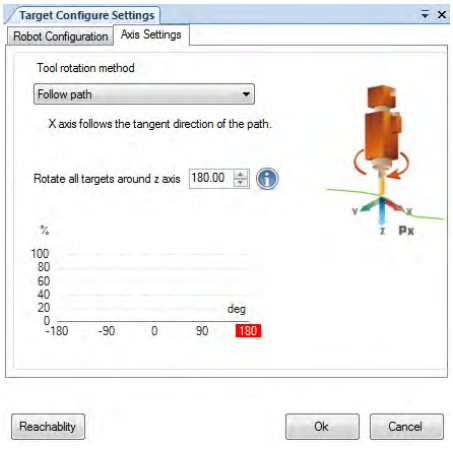
	Action	Illustration/Note
1	Use **Station** -> **Edit station**, then select the **linear track**.	
2	Select the **External Axis** on the second tab. Set the **Rall Value**. The first option is **Fixed**. Drag the slider's motion cursor or fill out the offset distance to change the distance between fixed position and linear track origin.	
3	The second option is **Distance**. Drag the slider's motion cursor or fill out the offset distance to change the distance between the robot base and robot target in linear track direction.	
4	The third option is **Projection**.Resize the slider's cursor to change the offset distance region of the robot base on linear track.	

Continues on next page

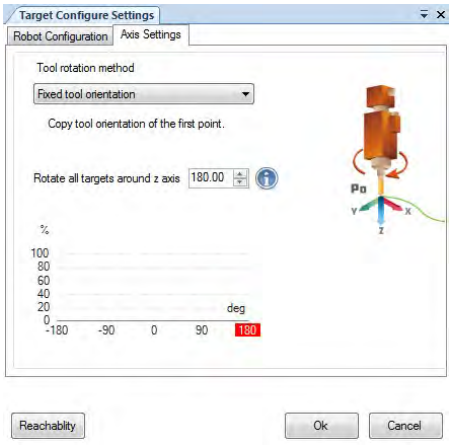
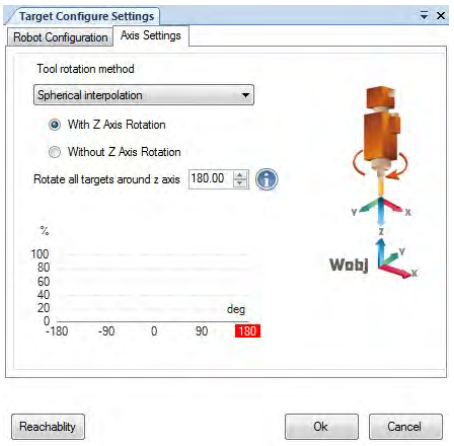
4.4.1 Configuring a target
Continued

	Action	Illustration/Note
5	The orange block following **Return value Description** represents that the swing robot targets in preview is in orange color. If you hover the mouse on **info** icon, the picture will show to depict the principle of projection mode.	

Editing Axis Settings

	Action	Illustration/Note
1	Select the **Axis settings** on the second tab. There are three types of Tool rotation method options: • Follow path • Fixed tool orientation • Spherical interpolation	
2	**Follow path**. It has no other settings.	

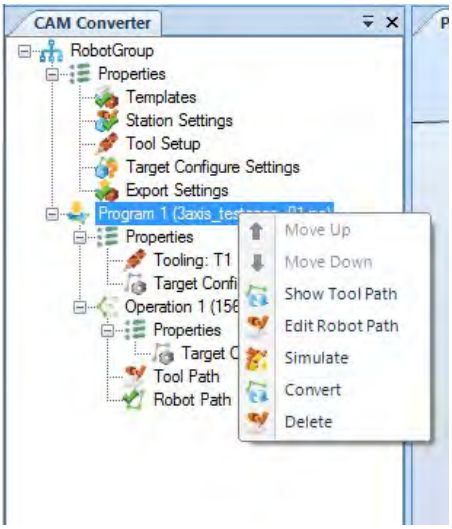
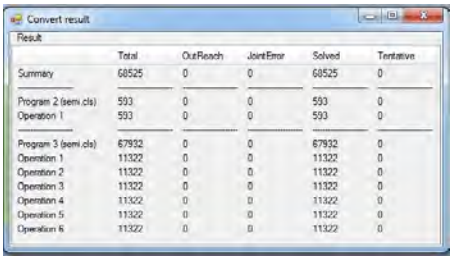
Continues on next page

	Action	Illustration/Note
3	**Fixed tool orientation**. It has no other settings.	
4	**Spherical interpolation**. It has one more setting: • With Z Axis Rotation • No Z Axis Rotation	
5	Confirm your choice by clicking **OK**.	

4.4.2 Converting

Procedure

> **Note**
>
> Import a valid CNC code before converting to make it work.

	Action	Note
1	On the **CAM Converter** ribbon tab, select **Convert** in the **Convert** group to start converting.	In the **Post Tree**, you can also right click on a program group and select **Convert** from the pop-up menu. 
2	The **Convert** window will appear. It shows the conversion result.	
3	Click **OK** to proceed.	

4.5 Simulating robot path

Overview

After conversion, run simulation of the imported file to ensure the program run properly on the robot controller. See *Converting tool path to robot path on page 51* for detailed information.

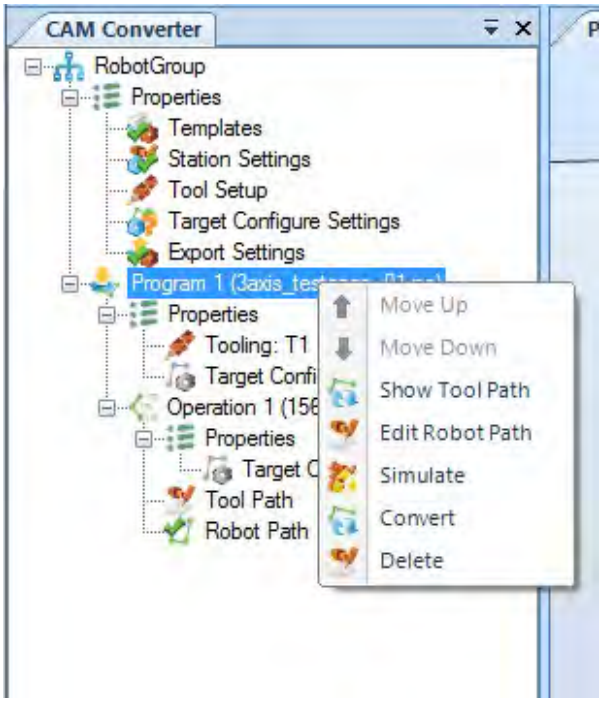
Continues on next page

4.5.1 Simulation window

Simulation window overview

Select **Simulation** -> **Simulate** in the **CAM Converter** ribbon tab to start simulation. A **simulation control** window and a **result** window will appear around the 3D graphics window.

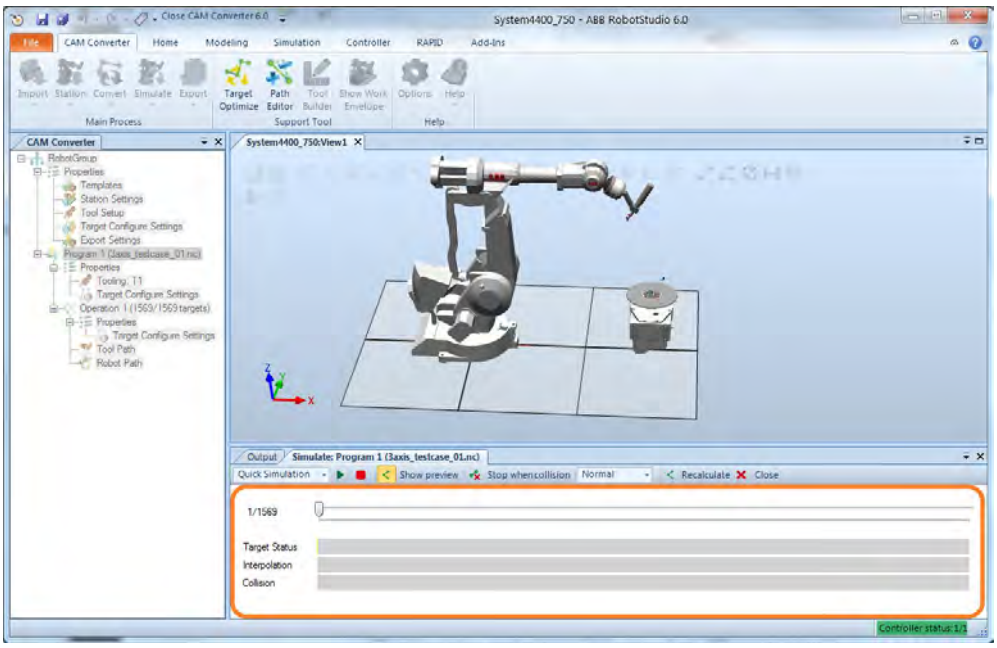
In the **Post Tree** view, right click a program group and then click **Simulate** to start a simulation of the specific program group.



xx1400000135

Continues on next page

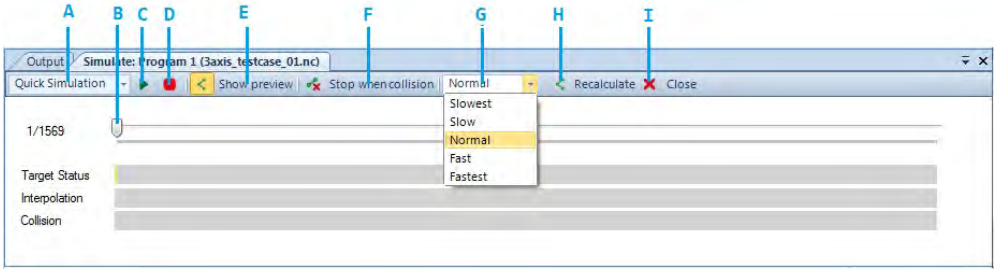
A separate simulation control panel appears at the bottom of the screen and a result window appears to the right of the 3D graphics window.



xx1400000138

Simulation controls

The simulation control panel consists of ten different elements.



xx1400000139

	Control	Description
A	Simulation method selection	Selects **Quick Simulation** for running with quick simulation. The Quick Simulation system is based on algorithms set as default by RobotStudio and cannot be modified in any way whereas running quick simulation allows customized calculation methods. **Note** Running with VC will be more precise but take more time in average.
B	Result toggle	The result toggle can be used to browse through parsed path points by dragging it along the horizontal axis.
C	Start/Pause	

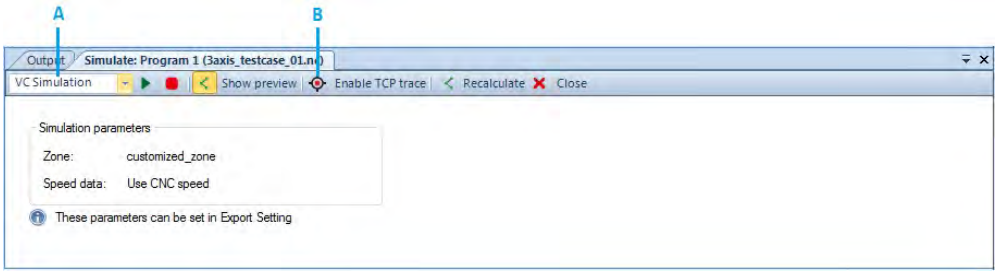
Continues on next page

4.5.1 Simulation window
Continued

	Control	Description
D	Stop	
E	Show Preview	Shows robot path when running simulation.
F	Stop when collision	Stops the simulation when collision occurs. This can only function when running Quick Simulation.
G	Speed selection	
H	Recalculate	When the robot path is modified by path editor or path optimization, the robot path need to be recalculated.
I	Close	Uses the X icon to exit simulation mode.

Simulation with a visual controller

Uncheck the **Quick Simulation** checkbox in the control panel, the simulation will run with a visual controller. **Enable TCP trace** button and **Show preview** checkbox will appear on the control panel.



xx1400000177

	Control	Description
A	Simulation method selection	Selects **VC Simulation** for running VC simulation.
B	Enable TCP trace	Enables tool path tracing. When this option is checked, the tool will leave a trace behind.

Note

Make sure the controller is ready before Quick Simulation.

If the controller is not ready, or the controller do not load the right RAPID file, there will be errors.

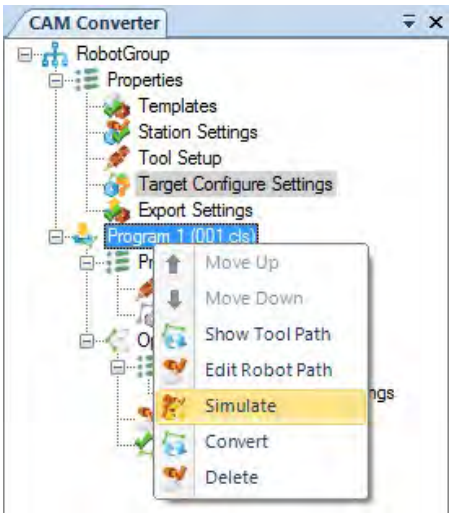
4.5.2 Simulating robot path

Procedure

After converting, use this procedure to simulate a robot path.

Note

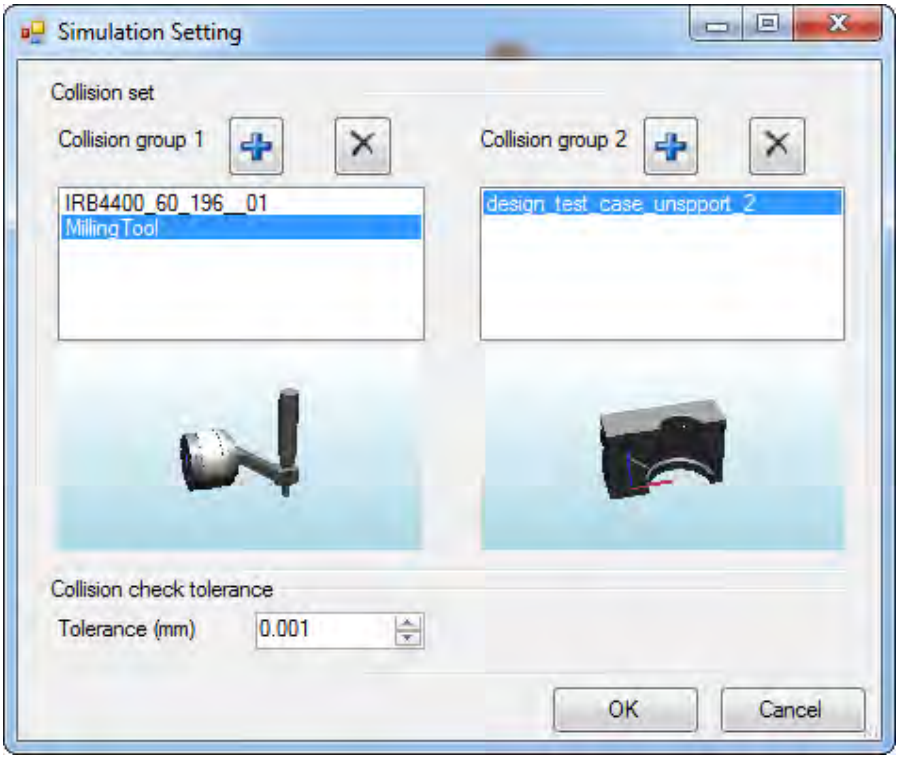
For simulation to work, you need to have a program group selected in the post tree.

	Action	Illustration/Note
1	Select a program group in the post tree, then select **Simulation** -> **Simulate** in the CAM Converter ribbon tab to start a simulation.	**Note** If clicking Simulate but without a program group selected, a message window will pop up: Please select a program group.
2	Alternatively you can right click a program group and select **Simulate Program Group** in the context menu.	
3	In the simulation control panel, click **Start** to start the simulation. The simulation will run at the given speed.	See *Simulation controls on page 61* for reference.
4	Click **Pause** button to pause the simulation, and click **Stop** to stop it. Resume a simulation by clicking **Start** button again.	
5	Use the **Result Toggle** in the simulation control panel to toggle among path points.	
6	To exit simulation mode, click X on the top-right conner of the simulate window or click **Close**.	

4.5.3 Creating collision set

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 a station, but each collision set can only contain two groups.



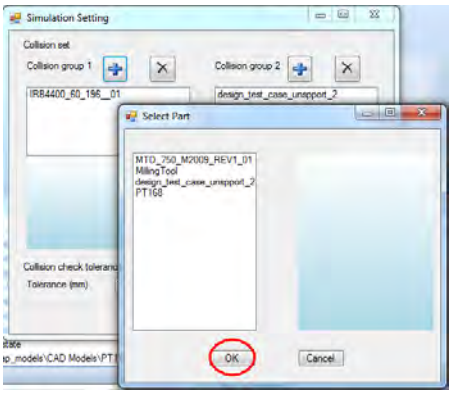
xx1400000144

Group	Item	Description
Collision Set	Collision group 1 and 2	The collision set component A and B.
	3D graphic view window	The selected parts in collision set are displayed in these windows.
Collision check toleration	Tolerance (mm)	For some pressure processes, the collision between the tool and part is allowed. In this case, you can specify a tolerance value so that the system will ignore all small collision checks within the tolerance during the simulation. You can focus on those poor collision checks.

Continues on next page

Procedure to create collision set

Use this procedure to create a collision set.

	Action	Illustration/Note
1	Click **Edit Collision Set** in the **Simulation** group to create a collision set in the Layout browser.	
2	Click the button **+** to expand the collision set.	
3	A new window **Select Part** will display. Choose the objects you want to use for collision testing by marking it with your mouse. Proceed by clicking **OK**. You can also add multiple objects to check for collision.	
4	Repeat step 3 for every object you want to check for collision.	

4.6 Path Editor

4.6.1 About the path editor

Overview

The Path Editor is used to manually edit path points after generated. In order to run Path Editor,user may run into path editor in both of the following cases.

- Path edit without simulation
- Path edit with simulation

From the simulation result window, pick a path point to edit. See *Simulating robot path on page 59* for more information on simulation.

The Path Editor has these features:

- Show the position and rotation based on the active workobject.
- Show how current robot joints.

4.6.2 Entering Path Editor

Entering methods

After conversion, users can change the robot instruction in path editor. To enter the path editor, select **Path Editor** from the **CAM Converter** ribbon tab. In the **Post Tree**, you can also right click on a operation group and select **Edit Robot path** from the pop-up menu.

xx1400000175

4.6.3 Working with instruction list

Icons and states in instruction lis

An instruction list view appears at the right of the 3D graphics window.There are four columns: **State** / **Instruction** / **Motion** / **Speed**.

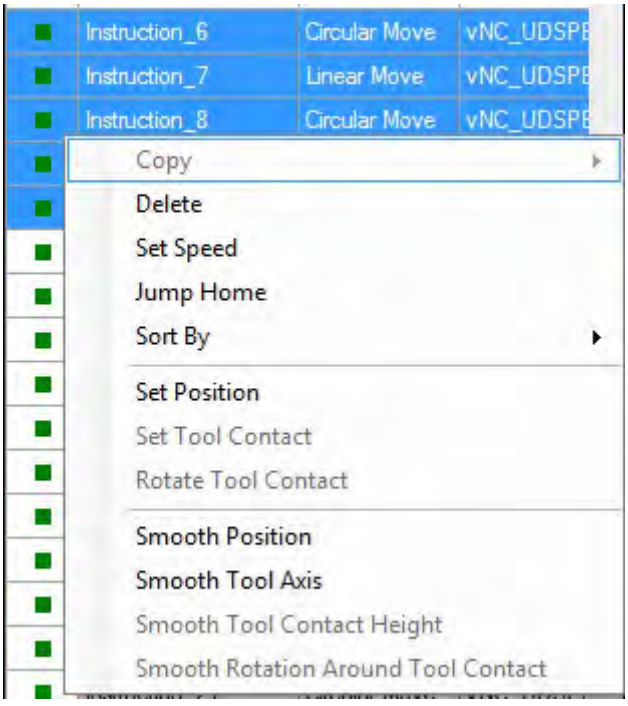
Every row in list view body represents an instruction node.

The robot target has 4 state.

Color	Description	Error Type
Red	Error	Singularity / Out of range / outside reach
Yellow	Warning	Large joint change / Wrist flip
Green	Normal	None error
Blue	Dynamic Error	Collision

Context menu for instruction nodes

Instruction nodes in the list can be selected for operating more functions. It is also possible to select multiple nodes using **Ctrl** and **Shift**. Right-clicking the selected node(s) in the **Instructions** window would display a context menu for target. The following figure shows the available functions.



xx1500000384

Continues on next page

	Menu Item	Description
1	Copy/Insert Above	Copies the selected robot instruction node and inserts above. **Note** The copy menu item is enabled when only one instruction selected. When user selects more than one instruction, the menu item will be disabled.
2	Copy/Insert Below	Copies the selected robot instruction node and inserts below. **Note** The copy menu item is enabled when only one instruction selected. When user selects more than one instruction, the menu item will be disabled.
3	Delete	Deletes the selected instruction(s).
4	Convert to Joint/Linear[i]	Converts the motion type of the selected instruction, joint movement or linear movement. For a joint target, the robot performs a joint movement from the previous target; otherwise, the robot runs in linear.
5	Set Speed	The **Set Speed** window will be shown in the left pane. Modifies the speed data of the target.
6	Jump Home	When the robot hides the robot path preview, users can click this menu item to make the robot jump to home position (zero position).
7	Sort By/Instruction Sequence	Lists the instruction nodes in the original instruction sequence.
8	Sort By/Speed Data	Lists the instruction nodes based on the speed data. If nodes are sorted by speed data, only **Set Speed** will be available in the context menu.
9	Set Position	The **Set Position** window will be shown in the left pane. Path editor reuses the set position tool of RobotStudio, so users can operate it easy. Sets the position and orientation based on the specified reference frame for the selected instruction. If multiple instructions have been selected, the position and orientation of the first instruction need to be set and other instructions will have a relative displacement.
10	Set Tool Contact	This command is unavailable for now.
11	Rotate Tool Contact	This command is unavailable for now.
12	Smooth Position[ii]	The **Set Position Smooth** window will be shown in the left pane. Selects one or more targets between a start and end target to adjust target positions for obtain a smooth position change. For details, see *Smoothing position on page 73*.
13	Smooth Tool Axis[ii]	The **Smooth Tool Axis** window will be shown in the left pane. Adjusts the tool axis of one or more targets to obtain a smooth tool axis change between targets. For details, see *Smoothing tool axis on page 74*.
14	Smooth Tool Contact Height	This command is unavailable for now.

Continues on next page

4.6.3 Working with instruction list
Continued

	Menu Item	Description
15	Smooth Rotation Around Tool Contact	This command is unavailable for now.

i Available only for one-node selection.
ii Available only when three or more instruction nodes are selected.

Path player

xx1400002313

Figure 4.1:

It looks like a simple audio player, but it plays with the robot instruction nodes.

1. Click the **Start** button (green) to start the player

   When the player starts, the button will be changed to the “Pause” icon, the selected node in list view will change one by one quickly, and the simulation tool will move to the first target of the selected node.
2. Click the **Pause** button

   The quick preview will be paused.
3. Click the **Stop** button (red block) to stop the player

   The quick preview will be stopped and the cursor will be reset to zero-position.

When mouse hover on the information icon, some tips on path editor will be shown.

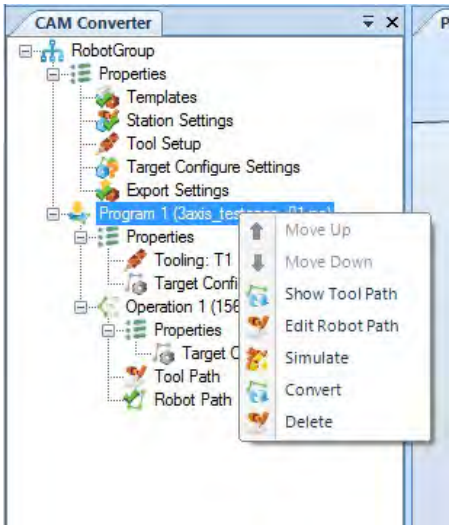
4.6.4 Editing a path

Procedure

After converting, use this procedure to edit a robot path.

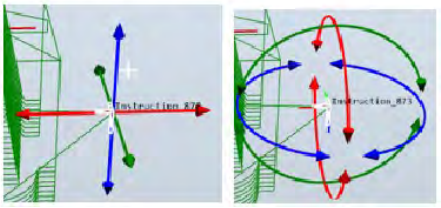
Note

For path editor to work, you need to have a program group selected in the post tree.

	Action	Illustration/Note
1	Select a program group in the post tree, then select **Path Editor** in the **CAM Converter** ribbon tab to open a path editor.	**Note** If clicking Path Editor but without a program group selected, a message window will pop up: Please select a program group.
2	Alternatively you can right click a program group and select **Edit Robot Path** in the context menu.	
3	Select instruction nodes in the instruction list view. Left-click mouse device to select one node. Multi-select by left-clicking and dragging mouse device or left-clicking mouse device with **Ctrl**/**Shift** pressed.	
4	Right-click on selected instruction nodes and select required command from the shortcut context menu.	

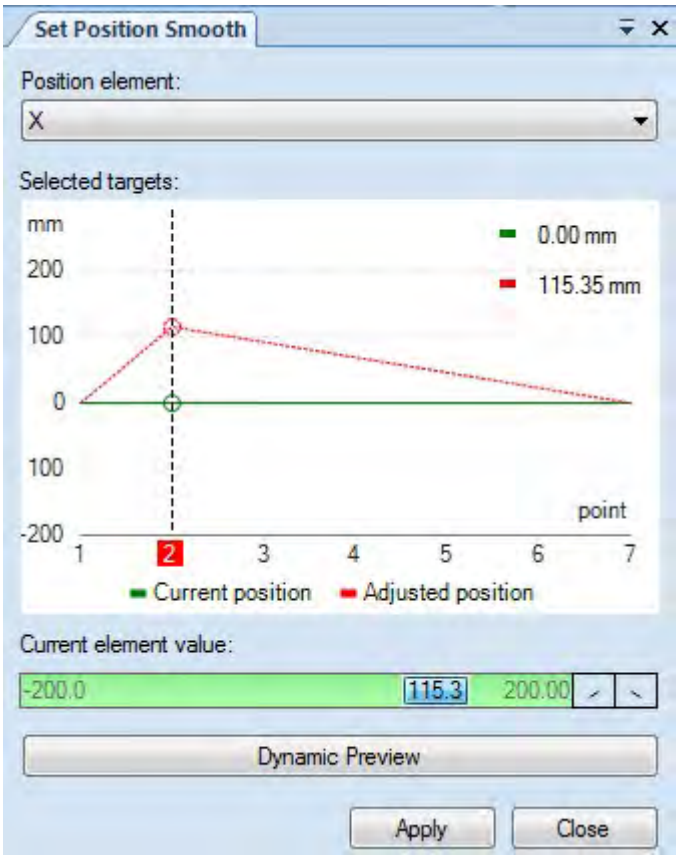
Continues on next page

4.6.4 Editing a path
Continued

	Action	Illustration/Note
5	The settings will be displayed in real time in the 3D graphics window or you can have a dynamic preview for some certain commands.	**Note** If you change the reference coordinate system to **World**, change the position or orientation then apply the modification, all the selected targets will be same in position and orientation. Except you change the **Position method** to **Relative** in **Options** > **Path Editor** in CAM Converter ribbon tab.
6	Alternatively, you can modify targets by freehand tool. Toggle on the **Move** or **Rotation** item in Freehand ribbon group. Select the expected reference coordinate system. Drag the linear arrows to change the targets' position, drag the arc arrows to change the targets' orientation.	  **Note** If **World** reference coordinate system is selected, the rotation tool is positioned at the origin of the reference coordinate system.

Continues on next page

Smoothing position



xx1500000194

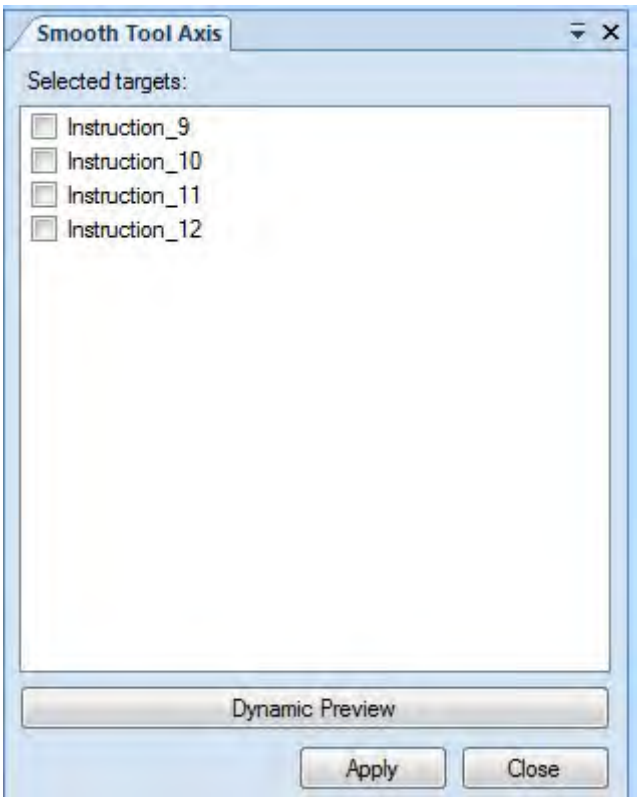
Positions of several targets can be adjusted based on the X/Y/Z axis of either original position or orientation to obtain a smooth position change between targets. If the contact height of a target changes, the target becomes a key target being circled. The key target can also be converted to a normal target by right-clicking the circle and choosing **Delete Key Node**. Note that modified value of the target keeps until the setting is all done.

The operation for smoothing position is similar to that for setting tool contact. Before applying the adjustment, use **Dynamic Preview** to preview the settings. A target can deviate from its original position by a maximum of 200 mm.

Continues on next page

4.6.4 Editing a path
Continued

Smoothing tool axis



xx1500000195

Check boxes are available for selecting one or more targets to obtain tool axis smoothness. Only tool axes of unselected targets will change to ensure smooth path from one selected target to its next selected target. However, the tool axes of the first and last targets never change. **Dynamic Preview** is also available.

4.7 Target Optimizing

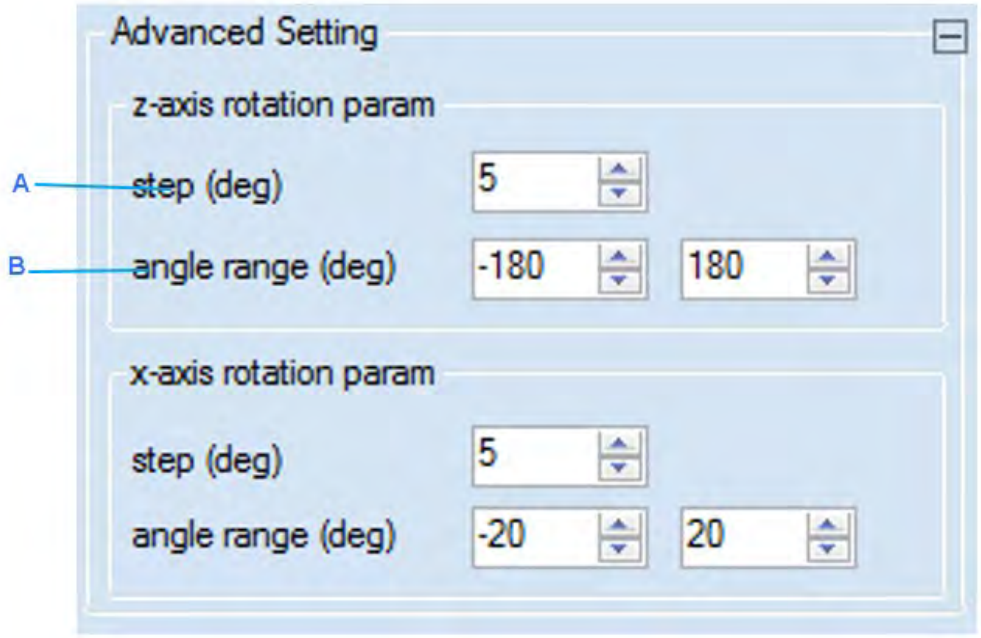
Overview

Target optimization will automatically rotate the robot targets to find error free solution. This operation only modifies the orientation of targets.

> **Note**
>
> The Target Optimize function is only active when in simulation mode.

Advanced setting

The advanced setting is default collapsed. You can modify the setting for specific requirement.



xx1400002315

Figure 4.2:

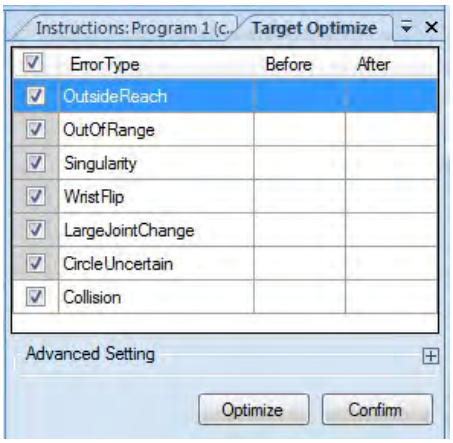
	Control	Description
A	Step	Optimization tries to find the error-free solution step by step. Step specifies the step value that changed every time.
B	Angle range	Axis angle range limitation of robot targets

Procedure

Use this procedure to target optimizing.

	Action	Illustration/Note
1	Select a program group and open simulation.	

Continues on next page

	Action	Illustration/Note
2	Click the **Target Optimize** in **CAM Converter** ribbon tab.	 xx1400002316
3	Toggle on or off the **Collision** check.	**Note** Currently, the other error types cannot be unchecked. They are always validated in optimization.
4	Click Optimize button to start the optimization and the result will be previewed in the table above. **Before** column displays the number of error targets before optimization. **After** column displays the number of error targets after optimization.	**Note** After click Optimize, the result will not be apply into robot targets automatically.
5	Click Confirm button to apply the optimization result into robot targets.	
6	Close target optimize window to exit this function.	The robot targets will be updated. And the process bar in simulation, the list view in path editor and the robot path will all be updated.

4.8 Exporting RAPID file

Overview

When the simulation is running well in CAM Converter, you can export your work as a RAPID program, and later download it to a robot controller to run.

> **Note**
>
> In order to export, convert the imported path first!

In the CAM Converter ribbon tab, the **Export** group has three funtions: **Export Template**, **Export Settings**, and **Export RAPID**.

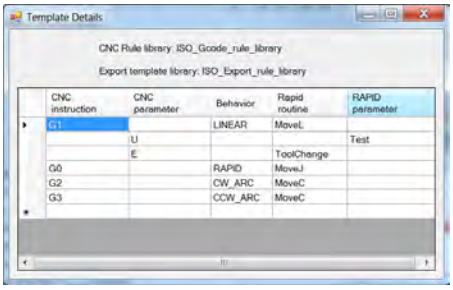
In the Post Tree, right click **RobotGroup** under the RobotGroup node and select **Export RAPID** to use this function.

Continues on next page

4.8.1 Exporting Template

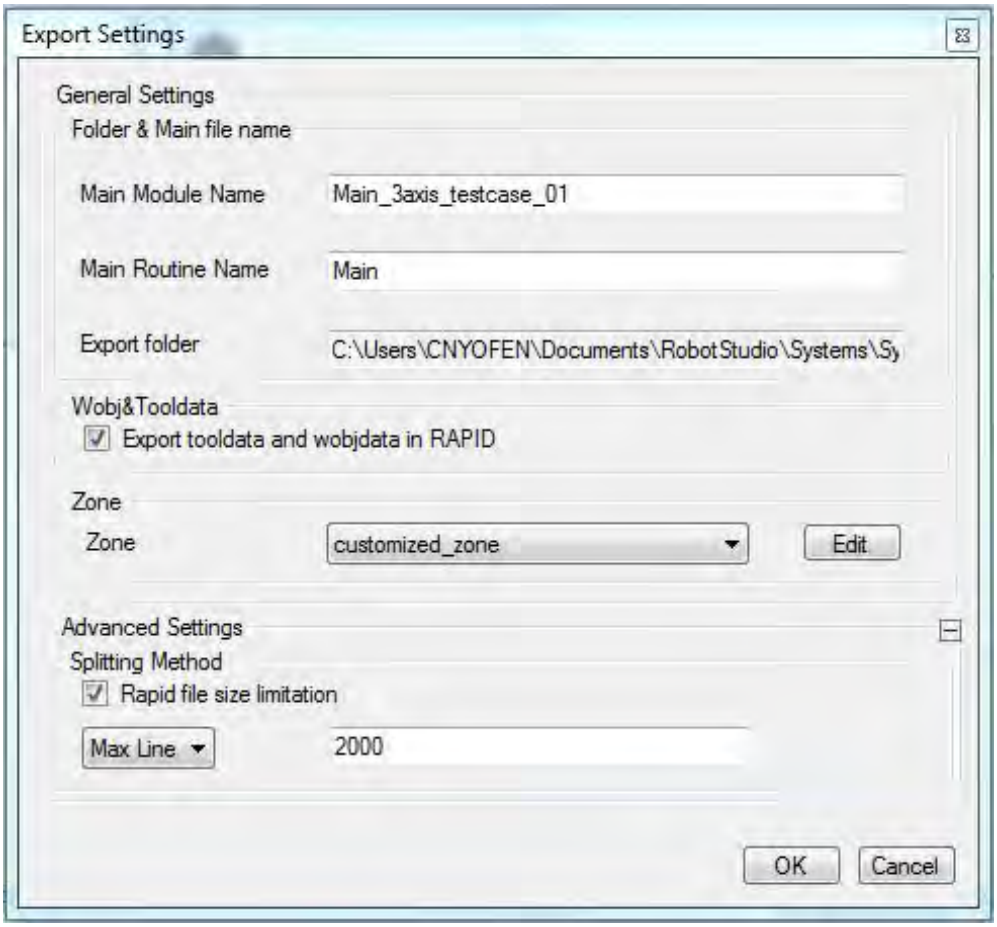
Procedure

Use this procedure to export a template:

	Action	Illustration/Note
1	Select **Export Template** in the **Export** group.	
2	An **Export template** window is displayed. In this window, select an active template from the drop-down list. Click the **Details** button to see its details. Proceed by clicking **OK**.	
3	Click the **Details** button, then a **Template Details** window is displayed. The template consists of CNC instruction, CNC parameter, behavior, rapid routine, and RAPID parameter.	
4	Close this window to exit.	

4.8.2 Export Settings

Export Settings window



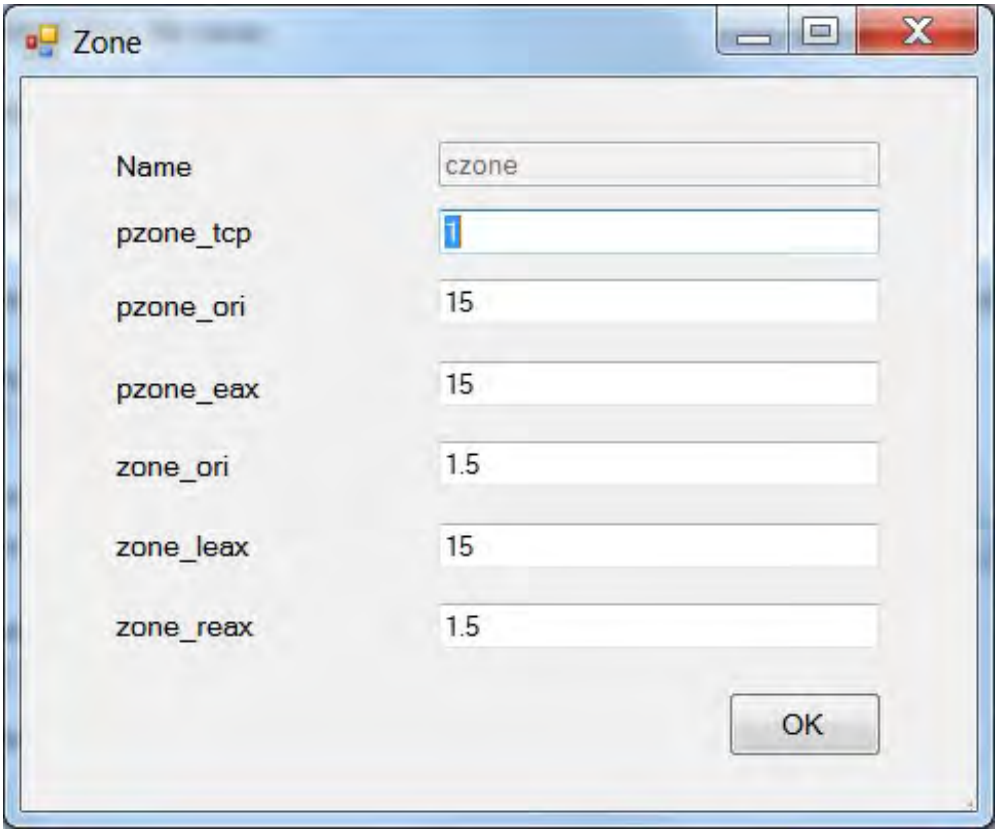
xx1400000148

Item	Description
Main Module Name	Enter a desired module name in the textbox.
Main Routine Name	Enter a desired routine name in the textbox. The output file will be named as: User inputted Routine Name.
Wobj& Tooldata and workobject	Select Export tooldata and workobject in RIPID when checked.
Zone	Select a zone. Click the drop-down list, select among z0, z1, z5, z10, z15, z20, z30, z40, z50, z60, z80, z100, z150, z200, and any user-defined zone. Each number behind z corresponds to the TCP path in mm, except z0, which is set at 0.3 mm. For details on the values set for different zones, see zonedata in RobotStudio manual.
Advance Settings/Splitting method	The export function will create some rapid codes which will be stored in some files, so this split method means the limit for this file, They are two ways to limit the file size, one is maxium code line limit and another is file size limit.

Continues on next page

4.8.2 Export Settings
Continued

New zone window



xx1400000149

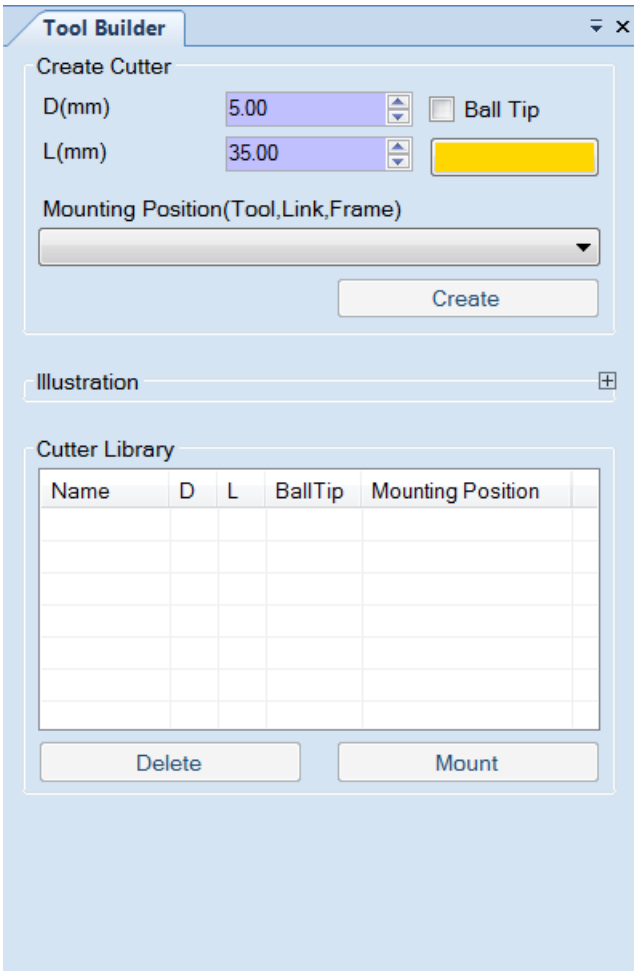
Item	Description
Name	Zone name.
pzone_tcp	The size (radius) of the TCP zone in mm.
pzone_ori	The zone size (radius) for the tool reorientation in mm. The size must be larger than the corresponding value for pzone_tcp.
pzone_eax	The zone size (radius) for external axes in mm. The size must be larger than the corresponding value for pzone_tcp.
zone_ori	The zone size for the tool reorientation in degrees. If the robot is holding the work object, this means an angle of rotation for the work object.
zone_leax	The zone size for linear external axes in mm.
zone_reax	The zone size for rotating external axes in degrees.

4.9 Managing tools

4.9.1 Tool builder

Overview

The tool builder is used to assemble a set of tools or customize existing tools from a list. Use this tool to add new tools in the tool library. These tools can be used for collision detection.



xx1400000156

Here is the elements in the tool builder window. See *Creating and modifying a cutter tool on page 84* for more information.

Item	Description
Name	Name for the new cutter. The name is given by the Tool Builder as ToolD_L.
D(mm)	New cutter diameter in mm.
L(mm)	New cutter length in mm.
Color	New cutter color. To choose a new cutter color, click the color tab and choose one color from the color palette.

Continues on next page

4.9.1 Tool builder

Continued

Item	Description
Ball Tip	New cutter tip shape. The cutter tip will be round when checked.
Create	Create a new cutter with the given properties.
Delete	Delete the current marked cutter.
Mount	Change current cutter to the marked one.

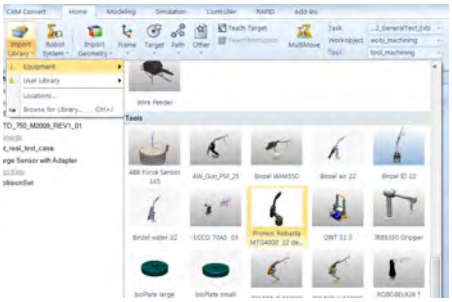
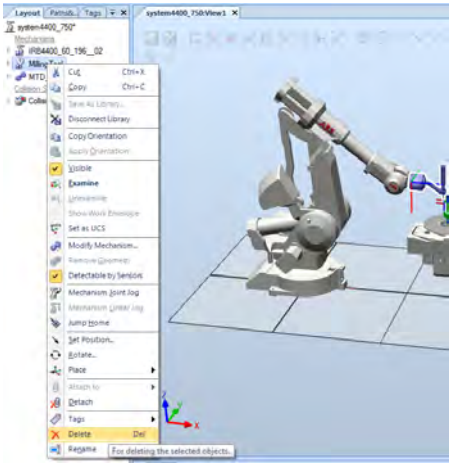
Note

To mark a cutter, click it in the **Cutter Library** list.

4.9.2 Importing Tool

Procedure

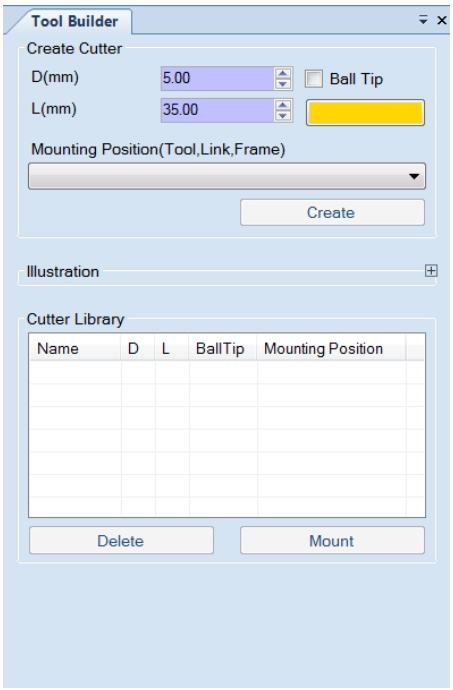
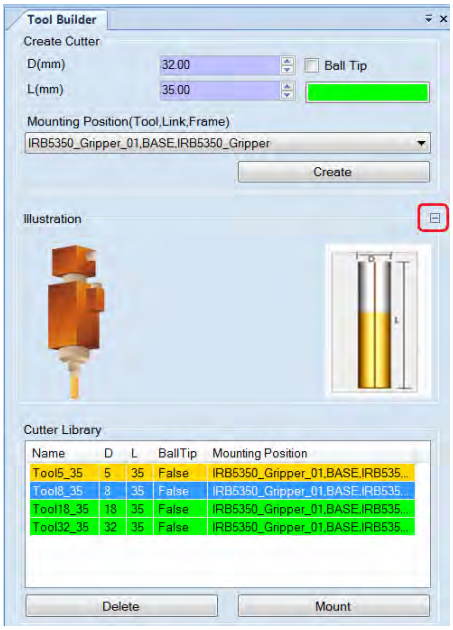
Before creating or modifying any tool, import a tool from the **Import Library**. Use this procedure to import a tool.

	Action	Illustration/Note
1	Click **Home** tab from the ribbon. Select **Import Library**, and click **Equipment**.	
2	Select a tool.	
3	In the Layout tab, find the imported tool and drag it to the robot controller.	
4	The new tool should now be properly attached to the robot.	
5	To delete the tool, right click the tool from the **Layout** window and select **Delete** or press **Delete** on your keyboard.	
6	Return to the **Tool Builder** window. The imported tool should now be in the CAM Converter tool database, ready for further modification.	

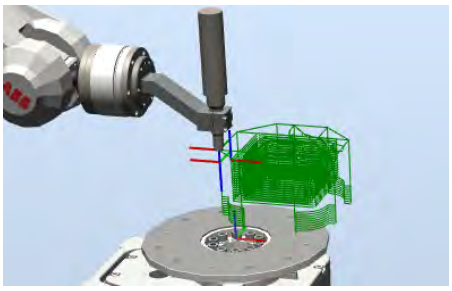
4.9.3 Creating and modifying a cutter tool

Procedure

Use this procedure to create and modify a cutter tool.

	Action	Illustration/Note
1	Begin by specifying the name, diameter, length and color of the tool in the boxes next to the belonging category. The default diameter and length is set at 5 mm and 35 mm.	
2	Click **Create Cutter** to create the tool. The new cutter tool will be displayed in the cutter library. The cutter name is given as ToolD_L.	
3	Click the + sign to the right to view illustration of the tool.	
4	Click a tool in the **Cutter Library** to select it and then click **Delete** to delete it.	

Continues on next page

	Action	Illustration/Note
5	Select a tool in the **Cutter Library** and click **Mount** to change cutter.	
6	The loaded tool will appear in the 3D graphics window.	
7	You can view different cutters by marking them in the cutter window. They will be displayed next to it. **Tip** Interact by scrolling in and out. Rotate by holding down mouse button 1 and middle mouse button and dragging around.	

This page is intentionally left blank

5 Reference information

5.1 Terminology

Keyword

Keyword of the G-code parameter or instruction. It is used to find the specific G-code/ field in the G-code program file, i.e. parameter or instruction. Must be a single word.

Behavior

Value type or action type for parameters or instructions.

Main file

The output RAPID file. If the output file is not split, then Main file would contain all information including program and user setting information in the Export -> Export Setting. If the output file is split, the output would be one main file and several sub files. The main file would contain export setting information.

Export Settings

If the output file is split, the output file will be one main file and several sub files. The main file will contain export settings information.

Sub file

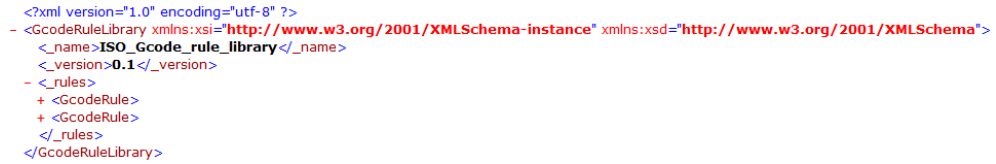
The output RAPID file. If the output file is split, instructions of the output file would be split into several parts.

5.2.1 G-code parse rule file structure overview

5.2 CAM Convert Parse rule

5.2.1 G-code parse rule file structure overview

G-code parse rule file structure



xx1400000165

Overall component description

Component	Description
`<?xml version="1.0" encoding="utf-8" ?>` `<GcodeRuleLibrary xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">`	Xml head information. **Note** Users can not change the term of `GcodeRuleLibrary`.
`<_name>ISO_Gcode_rule_library</_name>`	The G-code parse rule library name for this xml file.
`<_version>0.1</_version>`	Version no. of the rule file.
`</_rules>`	The node under which a set of G-code parse rules are listed.
`<GcodeRule>`	A rule for parse one type of CNC G-code to typical type of RAPID program.
`</GcodeRuleLibrary>`	End of the rule set

5.2.2 G-code parse rule component description

5.2.2.1 <GcodeRule>

Illustration

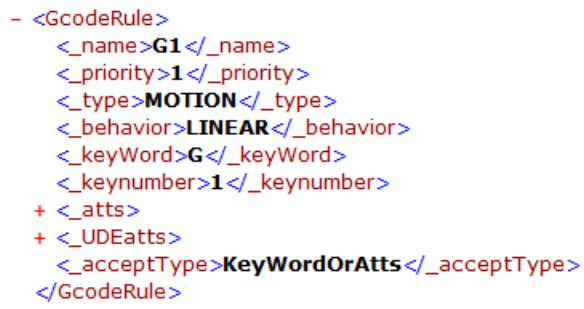


Figure 5.1: G-code parse rule component

Description

Component	Description
<_name>	Name of the G-code instruction
<_priority>	If there is no G-code instruction name in the G-code instruction, the priority order of the G-code parse rules to be used. Bigger number has higher priority.
<_type>	Type of the G-code instruction. There are three types: MOTION, NOMOTION, STATE. MOTION type is motive action. NOMOTION is also action related, but motion, i.e. tool change. STATE is the motion parameter.
<_behavior>	Behavior type of the G-code instruction.
<_keyWord>	Keyword of the G-code instruction. It is used to find the specific Must be a single word.
<_keynumber>	Key number of the G-code instruction.
<_atts>	A set of parameters in a G-code instruction, which is defined in the G-code instruction parse rule. For detailed information about <_atts>, see <_atts> on page 90.
<_UDEatts>	A set of user defined parameters in a G-code instruction. User can define user defined parameters for a user defined G-code parameter in the G-code instruction parse rule.
<_acceptType>	The condition that a G-code instruction is valid. There are three types of <_acceptType>: KeyWord, KeyWordOrAtts and KeyWordAndAtts. KeyWord: If this line of G-code instruction contains the matching KeyWord, then this instruction is valid. KeyWordOrAtts: If this line of G-code instruction contains the matching KeyWord or Keynumber, then this instruction is valid. KeyWordAndAtts: If this line of G-code instruction contains the matching KeyWord and Keynumber, then this instruction is valid.

5.2.2.2 <_atts>

Illustration

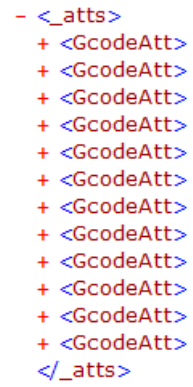


Figure 5.2: G-code parse rule atts

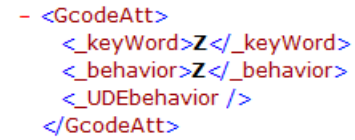
Description

<_atts> contains a set of attributes-<GcodeAtt> for the G-code parameters in a G-code instruction.

Component	Description
<GcodeAtt>	Attribute of a parameter in the G-code instruction.

5.2.2.3 <GcodeAtt>

Illustration



xx1400000168

Figure 5.3: G-code atts component

Description

Component	Description
<_keyWord>	Keyword of the parameter in the G-code instruction. Must be a single word.
<_behavior>	Behavior of the parameter, i.e. value type or executed action type of the parameter.
<_UDEbehavior />	Only when <_behavior> is UDE, <_UDEbehavior /> is valid. User can add user defined behavior in this place. For how to add a user defined behavior, see How to add a user defined behavior for an existing parameter?

5.2.3 Existing G-code rules and valid parameters

Existing G-code rules

The following G-code instructions can be parsed by default.

G-code instruction name	<_type>	<_behavior>	<_keyWord>	<_keynumber>	Description
G0	Motion	RAPID	G	0	Rapid move
G1	Motion	Linear	G	1	Linear move
G2	Motion	CW_ARC	G	2	Clockwise circular move
G3	Motion	CCW_ARC	G	3	Counter clockwise circular move
G17	STATE	PLANE_XY	G	17	Select plane XY
G18	STATE	PLANE_ZX	G	18	Select plane ZX
G19	STATE	PLANE_ZY	G	19	Select plane ZY
G53	STATE	MachineCoord	G	53	Dimension shift cancel (The position is in the machine coordinate system)
G54	STATE	SubCoord0	G	54	Zero shift. They can define the work coordinate systems. Each tuple of axis offsets relates program zero directly to machine zero. Standard is 6 tuples(G54 to G59).
G55	STATE	SubCoord1	G	55	
G56	STATE	SubCoord2	G	56	
G57	STATE	SubCoord3	G	57	
G58	STATE	SubCoord4	G	58	
G59	STATE	SubCoord5	G	59	
G70	STATE	UNIT_INCH	G	70	Input inch
G71	STATE	UNIT_MM	G	71	Input metric
G90	STATE	ABSOLUTE	G	90	Absolute
G91	STATE	INCREMENTAL	G	91	Incremental
G92	STATE	ZERORESET	G	92	Preload register. Modify the current zero shift, but not invoke any motion.
G93	STATE	SPEED1	G	93	Inverse time federate
G94	STATE	SPEED2	G	94	Feed per minute
G95	STATE	SPEED3	G	95	Feed per revolution
G96	STATE	SPINDLE_CW	G	96	Constant surface speed
G97	STATE	SPINDLE_CCW	G	97	Revolutions per minute

Continues on next page

Default valid parameters for G-code instruction

Currently only MOTION type G-code instructions contains parameters: X, Y, Z, I, J, K, A, B, C, F, S.

Item	Description
X/Y/Z	Position data
A/B/C	Rotation data. **Note** The prototype assures the A/B/C values as the rotation around the X/Y/Z axis of the machine coordinate, and the order is ZYX.
I/J/K	Center offset for circular move
R	Radius for circular move
F	Feed rate value
S	Spindle value

5.2.4 Customizing the parse rule

Adding a user defined parameter in a motion type G-code instruction

Procedure	Description
1	Add a <GcodeAtt> block under the node of <_UDEatts>.
2	Fill the fields of <_keyWord>, <_behavior> and <_UDEbehavior>. For description of these fields, see <GcodeAtt> on page 91.

Adding a user defined behavior for an existing parameter

Procedure	Description
1	Change the <_UDEbehavior /> in the <GcodeAtt> block into <_UDEbehavior>UDEbehaviorName</_UDEbehavior>.  **Note** <_behavior /> must be UDE.
2	Fill the fields of <_keyWord>, <_behavior> and <_UDEbehavior>. For description of these fields, see <GcodeAtt> on page 91.

5.3 Export rule

5.3.1 Export template file structure overview

Export template file structure

xx1400000170

Overall component description

Component	Description
`<?xml version="1.0" encoding="utf-8" ?>` `<ExportRuleLibrary xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">`	Xml head information. **Note** Users can not change the term of `ExportRuleLibrary`.
`<_name>ISO_Export_rule_library</_name>`	The name for this rule file
`<_version>VersionNo.</_version>`	Version no. of the rule file
`<_rules>`	The node under which a set of rules for exporting RAPID program file are listed
`<ExportRule>`	A rule for export a typical RAPID program. Each rule contains one or more ABB default export instructions. Users can also add user defined export instructions in the rule. For detailed information about `<ExportRule>`, see *`<ExportRule>` on page 106*.
`</_rules>`	End of the rule set
`</ExportRuleLibrary>`	End of the export rule library

5.3.2 Existing ABB export rules and valid parameters

Overview

The existing export rules are default export rules created by ABB. Users can modify the rules in accordance with the rule structure. Following items can be defined by users:

- Add user defined parameters
- Add or modify existing instruction

Note

Users can change the existing instruction except Keyword in the instruction.

For information about <ExportRule>, see *<ExportRule> on page 106*.

Existing export rules

Export Rule type	Name	Description	Export RAPID example
FILEHEAD	-	File head of the exported RAPID program file if the export file is not split. By default, the file head contains information as version no. , module name and robot type etc.	%%% **VERSION: 1** **LANGUAGE:ENGLISH** %%% **MODULE Main_mitte** **! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3** PERS wobjdata Workobject_1 := [FALSE,TRUE,"STN1", [[1,2,3],[1,0,0,0]],[[4,5,6],[1,0,0,0]]]; PERS tooldata Tooldata_1 := [TRUE, [[0.001,2,3],[1,0,0,0]],[1,[0,0,0.001],[1,0,0,0],0,0,0]]; PROC mitte_udeMainPath() AccSet 10, 10; ConfL\Off; SingArea\Off; MoveL [[-317.874, -102.467, 568.623],[0, 6.12303176911189E-17, 1, 0],[0, 0, 0, 0],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10,Tooldata_1 \wobj:=Workobject_1 3; toolchange 2; ConfL\On; ENDPROC ENDMODULE

Continues on next page

Export Rule type	Name	Description	Export RAPID example

SPLITMAIN FILEHEAD	-	The file head of the main file if the export file is split into several sub files.
		By default, it contains information about where the output files would be saved, prefix of saved export files.
		  **Tip**
		Users can change its information except for Keyword.
			%%%
			VERSION: 1
			LANGUAGE:ENGLISH
			%%%
			MODULE Main_mitte
			! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3
			PERS wobjdata Workobject_1 := [FALSE,TRUE,"STN1", [[1,2,3],[1,0,0,0]],[[4,5,6],[1,0,0,0]]];
			PERS tooldata Tooldata_1 := [TRUE, [[0.001,2,3],[1,0,0,0]],[1,[0,0,0.001],[1,0,0,0],0,0,0]];
			PERS string
			FilePath:="C:\\Users\\xyz@cn.abb.com\\Desktop\\outsourcing\\output";
			PERS string FilePrefix:="mitte";
			PERS string PathPrefix:="path_mitte";
			PROC mitteMainPath()
			FOR i FROM 1 TO 5 DO
			Load\Dynamic, FilePath\File:=FilePrefix + NumToStr(i,0) +".mod";
			TPWrite PathPrefix\Num:=i;
			CallByVar PathPrefix,i;
			UnLoad FilePath\File:=FilePrefix+NumToStr(i,0)+".mod";
			ENDFOR
			ENDPROC
			ENDMODULE

SUBFILEHEAD	-	The file head of the several sub files if the export file is split into several sub files. By default, the information is the same with FILEHEAD except for sub module name. The sub module name is automatically generated according to the user setting in the UI.	%%% **VERSION: 1** **LANGUAGE:ENGLISH** %%% **MODULE mitte1** **! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3** PROC path_mitte1() AccSet 10, 10; ConfL\Off; SingArea\Off; MoveL [[-317.874, -102.467, 568.623],[0, 6.12303176911189E-17, 1, 0],[-2, -3, -2, 1],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10, Tooldata_1 \wobj:=Workobject_1 ; MoveJ [[-313.838, -102.467, 107.009],[0, 6.12303176911189E-17, 1, 0],[-2, 0, -4, 1],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10, Tooldata_1 \wobj:=Workobject_1 ; ENDPROC ENDMODULE

Continues on next page

5.3.2 Existing ABB export rules and valid parameters
Continued

Export Rule type	Name	Description	Export RAPID example
SUBPROHEAD	-	The sub pro head of a pro in a sub export file if the export file is split into several sub files.	%%% VERSION: 1 LANGUAGE:ENGLISH %%% MODULE mitte1 ! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3 **PROC path_mitte1()** AccSet 10, 10; ConfL\Off; SingArea\Off; MoveL [[-317.874, -102.467, 568.623],[0, 6.12303176911189E-17, 1, 0],[-2, -3, -2, 1],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10, Tooldata_1 \wobj:=Workobject_1 ; MoveJ [[-313.838, -102.467, 107.009],[0, 6.12303176911189E-17, 1, 0],[-2, 0, -4, 1],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10, Tooldata_1 \wobj:=Workobject_1 ; ENDPROC ENDMODULE
PROHEAD	-	The pro head of a pro in the export file	%%% VERSION: 1 LANGUAGE:ENGLISH %%% MODULE Main_mitte_ude ! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3 PERS wobjdata Workobject_1 := [FALSE,TRUE,"STN1", [[1,2,3],[1,0,0,0]],[[4,5,6],[1,0,0,0]]]; PERS tooldata Tooldata_1 := [TRUE, [[0.001,2,3],[1,0,0,0]], [1,[0,0,0.001],[1,0,0,0],0,0,0]]; **PROC mitte_udeMainPath()** AccSet 10, 10; ConfL\Off; SingArea\Off; MoveL [[-317.874, -102.467, 568.623],[0, 6.12303176911189E-17, 1, 0],[0, 0, 0, 0],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10, Tooldata_1\wobj:=Workobject_1 3; toolchange 2; ConfL\On; ENDPROC ENDMODULE

Continues on next page

Export Rule type	Name	Description	Export RAPID example
PROCODE HEAD	-	Head of the program code in the main file that is not split. If the exported file is split, then this part is in the first sub file.	%%% VERSION: 1 LANGUAGE:ENGLISH %%% MODULE Main_mitte_ude ! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3 PERS wobjdata Workobject_1 := [FALSE,TRUE,"STN1", [[1,2,3],[1,0,0,0]],[[4,5,6],[1,0,0,0]]]; PERS tooldata Tooldata_1 := [TRUE, [[0.001,2,3],[1,0,0,0]],[1,[0,0,0.001],[1,0,0,0],0,0,0]; PROC mitte_udeMainPath() **AccSet 10, 10;** **ConfL\Off;** **SingArea\Off;** MoveL [[-317.874, -102.467, 568.623],[0, 6.12303176911189E-17, 1, 0],[0, 0, 0, 0],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10, Tooldata_1\wobj:=Workobject_1 3; toolchange 2; ConfL\On; ENDPROC ENDMODULE

Continues on next page

5.3.2 Existing ABB export rules and valid parameters
Continued

Export Rule type	Name	Description	Export RAPID example

SPLITMAIN PROBODY	-	The program code of the main export file if the export file is split into several sub files.	%%%
			VERSION: 1
			LANGUAGE:ENGLISH
			%%%
			MODULE Main_mitte
			! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3
			PERS wobjdata Workobject_1 := [FALSE,TRUE,"STN1", [[1,2,3],[1,0,0,0]],[[4,5,6],[1,0,0,0]]];
			PERS tooldata Tooldata_1 := [TRUE, [[0.001,2,3],[1,0,0,0]],[1,[0,0,0.001],[1,0,0,0],0,0,0]];
			PERS string
			FilePath:="C:\\Users\\xyz@cn.abb.com \\Desktop\\outsourcing\\output";
			PERS string FilePrefix:="mitte";
			PERS string PathPrefix:="path_mitte";
			PROC mitteMainPath()
			FOR i FROM 1 TO 5 DO
			Load\Dynamic, FilePath\File:=FilePrefix + NumToStr(i,0) + ".mod";
			TPWrite PathPrefix\Num:=i;
			CallByVar PathPrefix,i;
			UnLoad FilePath\File:=FilePrefix+NumToStr(i,0)+".mod";
			ENDFOR
			ENDPROC
			ENDMODULE

Continues on next page

Export Rule type	Name	Description	Export RAPID example
PROCODEEND	-	End of the program code in the main file that not split. If the exported file is split, then this part is in the last sub file.	%%% VERSION: 1 LANGUAGE:ENGLISH %%% MODULE Main_mitte_ude ! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3 PERS wobjdata Workobject_1 := [FALSE,TRUE,"STN1", [[1,2,3],[1,0,0,0]],[[4,5,6],[1,0,0,0]]]; PERS tooldata Tooldata_1 := [TRUE, [[0.001,2,3],[1,0,0,0]],[1,[0,0,0.001],[1,0,0,0],0,0,0]]; PROC mitte_udeMainPath() AccSet 10, 10; ConfL\Off; SingArea\Off; MoveL [[-317.874, -102.467, 568.623],[0, 6.12303176911189E-17, 1, 0],[0, 0, 0, 0],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10, Tooldata_1\wobj:=Workobject_1 3; **ConfL\On;** ENDPROC ENDMODULE
PROEND	-	End of the Proc.	%%% VERSION: 1 LANGUAGE:ENGLISH %%% MODULE Main_mitte_ude ! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3 PERS wobjdata Workobject_1 := [FALSE,TRUE,"STN1", [[1,2,3],[1,0,0,0]],[[4,5,6],[1,0,0,0]]]; PERS tooldata Tooldata_1 := [TRUE, [[0.001,2,3],[1,0,0,0]],[1,[0,0,0.001],[1,0,0,0],0,0,0]]; PROC mitte_udeMainPath() AccSet 10, 10; ConfL\Off; SingArea\Off; MoveL [[-317.874, -102.467, 568.623],[0, 6.12303176911189E-17, 1, 0],[0, 0, 0, 0],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10, Tooldata_1\wobj:=Workobject_1 3; ConfL\On; **ENDPROC** ENDMODULE

Continues on next page

5.3.2 Existing ABB export rules and valid parameters
Continued

Export Rule type	Name	Description	Export RAPID example
FILEEND	-	End of the exported file.	%%% VERSION: 1 LANGUAGE:ENGLISH %%% MODULE Main_mitte_ude ! Generated by ABB G-code converter for ABB Robot IRB140_6_81_C_01_3 PERS wobjdata Workobject_1 := [FALSE,TRUE,"STN1", [[1,2,3],[1,0,0,0]],[[4,5,6],[1,0,0,0]]]; PERS tooldata Tooldata_1 := [TRUE, [[0.001,2,3],[1,0,0,0]],[1,[0,0,0.001],[1,0,0,0],0,0,0]]; PROC mitte_udeMainPath() AccSet 10, 10; ConfL\Off; SingArea\Off; MoveL [[-317.874, -102.467, 568.623],[0, 6.12303176911189E-17, 1, 0],[0, 0, 0, 0],[9E9,9E9,9E9,9E9,9E9,9E9]], v100, z10, Tooldata_1\wobj:=Workobject_1 3; ConfL\On; ENDPROC **ENDMODULE**
LINEAR	MoveL	Export to MovL instruction	
RAPID	MoveJ	Export to MovJ instruction	
ARC	MoveC	Export to MovC instruction	

Parameters in existing export rules/Behaviors

Export Rule type	Name	Description	
FILEHEAD	-	%ModuleName	Name of the module where the export RAPID program is saved. This is set in Export -> Export Setting.
		%RobotName	Name of the robot that would execute machining work. This is set in Station -> Edit Station -> Robot.
SPLITMAIN FILEHEAD	-	%filePath	Location where the export RAPID files would be saved. This is set in Export -> Export Setting.
		%SubModuleName	Name of modules which split export RAPID routines belong to. This is set in Export -> Export Setting.
		%SubRoutineName	Name of routines which split export RAPID programs belong to. This is set in Export -> Export Setting.

Continues on next page

Export Rule type	Name	Description	
SUBFILEHEAD	-	%SubModuleName	Name of modules which split export RAPID routines belong to. This is set in Export -> Export Setting.
		%RobotName	Name of the robot that would execute machining work. This is set in Station -> Edit Station -> Robot.
SUBPROHEAD	-	%SubRoutineName	Name of routines which split export RAPID programs belong to. This is set in Export -> Export Setting.
PROHEAD	-	%RoutineName	Name of the routine which the export RAPID program belongs to. This is set in Export -> Export Setting.
PROCODE HEAD	-	*	Instructions that by default would be put at the beginning of the export RAPID file.
SPLITMAIN PROBODY	-	%SplitNum	Quantity that the export RAPID file would be split. This is decided by how many lines of instructions that each sub RAPID file can contain which is calculated by the definition from users in Export -> Export Setting.
PROCODEEND	-	*	Instructions by default at the end of the export RAPID file.
PROEND	-	*	Instruction indicating the end of the program.
FILEEND	-	*	Instruction indicating the end of the module.
LINEAR	MoveL	%Target1_x	Position data
		%Target1_y	Position data
		%Target1_z	Position data
		%Target1_q1	Rotation data
		%Target1_q2	Rotation data
		%Target1_q3	Rotation data
		%Target1_q4	Rotation data
		%Target1_Config0	Robot configuration at the target
		%Target1_Config1	Robot configuration at the target
		%Target1_Config2	Robot configuration at the target
		%Target1_Config3	Robot configuration at the target
		%Speed	Speed data
		%Zone	Zone data
		%ActiveTool	Active tool data
		%ActiveWobj	Active Wobj data
		%AxisIndex1	External axis value at the target
RAPID	MoveJ	%Target1_x	Position data
		%Target1_y	Position data

Continues on next page

5.3.2 Existing ABB export rules and valid parameters
Continued

Export Rule type	Name	Description	
		%Target1_z	Position data
		%Target1_q1	Rotation data
		%Target1_q2	Rotation data
		%Target1_q3	Rotation data
		%Target1_q4	Rotation data
		%Target1_Config0	Robot configuration at the target
		%Target1_Config1	Robot configuration at the target
		%Target1_Config2	Robot configuration at the target
		%Target1_Config3	Robot configuration at the target
		%Speed	Speed data
		%Zone	Zone data
		%ActiveTool	Active tool data
		%ActiveWobj	Active Wobj data
		%AxisIndex1	External axis value at the target
ARC	MoveC	%Target1_x	Position data for target 1
		%Target1_y	Position data for target 1
		%Target1_z	Position data for target 1
		%Target1_q1	Rotation data for target 1
		%Target1_q2	Rotation data for target 1
		%Target1_q3	Rotation data for target 1
		%Target1_q4	Rotation data for target 1
		%Target1_Config0	Robot configuration at target 1
		%Target1_Config1	Robot configuration at target 1
		%Target1_Config2	Robot configuration at target 1
		%Target1_Config3	Robot configuration at target 1
		%Target2_x	Position data for target 2
		%Target2_y	Position data for target 2
		%Target2_z	Position data for target 2
		%Target2_q1	Rotation data for target 2
		%Target2_q2	Rotation data for target 2
		%Target2_q3	Rotation data for target 2
		%Target2_q4	Rotation data for target 2
		%Target2_Config0	Robot configuration at target 2
		%Target2_Config1	Robot configuration at target 2
		%Target2_Config2	Robot configuration at target 2
		%Target2_Config3	Robot configuration at target 2
		%Speed	Speed data

Continues on next page

Export Rule type	Name	Description	
		%Zone	Zone data
		%ActiveTool	Active tool data
		%ActiveWobj	Active Wobj data
		%AxisIndex1	External axis value at target1
		%AxisIndex2	External axis value at target 2

*: No parameter included.

5.3.3.1 <ExportRule>

5.3.3 Export rule component description

5.3.3.1 <ExportRule>

Illustration

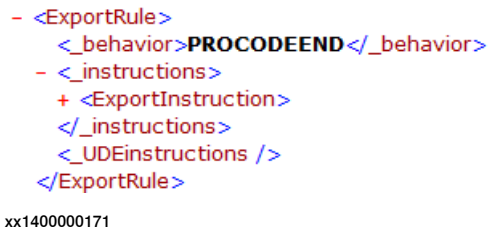


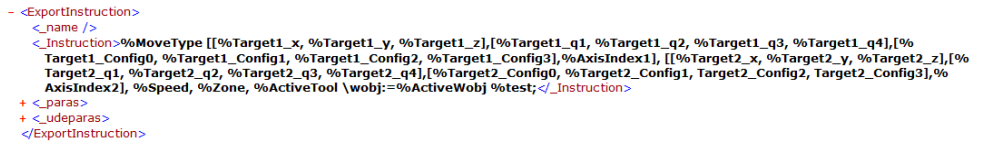
Figure 5.4: Export rule component

Description

Component	Description
<ExportRule>	An export rule block
<_behavior>	Component type for components in the export rule
<_instructions>	Node where all the export instructions for this rule would be put.
<ExportInstruction>	An export instruction with instruction code and parameter definition. For detailed information, see <ExportInstruction> on page 107.
</_instructions>	End of <_instructions>
<_UDEinstructions />	Where to put user defined instructions
</ExportRule>	End of <ExportRule>

5.3.3.2 <ExportInstruction>

Illustration



xx1400000172

Figure 5.5: Export instruction

Description

Component	Description
<ExportInstruction>	An export instruction block
<_name />	n/a. This function is to be extended later.
<_Instruction>	The export RAPID instruction. It would be displayed in the export RAPID program file.
<_paras>	Contains the definition for the instruction parameters which are defined in <InstructionPara>. Detailed content and structure for <InstructionPara>, see *<InstructionPara> on page 108*.
<_udeparas>	User defined parameters. User only add a <InstructionPara> block under this node, then define the user defined parameter. A valid user defined parameter must have corresponding parameter in the input G-code file and the parse rule file.

5.3.3.3 <InstructionPara>

Illustration

Here is the instruction parameter in an existing instruction:

```
- <InstructionPara>
    <_keyWord />
    <_behavior />
    <_paraStr>%Target1_x</_paraStr>
  </InstructionPara>
```

xx1400000173

Figure 5.6: Export instruction parameter

Here is user defined parameter in an existing instruction:

```
- <InstructionPara>
    <_keyWord>U</_keyWord>
    <_behavior>Test</_behavior>
    <_paraStr>%test</_paraStr>
  </InstructionPara>
```

xx1400000174

Figure 5.7: User defined export instruction parameter

Description

Component	Description
<_keyWord />	The keyword is used to match the corresponding keyword in the G-code file and help the post-processor to locate the instruction parameter or the instruction in the export template and find out the corresponding export method.
<_behavior />	The value type of the parameter. It is used to help the post processor to confirm what type of execution should be done to a parsed parameter regarding to its behavior, execute the corresponding calculation or action and then assign the result to the corresponding parameter in the exported RAPID program.
<_paraStr>	The parameter value. This value can be defined by the users or calculated by the post processor. The value would be applied to its corresponding parameter in the exported RAPID instruction.

Tip

Only motion instructions are directly converted to RAPID instructions. Status data from the G-Code program would be parsed, calculated and converted as parameters in the RAPID instructions.

5.3.4 Customizing export rule

Adding a customized rule

Procedure	Description
1	Add a node of new rule <ExportRule> under the node of <_rules> .For detailed content and structure about <ExportRule>, see <ExportRule> on page 106.
2	Name the new rule in <_behavior>NameofTheNewRule</_behavior>.
3	Add program code node <ExportInstruction> under the node of <_UDEinstructions>. For detailed content and structure about <ExportInstruction>, see <ExportInstruction> on page 107.

Adding a customized parameter in an existing instruction

Procedure	Description
1	Find <_udeparas /> under the node of <ExportInstruction> .
2	Add the node of <InstructionPara>. For detailed content and structure about <InstructionPara>, see <InstructionPara> on page 108.

Note

Only MoveL, MoveJ, MoveC instructions can add customized parameters in the instructions.

5.4 Components of the export RAPID file

Export RAPID file

The export RAPID file contains the following information by default:

- Version information
- Language type
- Module Headname
- !Generated by ABB G-code converter for ABB Robot IRBXXXX
- Definition of Wobjdata
- Definition of Tooldata
- FilePath definition. Where the output file would be saved.
- Define the default prefix of the output file name.
- Define the default prefix of the output path name.
- Define proc name.
- Code of the program.
- End of the program.
- End of the function.
- End of the module.

Note

The exported RAPID file must contain tool and wobj information if tool and wobj beside Tool0 and Wobj0 are selected in Station -> Edit Station. For example:

```
PERS wobjdata Workobject_1 :=
[FALSE,TRUE,STN1,[[1,2,3],[1,0,0,0]],[[4,5,6],[1,0,0,0]]];
PERS tooldata Tooldata_1 :=
[TRUE,[[1,2,3],[1,0,0,0]],[1,[0,0,0.001],[1,0,0,0],0,0,0]];
```

5.5 Supported APT instructions list

Supported APT instructions

Keyword	Description
GOTO/X,Y,Z	Linear 3-axis moves.
GOTO/X,Y,Z,I,J,K	Linear 5-axis moves. (I,J,K) is the tool vector. Default tool vector is (0,0,1).
CIRCLE/XC,YC,ZC,I,J,K,R	Arc/Circle move where is (XC,YC,ZC) is the circle centre, (I,J,K) is a vector normal to the arc/circle plane, and R is the radius.
RAPID	Rapid (no contract) move.
FEDRAT/MMPM,V	Feed rate V. Units can be MMPM or IPM.
LOADTL/x	Tool number x.
SPINDL/RPM,x	Spindle speed x.
MSYS/(12 number array)	Coordinate transformation. MCS.

This page is intentionally left blank

Index

Contact us

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

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

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

www.abb.com/robotics

3HAC050262-001, Rev B, en