Release Notes RobotStudio® 5.12

GENERAL **3**

Installation **3**

Activate RobotStudio **4**

How to proceed when contacting ABB **6**

RELEASE INFORMATION **7**

Release Name **7**

Release Information **7**

Release Date **7**

ROBOTSTUDIO 5.12 ..... **8**

Supported Operating Systems **8**

Recommended Hardware **8**

Supported RobotWare Versions **9**
Compatibility Limitations 9

Safety Configuration **10**

Support for future RobotWare versions **10**

CAD Converter **10**

Demo stations **12**

Tutorials **12**

Robot Libraries **13**

Robot Libraries Paint **14**

Track Libraries **14**

Positioner Libraries **14**

Language Support **14**

Documentation **15**

New Functionality in RobotStudio 5.12 (since RobotStudio 5.11) **16**

Solved Limitations **21**
VSTA 21
Offline 21
Online 22

Known Limitations **22**
General 22
Online 22
Offline 23
Graphics and Geometry 27

Visual Studio Tools for Applications **29**

Corrected "Product Defect Document" RW5.11 PDD **31**

SUPPORTING INFORMATION **33**

How to automatically create a system with external axes **33**

How to manually set up a system with track motion of type RTT or IRBTx003? **34**

How to manually set up a system with track motion of type IRBTx004? **35**

Code Snippets **36**

Instruction Templates **37**

API IMPROVEMENTS **38**

New types* **38**

Improved types **39**

General

Installation

Upon installation of RobotStudio 5.12, the user will be asked if any existing installation of RobotStudio 5.XX or 2008 should be uninstalled automatically or the existing installation should remain untouched. Also any previous installation of RobotStudio Online will remain untouched.

Note: the user must have administrator privileges on the PC in order to install RobotStudio.

There are three possible options when installing RobotStudio:

- **Minimal** – installs the functionality corresponding to RobotStudio Online. Only the Online tab will be available with this option.
- **Full** – Installs everything needed to run the complete RobotStudio. This is also the option to use to take advantage of the additional features in the Basic and Premium functionality mode.
- **Custom** – Allows the possibility to exclude unwanted robot libraries and CAD converters.

How to install RobotStudio on a PC

1. Insert the robot software DVD in the PC.
    a. If a menu for the DVD is opened automatically, continue with step 5.
    b. If no menu for the DVD is opened, continue with step 2.
2. On the Start menu, click Run.
3. In the Open box, type the drive letter for your DVD drive followed by: `\launch.exe`

    ***Example**: If your DVD drive has the letter D, then type: D:\launch.exe*
4. Click OK.
5. Select language for the DVD menu.
6. On the DVD menu, click Install.
7. On the installation menu, click RobotStudio. This opens the installation wizard, which will guide you through the rest of the software installation.
8. Follow the instructions in the installation wizard.
9. After installing RobotStudio, proceed with activating RobotStudio, see below.

***Note:** For an immediate trial period of 30 days, RobotStudio will work without activation.*

Activate RobotStudio

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

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

How do I activate RobotStudio?

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

Trial period: Before entering a valid Activation Key, you can run the software, in Premium functionality mode, with all the features enabled, for a trial period of up to 30 days. Please note that the trial period days start immediately after installation. After entering a valid Activation Key, you will see only the features you have purchased (if installed during the trial period you will loose the trial period time).

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

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

Activate automatically over the Internet or manually

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

Automatic activation by using the Internet (recommended): Once you have selected the option *I want to activate RobotStudio over the Internet*, and proceeded through the Wizard, the Activation Wizard automatically contacts the ABB licensing servers over your Internet connection. If you are using a valid Activation Key that has not exceeded the number of installations allowed, your product is activated immediately.

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

Manual activation: If the computer does not have an Internet connection, you must create a license file by selecting the option *I want to request a license file.* Proceed through the wizard, enter your Activation Key and save the License Request File to your computer. Use a removable medium, such as a USB stick or floppy disk, to transfer the file to a computer with an Internet connection. Go to *www.robotstudio.com/community* and click on *My Subscriptions*. Use the login sent to you via e-mail at an earlier date to be able to see your subscriptions. In the menu to the left you will find a link *Activate Product*. Click on it and follow the instructions. The result will be a License File that should be saved and transferred back to the computer holding your product. Relaunch

the Activation Wizard and select the option *I have a license file I want to install.* Proceed through the wizard, selecting the License File when requested. Upon completion, RobotStudio is activated and ready for use.

How do I activate later?

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

1. Click the **RobotStudio button** **,** and then click on the **RobotStudio options** button beside **Exit**, and select the **Licensing** section.
2. Click **Activation Wizard** to launch the activation wizard.
3. If your RobotStudio installation has been activated, you will have valid licenses for the features covered by your subscription.

Which RobotStudio version are you using?

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

How can I tell whether my RobotStudio installation has already been activated?

1. Click the **RobotStudio button** **,** and then click on the **RobotStudio options** button beside **Exit**, and select the **Licensing** section.
2. Click **View Installed License Keys** to see the status of your current license.
3. If your RobotStudio installation is activated, you will have valid licenses for the features covered by your subscription.

Network licenses

Network licenses are not available for RobotStudio 5.12.

How to proceed when contacting ABB

If you have any questions or problems with your RobotStudio installation, please get in touch with your local ABB contact, see *http://www.abb.com/robotics*.

Have the following in mind when contacting ABB

1. Give a brief description of how to reproduce your problem.
2. Create screenshots if applicable. (Use ALT + PRINT SCREEN to get an image of the active window instead of the entire screen.)
3. Generate a Full Scan with the RobotStudio Support Tool available next to RobotStudio in the Start menu, save the report and attach it with your problem description. (Click *Start → Programs → ABB Industrial IT → Robotics IT → RobotStudio → RobotStudio Support Tool*, click on *Run Full Scan* and then *Save Report.*
4. We also need the following user information:
   i. name
   ii. company
   iii. contact information
   iv. what operating system you are running (incl. language)
   v. subscription ID for your purchased license.

Note: *When sending large (> 1 Mb) files, please compress them with WinZip® or WinRAR.*

License support

For license-related questions, please contact the team responsible for license support directly at *softwarefactory_support@se.abb.com*

Release Information

Release Name

The release name is RobotStudio 5.12

The release contains the following products:

- RobotStudio 5.12 build 3076
  (built with RobotWare 5.12 build 3076

Release Information

The information should be considered as last-minute and most up-to-date.

For more information, please visit the support web site at *http://www.robotstudio.com/community*. There you can find a discussion forum dedicated to RobotStudio.

Release Date

Release date **2009-April-09**

RobotStudio 5.12

Supported Operating Systems

Microsoft Windows XP Professional with Service Pack 3

Microsoft Windows Vista SP1 Business or Enterprise

- Windows Media Encoder Hotfix KB929182

Note: *RobotStudio 5.12 does not support 64-bit editions of Windows XP Professional or Vista Business or Enterprise.*

Note*: The Windows Firewall will try to block features necessary to run RobotStudio. Make sure to unblock these features when asked (Industrial Robot Discovery Server, RobotStudio StudioAppFramework module, Virtual RobotController (all published by ABB)). The blocking state of a certain program can be viewed and changed at Start/Control Panel/Windows Security Center/Windows Firewall. Read more on www.microsoft.com.*

Recommended Hardware

High-performance desktop or laptop workstation:

CPU:	2.0 GHz or faster processor
Memory:	1 GB system memory at minimum, 2 GB if running Windows Vista, stations with several robot systems, or large CAD-models.
Free disk-space:	5+ GB free space
Graphics card:	High-performance DirectX 9 or OpenGL-compatible graphics card with the corresponding **up-to-date drivers** installed
Screen resolution:	1280 x 1024 pixels or higher
DPI:	Normal size (96 dpi)
Mouse:	Three-button mouse
3D Mouse	Any 3D mouse from 3DConnexion, see http://www.3dconnexion.com.
DVD-ROM Drive	

Supported RobotWare Versions

RobotStudio 5.12 is distributed with RobotWare 5.12 and works with RobotWare 5.05 up to 5.12. Please see below for details.

Compatibility Limitations

RobotWare 5.05 and 5.06 Compatibility

RobotWare 5.05 and 5.06 and revisions of those versions are supported with the following limitations:

General

- The RAPID Editor does not support RobotWare 5.05 or 5.06, but requires RobotWare 5.07 or later.
  Workaround: *Save the RAPID code to a text file and edit the code using any text editor. RAPID code can also be edited using the Virtual FlexPendant.*
- The RAPID debugging features that are available in the Premium edition of RobotStudio (Step In, Step Over, Step Out, Breakpoints, Watch Window) are not available for 5.05 or 5.06 systems.
  Workaround: *None.*

Offline

- The function *Sync to VC* may cause corrupt RAPID programs. The problem appears when lines (e.g. targets, paths) are removed from the RAPID program and paths are added to the RAPID program in the same *Sync to VC* operation. As a consequence, the new path may be added after the ENDMODULE statement. This problem does not appear when running RobotWare 5.07 or later.
  Workaround*: Do not add and remove RAPID paths and targets in the same operation. If the problem has appeared, resolve the syntax error using the Virtual FlexPendant or any text editor.*
- The function *System from Layout* does not support RobotWare 5.05 or 5.06, but requires RobotWare 5.07 or later.
  Workaround*: Use a supported RobotWare version or create the system manually by using System Builder.*

RobotWare 5.07 Compatibility

RobotWare 5.07 and its revisions of are supported with the following limitations:

General

- The location of the program pointer is not updated in the RAPID Editor during program execution.

Offline

- A limitation in the versions 5.07.02, 5.07.03, and, 5.07.04 of RobotWare may cause the Virtual Controller to System Failure state during I-start on certain computers. The problem is due to the ctrl.bin-file not being correctly created.
  Workaround*: Create an empty ctrl.bin file in the INTERNAL folder of the*

controller system, and then perform a warm start.
***Note**: The problem will reappear if the system is I-started.*

RobotWare 5.08 Compatibility

RobotWare 5.08 and its revisions of are supported with the following limitations:

Offline

- RobotWare 5.08 is not supported.
  ***Workaround**: Use RobotWare 5.08.01 or later.*

RobotWare 5.10 Compatibility

- RobotWare 5.10 and its revisions of are supported with the following limitations:

Offline

- Starting a controller will generate internal UAS error in controller error log.

Safety Configuration

Safety configuration of a track motion IRC5 system equipped with a safety controller of type EPS or SafeMove can be done without the need to read track motion parameters manually when using RobotStudio 5.11.01 or later and RobotWare 5.11.01 or later. Encrypted parameters needed by the safety controller will be automatically read by EPS Wizard and SafeMove Configurator, respectively.

Support for future RobotWare versions

RobotStudio 5.12 supports all future minor revisions of RobotWare, but no future major releases. For example, RobotStudio 5.12 will support RobotWare 5.12.01 (if, and when available) but not RobotWare 5.13.

CAD Converter

Supported CAD Formats and Versions

RobotStudio includes advanced CAD import capabilities such as:

- ACIS (reads/writes versions v6 to R19)
- IGES (reads versions up to 5.3, writes version 5.3)
- STEP (reads versions AP203 and AP214 (geometry only), writes version AP214)
- VDAFS (reads VDAFS up to 2.0, writes VDAFS 2.0)
- CATIA V4 (reads versions 4.1.9 to 4.2.4)
- CATIA V5 (reads CATIA V5 R6 to R19)
- Inventor (reads versions 6 to 2009)
- Pro/Engineer (reads versions 16 to Wildfire4)

***Note**: The CAD converters require separate licenses (except ACIS).*

CAD Converter Options

The CAD Converter options can be set by using the *Advanced* button of the *Settings* dialog of the CAD Converter. By pressing the *Advanced* button, the *CADConverter.ini* file is opened. The file specifies all available options for CAD conversion. To change an option, simply uncomment the line by removing the semicolon and modify the option as desired. All options are described in the file *AcisInterOpConnectOptions.pdf* in the RobotStudio folder of the RobotWare DVD.

Demo stations

There are four demo stations included in this version.

- Demo Two Robots and Conveyor
- Demo FlexLoader
- Demo Exhaust Pipe
- Demo Palletizer

They are stored in the **Pack & Go** format and can be opened with the command **Unpack & Work** on the **Collaborate** section of the RobotStudio menu.

Tutorials

Tutorials are available at the RobotStudio Community at http://www.robotstudio.com/community.

Robot Libraries

The folder *ABB Library* contains libraries of robots, tools, external axes, positioners and equipment. The ABB Library folder also contains template robot systems for all included robot models. Updated robot libraries are published on *http://www.abb.com/robotics* as they become available.

ABB Robot Libraries supported by RobotStudio 5.12

Variant	Library name
140 5kg/0.8m Type A/B	IRB140_5_81__01.rslib
140 5kg/0.8m Type C	IRB140_5_81_C_01.rslib
140 6kg/0.8m Type C	IRB140_6_81_C_01.rslib
1400 Type A/B	IRB1400_5_144__01.rslib
1400H Type A/B	IRB1400H_5_128__01.rslib
140T 5kg/0.8m Type C	IRB140T_6_81_C_01.rslib
1410	IRB1410_5_144__01.rslib
1600 5kg/1.2m	IRB1600_5_120__01.rslib
1600 5kg/1.2m Type A	IRB1600_5_120_A_01.rslib
1600 5kg/1.45m	IRB1600_5_145__01.rslib
1600 5kg/1.45m Type A	IRB1600_5_145_A_01.rslib
1600 6kg/1.2m	IRB1600_6_120__02.rslib
1600 6kg/1.45m	IRB1600_6_145__02.rslib
1600 7kg/1.2m	IRB1600_7_120__01.rslib
1600 7kg/1.2m Type A	IRB1600_7_120_A_01.rslib
1600 7kg/1.45m	IRB1600_7_145__01.rslib
1600 7kg/1.45m Type A	IRB1600_7_145_A_01.rslib
1600 8kg/1.2m	IRB1600_8_120__02.rslib
1600 8kg/1.45m	IRB1600_8_145__02.rslib
1600ID 4kg/1.5m	IRB1600ID_4_150__02.rslib
2400 10kg	IRB2400_10_150__01.rslib
2400 16kg	IRB2400_16_150__01.rslib
2400L	IRB2400L_7_180__02.rslib
260	IRB260_30_150__01.rslib
340	IRB340__01.rslib
360 1kg Std No axis 4	IRB360_1_1130_3D_STD_03.rslib
360 1kg Wash-down No axis 4	IRB360_1_1130_3D_WD_03.rslib
360 1kg Standard	IRB360_1_1130_4D_STD_03.rslib
360 1kg Wash-down	IRB360_1_1130_4D_WD_04.rslib
360 1kg Stainless	IRB360_1_1130_4D_WDS_03.rslib
360 1kg/0.8m Std No axis 4	IRB360_1_800_3D_STD_03.rslib
360 1kg/0.8m Wash-down No axis 4	IRB360_1_800_3D_WD_03.rslib
360 1kg/0.8m Std	IRB360_1_800_4D_STD_03.rslib
360 1kg/0.8m Wash-down	IRB360_1_800_4D_WD_04.rslib
360 3kg Std No axis 4	IRB360_3_1130_3D_STD_03.rslib
360 3kg Wash-down No axis 4	IRB360_3_1130_3D_WD_03.rslib
360 3kg Standard	IRB360_3_1130_4D_STD_03.rslib
360 3kg Wash-down	IRB360_3_1130_4D_WD_04.rslib
360 3kg Stainless	IRB360_3_1130_4D_WDS_03.rslib
4400 45kg	IRB4400_45_196__01.rslib
4400 60kg	IRB4400_60_196__01.rslib

Variant	Library name
4400L 10kg	IRB4400L_10_255__01.rslib
4400L 30kg	IRB4400L_30_243__01.rslib
4400S 30kg	IRB4400S_30_243__01.rslib
4450S 30kg	IRB4450S_30_240__01.rslib
4600 20kg/2.5m	IRB4600_20_250__02.rslib
4600 40kg/2.55m	IRB4600_40_255__02.rslib
4600 45kg/2.05m	IRB4600_45_205__02.rslib
4600 60kg/2.05m	IRB4600_60_205__02.rslib
6400R 200kg/2.5m	IRB6400R_200_250__01.rslib
6400R 200kg/2.8m	IRB6400R_200_280__01.rslib
660 180kg/3.15m	IRB660_180_315__01.rslib
660 250kg/3.15m	IRB660_250_315__01.rslib
6600 175kg/2.55m	IRB6600_175_255__01.rslib
6600 175kg/2.80m	IRB6600_175_280__01.rslib
6600 225kg/2.55m	IRB6600_225_255__01.rslib
6600ID 185kg/2.55m	IRB6600ID_185_255__01.rslib
6620 150kg/2.2m	IRB6620_150_220__01.rslib
6640 130kg/3.2m	IRB6640_130_320__03.rslib
6640 180kg/2.55m	IRB6640_180_255__03.rslib
6640 185kg/2.8m	IRB6640_185_280__03.rslib
6640 205kg/2.75m	IRB6640_205_275__03.rslib
6640 235kg/2.55m	IRB6640_235_255__03.rslib
6640ID 170kg/2.75m	IRB6640ID_170_275__04.rslib
6640ID 200kg/2.55m	IRB6640ID_200_255__04.rslib
6650 125kg/3.2m	IRB6650_125_320__01.rslib
6650 200kg/2.75m	IRB6650_200_275__01.rslib
6650ID 170kg/2.75m	IRB6650ID_170_275__01.rslib
6650S 125kg/3.5m	IRB6650S_125_350__01.rslib
6650S 200kg/3.0m	IRB6650S_200_300__01.rslib
6650S 90kg/3.9m	IRB6650S_90_390__01.rslib
6660 130kg/3.1m	IRB6660_130_310__02.rslib
6660 205kg/1.9m	IRB6660_205_190__01.rslib
7600 150kg/3.5m	IRB7600_150_350__01.rslib
7600 325kg/3.1m	IRB7600_325_310__01.rslib
7600 340kg/2.8m	IRB7600_340_280__01.rslib
7600 400kg/2.55m	IRB7600_400_255__01.rslib
7600 500kg/2.3m	IRB7600_500_230__01.rslib
7600 500kg/2.55m	IRB7600_500_255__01.rslib
940	IRB940__01.rslib

Robot Libraries Paint

RobotStudio is distributed with the following Paint robot types that are available in the Robots folder of the ABB Library.

ABB Paint Robot Libraries supported by RobotStudio 5.12

Variant	Library name
52 short vertical arm	IRB52_12_475_1005__01.rslib
52 std vertical arm	IRB52_12_700_1005__01.rslib
580-12 std arm	IRB580_12_1000_1620__02.rslib
580-12 short arm	IRB580_12_1000_1220__01.rslib
5400-12 std arm	IRB5400_12_1200_1620__02.rslib
5400-13 std arm	IRB5400_13_1200_1620__02.rslib
5400-14 std arm	IRB5400_14_1200_1620__02.rslib
5400-22 process arm	IRB5400_22_1200_1620__02.rslib
5400-23 process arm	IRB5400_23_1200_1620__02.rslib
5400-24 process arm	IRB5400_24_1200_1620__02.rslib
5400-12 std arm axis 2 +60 deg	IRB5400_12_1200_1620_60P_01.rslib
5400-13 std arm axis 2 +60 deg	IRB5400_13_1200_1620_60P_01.rslib
5400-14 std arm axis 2 +60 deg	IRB5400_14_1200_1620_60P_01.rslib
5500 35A b_80	IRB5500_35A_1300_1720__01.rslib
5500 35B b_80	IRB5500_35B_1300_1720__01.rslib

Track Libraries

RobotStudio is distributed with the following track types that are available in the Track folder of the ABB Library.

Note: in order to use the IRBTX004 tracks the user must install the appropriate Trackmediapool from the RobotWare DVD.

Track family	Length
IRBT4003	1.7 m to 10.7 m
IRBT4004	1.9 m to 19.9 m
IRBT6003	1.7 m to 10.7 m
IRBT6004	1.7 m to 19.7 m
IRBT7003	1.7 m to 10.7 m
IRBT7004	1.7 m to 19.7 m
RTT_Bobin	1.7 m to 11.7 m
RTT_Marathon	1.7 m to 11.7 m
Paint Rails	2 m to 20 m

Positioner Libraries

RobotStudio is distributed with the standard positioners of type IRBP A, B, C, D, K, L and R. The section *Supporting Information* contains more information about the combinations of robots, track motions and positioners that RobotStudio supports.

Language Support

RobotStudio 5.12 is available in the following six languages:

- English

- French
- German
- Spanish
- Italian
- Japanese

Documentation

User documentation for RobotStudio is available from the *Help* button () in the upper-right corner of RobotStudio.

The complete documentation in PDF for RobotWare including RobotStudio is available on DVD and can be ordered separately

New Functionality in RobotStudio 5.12 (since RobotStudio 5.11)

Conveyor Tracking

The new Conveyor Tracking feature allows the user to simulate a system with the option *606-1 Conveyor Tracking*.

When using Conveyor Tracking, the robot tracks a moving object on a conveyor by coordinating its motion with a moving workobject. A physical robot controller system gets information about the workobject position from a Conveyor Encoder Unit connected to a sensor and an encoder.

RobotStudio simulates the physical Conveyor Encoder Unit equipment and manages the I/O signal communication with the virtual system.

The Create Mechanism tool has been enhanced to allow the creation of conveyor mechanisms.

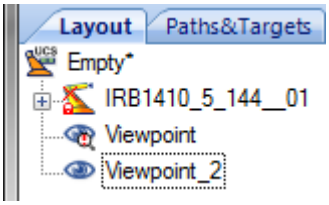
Parts representing the tracked objects can be attached to the conveyor, and the conveyor related system parameters can be accessed directly though the RobotStudio user interface.

There is a demo station available that demonstrates Conveyor Tracking with two robots, see section *Demo stations* above.

View Points

A Viewpoint stores the location and direction of the virtual “camera” in the 3D environment. Viewpoints can be used to store points of interests in a station and to create camera movements during simulation.

Viewpoints in the station are displayed as an eye icon in the Layout browser:



The position and direction of the Viewpoint can also be visualized as an arrow in the 3D graphics. Note: newly created Viewpoints are by default not visible. Viewpoints cannot be selected by clicking in the graphics.

A new action type “Move to Viewpoint” is available in the Event Manager.

When this action is executed, the active 3D view is moved to the Viewpoint in the same way as when executing the corresponding command. In addition, the time for the movement can be specified.

This action is also executed when replaying the simulation in a Station Viewer

Station Viewer

The Station Viewer is a standalone executable that contains a station file and the binaries required to load and view a station in a 3D environment. It can also play back a prerecorded simulation.

When the target user starts the executable, the embedded station file is automatically loaded and presented in a 3D view. The user interface is ribbon based with a single tab.

The RobotStudio user creates a Station Viewer by selecting “Save Station as Viewer” beneath “Collaborate” in the application menu. If a simulation has been executed, an option to include a recording of that simulation is available. The user can also add a comment to be shown when the viewer is started, for example confidentiality information.

In addition to the embedded station, .rsstn files can be loaded by selecting “Open” in the application menu. Note that if the station contains connected library files they must be available on the target system, otherwise the station will be incomplete.

When the RobotStudio user executes a simulation, the movements and visibility of objects are recorded. This recording can optionally be included in the Station Viewer.

If the Station Viewer contains a recorded simulation, the Simulation controls in the ribbon are enabled. These are:

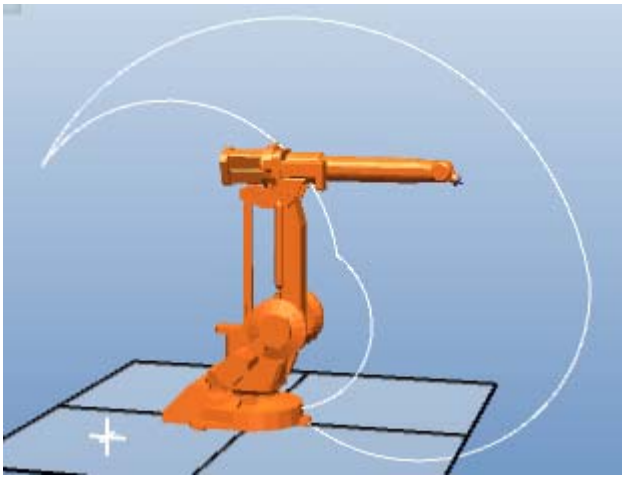
- Play: Starts or resumes simulation playback
- Stop: Stops simulation playback
- Reset: Resets all objects to their initial state. Resets the process time display to zero.
- Run mode: Select to run the simulation once or continuously
- Process time: Displays the current simulation time.

The recording only stores events for objects of Part level and higher. Other events such as VSTA macros are not included. For example, if the RobotStudio simulation executes a VSTA macro that sets the visibility of an individual Body, this is not reflected when playing back the simulation in the Station Viewer.

Note: cycle time displayed in the station viewer only matches cycle time from RobotStudio if the simulation was executed with Time Slice option (under RobotStudio options – Virtual Time).

Working Envelope

The work envelope for a robot is the maximum range of movement. It can be visualized in the 3D graphics by selecting “Show Work Envelope” in the context menu for a robot.



The work envelope is visualized as the outermost trajectory of the active TCP when jogging joints 2 and 3 through all possible values. The visualization is attached to the first link of the robot, so it follows the movement when rotating joint 1.

Work envelopes are only available for articulated robots, e.g. not for picker robots or external axes.

To display the correct work envelope for parallel arm robots (such as the IRB1410 above), some additional metadata in the robot model is needed. This metadata is only included in models shipped with RS 5.12 and higher.

The visualization is temporary, e.g. it is not part of the station data. This means that it is not persisted when the station is saved

Linked Geometry

The Linked Geometry feature allows the user to load geometry from a shared repository (such as a file share) and let RobotStudio remember the source location. If the source file is updated, the RobotStudio user will be able to update the RobotStudio station with a simple click. The following functions are available for linked geometry:

- Add Link:

There are two ways to add a link to a geometry file:

1. Select the “Link to Geometry” checkbox in the “Import Geometry” dialog
2. Select an existing part in the station and select the “Add Link” command from the context menu. This will open a file dialog where the user can select the CAD file to link to.

The tooltip for a linked part in the layout browser will display the path of the file that has been linked to.

- Edit Link:

An existing link can be edited from the Part context menu by selecting “Edit Link”. The behavior is identical to the “Add Link” command.

- Delete Link:

An existing link can be deleted by selecting “Delete Link” from the context menu.

- Update Linked Geometry:

By selecting “Update Linked Geometry” from the Part, Component group or Station context menus, all corresponding parts will be updated from the source location if the timestamp on the file is newer than the timestamp stored in the station.

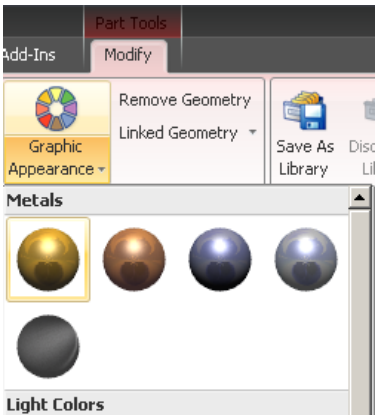
The result of the update operation will be displayed in the output window

Event Manager

It is now possible to export and import Event Manager tables as xml files. So the user can now transfer complex Event manager scenarios from one project to another.

Graphic Appearance

It is now possible to select from predefined materials:

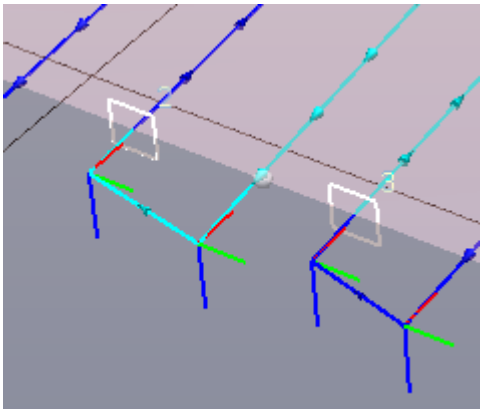


Visualization and manipulation of SetBrush instructions

Trig planes defined by the SetBrush instruction in RobotWare Paint are now visualized in the graphics. Each SetBrush instruction is drawn as a rectangle together with the brush number. It is possible to change the trig plane position by freehand move, using selection level *Taget/Frame*, or by using the Modify Instruction tool.

The path segments between trig planes are drawn with different colors. Each brush number has a color associated with it.

Note: In the RobotStudio user interface it is not possible to change the color that is associated with a brush number, but it is possible from Painting PowerPac.



Screen Recorder

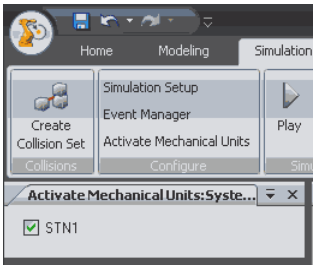
It is now possible to record a simulation in time slice mode. For each time step the recorder shall grab a frame.

This will allow smooth simulation movies to be recorded independently of the hardware used.

The user is able to select which codec he wants to use for the recording. The codec installed on the PC shall be available for selection

Reachability & AutoConfiguration

Reachability and AutoConfiguration is now supported for external axis (positioners) as well. **Note:** it important to active the external axis Mechanical Units before using those functions!



HOME, BACKUP, SYSPAR Shortcuts

There are shortcuts in the dialogs for saving/loading modules, backups and configuration files for the respective default directories for the VC.

Support for new SxTPU3

RobotStudio now supports the new SxTPU3. It will automatically detect which TPU version that is used by the system and start the right one when the user selects to start the Virtual FlexPendant.

Operator Window

The FlexPendant Operator Window is available in RobotStudio. It will display in Offline mode the same output as displayed on the Virtual FlexPendant Operator Window.

Online: EPS Wizard

It is now possible to upload the EPS configuration to the EPS Wizard from the controller to the EPS Wizard.

Application Development

Add-Ins can be loaded from Common Files

RobotStudio now scans the folder *C:\Program Files\Common Files\ABB Industrial IT\Robotics IT\RobotStudio\Addins* for *.rsaddin* files.

In the *.rsaddin* file it is possible to specify the location of the Add-In DLL.

If Add-Ins are located in a folder outside the RobotStudio installation folder, they will be able to be discovered and loaded when you upgrade to a newer RobotStudio version.

API Improvements

For a detailed list of new and improved classes please refer to the chapter API Improvements.

Solved Limitations

VSTA

Exception is always thrown when calling Body.CreatePolyline and Body.CreateSpline from VSTA (PDD7634)

When calling a method which takes an array of structures as parameter (for example Vector3[]), an exception is thrown. This has now been corrected.

A method that take an array as parameter, or returns an array can now be called from VSTA.

Offline

Kinematic baseframe in Mechanism Modeler

Defining a joint which is not perpendicular to any of the baseframe vectors causes the moved link to be positioned in wrong way.

Workaround:
Select the mechanism and execute modify mechanism.
Select the Frame node and select "Add Frame"
In the "Create Frame"dialog, check "Set as Kinematic BaseFrame" and leave the transform as is (position = 0,0,0 rotation = 0,0,0)
Compile mechanism.

*The Encoder can connect to the wrong part when there are more than two parts on the Conveyor

The system parameter Start Window is not handled correct. If there is more than one part inside the Start Window when dropping a workobject, the encoder will not connect to the next part, but instead connects to the part closest to the conveyor base frame.

UIShow stops program execution

Running a program in RobotStudio with the RAPID instruction *UIShow*, stops with an error, if the Virtual FlexPendant is not started. The error message *Unsupported instruction UIShow* will appear in the log window.

Limitation for the function *Create Workobject from Frame* for non-zero task frame

The function *Create Workobject from Frame* will not behave as expected in a station with a non-zero task frame.

The Check Reachability function lacks support for track motions

Copy to path in another Task creates new data even if it already exists

When you copy a path to another task, then new wobj/tooldata/weldata/ seamdata is created even if the data already exists.

AutoConfiguration does not support positioners

Wobj argument not updated, when dragging wobj0 targets to a new workobjec

VC synced Robtargets used in Moveinstructions without \Wobj:=xxx argument, did not update the Moveinstruction when dragged to another workobject.

Online

Error Message: *You are denied write access. …*

If the following error message appears "*You are denied write access. Access to <system name> on <controller name> is granted to user Unknown (user location unknown)*" then log off and log on again to workaround the problem.

Controller connection is lost after computer recovers from sleep mode

The connection to the virtual or real controller may be lost or in an undefined state after the computer recovers from sleep mode. However, obtaining mastership of the controller again is only possible through restart of RobotStudio..

RAPID Watch Online requires focus change to update

There is a new refresh function available in the context menu of the RAPID watch window, which allows to update the variable values.

Go Offline is not working

Known Limitations

An asterix (*) indicates new information since Release Notes 5.11.01.

General

Installing a license for RobotStudio Premium removes trial licenses for PowerPacs

When installing a RobotStudio license for the Premium functionality, the trial licenses are removed. This means that possible remaining trial time for features not part of the installed license, e.g. PowerPacs, will no longer be available. The current behaviour implies that in order to test a PowerPac for free you must do it within the trial time of RobotStudio (30days).

Compatibility of RobotStudio Library and Stations with older RobotStudio versions

RobotStudio is not upward compatible, i.e. it is not possible to load RobotStudio 5.12 libraries or stations into RobotStudio 5.11.

Online

FlexPendant Viewer running with automatic reloading

When having FlexPendant Viewer running with automatic reloading of the screens and at the same time jogging the robot with the joystick the robot jogging might halt when the FlexPendant Viewer reloads.

Backup for Paint systems does not create backup of the PIB board

The Backup function of RobotStudio does not create a backup of the PIB board of the IRC5P system.

***Workaround**: Create the backup of the PIB board with the FlexPaint Pendant using an USB-stick.*

Go Offline does not work for Paint systems

The Go offline function will not create a working Virtual controller system for Paint system unless the Paint package I/O option is set to Simulated.

Restart of Controller when connected through the service port

Re-connection of controller may fail when a controller is restarted from a service port connection.

Generating many signals using Add Signals tool may fail.

Adding many signals in one operation (>100) may fail and display the error message *Failed to apply changes to controller* .

***Workaround**: After start of RobotStudio, the first thing to do is to launch the Add Signals tool and generate the amount of signals you want. Do **not** expand the Configuration node of the browser and do **not** launch the Configuration Editor until the signals have been generated.*

The connection type of the Controller Status window always shows LAN

The connection type of the Controller Status window always shows LAN, also when the controller is connected through the service port .

I/O Viewer is not refreshed after controller restart

When looking at I/O Signals launched for the entire I/O System this works just fine. However, due to a design limitation it is not possible for I/O Windows launched by Bus or Unit to be updated after a controller restart .

Task activation in Offline and Online

When starting program execution from the RAPID Editor, the tasks currently activated in the controller will be started. This applies both to Offline and Online controllers.

For Offline controllers, the active tasks are defined in the *Setup Simulation* dialog. This setting only applies to the Simulation Play button. The task settings of the controller will not be used in the Offline case.

Offline

Lack of Virtual Controller support for the Paint systems

Paint systems that are configured using the Paint package I/O option Discrete, Compact or Fieldbus, will result in a SysFail state.

***Workaround**: Re-create the system with the simulated I/O option.*

No Virtual FlexPaint Pendant available

There is no Virtual FlexPaint Pendant available for Virtual Controller systems with paint robots.

***Workaround**: Use the regular Virtual FlexPendant instead.*

Hidden main entry point for Paint systems

Controller systems for Paint robots (IRB5XXX) has a hidden main procedure to handle the so-called *job-queue*. This is why the user must define a new entry point, e.g. *main2,*

using the *Setup Simulation* tool to avoid conflicts when working with a paint robot in RobotStudio.

Only single robot setups supported for Paint systems

It is possible to create the system with System builder for both single & multi paint robot systems using Virtual Key & Paint option. But a System failure will occur starting the Multi paint robot system. Only Single paint robot system can be started.

r

Not possible to open VirtualFlexpendant when mediapool and virtual system on networked drive

By default, the .NET Framework 2.0 does **not** allow execution of code from a remote file system. This means the VFP will cause a security exception and crash if the media used by the system resides on a network share.

***Workaround**: To resolve this, the user must explicitly grant the required permissions:*
*1. Open Control Panel / **Administrative Tools** / **Microsoft .NET Framework 2.0 Configuration***
*2. Open **My Computer** / **Runtime Security Policy** / **Machine** / **Code Groups** / **All_Code***
*3. Right-click on **All_Code** and select **New**...*
*4. Type a name for the code group (for example "RobotWare") and click **Next***
*5. In the condition dropdown select **URL**. Type the path to the RobotWare location, for example Z:\RobotWare* and click **Next***
*6. Accept the default (**FullTrust**) and click **Next**.*
*7. Click **Finish***

Not possible to use Create Path from Curve with a jointtarget instruction template*

It is not possible to create a path from a curve when a jointtarget instruction template is active, for example MoveAbsJ, using the the Create Path from Curve tool. The Apply button will be disabled until an instruction template based on robtarget is selected.

This behavior is by design but is not documented.

The Virtual Controller may fail to start on 64-bit editions of Windows XP and Vista

The Virtual Controller may fail to start on 64-bit editions of Windows XP and Vista (PDD DSE8481).

*Circular Conveyor Tracking not supported

RobotStudio does not support tracking of circular conveyors. Only linear conveyors are supported.

*Compiling a Conveyor Mechanism does not disable the Compile button

After compiling a conveyor mechanism, using the *Create Mechanism* tool, the *Compile Mechanism* button is not disabled. If the user presses the *Compile* button again, without changing anything, another identical conveyor mechanism will be created .

*System in *Guard Stop* state in *Automatic* mode after run-time error

Certain run-time errors may cause the controller system to enter *Guard Stop* state in *Automatic* mode. This is the same behavior as in a physical robot controller system. This typically happens when a run-time error related to Conveyor Tracking occurs. A simulation cannot be started when the controller is in this state.

Workaround: *To reset the controller state, open the Control Panel window and first switch to Manual mode, and then back to Automatic mode.*

***Conveyor Tracking programs must be started with the *Simulation-Play* button**

It is not possible to successfully run a RAPID program with Conveyor Tracking from the Virtual FlexPendant or from the RAPID Editor. The reason is that RobotStudio must simulate the *Conveyor Encoder Unit* in order to provide the required I/O signals to the system. This is only possible when running a simulation.

Workaround: *Start the simulation with the Simulation-Play button of RobotStudio instead of the Virtual FlexPendant or the RAPID Editor.*

***The same part can only be attached once on a Conveyor**

It is not possible to attach the same part on a conveyor more then once.

Workaround: *Import the same part several times, or copy and paste the part in the Layout browser, before attaching them to the conveyor.*

Note: *The part must not be attached to the conveyor during the copy and paste operations, then the copy will get the wrong transform.*

Not possible to Modify System for Pack&Go file.

It is not possible to use the function Modify System of the System Builder for a system that uses a mediapool embedded in a Pack&Work file.

Workaround*: Copy the mediapool to the common Mediapool folder, and create the system from the backup.*

Cannot create a system from layout with IRBx and IRBTX004

The function Create System from Layout will use the installed Trackmediapools on the users system. This might fail due to incompatibility between RobotWare vs. Trackmediapool version. The user should make sure to use the latest Trackmediapool delivered on the RobotWare 5.12 DVD.

Workaround*: Use System Builder to manually create a system with the correct mediapool combination .*

Opening a station with a non-coordinated track requires re-attaching the robot

When opening a station with a non-coordinated track motion, RobotStudio will not recognize the setup correctly.

Workaround: *The robot needs to be detached and re-attached to the track motion to make the system work as expected*

The FlexPendant Operator Window may behave unexpectedly

The FlexPendant Operator Window is available in the Offline mode and shows the output of the FlexPendant operator window. In certain cases, it may not display the complete text displayed on the FlexPendant. In addition, user interaction with the operator window may cause a different result than on the FlexPendant. The operator window is disabled by default, but can be enabled from the RobotStudio Options dialog.

Absolute Accuracy may cause the VC to miss the programmed position

The robot will not go to the programmed location if the controller has the Absolute Accuracy option activated and parameters from a real robot. The virtual robot in RobotStudio will move to fake targets in the same way as the real robot. The reason is

that the robot models in RobotStudio are nominal and do not correspond to the real, physical robots calibrated with Absolute Accuracy parameters.

***Workaround**: Reset the Absolute Accuracy parameters for the virtual system.*

Error Message: *Sync. to Station completed with errors*

Error Message: Sync to Station completed with errors: New data <name> <type> has same name as existing object in same block <routine>.

When this error message appears, there is a storage type mixup between data already stored in RS and in the VC. Because of this, and per design, the data is not considered the same data.

***Workaround**: 1. Ensure all data declarations have the same definition in RS as in RAPID (there is no user interface for this).*

2. Sync to station should now work.

3. Sync back to controller, and remember to change the data declarations back to what you want.

Move/Copy of Virtual Controller systems

Warm-started systems cannot be moved to another location and/or PC. This will result in a non-working VC.

A typical symptom of the problem is that the Virtual Controller reports *Failed to retrieve procedure.*

Workaround and recommended method of working:

1. Use *'Pack & Go'* to pack the station and system backups in a zip file.
2. Use *'Unpack & Work'* to unpack the zip file created by *'Pack & Go'*.

Array of robtargets, tooldata and workobjects are not supported

RAPID programs containing arrays of tooldata, robtargets and workobjects are not supported, i.e. they will not be synchronized to the station.

LOCAL declarations in RAPID are not supported the Paths & Targets browser

RobotStudio does not support LOCAL declarations of data or routines. RobotStudio will show an error message if such declarations are used.

The RAPID functions Offs and RelTool are not fully supported

RobotStudio doesn't fully support instructions using Offs or RelTool functions. They will be synchronized and will appear in the element browser, but commands such as "View Tool at Target" and "Locate Target" will not work. Targets used in the instructions will not be visible in the graphics.

Error message starting system with IRB260/660

Starting a system with IRB260/660 gives you an error message: "The number of joints is different between the model and VC". The reason is that the IRB260/660 is modeled with six joints in RobotStudio of which two are locked, but has four joints in the VC .

Incorrect error message "IRBxxx: Could not change motor state"

When starting the VC, the error message "IRBxxx: Could not change motor state" may appear in the output window. This message may be displayed even though the VC has started successfully.

Working range of IRB340

In some cases, it may be possible to Jump to Target and get Configurations for targets that are outside the working range of IRB340. This is due to the working range being defined as a cylinder and not only defined by the joint limits. It is however not possible to jog the robot to these targets.

Path handling of instructions with multiple joint targets

The path functions Rotate, Translate, and Mirror do not work as expected with instructions containing via points as jointtargets. The functions will leave the jointtargets as is. Interpolate Path gives an Unknown Error and Tool Compensation reports an error message.

Process time is displayed only for Simulation - Play in Time Slice mode

This is the only combination for which a correct cycle time can be guaranteed when custom mechanisms are involved in the simulation. It is only in Time Slice mode that RobotStudio controls the time and can synchronize the execution of the Virtual Controller with custom mechanisms. For simulations that only involve robot motion, the cycle time is correct for other combinations as well (RAPID Editor – Play and FreeRun). The Process Timer will turn yellow if the process time cannot be guaranteed.

Minor difference in process time of "*Simulation Play*" and "*RAPID Editor Play*"

The cycle time deviation between "Simulation Play" and "RAPID editor" is 0.05 s (constant). The difference is due to the program execution starting in different ways in the two scenarios. The play button of the RAPID Editor starts program execution in the same way as the FlexPendant, whereas the play button of the Simulation toolbar uses a slightly different mechanism. When executing program from the RAPID editor, it takes a small amount of time for RobotStudio to be aware that the simulation has started, which is why the "RAPID Editor" cycle time is 0.05 s smaller. The process time of the "Simulation" play is more accurate.

Event Manager: Simulation cannot be triggered by analog system signals

The event manager only supports analog station signals, not analog system signals.

Virtual Flex Pendent: Emergency Stop button

When the emergency stop button is pressed on the Virtual FlexPendant, it cannot be reset through the VC Control Panel. The button must be reset on the Virtual FlexPendant.

System From Layout requires custom made track motion to be saved as library

The System From Layout requires the any custom made track motions used to be saved as library.

Baseframe incorrect for robot with pedestal on track motion

Having a robot on track with a pedestal causes a wrong baseframe written into the controller configuration database (MOC). The track must be rebuilt with Mechanism Modeler if a pedestal not part of the distributed track motion libraries is to be used.

***Workaround**: Adjust the track position manually in RobotStudio and answer No to the baseframe update question that appears when restarting a VC.*

Graphics and Geometry

The Healing option may increase size of CAD models

The healing option may be used during CAD import to try and heal CAD-models. For some CAD-models the size is increased a factor of ten.

***Workaround**: Uncheck the Healing option in the Import Geometry dialog or the CAD-converter.*

DirectX may require manual installation

The DirectX components that are installed with the Full installation of RobotStudio have been seen to require manual installation on certain computers.

***Workaround**: Install DirectX manually. It can be downloaded from http://www.microsoft.com.*

Virtual FlexPendant impairs performance when on top of graphics viewer

The control panel of the Virtual FlexPendant (VFP) might affect the performance of the graphical window if placed inside it. If this is the case on your computer, make sure to set the display mode of the VFP to simple mode. This is done by clearing the *'Enable transparency'* option in the *RobotStudio Option* dialog (a restart of the VFP is required after changing mode). Refreshing the graphical view might however still be somewhat delayed, especially when moving the VFP rapidly over the screen.

Problems when undoing Boolean operations on Geometry

Undoing a Boolean operation might not succeed. To recover from these problems, you need to delete the items that caused the problem.

Out of memory

The application might fail when out of memory due to the import of very large ACIS files or load of very large stations. There is no immediate workaround for this problem.

JointTargets for external axis

JointTargets for external axis are not visualized in the graphical window.

Direct3D limitations

The following two settings in the *'Graphics Performance'* dialog (Tools/Options) have no effect

- Cull back-facing triangles.
- Enable two-sided lighting.

***Workaround**: Select the graphical object in the object browser and open the "Graphics Appearance" dialog box (context menu) that handles these options per object instead.*

Use Direct3D on Windows Vista for improved performance

Windows Vista is optimized for Direct3d, which is why it is recommended to use it as the graphics renderer for RobotStudio. This can be changed in *RobotStudio → Main Menu → Tools → Options → Graphics → Renderer → Direct3D.*

Use CAD Converter when converting CATIA V4 files

It is recommended to use the CAD Converter when converting CATIA V4 files, instead of importing the files directly into RobotStudio using “Import Geometry”.

***Note:** The CATIA V4 converter requires a separate license.*

Visual Studio Tools for Applications

The RobotStudio API is not thread safe*

Access to the RobotStudio API is not inherently thread safe. Only access the API from the thread that your Add-In was called from by RobotStudio. If multiple threads manipulate the object model it can be left in an inconsistent state.

Problem with using PC-SDK event handlers from VSTA*

The PC-SDK events are raised on a separate thread. When running a VSTA Add-In from RobotStudio it is not possible to call back to the RobotStudio API from a PC-SDK event handler directly as the call will then be made from another thread.

There is no verified method to dispatch the call to the RobotStudio main thread.

Workaround:

For the events DigitalSignal.Change and AnalogSignal.Change, the workaround is to use the EvenTable class in RobotStudio. Add an EventTableTriggerIO object to the event table, and connect it an action that should be done when a digital or analog signal is changed. The action can be to execute a VSTA macro, or set another signal.

From a hosted Add-In it is possible dispatch the call from the event handler to the main thread by using for example Control.Invoke.

Properties and methods that use the type System.Drawing.Color will not work in VSTA.

This is a limitation on the Visual Studio Tools for Applications (VSTA) environment.

***Note**: There is a new VSTA-class VSTABridge that can be used to work around this problem, see API documentation.*

Static events cannot be called from applications developed in VSTA.

This affects for example the Simulation – Tick event.

***Workaround**: Create a standard add-in if static events are to be used. Alternatively, use the VSTABridge class that can workaround this problem, see API documentation.*

Debugging of VSTA Applications

When debugging a VSTA application that adds menu items to the RobotStudio environment, then the menu will not be removed when the program execution stops. This may cause multiple entries of the same menu to be added in RobotStudio. This only affects VSTA add-ins being debugged and not completed VSTA add-ins.

***Workaround**: Restart RobotStudio to remove the extra menus.*

VSTA Library add-ins not available

In the Add-ins browser there is a folder for so-called VSTA Library add-ins. This feature is not available.

Use Visual Studio 2005 Express for advanced add-in

The purpose of VSTA is to write custom actions and minor utilities. For advanced add-in development use Visual Studio 2005 Express that can be downloaded free of charge from *http://msdn.microsoft.com/vstudio/express/.*

RsLoadData does not work from VSTA

(CQ7935)

VSTA limitation

The 'FindDataDeclarationsByType' method used in VB.net throw an exception Limitation added to API doc.

Corrected "Product Defect Document" RS5.12 PDD

Defining a joint which is not perpendicular to any of the baseframe vectors causes the moved link to be positioned in wrong	DSE6788
VSTA	DSE7634
Programpointer may not be set in program editor	DSE7754
Creating a pose (Modify Mechanism) for a device with more than 6 joints fails	DSE8158
Moving or copying instructions with Drag&drop changes order	DSE8164
5.11 - Bad Performance on RSO-part	DSE8266
RS5.11: ambigous HMI status after PC has been in sleep mode (5.12)	DSE8255
Create System From Layout restart problems	DSE8287 & DSE8288
RS 5.11 / Version Number EPS Addin is 5.10.9000.6 (5.12)	DSE8291
Install RS in 64 bit platform, try to create a system, an error message throws.	DSE8481
RS 5.11 / Wrong required diskspace computation in customized installation	DSE8254
Status in the Recent Controllers list not refreshed (5.12)	DSE8319
RS closes window when aborting	DSE8344
RobotStudio crashes when restarting after a laptop PC has been in sleep mode (5.12)	DSE8353
unexpected behaviors with output FP Operator window (5.12)	DSE8356
informations missing in output FP Operator window	DSE8359
new output window "FP	DSE8360
RAPID Watch Online requires focus change to update	DSE8362
Screenshot option not enabled until RS Options dialog is opened	DSE8366
No list of last virt. Controllers in RS (in Free mode)	DSE8378
RAPID Watch Online does not update in continuous execution mode	DSE8361
RS 5.11 missing info about imported controller list	DSE8397
Very long response time while pressing the 'Import geometry' but	DSE8406
RS 5.11 - While defining a work object using the 3 pts method, it is not possible to change the selection level nor the snap	DSE8476
Import Instructiontemplate throws exception	DSE8487
Create Wobj from Frame wrong	DSE8488
RAPID Editor has problems displaying latebinding syntax	DSE8489
No menu option to delete individual RAPID data watches	DSE8520
Compare To function allows display of NoView files	DSE8521
Cannot create a system from layout	DSE8530
Open station with cold started controller does not contain a task in station	DSE8541
RS file transfer window shows hidden files on PC	DSE8568
RS controller status connection type shows always LAN	DSE8570
Position update problem in RobotStudio	DSE8578
Mechanism Joint Jog with external axis	DSE8579
IRB7600 325kg 3.1m not fully supported in RS 5.11.01	DSE8580

Crash of RobotStudio when adding station signals	DSE8602
Drag out of Componentgruoup hangs RS	DSE8604
RS 5.11.01 - Error in collision detection function with IRB660	DSE8614
Inventor Cad converter does not recognize entities	DSE8624
RobotStudio 5.11 does not reconnect to the controller	DSE8625
Shift Taskframe crash RS	DSE8644
RS5.11 not possible to setup a IRB6620 with Track (IRBT6004)	DSE8681
RS5.11	DSE8682
RS5.11	DSE8685
UAS application grants not listed in RS 5.11.xx GrantEditor	DSE8700

Supporting Information

How to automatically create a system with external axes

The recommened way to create a system with external axis is to use the wizard *Create System From Layout* that can be started by pressing the button *Add Robot System -> --- from Layout.* The function will analyze the contents of your station and create a matching system. Simply import the desired robots, positioners and track libraries, and run through the wizard.

Tracks

The following tracks with lengths from 1.7 to 19.7 meters are supported. The track can run in a separate task or in a robot task. The system allows 1-3 tracks per task (dependent of the TCP manipulator type).

IRBT4003
IRBT4004
IRBT6003
IRBT6004
IRBT7003
IRBT7004
RTT_Bobin
RTT_Marathon
Paint Rail

Note: *IRBTx004 can only be used in the first task. Only one track of this type can be used per system. Also the mediapool Track.5.10.0003 is used for by the feature "Create System From Layout" regardless of the selected RobotWare version. Create the system manually if another version of the Track mediapool is desired, see Section "How to manually set up a system with track motion of type IRBTx004?"*

Supported external axis configurations

Combination of IRB, Track Motion & Positioner	A	B	C	D	K	L	2xL	R
1 IRB (positioner in same task)	✓	✓	✓	✓	✓	✓	✓	✓
1 IRB (positioner in separate task)	✓	✓	✓	✓	✓	✓	✓	✓
2 IRB (positioner in separate task)	✓	✓	✓	✓	✓	✓	✗	✓
1 IRB on Track Motion (positioner in same task)	✓	✗	✗	✗	✓[1]	✓	✓	✗
1 IRB on Track Motion (positioner in separate task)	✓	✗	✗	✗	✓[1]	✓	✓	✗
2 IRB on Track Motion (positioner in separate task)	✓	✗	✗	✗	✓[1]	✓	✗	✗

[1]) Manual mapping of mechanical units and joints required. Select the models from the station and combine them with the appropriate mechanical units with the System Configuration windows. Select OK to close the window before the next step. Open the window again and click on *Change...* in the lower right corner. Map the joints 1 to 3 for the positioner.Press OK to close the window. Reopen the window and point out the baseframe (use current station values).

✓ Combination is supported

✗ Combination is not supported

Track Motion Create System From Layout only supports tracks of type RTT and IRBTx003 in combination with positioners., i.e. IRBTx004 is not supported in combination with positioners.

Single task: Positioner in same task as robot

Separate task Positioner in separate task from robot. (Track motion is always in same task as robot.)

How to manually set up a system with track motion of type RTT or IRBTx003?

Track configuration files to be used for track motions or type RTT or IRBTx003 (x = 4, 6, or, 7) can be found in the Track folder.

Follow the instructions below to manually create a system with track motion of type RTT Bobin, RTT Marathon or IRBT4003, IRBT6003 or IRBT7003.

1. Create a system for the desired robot variant using *System Builder*, which can be found in the *Offline* tab of RobotStudio. The system must include the corresponding *Additional axes configuration option*:
   a. Select the desired robot variant
      i. In the *Create New Controller System Wizard* of the System Builder, go to the Modify Options page,
      ii. expand *DriveModule1 → Drive module application → ABB standard manipulator*, and,
      iii. choose the appropriate robot variant, e.g. *IRB6620*.
   b. Select the corresponding *Additional axes configuration*
      i. Further down on the *Modify Options* page expand *Additional axes configuration*,
      ii. expand *Add axes IRB/drive module 6600*, (or whatever option that matches your selected robot variant), and
      iii. select *770-4 Drive W in pos Y2* or similar option. The options differ slightly depending on the Additional axes configuration used: the exact drive and position may differ. The important thing here is to select one drive in any position.
   c. Press Finish.
2. Add the system to the station.
3. Open *System Configuration* in RobotStudio and add the corresponding track configuration file of the desired track motion model by pressing the *Add* button, see below*. The system will restart when pressing the *OK* button.
   ***Note**: The system will end up in System Failure state unless the matching "Additional axes configuration" is selected.*
4. Point out the desired track model when the system asks for a mechanical unit for TRACK. Either choose one of the suggested models or browse for a different track.
5. Open System Configuration again, select the *ROB_1* node, and set *BaseFrame Moved By* to *TRACK*. The system will restart when pressing the *OK* button.

*** How to select the correct track configuration file?**

The track configuration files are available for different track lengths and different tasks. The track length is encoded in the name of the configuration file, e.g. if the track length is 4.7 m then the configuration file to pick is called *TRACK_4_7.cfg*.

If the track is used for a robot in a MultiMove system, then the task number must also be taken into account, e.g. if the track length is 19.9 m and the robot attached to the track is connected to task 4 of the MultiMove system, then the configuration file to pick is called *TRACK_19_9_Task4.cfg*.

How to manually set up a system with track motion of type IRBTx004?

For configuration of tracks of type IRBT4004, IRBT6004 or IRBT7004, there are additional options mediapools in the *Mediapool* folder installed in the same folder as RobotWare (..*%ProgramFiles%\ABB Industrial IT\Robotics IT\MediaPool*\ ). There are three versions of the Track mediapool that are installed with RobotStudio:
Track.5.09.0012, which supports RobotWare 5.09, *Track.5.10.0003* and
Track.5.10.0005, which supports RobotWare 5.10, and
Track.5.11.0001, which supports RobotWare 5.11 and
Track 5.12.xxxx which supports RobotWare *5.12 and* revisions of those releases.

Follow the instruction below to manually create a system for the IRBTx004 track motion.

1. Create a system for the desired robot variant using *System Builder*, which can be found in the *Offline* tab of RobotStudio. The system must include the corresponding *Additional axes configuration option*:
   a. Add the additional options mediapool for IRBTx004
      i. In the *Create New Controller System* Wizard of the System Builder, go to the *Add Additional Options* page,
      ii. select the "…" button and browse to the key (.kxt file) located in the mediapool *Track.5.11.0001*. The mediapool supports both RobotWare 5.10 and 5.11. (Use *Track.5.09.0012* toghether with RobotWare 5.09, and *Track.5.10.0005* with RobotWare 5.10).
      iii. Press the arrow → to add the option.
   b. Select the desired robot variant and track motion to use
      i. Proceed to the *Modify Options* page,
      ii. expand *DriveModule1 → Drive module application → ABB standard manipulator*, and, choose the appropriate robot variant, e.g. *IRB6600*.
   c. Select the corresponding additional axes configuration
      i. Further down on the *Modify Options* page, expand *Additional axes configuration,*
      ii. expand *Add axes IRB/drive module 6600*, (or whatever option that matches your selected robot variant), and

iii. select *770-4 Drive W in pos Y2* or similar option. The options differ slightly depending on the Additional axes configuration used. The drive and position may differ. The important thing here from a RobotStudio point-of-view is to select one drive in any position.

iv. Select the desired robot variant and track motion to use

i. Scroll down to TRACK, expand it and select *Drive Module 1 → Track Motion Type → IRBT 6004 → Irb Orientation on Track → Inline → Select Track Motion Length → 1.7 m*, or whatever variant you prefer.

v. Press Finish.

2. Add the system to the station. The system will start.
3. When the *Select Library* dialog appears, press the *Other* button, browse to the folder *ABB Library/Track* and select the track motion library corresponding to your system.
4. Done!

Note: *RobotStudio will adjust the joint limits of the mechanisms to the limits of the VC. If the incorrect track configuration file is selected, this may cause mismatch between the used joint limits of the RobotStudio/VC and the geometry of the track motion model in RobotStudio.*

Code Snippets

The RAPID Editor of RobotStudio contains Code Snippets that are integrated with the Pick List. Code Snippets are pieces of code that can be inserted into the editor on user request. RobotStudio comes with a number of predefined Code Snippets, such as:

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

In addition, the user can create customized Code Snippets and add to the existing list of Code Snippets. The Code Snippets adhere to the Code Snippet format of Visual Studio

2005. Code Snippets can be edited in any XML editor such as Visual Studio 2005 Express, which can be downloaded free of charge from Microsoft at *http://msdn.microsoft.com/vstudio/express/*. Read more about creating your own Code Snippets in the MSDN article *'How to: Create a Basic Code Snippet'* at *http://msdn.microsoft.com/en-us/library/ms165394.aspx*

The RobotStudio .snippet files are stored per user and located in the folder

C:\<Documents and Settings>\<user name>\RobotStudio\Code Snippets,

where the folder *<Documents and Settings>* may be configured to have different names, e.g. *Data*. It may also be translated on localized versions of Windows.

***Note**: The language specified in the .snippet file to be used in the RobotStudio RAPID Editor must be RAPID, whereas the Microsoft examples are targeted towards other programming languages. However, the structure and format are the same. See also the pre-defined .snippet files installed with RobotStudio 5.12.*

Instruction Templates

The *'Instruction Template Manager'* can be used to add support for instructions other than the pre-defined set that comes with RobotStudio by default. For example, a robot controller system with the RobotWare Dispense option comes with specialized move instructions related to glueing, e.g. DispL and DispC. The user can manually define instruction templates for these using the *'Instruction Template Manager'*. The instruction templates can be exported to XML format for later reuse.

For some common processes RobotStudio 5.12 comes with pre-defined XML files that can be imported and used for robot controller systems with the appropriate RobotWare options. The instruction templates provided add support for the following RobotWare options:

- Cap (Continuous Application Process)
- Disp (Dispense)
- Trigg (Fixed Position Events)
- Spot Pneumatic
- Spot Servo
- Spot Servo Equalizing
- Paint

The XML files provided contain both Move and Action instructions.

The instruction template files can be found in the *'Instruction Templates'* folder of the users RobotStudio folder.

***Note**: RobotStudio ArcWelding PowerPac is recommended when using RobotWare Arc.*

API Improvements

*New types**

ABB.Robotics.RobotStudio

`class CompressorInfo`	Contains information about a specific compressor (codec)

ABB.Robotics.RobotStudio.Stations

`class Irc5ConveyorInfo`	Represents conveyor tracking specific information about a Mechanism of type MechanismType.Irc5Conveyor. Such a mechanism is used to model a conveyor tracking device. It has only one joint, and this joint holds a number of attachment frames where tracked parts can be attached. Jogging this joint will move the attachement frames in the conveyor mechanical unit Base Frame X-axis direction.
`struct Irc5ConveyorWorkpieceInfo`	This struct contains the information needed to attach a workpiece part to a conveyor mechanism.
`class RsIrc5ConveyorEncoderUnit : ProjectObject`	Represents the IRC5 Conveyor Encoder Unit that is connected to a controller that has the option Conveyor Tracking. This class provides access to conveyor tracking specific controller system parameters and I/O signals.

** The following new types are not listed:*

Event argument classes corresponding to new events. Please refer to the documentation of the new events.

Enumeration types used by new types and members. Please refer to the documentation of the new types and methods.

Methods and properties of the new classes. Please refer to the documentation of the new classes.

Improved types

ABB.Robotics.RobotStudio.Environment

CommandBarButton	
`bool ResolveSubscriber { get; set; }`	Gets or sets whether the UpdateCommandUI and ExecuteCommand are sent to all subscribers (false) or just active window (true)

DisplayCommandGroupEventArgs	
`bool ResolveSubscriber { get; set; }`	Gets or sets whether the UpdateCommandUI and ExecuteCommand are sent to all subscribers (false) or just active window (true)

`public void AddItem(`
`  string caption,`
`  Object tag,`
`  Image image,`
`  bool isChecked,`
`  bool isEnabled,`
`  string helpText`
`)`
	Adds a menu item to the command group.

`public string Id { get; }`	ID of the corresponding CommandBarButtonGroup

WindowCollection	

`public Window[] FindWindows(`
`  Predicate<Window> match`
`)`
	Find all windows that matches a given predicate

`public void RemoveAll(`
`  Predicate<Window> match`
`)`
	Removes all windows that matches a given predicate

ABB.Robotics.RobotStudio

Options	

`static void SetDefaultValue(`
`  string section,`
`  string key,`
`  long value`
`)`
	Sets a default 64-bit integer value of for an option with the specified key.

`static void SetValue(`	Sets an 64-bit integer value for an

`string section,`
`string key,`
`long value`
`)`
	option with the specified key.

ScreenRecorder	
`static void AddFrame()`	Adds a screenshot of the window specified in Start() to the current recording.
`static void AddFrame(` `Bitmap bitmap)`	Adds a frame to the output video. The bitmap will be automatically disposed once it has been encoded.
`static CompressorInfo[]` `GetCompressors(ScreenRecorderFileFormat fileFormat)`	Returns information about installed and available compressors (codecs) for a given file format.
`static ScreenRecorderFileFormat` `FileFormat { get; set; }`	Gets or sets the file format to record to.
`static double StartTime { get; set; }`	Gets or sets the time that shall elapse before the recording starts.
`static double StopTime { get; set; }`	Gets or sets the time that shall elapse before the recording stops.
`static long TotalFrames { get; }`	Returns the total number of frames recorded in this recording session.

`static bool Start(`
`string filename,`
`IntPtr hWnd,`
`bool automatic`
`)`
	Starts recording the content of window specified by its handle, to the specified file.

ABB.Robotics.RobotStudio.Stations

Station	
`ProjectObject[] GetAllObjects()`	Returns an array of all ProjectObjects that are a part of this station (including the Station itself). For performance reasons, geometry objects (all objects below Part) are excluded.

Texture	
`Texture(string fileName,TextureType type)`	Creates a Texture from an image file.
`Texture(Bitmap bitmap, TextureType type)`	Creates a Texture from a Bitmap.
`static Texture[] GetAllTextures()`	Returns an array containing all textures that are currently in memory.

Simulator	
`static event EvaluateTriggerEventHandler EvaluateTrigger`	Occurs when an event table trigger should be evaluated.
`static event ExecuteActionEventHandler ExecuteAction`	Occurs when an event table action should be executed.

RsIRc5Controller	
`BrushColorCollection BrushColors { get; }`	Returns a collection of colors that is used to visualize SetBrush RsActionInstructions in the graphics.
`RsIrc5ConveyorEncoderUnitCollection ConveyorEncoderUnits {get;}`	

RsActionInstruction	
`double FrameSize { get; set; }`	Gets or sets the size of RsActionInstructions that have a graphical representation, e.g. SetBrush.

RsLoadData	

`RsLoadData(`
`  double mass,`
`  Vector3 cog,`
`  Quaternion aom,`
`  Vector3 inertia`
`)`
	Initializes a new RsLoadData instance with the specified parameters.

Part	
`string Source { get; set; }`	Gets or sets the file that was the source of this part.
`DateTime SourceDateTime { get; set; }`	Gets or sets a time stamp of the last update from Part.Source (in UTC)

MeshFace	
`List<Vector3> MidPoints { get; }`	Gets or sets the midpoints used for snap.
`List<Vector3> CenterPoints { get; }`	Gets or sets the center points used for snap.

MechanismBuilder	

`public void SetAttachmentPoints(`
`  Matrix4[] attachmentPoints`
`)`
	Sets an array of matrices that will become the frames used for attaching parts and workobjects on a conveyor tracking mecahnism.

Mechanism	
`bool UsesCfx { get; }`	Returns true if the Cfx parameter is used by this robot model, otherwise false.
`Irc5ConveyorInfo Irc5Conveyor { get; }`	Gets conveoyr tracking specific extensions of a Mechanism of type MechanismType.Irc5Conveyor.

MechanismType	
`MecahnismType.Irc5Conveyor`	The mechanism represents a conveyor tracking device.

Material	
`void SetSimpleColor(` `  Color color)`	Sets the basic color and automatically calculate material properties like specular color, diffuse color etc.
`Color GetSimpleColor()`	Gets the basic color from the material.

Irc5ConveyorInfo	
`event EventHandler TrackingStateChanged`	Occurs when the conveyor changed the tracking state.

GraphicComponent	
`void DeleteGeometry(` `  bool preserveSnapPoints)`	Recursively deletes all CAD geometry.
`FrameCollection Frames { get}`	Gets a FrameCollection object that represents all the Frames on this Graphic Component object.

Face	

`void SetMaterial(Material material,SizeF scale)`	Sets the material of the face and re-scales the texture cooordinates.

Camera	
`bool Visible { get; set; }`	Gets or sets a value indicating whether the camera will be displayed in the graphics.

Frame	
`class Frame : ProjectObject, IHasTransform, IAttachableChild, IAttachableParent`	Frame now inherits IAttachableParent

Body	
`public BodyType BodyType { get; }`	Returns the type of the body.

ABB.Robotics.RobotStudio.Station.Forms

GraphicControl	
`class GraphicControl : Control, IDisposable`	Now inherits IDisposable.
`public Nullable<Vector3> ProjectPoint( int x, int y )`	Returns the point in 3D that corresponds to a window coordinate.
`public Nullable<Vector3> ProjectRay( Vector3 start, Vector3 direction, out ProjectObject hitObject )`	Performs a ray test

GraphicPickEventArgs	
`public TemporaryGraphic PickedTemporaryGraphic { get; }`	Gets the picked project object (can be NULL for example when picking the floor or a TemporaryGraphic object)

TemporaryGraphics	
`public TemporaryGraphic Copy()`	Returns a copy of this TemporaryGraphic object.
`public void GetBoundingBox( bool global, out Vector3 min, out Vector3 max`	Computes the (axis-aligned) bounding box for this object.

) 	

public void Highlight(
  bool highlight,
  Color color
)
	Highlights this object in the specified color.

public bool Pickable { get; set; }	Gets or sets a value indicating whether this object can be picked in the graphics.
public Object Tag { get; set; }	Gets or sets an object associated with the temporary graphic object. This provides the ability to associate an arbitrary object to a temporary graphic object.