

Release Notes
RobotStudio SDK

5.14.01

Revision: -

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

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

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

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

Additional copies of this document may be obtained from ABB at its then current charge.

© Copyright 2011 ABB All rights reserved.
ABB AB
Robotics Products
SE-721 68 Västerås
Sweden

Overview	5
1 Release Information	6
1.1 General	6
1.2 Contents	6
2 What's New in this Release?	7
2.1 Energy Quantity	7
2.2 Multiline string properties on SmartComponents	7
2.3 Controller I/O Signals does not need to have Access Level ALL	7
2.4 Face.GetU/VRange not limited to spline faces	7
3 What's New in 5.14?	8
3.1 General	8
3.2 Leveraging new features of RobotStudio	8
3.3 New types and methods	11
3.3.1 ABB.Robotics.Math	11
3.3.2 ABB.Robotics.RobotStudio.Environment	12
3.3.3 ABB.Robotics.RobotStudio	12
3.3.4 ABB.Robotics.RobotStudio.Controllers	14
3.3.5 ABB.Robotics.RobotStudio.Diagnostics	14
3.3.6 ABB.Robotics.RobotStudio.Stations	14
3.3.7 ABB.Robotics.RobotStudio.Stations.Forms	18
4 Late-breaking information	19
4.1 Overview	19
5 Problems Corrected	20
5.1 Solved Product Defect Documents (PDD) since RobotStudio SDK 5.14	20
5.2 Solved Product Defect Documents (PDD) since RobotStudio SDK 5.13	20

Table of Contents

6 Known Limitations	21
6.1 Development Environment	21
7 Installation Information	22
7.1 Hardware and Software requirements	22
7.2 Compatibility	22

Overview

Description

This document provides information about the new features, problems corrected, and installation of RobotStudio SDK.

Product Overview

RobotStudio SDK is included in the RobotStudio installation and can be used for free by anyone who wants to develop on the RobotStudio platform. It could be scripts, Add-Ins or applications that can provide new functionality and/or customized user interfaces in RobotStudio.

It is also possible to call PC SDK to access real or virtual IRC5 controller functionality from the Add-In.

Custom Smart Components with Code Behind can also be developed using the RobotStudio SDK.

Visit our web site at <http://www.robotstudio.com/community> for information and updates. RobotStudio, including RobotStudio SDK, can be downloaded from the Download section.

To learn more and get hints, visit the User Forum, where developers discuss software problems and solutions online.

1 Release Information

1.1 General

Release Name

The release name is RobotStudio SDK 5.14.01 and the build number is 5.14.4367.1032.

For information about RobotStudio, please refer to the document '*Release Notes RobotStudio.pdf*'.

Release Date

The release date is 22nd of June, 2011.

1.2 Contents

Overview

The installation includes software, documentation and tools as specified below.

Software

ABB.Robotics.Math.dll

ABB.Robotics.Environment.dll

ABB.Robotics.RobotStudio.Documents.dll

ABB.Robotics.RobotStudio.dll

ABB.Robotics.RobotStudio.Controllers.dll

ABB.Robotics.RobotStudio.Stations.dll

ABB.Robotics.RobotStudio.Stations.Forms.dll

Documentation

Reference Manual – RobotStudio SDK (Html Help).

After installation the documentation can be launched from the RobotStudio Help Menu, and Windows Start Menu\Programs\ABB Industrial IT\Robotics IT\Robot Studio 5.14\SDK.

Tools

LibraryCompiler.exe – A tool for batch creation for RobotStudio libraries such as SmartComponents and Mechanisms.

Visual Studio Project Templates

RobotStudio Add-In – template for a standard Add-In

RobotStudio Smart Component – template for a Smart Component with Code Behind and XML description.

2 What's New in this Release?

Overview

This section contains information about the new features in this release. Corrected problems are listed in chapter 5.1.

2.1 Energy Quantity

There is a new quantity representing Energy. You can use it by the new property `Quantity.Energy` and the enumeration value `BuildQuantity.Energy`.

2.2 Multiline string properties on SmartComponents

It is now possible to have a dynamic property on a SmartComponent that is of type String but contains multiple lines of text.

When a `DynamicProperty` of type String is tagged with the attribute `KnownAttributeKey.Multiline` it will be displayed using a multi line text box in the RobotStudio User Interface. Lines are separated by the Environment.NewLine character.

When building your SmartComponent with the library compiler use the `Multiline` attribute of the `DynamicProperty` element to specify a multi line string property.

2.3 Controller I/O Signals does not need to have Access Level ALL

RobotStudio can now change the value of an input signal in the Virtual Controller even if the Access Level is not ALL. For example you can now always change the value of a Station Logic signal that is connected to a signal in the Virtual Controller regardless of the access level.

Setting a signal in this manner is identical to setting a physical I/O signal (that is to apply +24V or 0V). The Access Level cannot stop +24V to be connected to a physical signal. Now the Virtual environment behaves the same.

Hint: Using PC SDK you can change a signals value in the same manner by using the property `Signal.InputAsPhysical`.

2.4 Face.GetUVRRange not limited to spline faces

The methods `Face.GetUVRRange` and `Face.URange` are no longer limited to return a valid result for spline faces.

3 What's New in 5.14?

3.1 General

Overview

This section contains information about the new features and problems corrected in 5.14.

Section 2.2 describes how you can take advantage of new RobotStudio features in your application. Section 2.3 contains a list of new methods and types sorted by namespace.

3.2 Leveraging new features of RobotStudio

Fluent UI

The RobotStudio 5.14 user interface is redesigned and now uses the Microsoft Office 2010 Fluent user interface. Applications built for previous version of RobotStudio that added user interface elements in the Ribbon does not need to be updated in order to take advantage of the new style.

The main difference compared to Office 2007 look and style, besides the visual appearance and color scheme, is the Backstage tab.

The backstage tab replaces the application menu of RobotStudio 5.13.

It is currently not possible for applications to add their own content to the backstage tab.

Document Manager References

References to files and folder can be added to a station through the new Station.Documents property.

You can use the enumeration ProjectDocumentType to decide if a reference to a file or folder, or an entire file, shall be added to the station.

Accelerated simulation

An option to run the simulation as fast as possible has been added to the Simulation Accuracy section of the RobotStudio options.

This option corresponds to the new property Simulator.FullSpeed.

It is possible to set the resolution (virtual time step size) and simulation speed (how fast the virtual clock is advancing) using the properties Simulator.Resolution and Simulator.SimulationSpeed.

When the simulator is running in full speed it will execute the next time step as soon as the calculation for the previous one is completed, as fast as the CPU allows. The predicted execution times for the Virtual Controllers will not be affected.

Simulation Step / Pause / Resume

The methods Simulator.Step, Simulator.Pause and Simulator.Resume, can be used to be used to step the simulation forward one time step at the time, and to pause and then resume the simulation.

ScreenRecorder support for Microsoft Windows Media format

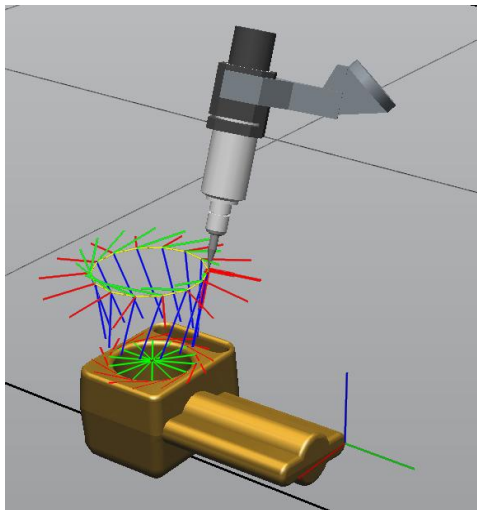
The ScreenRecorder class now has support Windows Media version 8 and 9. In addition, movies can be recorded using the H.264 / MPEG-4 encoder that creates high-quality recordings with a high compression ratio.

The ScreenRecorderFileFormat enumeration has the new elements Wmv and mp4.

Target Reference Frame

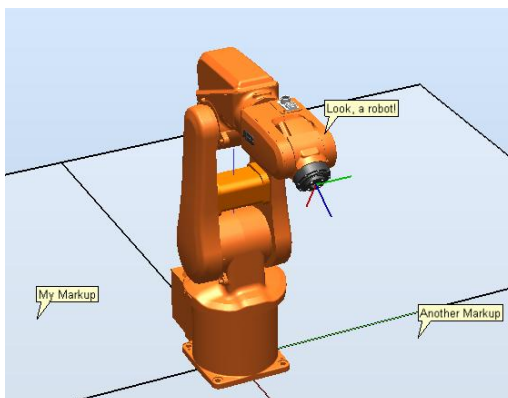
The reference frame of a target has now been exposed in the RobotStudio user interface.

Applications can access modify the reference frame using the property RsTarget.ReferenceFrame. The target reference frame allows a target to be modified relative to the original position. This is useful if your application lets the user create targets using a CAD-model. Your application can define its own specific relations between the current and original position. For example spin angle, and travel angle are terms used in arc welding.



Markups

The new Markup class together with the Station.Markups property can be used to annotate your station. The markups will be persisted in the station file, as opposed to temporary graphic text boxes.



AutoConfig options

The performance of the methods that retrieves a list of all available configurations has been greatly improved. It is now possible for applications to have interactive user interfaces which needs to constantly retrieve the available configurations.

There new overload

Mechanism. GetAllConfigurations(RsTarget target, RsToolData tool, Boolean includeTurns) lets you decide if you want to get only the base configuration, or also all configurations resulting from rotations of 4 and 6.

DataRecorder

The new Signals Analyzer feature is built on top of the DataRecorder API which allows applications to both provide their own signals which then be selected and analyzed in the user interface, and also lets applications subscribe to signal changes.

The DataRecorder class is the hub which collects data from all its sources, and distributes it to its clients, or sinks.

Use the class DataRecorderSource if you would like to provide your own signals to users, and inherit the base class DataRecorderSink to get notified when a signals changes value.

The class BuiltInDataRecorderSignals can be used to programmatically access the signals built in to RobotStudio.

In addition to I/O signals and the EventLog, the following signals can be analyzed:

Signal name	RW availability
<i>Controller Signals / <system name> / EventLog</i>	
<domain>	Any RW version
<i>Controller Signals / <system name> / Mechanical Units / I/O System</i>	
<all signals>	Any RW version
<i>Controller Signals / <system name> / Mechanical Units / ROB_x / Joint</i>	
J1-J6	Any RW version
Near Limit	Any RW version
<i>Controller Signals / <system name> / Mechanical Units / ROB_x / Target</i>	
Fine Point	Any RW version
Target Reached	Any RW version
<i>Controller Signals / <system name> / Mechanical Units / ROB_x / TCP</i>	
Maximum Linear Acceleration in World ¹	RW version ≥ 5.14
Orientation Q1-Q4 Current Workobject	RW version ≥ 5.14
Orientation Speed in Current Workobject	RW version ≥ 5.14
Pos X, Y, Z in Current Workobject	RW version ≥ 5.14
Robot Configuration cf1, cf4, cf6, cfx	RW version ≥ 5.14
Speed in Current Workobject	RW version ≥ 5.14
Total Motor Power	RW version ≥ 5.14
<i>Smart Components</i>	
<all signals>	<no RW needed>

¹ The signal 'Maximum Linear Acceleration in World' provides the maximum acceleration of the current move instruction.

3.3 New types and methods

Overview

This section contains information about the new types in the API, and existing types which has been extended with new methods.

3.3.1 ABB.Robotics.Math

New Types

<code>class LogicExpression</code>	Parses and evaluates a logic (boolean) expression
<code>class Matrix</code>	Arbitrary-sized matrix
<code>struct Ray</code>	Represents a 3D ray with an origin and a direction.

Extended Types

<code>class BoundingBox</code>	
<code>Double Distance(Vector3 point)</code>	Returns the distance between this BoundingBox and a point.
<code>static Double Distance(BoundingBox lhs, BoundingBox rhs)</code>	Returns the distance between two bounding boxes.
<code>BoundingBox Expand(Double amount)</code>	Returns a BoundingBox expanded by an amount in all directions.
<code>static Boolean Intersects(BoundingBox lhs, BoundingBox rhs)</code>	Returns true if two bounding boxes intersect.
<code>static Boolean Intersects(BoundingBox lhs, BoundingBox rhs, Double tolerance)</code>	Returns true if two bounding boxes intersect.
<code>class MathExpression</code>	
<code>String ErrorText { get; }</code>	If the expression is invalid (IsValid returns false), this property may return more information why.
<code>Double[] GetLinearCoefficients(String[] variables)</code>	Returns the linear coefficients, if this expression is linear in the set of given variables. Otherwise it returns null.
<code>Boolean TrySetVariableValue(String variable, Double val)</code>	Sets the value of a variable, which will be used the next time the expression is evaluated.
<code>struct Matrix4</code>	
<code>Matrix4(Double[] values, Boolean rowMajor);</code>	Creates a matrix from an array of 16 values.
<code>Boolean AlmostEquals(Matrix4 rhs, Double tolerance)</code>	Comparison method for vectors with tolerance.
<code>static Matrix4 RelativeMatrix(Matrix4 from, Matrix4 to)</code>	Returns the relative transform, between the two specified transforms.
<code>struct Plane</code>	

<code>static Boolean operator==(Plane lhs, Plane rhs)</code>	Equality operator
<code>static Boolean operator!=(Plane lhs, Plane rhs);</code>	Inequality operator
<code>struct Quaternion</code>	
<code>Double this[Int32 index]</code>	Array access.
<code>Vector2</code>	
<code>static Boolean operator==(Vector2 lhs, Vector2 rhs);</code>	Equality operator
<code>static Boolean operator!=(Vector2 lhs, Vector2 rhs);</code>	Inequality operator
<code>struct Vector3</code>	
<code>Double[] ToArray();</code>	

3.3.2 ABB.Robotics.RobotStudio.Environment

Extended Types

<code>class ApplicationMenu</code>	
<code>CommandBarControl</code> <code>ActiveBackstageControl</code>	Gets or sets the active tab of the Backstage (file) menu
<code>event EventHandler</code> <code>ActiveBackstageControl Changed</code>	Raised when the ActiveBackstageControl property changes
<code>class UIEnvironment</code>	
<code>static Boolean BackstageVisible</code>	True if the Backstage (File) tab is currently active, false otherwise.
<code>static event EventHandler</code> <code>BackstageVisibleChanged</code>	Raised when the BackstageVisible property has been changed
<code>static event ExceptionEventHandler</code> <code>Exception</code>	Raised when an exception is thrown by an event handler and caught by the environment.

3.3.3 ABB.Robotics.RobotStudio

New Types

<code>class DataRecorderBase</code>	A data recorder collects data from all its sources, and distributes it to its sinks.
<code>class DataRecorderSignal</code>	Represents a piece of information that can be recorded by a DataRecorder
<code>struct DataRecorderSignalInfo</code>	Represents information such as data Type and quantity of a DataRecorderSignal.
<code>class DataRecorderSinkBase</code>	A data recorder sink will get notified when signals subscribed by the data recorder is updated.
<code>class DataRecorderSourceBase</code>	A data recorder source provides

	information to a data recorder.
class DataRecorderSubscribeSignalResult	Represents information about the reason for success or failure to subscribe to a signal.
class ProjectDocument	Represents a document associated with a project.
enum ProjectDocumentFlags	Specifies values for the Flags property.
enum ProjectDocumentType	Specifies the type of a ProjectDocument
enum SignalDataType	Specifies the data type of a DataRecorderSignal
enum SignalInterpolationType	Specifies the type of interpolation for a DataRecorderSignal

Extended Types

class AttributeCollection	
Boolean TryGetValue<T>(String key, out T value)	Gets the value with the specified key.
Boolean TryGetValue(String key, out Object value)	Gets the value with the specified key.
enum BuiltInQuantity	
Power	
class ExceptionLogMessage	
String GetErrorString()	Returns a string containing the aggregated stack traces and exception messages of the associated exception, and its inner exceptions.
class Logger	
static void AddMessage(String message)	Adds an log message to the logger.
class Options	
static void Save()	Commits all options to disk.
class ProjectObject	
IEnumerable<ProjectObject> Children	Enumerates all child objects.
enum ProjectObjectChangeType	
UndoRedo	The object was changed as a result of an undo or redo operation.
class Quantity	
static Quantity Power	
enum ScreenRecorderFileFormat	
Wmv	Windows Media Video
Mp4	MPEG-4

<code>struct Vector4</code>	
<code>Boolean AlmostEquals(Vector4 vec, Double tolerance)</code>	Comparison method for vectors with tolerance.
<code>Double[] ToArray();</code>	Returns the elements (x,y,z,w) as an array.

3.3.4 ABB.Robotics.RobotStudio.Controllers

New Types

<code>class ControllerObjectSelectionTypes</code>	Gives access to the controller object types which can be selected in the Online or Offline browser.
---	---

Extended Types

<code>class ControllerObjectReference</code>	
<code>ControllerObjectType ObjectType</code>	Specifies the type of controller object that is referenced by a ControllerObjectReference.
<code>class ControllerReferenceCollection</code>	
<code>ControllerObjectReference this[Guid systemId]</code>	Gets the controller reference at the specified index.

3.3.5 ABB.Robotics.RobotStudio.Diagnostics

Extended Types

<code>class ApplicationLogger</code>	
<code>static String CurrentLogFile</code>	Returns the location of the log file for the current session.
<code>static void LogException(Exception ex)</code>	Adds an exception error message and flushes the log to disk.
<code>static void LogException(String message, Exception ex)</code>	Adds an exception error message and flushes the log to disk.
<code>static void LogException(Exception ex, Boolean toOutput)</code>	Adds an exception error message and flushes the log to disk.
<code>static void LogException(String message, Exception ex, Boolean toOutput)</code>	Adds an exception error message and flushes the log to disk.
<code>static void LogExceptionOnce(String message, Exception ex, Boolean toOutput)</code>	Adds an exception error message and flushes the log to disk.

3.3.6 ABB.Robotics.RobotStudio.Stations

New Types

<code>class BuiltInControllerSourceSignals</code>	Gives access to built in data recorder signals, that represents information
---	---

	from the controller.
enum BuiltInDataRecorderMotionSignal	Specifies information from the robot motion system that can be recorded.
class BuiltInDataRecorderSignals	Gives access to the identities of the built in data recorder signals.
class BuiltInSmartComponentSourceSignals	Gives access to built in data recorder signals, that represents information from SmartComponents.
class Markup	Represents a text and pointer markup displayed in the 3D graphics.
class SimulationDataRecorder	A data recorder than can record signals from a simulation. It is synchronized with virtual time.

Extended Types

class Body	
FaceCollection Faces	Gets the FaceCollection that the Shell belongs to.
WireCollection Wires	Gets the WireCollection that the Shell belongs to.
static Body CreateFromFace(Face face)	Creates a new Body consisting only of a copy of the given Face
class Camera	
Double FieldOfView	Gets or sets the field of view, in degrees.
Vector3 UpDirection	Gets or sets the up direction vector.
class CollisionDetector	
static CollisionType CheckCollision(GraphicComponent object1, GraphicComponent object2, Double nearMiss, DetectableUsage detectableUsage, out Part part1, out Part part2)	Calculates whether two objects intersect, or whether an object intersects any other object, and returns the intersecting Parts.
enum DetailLevels	
Default	
class Face	
Boolean GetCenterPoint(out Vector3 centerPoint)	Returns the center point of a planar surface.
Edge[] GetEdges()	Returns an array containing the edges of this face.
class GraphicComponent	
Boolean Detectable	Gets or sets a value indicating the the component can be detected by sensors.
void Highlight(Boolean value, Color color)	Highlights the object using a specified color.
enum Irc5ConveyorTrackingState	
PassedStartWindow	

class Material	
Material Clone()	
class Mechanism	
Matrix4 Model BaseFrame	Returns the nominal base frame of this model.
ConfigurationData[] GetAllConfigurations(RsTarget target, RsToolData tool, Boolean includeTurns)	Returns reachable arm configurations for the specified target.
Matrix4 GetCalibrationPosition(Int32 jointIndex)	Returns the calibration transform for a joint.
Matrix4 GetJointTransform(Int32 jointIndex)	Get the transform for a joint The index of the joint for which the transform shall be retrieved.
Boolean GetParentJoint(GraphicComponent link, out Int32 jointIndex)	Returns the joint index of a link, or -1.
Boolean GetParentLink(Int32 jointIndex, out GraphicComponent link)	Returns the parent link of a joint.
class Mesh	
MeshPart Closest(DetailLevel s detailLevel)	Returns the available detail level, which is close as possible to the one specified.
void Save(String fileName)	Saves this Mesh to an .rsgfx file.
static Mesh Load(String fileName)	Loads a Mesh from an .rsgfx file.
class MeshBody	
MeshBody Clone()	Returns a deep clone of this MeshBody.
class MeshFace	
MeshFace Clone()	Returns a deep clone of this MeshFace.
class MeshPart	
MeshPart Clone()	Returns a deep clone of this MeshPart.
class Part	
void Facet(DetailLevel s detailLevel s)	Re-creates the graphical representation of this Part.
void Facet(DetailLevel s detailLevel s, Boolean surfaceModel)	Re-creates the graphical representation of this Part.
void Facet(DetailLevel s detailLevel s, Boolean surfaceModel, ProgressNotification progress)	Re-creates the graphical representation of this Part.
class PropertyBindingCollection	
void Clear()	
class RsIrc5ConveyorEncoderUnit	

Boolean ConnectionState	Sets or gets the connection status of the encoder, if true an object is connected.
Double Speed	Gets or sets the speed of the conveyor.
class RsTarget	
void JumpTo(RsToolData tool, Configurati onData confi gurati on)	Jumps the tool and the attached robot to the target with specified configuration. Completely ignores the configuration already defined.
class Simulator	
static Boolean FullSpeed	Gets or sets a value indicating if the simulation should run at maximum possible speed.
static Double CurrentTime	Gets the current simulation time (in milliseconds).
static SimulationDataRecorder DataRecorder	Gets the data recorder that records simulation data.
static void Step()	Advances the current simulation one time step.
static event EventHandler ActiveConfigurati onChanged	Raised when the ActiveConfiguration property changes.
class SmartComponent	
Boolean IsBaseComponent	Returns true if this is a built-in base component.
class SmartComponentCodeBehi nd	
virtual void OnLibraryRepl aced(SmartComponent component)	Called if the library containing the SmartComponent has been replaced
virtual void OnLoad(SmartComponent component)	Called when the library or station containing the SmartComponent has been loaded.
class Station	
BuiltInDataRecorderSi gnals BuiltInDataRecorderSi gnals	Gives access to the identities of the built in data recorder signals.
MarkupCollection Markups	Gets a MarkupCollection object that represents all the markups on this Station object.
ProjectDocumentCol lecti on Documents	Gets a the collection of ProjectDocuments associated with this Station.
class Texture	
static Texture CreateParal laxBumpMap	Creates a combined normal map and bump map from two images, suitable for parallax bump mapping
class TrackingStateEventArgs	
Irc5ConveyorWorkpi ecel nfo ConveyorWorkpi ecel nfo	

3.3.7 ABB.Robotics.RobotStudio.Stations.Forms

New Types

class RefFrameCoordSys	
------------------------	--

Extended Types

class Graphi cPi cker	
static event EventHandl er<AfterFreehandEventArgs> AfterFreehand	Occurs when an object has been moved or rotated by freehand manipulation.
class TemporaryGraphi c	
String Text	Sets the text of this TemporaryGraphic object
class TemporaryGraphi cCol l ection	
TemporaryGraphi c DrawMesh(Matri x4 origi n, MeshPart mesh)	Creates a mesh.

4 Late-breaking information

4.1 Overview

This section contains late-breaking information that will be included in the appropriate documents in the subsequent releases.

5 Problems Corrected

Overview

This section describes the problems solved in RobotStudio SDK.

5.1 Solved Product Defect Documents (PDD) since RobotStudio SDK 5.14

ID	Title
PDD10178	GetURange and GetVRRange
PDD10459	Body.scale method not working
PDD10707	installation removes 5.13 SDK files

5.2 Solved Product Defect Documents (PDD) since RobotStudio SDK 5.13

ID	Title
DSE9243	RS API to set position of CAD parts is not working properly.

6 Known Limitations

6.1 Development Environment

No known limitations

7 Installation Information

Overview

In order for the Visual Studio project templates to be installed, Visual Studio 2010 needs to be installed on the computer before you install RobotStudio SDK.

To install RobotStudio SDK click RobotStudio on the RobotWare & RobotStudio DVD. If you select the default installation option Complete, RobotStudio SDK will be installed. The .NET assemblies and Visual Studio template can be optionally installed, while the documentation is always installed. If you do not want to install the assemblies and templates, select the installation option Custom and uncheck the feature RobotStudio/SDK.

7.1 Hardware and Software requirements

Software requirements

Operating system:

Same as RobotStudio – Please refer to RobotStudio Release Notes

Supported development environment:

Microsoft Visual Studio 2010.

The Visual Studio project templates works only for the English version of Visual Studio.

NOTE!

As RobotStudio 5.14 is built on .NET Framework 4.0, Microsoft Visual Studio 2008 can not be used for RobotStudio 5.14 application development.

It is possible, but not supported, to develop RobotStudio Add-Ins with any .NET development environment, such as Visual Studio 2010 Express or SharpDevelop, but there will be no project templates available. The RobotStudio API documentation also assumes that Visual Studio 2010 is used, which means that information about VisualStudio project settings etc has to be adopted to the other development environment.

7.2 Compatibility

The APIs in the RobotStudio SDKs 5.14, 5.13.03, 5.13.02, 5.13.01, 5.13, and 5.12 are compatible, there are no breaking changes.