

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 1/13
	Dealt with by, telephone		

Table of Content

GENERAL 2

Release Information..... 2

Introduction..... 2

Installation..... 3

Hardware and Software requirements..... 4

Deployment..... 5

Compatibility..... 5

Updates in PC SDK 5.14.02 vs. 5.14.01 7

Updates in PC SDK 5.14.01 vs. 5.14..... 7

Updates in PC SDK 5.14 vs. 5.13..... 7

Information 5.14.02 7

Information 5.14.01 9

Information 5.14 10

Information 5.13.02 10

Information 5.13.01 10

Information 5.13 11

Corrected Issues 5.14.02 PDD..... 11

Corrected Issues 5.14.01 PDD..... 12

Corrected “Product Defect Document” 5.14 PDD 12

Corrected “Product Defect Document” 5.13.02 PDD 12

Corrected “Product Defect Document” 5.13.01 PDD 12

Corrected “Product Defect Document” 5.13 PDD 12

Known Limitations..... 12

Product Support 13

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 2/13
	Dealt with by, telephone		

General

Release Information

The information should be considered as last minutes information and most up-to-date. For more information please visit Robot Release Information (RRI) Homepage:

<http://roboticsportal.se.abb.com/prod/!display?action=page&id=351>

Introduction

This file contains release notes for PC SDK 5.14.02.

Previously PC SDK and FlexPendant SDK were distributed as the product Robot Application Builder (RAB). Starting with RobotWare 5.13, PC SDK and FlexPendant SDK are distributed together with RobotStudio. The SDKs are not affected by this change.

PC SDK 5.14.02 is included in the RobotStudio installer, which is distributed on the RobotWare 5.14.02 DVD. It can be used for free by anyone who wants to develop a customized PC operator interface, for the IRC 5 controller. It can also be used to develop RobotStudio add-ins that interacts with the controller. With PC SDK it is possible to create an application that connects to one or several IRC5 controllers, real as well as virtual. No license is required to develop applications using PC SDK. The end user of a PC SDK application, needs to have the option PC Interface on the targeted controller.

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

Software

PC SDK (5.14.02)

Documentation

Application manual – PC SDK (5.14.02), Html Help and Pdf).

PC SDK Reference Documentation (5.14.02), documentation of class libraries with method signatures in C# and Visual Basic (Html Help).

Tools

ABBControllerAPI.msm - merge module including the PS SDK dlls to be used when a PC SDK application is deployed to a customer's PC

ABB Industrial Robot Communication Runtime.msi - to be used when a PC SDK application is deployed to a PC without RobotStudio

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

In no event shall ABB be liable for incidental or consequential damages arising from use of this product, of the software and hardware described in relating product documentation.

Visit our web site at <http://www.robotstudio.com/community> for information and updates. Click the Download symbol to the right in order to download RobotStudio 5.14.02, including PC SDK 5.14.02 for free.

Click Developer Tools to learn more about the PC SDK and visit the User Forum, where developers discuss software problems and solutions online.

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 3/13
	Dealt with by, telephone		

Installation

To install PC SDK click *RobotStudio* on the RobotWare & RobotStudio DVD. If you select the default installation option **Full**, PC SDK will be installed. If you want to install only PC SDK and not RobotStudio select the installation option **Custom**.

PC SDK 5.xx will be installed side by side with any previous major version of PC SDK 5.xx, while minor versions within a release will update to the latest one.

Before you start the installation you are recommended to read chapter 2, *Installation and development environment*, in the PC SDK Application Manual which is available on the Documentation DVD in PDF format.

Microsoft Visual Studio development environment is used to develop PC SDK applications. Visual Studio 2005 Express or better, or Visual Studio 2008 Express or better is required.

The default installation path is C:\Program Files\ABB Industrial IT\Robotics IT\SDK\PC SDK 5.14.01

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 4/13
	Dealt with by, telephone		

Hardware and Software requirements

Software requirements

Microsoft Windows XP SP3 – 32bit edition

Microsoft Windows 7 – 32bit edition

Microsoft Windows 7 – 64bit edition

***Note!** PC SDK application that shall be executed on Windows 7 – 64 bit edition, must be built with target platform set to “x86”. The default setting in VisualStudio 2008 is “AnyCPU”*

Microsoft Visual Studio 2008

PC SDK requires Express edition or better.

Microsoft Visual Studio 2010

PC SDK requires Express edition or better.

***Note!** PC SDK applications requires Robot Communications Runtime to be installed on your PC. It comes with the RobotStudio installation, but can also be installed separately from C:\Program Files\ABB Industrial IT\Robotics IT\SDK\PC SDK 5.14.\redistributable\RobotCommunicationRuntime after you have installed PC SDK.*

Recommended hardware

10 MB free disk-space on the installation hard disk

IRC5 requirements

- RobotStudio 5.14.xx for building a test system and for debugging and testing in the virtual environment
- RobotWare option "PC Interface" for communication with a real IRC5 controller.

***Note!** PC applications developed with PC SDK 5.14.xx require RobotWare version 5.07 or higher on the IRC5 controller.*

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 5/13
	Dealt with by, telephone		

Deployment

When you build a PC SDK stand alone executable (not a RobotStudio Add-In), and want to deploy it to the end users computer, you must ensure that all necessary dependencies are also deployed.

Your application will not function without those dependencies.

If RobotStudio and the PC SDK are already installed on the end users computer, the required assemblies may already be in place, but your application installer should not rely on that.

- To make sure that the PC SDK assemblies are deployed, include the merge module `ABBControllerAPI.msm` in your setup project. The merge module can be found in the folder `ABB Industrial IT\Robotics IT\SDK\PC SDK 5.14\Redistributable`. As an alternative you can include the individual assembly files and deploy them in the same folder as your executable.
- To make sure that Robot Communication Runtime is deployed, inform the end user to install it. A complete installer which you can distribute together with your application can be found in the folder, `ABB Industrial IT\Robotics IT\SDK\PC SDK 5.14\Redistributable`.

Note 1: Deploy the same versions as your application was built with

Please note. It is a requirement of the .NET Common Language Runtime that the exact same version of the referenced assemblies, as your application was built with, are available on the target computer.

For example if your application was built with `ABB.Robotics.Controller.dll` version 5.14.1032, the exact same version of this assembly must be deployed to the end users computer.

Note 2: Getting older versions of the PC SDK assemblies and `ABBControllerAPI.msm`

Contact ABB on the Developer Tools User Forum if you need to get older versions of the PC SDK assemblies, or `ABBControllerAPI.msm`.

Compatibility

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 6/13
	Dealt with by, telephone		

Compatibility

PC SDK 5.14 and 5.13 are compatible, there are no breaking changes.

PC SDK communicates with the robot controller using the Robot Communication Runtime, which is designed to be backwards compatible with earlier versions of RobotWare.

Note! Functionality introduced in later versions of RobotWare will not be available for a PC SDK application that is connected to a controller with an older version of RobotWare.

For example the Messaging functionality is only supported on RobotWare 5.10 and above. This means that a PC SDK application cannot use the `Ipc` class when communicating with a controller with RobotWare 5.09.

The code will compile, but an exception will be thrown at runtime. Application developers are responsible for handling this scenario in their applications. Please refer to the PC SDK Reference Documentation for details.

Note! Compatibility between RobotWare revisions is guaranteed (PC SDK 5.14 will be compatible with PC SDK/RW 5.14.02 etc).

EventHandlers and events

For 5.09 the internal event architecture was completely redesigned.

1. All previous event handlers are [Obsolete] and existing events are changed to the `EventHandler<TEventArgs>` generic delegate.

Ex: This example shows how to change from the old to the new event handler type.

// This old line will fail...

```
myEventLog.MessageWritten += new MessageWrittenEventHandler( OnMessage );
```

// and should be replaced like this...

```
myEventLog.MessageWritten += new EventHandler<MessageWrittenEventArgs>(
    OnMessage );
```

//or preferably...

```
myEventLog.MessageWritten += OnMessage;
```

2. Previous versions of PC SDK used the Windows Thread Pool internally to raise events.

From 5.09 a single thread is used to dispatch all events. This reduces the risk of race conditions in client code. However, it makes it even more important to use the `Control.BeginInvoke(...)/Control.EndInvoke(...)` pattern to avoid event starvation.

Additionally, we now raise all events internally prior to any external subscribers, as this will reduce the risk of race conditions between inner and external subscribers.

If your application is based on `Control.Invoke(...)` and not on `Control.BeginInvoke(...)` all events will be serialized, both internally and externally. However in previous PC SDK versions you may encounter "out-of-order" events and thread pool starvation through the use of `Control.Invoke(...)`.

Some public events raise an initial event immediately when the subscription is activated; however this is not consistent or by design and should therefore be avoided. Later versions of PC SDK will remove all initial events.

3. `Mastership.Request(...)` throws an `InvalidOperationException` if the user is not authenticated against the controller, previous versions raised an `ArgumentException`.

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 7/13
Dealt with by, telephone			

Updates in PC SDK 5.14.02 vs. 5.14.01

No changes

Updates in PC SDK 5.14.01 vs. 5.14

No changes

Updates in PC SDK 5.14 vs. 5.13

No changes

Information 5.14.02

PC SDK 5.14.01 assemblies are removed from the GAC

When PC SDK 5.14.02 is installed, the 5.14.02 version of the PC SDK assemblies are installed in the GAC. The 5.14.01 assemblies are removed from the GAC, if not in use by another installed application.

Hint:

To force a PC SDK application built with an earlier version of the SDK, for example 5.14.01, to use the 5.14.02 assemblies, without first rebuilding it, you can include an assembly binding element in your app.config file. This may be useful on your development machine. On the end users machine it is recommended to deploy the correct assemblies. See section **Deployment** for more information.

```
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
      <assemblyIdentity name="ABB.Robotics" publicKeyToken="1da709b7d1f14b7b" culture="neutral" />
      <bindingRedirect oldVersion="5.0.0.0-5.65535.65535.65535" newVersion="5.14.2027.0"/>
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="ABB.Robotics.Controllers" publicKeyToken="1da709b7d1f14b7b"
culture="neutral" />
      <bindingRedirect oldVersion="5.0.0.0-5.65535.65535.65535" newVersion="5.14.2027.0"/>
    </dependentAssembly>
  </assemblyBinding>
</runtime>
```

Manage mastership for all controller resources

It is now possible to request and release mastership over all resources of the controller at the same time.

Example:

```
Controller controller;
using(Mastership.Request(controller))
```

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 8/13
	Dealt with by, telephone		

```
{
    // add your code here
}
```

By calling `control.Ier.IsMaster` you can check if you have mastership over all the resources.

Event when `Task.Enabled` is changed

The new event `Control.Ier.EnabledChanged` is fired when the task selection state is changed.

`Task.Enabled` can be set – under the following circumstances

When using PC SDK from a RobotStudio Add-In, connected to a Virtual Controller, that is in Manual Mode, it is possible to set the task enabled property. Under all other conditions, setting the property will throw an exception.

`Task.Enabled` can be set – under the following conditions

When using PC SDK from a RobotStudio Add-In, connected to a Virtual Controller, that is in Manual Mode, it is possible to set the task enabled property. Under all other conditions, setting the property will throw an exception.

`Type.GetObject` renamed to `Type.GetInstance`

The methods `Type.GetObjects()` and `Type.GetObject(string name)` has been renamed `Type.GetInstances()` and `Type.GetInstance(string name)` for usability reasons. The old methods can still be used but are marked as obsolete.

Fill `ArrayData` from string

It is now possible to fill arrays from strings. If the string is not of the correct format, an exception of type `RapidDataFormatException` is thrown.

Example:

```
ArrayData data = ...
data.FillFromString2("....");
```

Note:

The fill from string implementations of all RAPID data type classes like "Bool", "RobTarget", "Num", etc, is documented to through a `RapidDataFormatException` if the string is not of the correct format. However the behaviour was different than the documentation and various .NET exceptions like `NullReferenceException`, as the general PC SDK `GenericControl.IerException` was thrown instead.

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 9/13
	Dealt with by, telephone		

When fill from string was implemented for array types, it was desired to have the same error handling as for the atomic types, mentioned above.

Throwing `RapidDataFormatException` instead of `GenericControllerException`, would break compatibility. In order to provide a consistent set of fill from string implementations the current `FillFromString` method was made obsolete in favour for `FillFromString2` which implements the correct functionality and error handling.

Class UserDefined initialized to a valid default value

When creating an instance of the `UserDefined` class, its `Components` property now returns an array of sub components, representing the default value for the record type represented by the instance.

In previous versions null was returned for a newly created `UserDefined` instance.

Instance.SetAttribute works for string data type attributes

It is now possible to set the value of attributes with string data type.

AuthorizationException thrown when not logged in

When calling methods that requires a logged on user, without a previous logon, an `AuthorizationException` exception is thrown, with a message text explaining that a log on is required.

Information 5.14.01

Eventlog.MessageWritten now works on Chinese system

There was a problem with listening to the event `Eventlog.MessageWritten` on a controller with Chinese language (culture code "zh"). This has now been corrected.

Creating a single installer for a PC SDK application using InstallShield Limited Edition using Custom Action

Using `InstallShield 2010 Limited Edition` and `Visual Studio 2010`, you can create an installer that installs both your PC SDK application and the Robot Communication Runtime. This means you can deliver one installer to your customers instead of two.

This solution makes use of a Custom Action in `InstallShield` which launches the embedded Robot Communication Runtime redistributable installer. Please refer to this post in the Developer Tools User Forum for more details:

http://www.robotstudio.com/forum/forum_posts.asp?TID=5047&SID=139z8358c3d313778a61366bzec384fc

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 10/13
	Dealt with by, telephone		

Information 5.14

Memory leakage in PC-SDK (DSE10353)

A memory leak in the PC SDK method Controller.Rapid.GetRapidData() has been corrected.

No simple installation for PC SDK apps because msm file RCR (DSE10406)

The Robot Communication Runtime is at present not available as a merge module (.msm file), only as a separate installer. If you are creating an installer for an application built with PC SDK, it is not possible to let it install the Robot Communication Runtime.

The Robot Communication Runtime must first be installed on the target computer, using the separate installer, which can be found in the Redistributable folder of the PC SDK installation.

A merge module containing the Robot Communication Runtime is planned for PC SDK 5.14.02.

Information 5.13.02

PC SDK exception in AWPP when running on Windows 7 - 64 bit (DSE9926)

PC SDK can now handle parenthesis in the system or release folders, e.g. "Program Files (x86)". Previous versions throw an exception.

Parameter values for x- p- and i-start missing in Restart com (DSE10140)

From revision 5.13.02 is it possible to do a I-Start via PC SDK.

A new enum value, IStart, has been added to the enum ControllerStartMode.

E.g.

```
using (Mastership.Request(m_controller.Rapid))
using (Mastership.Request(m_controller.Configuration))
{
    m_controller.Restart(ControllerStartMode.IStart);
}
```

PCSDK 5.13.255.0 Evenlog Class (DSE1069)

This is now corrected.

Information 5.13.01

Connect when controller is in SYS_FAIL state (DSE9830)

A bug in PC SDK 5.13 makes it impossible to connect to a controller which is in sys-fail state. This has now been corrected.

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 11/13
Dealt with by, telephone			

Information 5.13

It is now possible to access RAPID data that are declared in hidden modules and in hidden tasks.

E.g

```
RapidData data;
Controller controller;
```

```
// Access data declared in a hidden task.
```

```
// A hidden task has the -hidden attribute defined in the Controller Configuration.
```

```
data = controller.Rapid.GetRapidData("hidden_task", "module1", "num1");
```

```
// Access data declared in a hidden module
```

```
// A hidden module has the -hidden attribute defined in the Controller Configuration.
```

```
data = controller.Rapid.GetRapidData("T_ROB1", "hidden_module1", "num2");
```

```
// Access data declared in a shared hidden module
```

```
// A shared hidden module has the -hidden and the -shared attributes defined in the Controller Configuration.
```

```
data = controller.Rapid.GetRapidData("num3");
```

```
// Access data declared in an installed module
```

```
// An installed module is loaded from a install script using the -install option.
```

```
data = controller.Rapid.GetRapidData("T_ROB1", "#SYS", "num4");
```

```
// Access data declared in an installed hidden module
```

```
// An installed module is loaded from an install script using the -install and -hidden options.
```

```
data = controller.Rapid.GetRapidData("T_ROB1", "#SYS", "num5")
```

Simulate physical setting of input signals on a Virtual Controller

It is now possible to simulate the physical setting of the value of an input signal.

The previously existing method to set the value of a signal is similar to setting a signal from RAPID. Depending on the Access Level the controller safety system may or may not allow the signal value to be changed. When a physical signal is set by connecting it to +24V the controller cannot prevent it from happening, no matter what Access Level the signal has. By setting the property **-InputAsPhysical** on the **Signal** class, a subsequent write operation will simulate a physical write. This is important in a scenario where a PC SDK application shall set signal values in order to verify a RAPID program in a Virtual Controller

Dnum variable not accessible from PC SDK (DSE9020)

It is now possible to access RAPID dnum variables via PC SDK.

A new class Dnum has been added.

32 bit group signals is now supported by the PC SDK. (DSE9386)

32 bit group signals is now supported by the PC SDK.

Memory leak (DSE9493)

There was a memory leak in PCSDK when retrieving ipc messages. It leaked about 400 bytes per message it retrieves, this has now been corrected.

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 12/13
	Dealt with by, telephone		

Corrected Issues 5.14.01 **PDD**

RobotStudio installation removes 5.13 SDK files	DSE10707
ControllerFileSystemInfo.Exists throws exception	Forum
PC SDK CapiDevTest Update	DSE10702
No simple installation for PC SDK apps because msm file RCR is missing	DSE10406
UIInstructionEvent does not support more than 4 lines	Forum
Comma in UIMessageBox text is interpreted as newline	Forum

Corrected "Product Defect Document" 5.14 **PDD**

Memory leakage in PC-SDK	DSE10353
--------------------------	----------

Corrected "Product Defect Document" 5.13.02 **PDD**

PC SDK exception in AWPP when running on Windows 7 - 64 bit	DSE9926
Parameter values for x- p- and i-start missing in Restart com	DSE10140
PCSDK 5.13.255.0 Evenlog Class	DSE10169

Corrected "Product Defect Document" 5.13.01 **PDD**

Connect when controller is in SYS_FAIL state	DSE9830
--	---------

Corrected "Product Defect Document" 5.13 **PDD**

Dnum variable not accessible from PC SDK	DSE9020
32 bit group signals is now supported by the PC SDK.	DSE9386
Memory leak	DSE9493

Known Limitations

- PC SDK UIInstructionEvent / RAPID UIListView - if you are using characters like double quote (") and brackets ([]) in the RAPID array of list item strings, the UIListViewEventArgs.ListItems may return the wrong number of list items. Avoid using double quote (") and brackets ([]).
- In manual mode, when a PC SDK application releases master and immediately requests master again, the FlexPendant is locked up when the user presses the prompt to grant write access to the PC SDK application. In automatic mode there is no problem.

PRU SEROP	Title Release Notes PC SDK 5.14.02	Ref RP5.15	
Department DMRO/PRS	Date 2011-10-14	Filename Release Notes PC SDK 5.14.02.doc	Page 13/13
	Dealt with by, telephone		

- PC SDK IPC Messaging - the PC SDK erratically ALWAYS sends 444 byte. Data must therefore be null terminated. Also, there is a problem reading the data if the PC SDK thread is running as STA. Changing it to MTA will solve that problem.
- The property IsLocal on the class RapidData, returns true for shared data, even though such data is visible from all modules.

The declarations of default zonedata and speeddata has been moved from module BASE.SYS in each task, and are now installed as shared data.

If an application used to create a RapidData object representing for example "v1000", the IsLocal property returned false in RobotWare 5.12, but will return true in RobotWare 5.13."

Product Support

For technical support please turn to your local ABB office.

<http://www.abb.com/contacts>

For help, advice and the latest updates please visit the User Forum.

<http://www.robotstudio.com/forum>