



**CODESYS**

## **Release Information**

### **CODESYS**

Profile: V3.5 SP9

Release Date: 14 June 2016

# CONTENTS

	Seite
<b>1 General</b>	<b>3</b>
1.1 Responsible for the release	3
<b>2 Compatibility Information</b>	<b>4</b>
2.1 Automation Platform	4
2.2 Runtime System	4
2.3 Compatibility Information for End Users	4
2.3.1 Automation Platform	4
2.3.2 New Compiler Errors and Warnings	5
2.3.3 Runtime System	6
2.4 Compatibility Information for OEMs	6
2.4.1 Automation Platform	6
2.4.2 Runtime System	7
<b>3 Known limitations</b>	<b>9</b>
<b>4 Features and changes / addressed defects</b>	<b>10</b>
<b>5 Release Information</b>	<b>10</b>

# 1 General

This document contains information on a product version which was released in compliance with our quality guidelines.

## 1.1 Responsible for the release

Name	Role
Dieter Hess	Product Manager / CEO
Kevin Ketterle	Manager Product Development (Base Technology)
Hilmar Panzer	Manager Product Development (Application Technology)
Thomas Zauner	Manager Product Development (Systems)
Andrea Hein	Manager Quality Assurance

Responsibilities of the undersigned

- Product Management  
(Confirmation that the content of the version to be released meets the requirements of the Product Management)
- Product Development  
(Confirmation that the version to be released was developed in compliance with the relevant process description)
- Quality Assurance  
(Confirmation that the version to be released was released in compliance with the relevant process description)

\_\_\_\_\_ (Dieter Hess)

\_\_\_\_\_ (Hilmar Panzer)

\_\_\_\_\_ (Kevin Ketterle)

\_\_\_\_\_ (Thomas Zauner)

\_\_\_\_\_ (Andrea Hein)

**Note:** Please make sure that you read the release information documents for all releases since your version.

## 2 Compatibility Information

CODESYS V3.5 SP9 requires at least Version 4.1.0.0 of the **CODESYS Static Analysis** AddOn and will not work with older versions.

Likewise, the CODESYS Static Analysis AddOn Version 4.1.0.0 requires at least version 3.5 SP9 of CODESYS and will not work with older versions.

### Drive no longer supported by CODESYS SoftMotion

With the release of CODESYS V3.5 SP9, the following drive will no longer be supported by CODESYS SoftMotion: Nanotec SMC147S

All previously delivered standard versions of the CODESYS setup contain the required drivers and libraries for these drives.

### No more CODESYS product support for systems running a Windows version older than Windows 7

Along with the introduction of Windows 7, Microsoft has enhanced protection for Windows systems through certificate management. Due to this, Windows systems older than Windows 7 are no longer compatible.

Starting with CODESYS V3.5 SP9, the following CODESYS products will no longer be offered for systems running Windows versions older than Windows 7:

- CODESYS Control RTE
- CODESYS Control RTE (for Beckhoff CX)
- CODESYS Control SoftMotion RTE
- CODESYS Control SoftMotion RTE (for Beckhoff)
- CODESYS Control Win
- CODESYS Automation Platform SDK
- CODESYS OPC-Server
- CODESYS HMI
- CODESYS Gateway

## 2.1 Automation Platform

### DeviceApplication OnlineServices [CDS-46517](#)

The following AutomationPlatform API methods/delegates in the `_3S.CoDeSys.OnlineCommands` namespace have been deprecated:

- `ILoginService.ConfigureLogin`
- `OnlineChangeDialogDelegate`
- `PromptDialogDelegate`

Contents, order, number and sequence of values of the undocumented options array parameter may have changed.

## 2.2 Runtime System

-none-

## 2.3 Compatibility Information for End Users

### 2.3.1 Automation Platform

#### Online change message [CDS-42000](#)

There are two distinct mechanisms to show the changes that occur during download:

If at login, changes have been found, there will be a message box to confirm Download or Online Change. This message box contains a Details button, that opens the Application Information dialog.

This dialog contains a text field labelled "Changes" that contains all changes in the source code of the project. For example, if a global variable has changed, the global variable list will be listed as changed.

If the application content is part of the project and downloaded to the runtime, then a second tab is present in the "Application Content" dialog. This second tab contains the changes between the generated code in the project and the generated code in the runtime.

For example, if a global variable has changed, the global variable list will be listed as changed, and all POU's that access this global variable!

The changes for this issue affect both mechanisms:

- first of all, the "Changes" text will always be generated, even if the Application Content is present. (it used to be either-or).
- Some of the content in both lists contain cryptic names, especially those that are generated by the I/O Configuration. In most cases, it will be possible to show the original object in the I/O Configuration as origin of the change. For example, Global Variable lists with a generated name like G1234...HPS will be replaced by the name of a module in the IO-Configuration.
- The output in the content text is more detailed. If an interface has changed, the output will contain the name of the changed variable.

### Breakpoints [CDS-21720](#)

When performing a Reset Warm or Reset Cold, breakpoints are now re-enabled after the reset.

### 2.3.2 New Compiler Errors and Warnings

#### Online Change: Handle out of memory exception [CDS-47453](#)

Error C0398 introduced for System.OutOfMemory Exception.

#### No error appears if a located variable is defined in a VAR\_TEMP block [CDS-46022](#)

The error C0392 is reported if an AT declaration is used inside one of the following blocks: VAR\_TEMP, VAR\_IN\_OUT and VAR CONSTANT

#### Output assignment of interfaces [CDS-45977](#)

The assignment is not supported, a type mismatch error is reported.

#### \_\_ISVALIDREF returns TRUE for check of interface instance with value null [CDS-43812](#)

\_\_ISVALIDREF is a special operator that should only be used for checking references. Interfaces and pointers can be checked via

```
interf <> 0, pointer <> 0
```

Unfortunately, \_\_ISVALIDREF did not check the input for any useful type but accepted all inputs.

Therefore, there will now be a new error message if \_\_ISVALIDREF is used with any other type than REFERENCE types.

#### Online Change: wrong data after changed address [CDS-44576](#)

If an address is moved from one variable to another variable at an online change code, there will now be an error message, since we cannot guarantee correct copy code.

Example:

```
var1 : BYTE;
var2 AT %MB0 : BYTE;
->
var1 AT %MB0 : BYTE;
var2 : BYTE;
```

This will no longer be possible with online change, since we do not generate a specific order for copying values, and the old value of var1 may be copied to %MB0 before the old value of var2 is copied from %MB0.

#### Assertion at compilation of project with Array with length 0 [CDS-45776](#)

With Compiler Version >= 3.5.9.0 there will now be an error message for arrays like this:

```
arr[0..unsigned_const-1]
```

if (and only if) the unsigned\_const is 0 because the upper border is evaluated as unsigned operation and would result in a too large number for any signed array border.

A way to avoid the problem would be to use a signed constant instead, or to cast the constant to a signed type: `arr[0..TO_INT(unsigned_const)-1]`

#### PRIVATE should not be allowed for FB\_Init [CDS-45575](#)

A new error message will be produced, if an FB\_Init, FB\_Exit or FB\_ReInit Method is PRIVATE, or PROTECTED

#### No compile error reported if writing to an %I\* input variable [CDS-45684](#)

There will now be a new error message for writing to variables located at %I\* addresses

#### FB\_Exit should be called to destroy local instances [CDS-383](#)

FB\_EXIT is now called for instances allocated on the stack before the owning scope returns.

A warning has been added to inform users about the change of semantics in these cases. In order to suppress the warning, decorate the POU declaring the local instance with the warning disable macro for the warning code C0394 ("warning disable C0394")

### **Implicit variables generated for properties with "Exclude from build" [CDS-48066](#)**

The compiler now treats getters and setters marked with "Exclude from build" as if the corresponding accessor was missing from the object tree.

No implicit variable declaration is generated for properties without getters and setters.

A warning is generated if a property has neither a getter nor a setter.

## **2.3.3 Runtime System**

-none-

## **2.4 Compatibility Information for OEMs**

### **2.4.1 Automation Platform**

#### **Power PC: Stackframe [CDS-45213](#)**

Requires targetsetting "codegenerator\EABIStackFrame"

Attention:

The stack frame of the CODESYS PowerPC and PowerPC VLE codegenerator changed in a way that it is now according to EABI

and thus the same in CODESYS application context and in C context . This requires an adaptation of the runtime system layer, which implicitly relies on the stack frame of the CODESYS application.

Following functions of SysCpuHandlingOS.c must be adapted:

- SysCpuGetInstancePointer()

This function is used to get the instance pointer, which is placed on the stack by the CODESYS application relatively to the base pointer. The relative offset of the instance pointer to the base pointer changed from 16 byte to 8 byte.

- SysCpuGetCallstackEntry2():

To roll back the call stack in case of an exception or a breakpoint in the CODESYS IDE, this function gets the necessary information like the return address and the previous base pointer, by using the current base pointer. As the stack frame of CODESYS application context and the C context is usually different, there is an differentiation between those contexts. This is not necessary anymore and the implementation may be the same for both contexts now, which is usually implemented in the function SysCpuGetCallstackEntry().

- SysCpuSearchOnStack\_():

In case of an exception in an external C function and the case that SysCpuGetCallstackEntry2() was not able to roll back the call stack, it is tried to find a pattern, placed by the CODESYS application on the stack. This allows to roll back the call stack of the CODESYS application independent of the number of nested function call in the external C implementation. As the stack frame is the same for both contexts now, this function is not needed and should return ERR\_NOTIMPLEMENTED.

Please refer to the interface documentation of the SysCpuHandling interface in

Components/SysCpuHandlingI.h of your runtime toolkit for more detailed information and to the example implementation of the described functions in SysTemplates/Sys/SysCpuHandlingOS.c of your runtime toolkit.

### **Redundancy [CDS-44089](#)**

The services (write variable, force variable, release force list, online change) which are executed synchronously on both plcs cannot be selected any more in Redundancy Configuration editor. They are now activated per default when the settings are written to the plcs.

A custom setting of the supported services can be written by using `IRedundancySettings`. `TransmitSynchServices`. The services from `IRedundancySettings.SynchServices` are then written when the settings are written to the PLCs.

## 2.4.2 Runtime System

### M4-Export [CDS-47088](#)

Constant literals and enum initial values are exported now via constant macros (`#define RTS_IEC_type_C(constant)`) in the m4 file to be portable in the runtime.

These new constant qualifiers are available only in the runtime with v3.5.9.0 and later. So you will get compile errors if you use these exported m4 files in previous versions of the runtime system.

There are 2 workarounds for previous runtime versions:

1. You export the library with `CODESYS < v3.5.9.0`
2. You add the missing defines in your platform specific `sysdefines.h` of your runtime system:

```
#define RTS_IEC_BOOL_C(x) (x)
#define RTS_IEC_BYTE_C(x) (UINT8_C(x))
#define RTS_IEC_SINT_C(x) (INT8_C(x))
#define RTS_IEC_USINT_C(x) (UINT8_C(x))
#define RTS_IEC_INT_C(x) (INT16_C(x))
#define RTS_IEC_UINT_C(x) (UINT16_C(x))
#define RTS_IEC_DINT_C(x) (INT32_C(x))
#define RTS_IEC_UDINT_C(x) (UINT32_C(x))
#define RTS_IEC_LINT_C(x) (INT64_C(x))
#define RTS_IEC_ULINT_C(x) (UINT64_C(x))
#define RTS_IEC_WORD_C(x) (UINT16_C(x))
#define RTS_IEC_DWORD_C(x) (UINT32_C(x))
#define RTS_IEC_LWORD_C(x) (UINT64_C(x))
#define RTS_IEC_REAL_C(x) (x)
#define RTS_IEC_LREAL_C(x) (x)
#ifdef TRG_64BIT
#define RTS_IEC_XINT_C(x) (INT64_C(x))
#define RTS_IEC_UXINT_C(x) (UINT64_C(x))
#define RTS_IEC_XWORD_C(x) (UINT64_C(x))
#else
#define RTS_IEC_XINT_C(x) (INT32_C(x))
#define RTS_IEC_UXINT_C(x) (UINT32_C(x))
#define RTS_IEC_XWORD_C(x) (UINT32_C(x))
#endif
```

### CODESYSControl: Management of X.509 certificates [CDS-47581](#)

The new X509 Certificate store saves the certificates and private keys under the `$.pki$` placeholder. This folder structure is managed by the certificate store and not meant to be modified by hand.

The default folder will `.pki` in the working directory of the PLC.

There will be PLC shell commands to manage the certificates.

Use “?” to get a list of commands.

By now, this is a temporary solution for security reasons. In the future, this folder structure has to be protected from unauthorized access. See [CDS-49418](#).

### OPC UA / Server: Public component interface [CDS-47004](#)

There is a new OPC UA provider interface (`CmpOPCUAProviderInterfaceIcf.m4`) which allows for extending the OPC UA server with custom nodes. In this way, a new provider can be implemented in a separate component.

### CAA NetBaseServices [CDS-45946](#)

Uses `CmpAsyncMgr` for TCP, which needs to be implemented by the used runtime system, when `DEFINE_CAA_NETBASESERVICES_USE_ASYNCMGR` is set. In this case, two extra threads are used:

- 1.) for opening and closing sockets
- 2.) for send and receive.

The priority can be set by means of the library parameters `gc_udiTaskOpenerPriority` and `gc_udiTaskRWPriority`, which are default on 240 (low priority).

When this `DEFINE` is not set, the calls on the sockets will be executed on the calling thread context. This is

normally the IEC task, which may be blocked by the incorrectly implemented non-blocking sockets on the used system.



### 3 Known limitations

Component	Description	JIRA ID
<b>Profinet CIFX Controller</b>	CifX Driver for the new Profinet Stack from Hilscher Currently not available on Big-Endian Systems. Requires Firmware > V3.1.0.0, which is contained in setup, but is not referenced by default (has to be done manually, file cifxpmV3.nxf). Reconfigure not implemented (not possible in IRT Mode).	<a href="#">CDS-38707</a>
<b>Profinet CIFX Master</b>	Reconfiguration (at runtime) of enabled/diabled slave fails. The original enbaled/disabled state remains.	<a href="#">CDS-49512</a>

## 4 Features and changes / addressed defects

The link below opens a spreadsheet that contains information on the new implementations and changes in the current version:

Key, Summary, Description, Resolution, Priority, Issue Type, Components, Target User Group, Fix Version, Release Note

Special release information for OEM partners can be found in the RID note with the filter “Target User Group” for “OEM” or “OEM and End User”.

[Features and Changes Addressed Defects CODESYS V3.5 SP9.xls](#)

## 5 Release Information

All the tests conducted during the release of the above mentioned version and the corresponding results are documented in an internal document (test plan), which we will provide upon request.

The following tests have failed or could not be performed. No release could be issued for these components and/or functionalities.

Test	Environment	Result
Profinet (Autotest)	Beagle Bone Black	Failed / not released
Runtime - Full (Autotest)	Linux, SH4	Not tested / Not released (defect test hardware)
Soft Motion - ETC Servotronix_CDHD (Autotest)		Failed / not released

**Document release**

Created	Andrea Hein	QS	13.06.2016
Reviewed	Peter Zenker	QS	13.06.2016
Released	Peter Zenker	QS	13.06.2016