



Content

1	New SILworX Version 5.34	1
1.1	Compatibility with the PES Operating System	1
1.2	Compatibility with Existing Projects	1
1.3	Compatibility with the PC in Use	2
2	Improvements of Version 5.34	2
2.1	Improvements Compared to Previous Versions	2
3	Corrected Problems of Version 5.34	2
4	Restrictions	3
4.1	Restrictions of the Version 5.34	3
4.2	Basic Restrictions	4
4.2.1	General	4
4.2.2	Functions and Function Blocks	6
5	Upgrading from one Previous Version to Version. 5.34	7

1 New SILworX Version 5.34

This document describes the new functions of version 5.34, its restrictions and improvements compared to previous versions:

- Chapters 2 and 3 describe the new functions and improvements.
- Chapter 4 specifies the current restrictions of V.5.34.
- Chapter 5 describes the upgrade procedure from a previous version.

1.1 Compatibility with the PES Operating System

Version 5.34 can be used for the following PES with the mentioned operating system versions:

- HIMax controllers, firmware version 2.0 and beyond.
- HIMatrix controllers, all operating system versions suitable for SILworX (CPU operating system beyond version 7.0 and COM operating system beyond version 12.0).

New functions depending on the operating system can only be used with the according operating system release.

SILworX version 5.x is needed to use all functions of following HIMax operating systems:

- V5.14 for processor modules
- V5.10 for communication modules and I/O modules SIL 3
- V5.16 for I/O modules SIL 1 and standard

1.2 Compatibility with Existing Projects

The version can convert and edit projects created with a previous version. If a code generation is performed for the unchanged project, the existing CRC is maintained except for:

- X-OPC Server
- X-OTS

SILworX programming tool V.5.34

Export files (*.csv, *.xml) created with a previous version might be not completely imported, see Chapter 4.2.1, Point 13.

1.3 Compatibility with the PC in Use

The minimum requirements for the PC used to operate SILworX are specified in the corresponding installation DVD. In particular with very large projects, old PCs may require long processing times and thus be inappropriate for this task. Therefore, computers should comply with the stand of the art as soon as possible. Enhanced hardware features such as computing power and memory space ensure improved performance.

2 Improvements of Version 5.34

1 Programming of C++ function blocks

Support for creating function blocks programmed in C++. Refer to the C++ Function Block Manual (HI 801 319 E) for details.

2 Definable number of resends for Modbus TCP master

To prevent connection losses, the number of resends can be set for Modbus TCP master.

2.1 Improvements Compared to Previous Versions

Refer to the release notes for version 5 for more information on the extensions and improvements of version 5.30 compared to the previous versions.

3 Corrected Problems of Version 5.34

This chapter lists problems within previous versions, that have been resolved in this version 5.34.

1 Proper function of the Retain attribute in connection with structure or array variables

If the following circumstances occurred, the previous version did not function properly:

- The Retain attribute was assigned to a global variable with a user-defined data type (structure, array)
- The data type was extended by adding a structure element or the array range limit was increased

If this occurred, the Retain attribute was not assigned to the added structure or array element. This problem was removed. [HE22479]

2 Compiling of the OTS resource is possible

In the previous version, variables transferred from an OPC server of the OTS were displayed to the OPC interface of the OTS for reading and writing, i.e., with two sources. As a result, the OTS resource could no longer be compiled. This problem was removed. [HE22246]

3 SILworX no longer terminates if errors occur

If two transitions of an SFC branch were located, even if not allowed, on the same x-coordinate, the previous SILworX version terminated when reporting the error. [HE22354]

4 Restrictions

When using SILworX version 5.34, take the following restrictions into account.

If the following instructions are observed, the restrictions have no influence on safety and on the availability of the code generated for a controller.

4.1 Restrictions of the Version 5.34

- 1 Sequential function chart: No indication of deadlocks.
Shared use of selection and simultaneous nodes causes deadlocks, i.e., undefined states in which either all steps or no steps are active. SILworX does not warn the users. [HE17716]
- 2 Value of global variables used as VAR_EXTERNAL is not displayed
If global variable with Struct or Array data type are used as VAR_EXTERNAL, the FBD Editor does not display for sub-elements the values entered in the column *Initial Value*, *Description*, *Additional Comment* and *Technical Unit* [HE19688]
- 3 Cutting and pasting assigned comments or online test fields causes SILworX to terminate
The fault does not occur when cutting and pasting assigned comments or online test fields with a variable. [HE21719]
- 4 CRC change during OPC configuration
Generating a new code for X-OPC or X-OTS in a project converted from a previous version, the CRC of the opc.conf file changes.
- 5 Cross-project communication
Cross-project communication files between SILworX projects are now exchanged using the Archive and Restore functions.
Existing connections are converted to normal connections.
The assignment of global variables with other names to the transport variables is no longer supported during import procedures.
Proxy resources remain available, their semantic remain the same (resources without code generation).
- 6 The MUL function block provides erroneous values, in coincidence with the following circumstances:
 - HIMatrix standard resource
 - Data type LREAL
 - One input has the value $\pm\infty$, the other input the value nan (not a number)
 In this case the result is $-\infty$, and not *nan* as specified. [HE21924]
- 7 Communication is possible from OTS to OPC server, not in the opposite direction. If the OTS resource contains global variables, which read information about the connection with the OPC server, than the code generation is interrupted with an Error message. [HE22246]
- 8 Code generation terminates if only the second OPC server of an OPC server set exists
If only the second OPC server exists within an OPC server set, SILworX terminates during code generation.
Workaround: replace the two OPC servers in the safeethernet Editor. This ensures that the CRC checksum from version 3 is maintained. [HE22259]
- 9 The code generator terminates if the Multitasking Editor is open
SILworX terminates if a code generation is started while the Multitasking Editor is open. [HE22316]

SILworX programming tool V.5.34

- 10** The XML files that the Hardware Editor of SILworX V.5.30 reads from during import and writes to during export, are not compatible with the corresponding XML data from the previous version.
- 11** Termination of SILworX after a **Search and Replace** action
After a **Search and Replace** action, SILworX might terminate if table cells are in editing mode.
Workaround: Exit the editing mode either by clicking in another cell or using the Return key. [HE22543]
- 12** SILworX termination after changes to interface variables of functions/function blocks
SILworX might terminate if changes are performed to the interfaces of functions or function blocks and the instance(s) is(are) then updated (e.g. conversions of inputs to outputs or vice versa).
Workaround: Delete the instance and create a new one, instead of updating it. [HE22582]
- 13** Connections and ENO no longer exist when updating conflicts
After a function block is moved to another location within the structure tree, SILworX reports that all references between interface variables and local/global variables are removed. If the function is then updated, the connections and ENO are deleted.
Workaround: After moving a function block, start the **Connect References** function. [HE22532]
- 14** Termination of SILworX when restoring a project with PROFINET
SILworX terminates when restoring a project archive containing PROFINET.
Workaround: Use a project file backup instead of a project archive. [HE22384]
- 15** Updating an extendible function block not always possible
If the minimum extendibility of an extendible function block is increased and inputs/outputs are added, an existing function block instance can no longer be updated, if it is extended to a value less than the newly set extendibility value.
Workaround: Update the function block instance prior to increasing the minimum extendibility. [HE21622]
- 16** Faulty error detection during changes within the function block type of extendible function blocks
If a function block instance is extended up to a certain point, no errors are detected during validation if additional inputs to outputs are added within the extended area of the corresponding function block type.
Workaround: Deactivate the Extendibility option, verify and update the function block instance.

4.2 Basic Restrictions

4.2.1 General

- 1** In the HW Editor the scaling settings for an analog value are read as REAL
SILworX reads the values specified for the base points of an analog value as REAL (at 4 mA and 20 mA). They are, however, further processed as LREAL. LREAL can also be used in the user program. This restriction is only relevant with very large or very small base point values. [HE16388, restriction]
- 2** Logic operations of BOOL variables having values that do not originate from external systems, can provide results that differ from those expected.
The cause is that the coding of BOOL values used in the third-party system deviates from the coding used in the HIMax.

Two workarounds are possible:

- The external system only provides 0 for FALSE and 1 for TRUE
- A correction circuit is implemented in the user program for all relevant BOOL variables to normalize the value to 0 or 1:
non-normalized variable -> AtoByte function block -> AtoBOOL function block ->
normalized variable [HE13042, restriction]

3 Saving changes in a SILworX Editor is not possible

After specific changes are made within an editor, the message 'Impossible to save changes' appears while attempting to save ...". After confirming the message, however, the changes are saved.

If the SILworX Editor is then closed and re-opened, the message "The required data is being processed" appears.

Examples of changes in which this problem occurs are the cyclic renaming of variables (A => B, B => C, C => A).

Workaround: Avoid exchanging names.

If required, restart SILworX. [HE11613, restriction]

4 Variations of the cycle time during LREAL calculations.

The cycle times can vary strongly during calculations with variables of type LREAL. To measure the watchdog time, the cycle time must be determined under realistic conditions.
[HE12115, restriction]

5 Sequential function chart: Step-internal TON starts a cycle later than normal TON

A reload is performed and leads to the following changes:

- A new step is added and must be active immediately after the reload.
- A TON function block with the input set to TRUE is added.

The step-internal TON then starts one cycle later than the TON function block in the program logic. [HE16288, restriction]

6 Use of hardlocks

Licensing SILworX using Hardlocks (U3 USB sticks, standard USB sticks) is managed differently among the various operating systems:

- With Windows XP, administrator rights are required in the following cases:
 - a For installation
 - b For operation, if SILworX was licensed using U3 USB sticks.
The rights of a standard user are sufficient for operation, if SILworX was licensed using standard USB sticks.
- With Windows 7 administrator rights are required to perform the installation.
For operation, hardlocks can be used to license SILworX to all types users.

Workaround for Windows XP: Use softlock licenses or other standard USB sticks. [HE17056, restriction]

7 Handling of VAR_INPUT variables in connection with user-defined function blocks

With user-defined function blocks, SILworX handles VAR_INPUT variables differently, depending on how the inputs are connected:

- If the inputs are connected to variables of a default data type, SILworX transfers the value of the global variable to a copy within the function block (call by value). Changes to the original variables do not affect this copy (writing to the corresponding VAR_EXTERNAL). The VAR_INPUT retains its value.
- If the inputs are wired with variables of a user-defined data type, SILworX transfers a reference to the variable (call by reference).

SILworX programming tool V.5.34

If the variable is a global variable, take into account that it can be modified in the function block. Additionally, a global variable can be used in a function block as VAR_EXTERNAL. Changes to such variables (description of the VAR_EXTERNAL) are effective immediately when the corresponding VAR_INPUT variable is read in the function block. As such, the value of the VAR_INPUT can change while executing the function block. [HE17740, restriction]

- 8** If the diagnostic view is opened during a system login and the connection is closed, SILworX offers the module login when attempting to re-establish the connection. [HE11926, restriction]

- 9** Online help associated with a POU not printable

The document management cannot print the content of the online help associated with a user-defined POU.

Workaround: Use the display panel to display the online help content and print out the individual topics. [HE14244]

- 10** Various elements of a structure variable cannot be written simultaneously from different sources

The user program and the hardware or communication cannot simultaneously write to two different elements of the same structure variables.

Workaround: Use different structure variables for those elements written to by the user program and for those elements written to by the hardware or communication. [HE15700]

- 11** Elements of structure variables used as index

It is not possible to use elements of variables with structure data type as array index. [HE16159]

- 12** Value of system variables during the online test and offline simulation

The value of system variables is not displayed during the online test and offline simulation:

- The OLT field is empty
- The value of digital system variables is not represented by the color of the corresponding line
- The Process Value column in the System Variables tab of the Object Panel is empty
- The Force Editor contains no system variables

Workaround: Most of the information is displayed elsewhere, e.g., in the Control Panel. To display it in the OLT, connect the system variable to a global variable (VAR_TEMP) and connect this to an OLT field. Forcing can only be performed in HIMax, if the program is connected to the system variable via a variable. The variable can be forced. [HE15396, restriction]

- 13 Import of export files of a previous version**

If the data type column cannot be found in the file to be imported (*.cvs, *.xml), all variables are created with the default data type BOOL.

Workaround: Rename the Data Type column in the file to be imported to Data type. [HE21691]

4.2.2 Functions and Function Blocks

- 1** DIV_TIME with REAL typecast reports an error on ENO for divisor := +/-INF

The DIV_TIME function from the standard library improperly sets the ENO error output ENO to FALSE and reports therefore an error under the following conditions:

- The IN2 input (divisor) is of type REAL.
- The value of IN2 is +/-INF. [HE15199, Restriction]

- 2 ENO output in connection with user-defined function blocks may be overwritten during reload.
With user-defined function blocks, in which the ENO output is only dependent on the EN input, ENO may be set to FALSE during a reload. Such function blocks do not themselves write to ENO. [HE19129]
- 3 Timer function blocks cannot be correctly used with the retain attribute
If a timer function block is used with the retain attribute, the time counter may adopt any potential value after a warm start.
Workaround: Do not use timer function blocks with the retain attribute. [HE17252]

5 Upgrading from one Previous Version to Version. 5.34

Project data from previous versions can continue to be used in V.5.34.

No CRC changes occur as long as the **Minimum Configuration Version** setting of a resource remains unchanged. SILworX maintains the CRCs compatible provided that no changes occur or no new features are used.

Observe the following procedure to upgrade from version 2.36 to version 5.34:

- Generate all resources prior to conversion. This allows potential deviations after the conversion to be detected during generation.
- Generate code for all resources prior to conversion. This allows potential deviations after the conversion to be detected during generation.
- Open the project in version 5.34 and convert it.
- Since the conversion is extensive, check the project integrity after completing the conversion.
- Generate the code in version 5.34 to detect potential errors and check if CRCs changed.
- Remove detected errors and re-generate the code to detect changed CRCs.
- If no CRC changes are detected, the migration was completed successfully.
- If CRC changes are detected, verify whether they can be accepted.
- If this is the case, the migration was completed successfully.
- If they cannot be accepted, continue to work with corresponding previous version.

Conversion Notes:

- The procedure to convert versions prior to V.2.36 is described in the release notes to V.2.36.
- For very large projects, the conversion can take several hours.