



Content

1	New SILworX Version	1
2	Remarks about Version 3.38.....	2
3	Improvements of V.3.x Compared to V.2.x	3
3.1	Support of New HIMax V.3.x Features	3
3.2	Multitasking	3
3.3	User Management for SILworX Projects	3
3.4	Support of 3 GB Memory	3
3.5	Modified Handling of I/O Module Redundancy	4
3.6	Further Improvements	4
3.7	System Requirements	4
3.8	Improvements Compared to Previous Versions	5
4	Corrected Problems of Version 3.x	6
4.1	Improvements Potentially Resulting in a Changed CRC	6
4.2	Improvements of FBD Editor	7
4.3	Improvements of Communication.....	8
4.4	Improvements of Version Comparator	9
4.5	Further Improvements	9
5	Restrictions of the Version 3.38	12
5.1	Restrictions with Respect to the FBD Editor	12
5.2	Restrictions with Respect to Online View and Test.....	12
5.3	Restrictions with Respect to OPC Configuration	13
5.4	Restrictions with Respect to Communication	13
5.5	Restrictions with Respect to Structure Data Types	14
5.6	Restrictions for Projects with HIMax Controllers	15
5.7	Further Restrictions.....	15
5.8	Basic Restrictions.....	17
6	Migration from one Previous Version to Version 3.38	20

1 New SILworX Version

- **Important note:** HIMA strongly recommends to using this version 3.38 instead of the previous versions 2.x and 3.30, see Chapter 2.
- Version **3.38** for:
 - HIMax controllers, firmware version 2.0 and beyond.
 - HIMatrix controllers, CPU operating system version 7.0 and COM operating system version 12.0.

This document describes the new functions of version 3.38, its restrictions and improvements compared to previous versions:

- Chapters 2, 3 and 4 describe the new functions and improvements.
- Chapter 5 specifies the current restrictions of version 3.38.
- Chapter 6 describes the migration procedure from a previous version.

2 Remarks about Version 3.38

- This version removes a safety-critical problem of previous versions 2.36, 2.46, 2.52 and 3.30, that can only occur in very rare cases:
The compiler may calculate too small of a stack size and not reserve sufficient memory. The problem was noted when the following circumstances occurred in the user program or in the function blocks called by the user program:
 - The program or function block call many instances of the functions or function blocks.
 - The called instances have many inputs.

In most cases, if the problem occurs, it results in an error stop of the user program.

The CRC generation performed with version 3.38 can also be used for validating the projects from the previous versions. If the CRC generated with version 3.38 is identical to the CRC generated with the previous version, the project was not affected by the problem mentioned above. [HE17005]

3 Improvements of V.3.x Compared to V.2.x

This chapter describes the enhancements and extensions of main version 3 compared to main version 2.

3.1 Support of New HIMax V.3.x Features

SILworX supports the following additional HIMax version 3.x features:

- Multitasking
- New I/O modules
- New communication protocols
- Reload of I/O modules

3.2 Multitasking

In a HIMax controller with operating system version 3.x and below, up to 32 user programs can be run simultaneously. SILworX includes extensions for planning and handling these programs:

- Multitasking Editor
- Extended settings for resource and program
- Extensions in the Online View: Control Panel, Online Detail View, Force Editor

3.3 User Management for SILworX Projects

It is possible to configure the user management scheme in SILworX such that only authorized users are able to edit the projects. This user management scheme can work with the user management scheme of the controller such that SILworX users need not login each time they connect to the controller.

3.4 Support of 3 GB Memory

Version 3.x can manage 3 GB of virtual memory. This version is thus able to compile and print very large projects.

However, the 32-bit versions of the operating systems Windows XP and Vista must be configured such that they provide SILworX with up to 3 GB of virtual memory.

Windows XP:

- Add the option "/3GB" to the row containing "WINDOWS" in the boot.ini file located under "[operating systems]".

Windows Vista:

- Edit Boot Configuration Data (BCD):
BCDEDIT /Set increaseUserVa 3072

Note: This option is only needed in very rare cases.

Notes for Using the Additional Memory

Increasing the memory for SILworX reduces the memory available to the operating system. This can prevent Windows from loading some drivers or even from starting.

For this reason, the PC in use should preferably only contain devices that can be used with standard drivers.

As few programs as possible should simultaneously run with SILworX. These programs should be low memory intensive.

If such problems occur, SILworX can be provided with less than 3 GB memory.

Windows XP:

- Add the option "/UserVa=x" (with $2048 < x < 3072$) to the row containing "WINDOWS" in the boot.ini file located under "[operating systems]".

Windows Vista:

- Edit "Boot Configuration Data (BCD)":
BCDEDIT /Set increaseUserVa x
with $2048 < x < 3072$.

3.5 Modified Handling of I/O Module Redundancy

Beginning with version 3.x, a redundant I/O module must be added to an existing I/O module to configure it for redundancy. The two I/O modules form a redundancy group. The redundant module can be located anywhere. A table with the title Redundancy is located in the Object Panel and provides an overview of the existing redundancy groups and their respective members.

3.6 Further Improvements

- Amount of memory used is displayed within the version comparator
- Improved replacement of variables using drag&drop
- Improved POU replacement using drag&drop
- "Go to..." with error messages
- Non-used POU outputs displayed in auto OLT fields

3.7 System Requirements

The computer used to install SILworX must comply with the following system requirements:

Minimum requirements

- Intel Pentium 4
- 256 MB RAM memory available
- 500 MB hard disk space available
- 1024 x 768 pixel monitor resolution
- Ethernet interface
- Windows XP Professional, at least SP2
- Windows Vista Ultimate / Business
- (For more information on the system requirements for Windows, refer to: <http://support.microsoft.com/kb/919183/en-us>)

Recommended requirements

- Intel® Core™ 2 Duo
- 4 GB RAM
- Physical Address Extension (PAE) activated (uses 3 GB of memory)
- 10 GB hard disk space available

SILworX Programming Tool V.3.x

- 1600 x 1200 pixel monitor resolution (true color)
- Ethernet interface
- Windows XP Professional SP3 or
- Windows Vista Ultimate / Business

3.8 Improvements Compared to Previous Versions

Refer to the release notes for versions 2.36 and 2.46 for more information on the extensions and improvements of version 3 compared to the previous versions 1.12 and 1.20. The migration procedure from the previous versions 1.12 and V.1.20 to version 3.38 is identical with that specified in the release notes for versions 2.36 and 2.46.

4 Corrected Problems of Version 3.x

This chapter lists problems within previous versions that have been resolved in this version 3.x.

The first paragraph lists problems that may cause the CRC of converted projects to change after correction.

4.1 Improvements Potentially Resulting in a Changed CRC

If the problems of previous versions specified in this paragraph are eliminated in version 3.x, the CRC of converted project may change.

- "Retain" attribute with functions
A function may be assigned the Retain attribute.
The compiler accepted this and compiled accordingly in previous versions. In version 3.x, the compiler rejects this as an error and requires that it is corrected. This causes the CRC to be changed. This change can then be loaded by performing a reload. [HE15744]
- No configuration of state variables
In previous versions, state variables were not configured during event configuration in the following cases:
 - If "CPU" was selected as event source.
 - If all the threshold and hysteresis values were set to 0.Version 3.x configures the state variables also in these cases such that the state variables are calculated. This action can result in a higher CPU cycle load. The adjustment results in a changed CRC. This change can then be loaded by performing a reload. [HE15168, HE15170]
- Non-redundant I/O channel can be nevertheless redundantly assigned.
In previous versions, if a channel in a redundant module pair was defined as non-redundant, it could nevertheless be assigned with a global variable in the form for the redundant channels. Furthermore, SILworX did not prevent removing the channel redundancy from a channel already assigned with a global variable.
No error message was created by the previous version during the code generation, but the redundant assignment was ignored.
In version 3.x, such an assignment results in a warning message during the code generation. The configuration and thus the CRC can be maintained.
Note: To change the channel assignment, the channel must be redefined as redundant. The change can be loaded by performing a reload if the channel is not a component of a SOE module. [HE15558]
- CRC includes the initial value of user-defined data types
In previous versions, the CRC of user-defined data types did not include the initial value. In version 3.x, the CRC also includes the initial value of user-defined data types. [HE15821]
- Program memory requirements
In the previous version, excessive memory requirements were calculated for the program in certain rare cases. The proper calculation of the program memory

requirements in version 3.38 can result in a changed CRC if one of the functions or function blocks specified below is used in conjunction with the EN input. Even if this is the case, a CRC change is unlikely.

- CTU, CTD, CTUD
- ROL, ROR
- ABS, CHS, LIMIT
- SQRT, EXP, EXPT, LN, LOG
- ASIN, ACOS, ATAN, SIN, COS, TAN
- CHK_REAL, TRUNC
- DIV
- UNPACK2, UNPACK4, UNPACK8

[HE16094]

4.2 Improvements of FBD Editor

- Retain property with function block instances.
In previous versions, if the interface variables of type VAR_OUT were set for a function block to Retain = TRUE, they could be set to Retain = FALSE in one instance of the function block. This did not affect the code generated. The changed Retain property was adopted in the VAR_OUT and changed the property inherited from the POU type. This did not affect the code generated. In version 3.x, the Retain property can still be changed, but it does no longer affect VAR_OUT. [HE14182]
- Processing sequence view refresh
In previous versions, the processing sequence was displayed without changes even though the logic had been modified using the Function Block Editor. Version 3.x deletes the processing sequence displayed if the logic is changed such that the user, if desired, must redisplay the processing sequence. [HE13841]
- Verification of branch priorities in sequential function charts (SFC).
The following conditions must apply for an SFC branch (two transitions after a step):
 - The priorities for the individual transitions must all be set either manually or automatically.
 - These priorities must be different.Previous versions did not check these conditions during compilation. Version 3.x checks these conditions. [HE13911]
- Value of internal value fields
In previous versions, internal value fields of function blocks did not adopt the specified value, but instead assumed the initial value from the type definition. They adopted the initial value from the type definition instead. In version 3.x, they adopt the specified value. The conversion to version 3.x removes all specified values from the internal value fields to prevent the CRC and the program semantics from changing. [HE15544]
- Invisible feed-back loops.
In previous versions, directly connecting an output to an input created an invisible feedback loop, if SILworX was not able to find any suitable routing. For in-

stance, if an online test field was added, drawing a connection line between the two connectors could create an invisible connection line. The connection only became visible after deleting the variable.

Version 3.x displays the connection line in the foreground. [HE15365]

4.3 Improvements of Communication

- Data volume check for Modbus function code 23.
The check of permissible buffer volume for Modbus function code 23 did not work correctly in previous versions:
 - Lower limit of the read buffer was not checked.
 - Upper limit of the write buffer was 242 bytes and not 240 bytes as stated in the error message.
 - If the maximum data volume was exceeded for a function code, the error message did not disappear even if data was reduced to the permitted amount.

The problem has been eliminated in version 3.x. [HE12899]

- Deleting a P2P-connected signal with user-defined data types
Previous versions terminated when deleting a global variable with the following properties:
 - User-defined data type
 - Data exchange with other resources via **safeethernet**.

The problem has been eliminated in version 3.x. [HE14271]

- Import of exported proxy resources
Previous versions could only import the export files (PRS files) from proxy resources if the receive and resend timeouts were set to the default values during the creation of the exported proxy resources.
The problem has been eliminated in version 3.x. [HE15226]

- Adding CUT (ComUserTask) for each resource
The previous versions allowed the user to only add one CUT to the resource protocols even if a CUT for each communication module may exist.
The problem has been eliminated in version 3.x. [HE15624]

- Settings for **safeethernet** of an OPC server set
In previous versions, if the **safeethernet** parameters Event Priority and Condition Value Priority were modified when configuring the OPC server, the configuration was no longer consistent.
The problem has been eliminated in version 3.x. [HE14664]

- Code generation for OPC server
In previous versions, the code generation for the OPC server terminated if more than one view was used.
The problem has been eliminated in version 3.x. [HE15331]

- Global variables in the Object Panel
These variables were deleted from the project and needed to be created newly, if necessary. In previous versions, global variables could be deleted from the Object Panel of the **safeethernet** Editor for OPC server sets, by pressing the delete key and clicking "OK" to confirm.

The problem has been eliminated in version 3.x. [HE14575]

4.4 Improvements of Version Comparator

- Program information tab displayed in the version comparator
In previous versions, the version comparator displayed implausible data in the Program Information tab.
Version 3.x displays correct data and takes multitasking into account.
[HE14533]
- List of embedded data types in the version comparator
In previous versions, the list of embedded data types, i.e., data types only used with array or structure elements, was missing.
The problem has been eliminated in version 3.x. [HE16248]
- New elements in the version comparator represented as changed
Previous SILworX versions only marked elements (modules, POU's) that were changed in comparison to the previous project version, but it did not mark new elements.
The problem has been eliminated in version 3.x. [HE14685]

4.5 Further Improvements

- Changing the module name of a redundant module
With the previous versions, changing the name of a module redundantly connected caused SILworX to loop endlessly. This does no longer occur with version 3.x. [HE14738]
- Compiling remote I/Os
The compiler of previous versions terminated when translating the parameters for remote I/Os, if all the following conditions were met:
 - A user-defined data-type (Struct) was defined.
 - At least two global variables with this data types were contained in the parameters for the remote I/O.
 - The same sub-element was used with at least two of such variables.The problem has been eliminated in version 3.x. [HE15607]
- Memory requirements while printing documents
After printing documents, previous versions could terminate SILworX with the error message "out of memory" when the code generation was started.
Version 3.x does no longer terminate.
Note: Refer to Chapter 3.4 for the use of 3 GB memory. [HE15653, HE15676]
- Code generation with SOE modules with I/O events
In previous versions, the code generator was aborted with an error message if the project contained SOE modules in which I/O events were configured.
The problem has been eliminated in version 3.x. [HE15323]
- SILworX no longer restricted to users with administrator access.
When SILworX was used with a Windows user account without administrator rights, the following problems occurred in the previous versions:
 - No license despite the existence of an Windows XP license.
Only demo mode was possible.

SILworX Programming Tool V.3.x

- The SILworX log book log file was not written to in Windows XP.
- SILworX could not write to the INI file. For this reason, no settings could be saved with Windows XP and Vista.
- If a U3 USB stick is used for licensing, the OLicenseServer was terminated during start up. For this reason, SILworX had no valid license! No workaround existed under Windows XP!

The problem has been eliminated in version 3.x. [HE 14880]

- Performance

Compared to the previous version, the performance of version 3.x was enhanced in many ways:

- Opening the editors [HE14717, HE14829]
- Importing variables [HE12742, HE13203]
- Archiving resources [HE11954]
- FBD Editor [HE13720]
- Global Variable Editor [HE14888, HE14911]
- Deleting the objects, e.g., the logic [HE14014]
- Signal tables of the safe**ethernet** connection for OPC [HE14378]
- Large arrays in multiple resources [HE14555]
- Copying multiple signals in the Alarm&Event Editor [HE14646]
- Generating the code [HE15413, HE15648]
- Restoring libraries [HE15650]
- Importing/exporting function blocks [HE15430]
- Importing variables with protocols and event definitions [HE14874]

- Handling unwritten variables during reload

During reload, previous versions handled unwritten variables of different types in a non-uniform manner:

- Global variables that were not CONST were set to their initial value.
- Local variables that were not CONST were not set to their initial value.

Version 3.x sets unwritten local and global variables to their initial values. Exception: Version 3.x does not set the elements of a global variable of data type Structure or Array to their initial values if one of the elements was written to in the user program. [HE12058]

- Partial writing to structured variables

Previous versions did not recognize the writing of individual elements of Structure and Array variables as writing to the variables themselves. For this reason, such variables were treated as unwritten during the reload and reset to their initial values.

The problem as such does no longer exist in version 3.x, see exception specified in HE12058. [HE15560]

- Arrays and Structures in Editors

With the exception of the Global Variable Editor, if Array and Structure elements were used in the editors, previous versions were unable to access the individual elements of the variable. The elements had to be specified as text.

SILworX Programming Tool V.3.x

In version 3.x, the individual variable elements can be accessed in the Object Panel and dragged onto the editor. [HE13722]

- PDF files as online help for user-defined POUs
In previous versions, no PDF file could be attached as online help for a user-defined POU.
In version 3.x, a PDF file can be used as online help for user-defined POUs. [HE14246]
- Use of global variables as index
In previous versions, the use of a global variable as an array index was neither recognized as a read access in the cross-reference list nor during code generation. For this reason, forcing this variable may have had no effect.
Version 3.x recognizes that the use of a global variable as an index represents a read access. [HE14353]
- "Go to..." in the cross-reference list of the Global Variable Editor
In previous versions, the Go to... function in the editor's cross-reference list did not work with global variables used in a protocol.
The problem has been eliminated in version 3.x. [HE14410]
- Language dependencies with system variables
Due to the language dependencies associated with the system variables, if the language was changed in SILworX, the version comparator reported divergences and the project could not be reloaded in previous versions.
The language dependencies were eliminated from version 3.x. [HE15036]
- In version 3.38, not existing **safeethernet** variables caused error messages during code generation.
Deleting a global variable used as **safeethernet** variable did not result in an error message in previous versions.
SILworX V.3.38 generates error messages for such variables. [HE17625]

5 Restrictions of the Version 3.38

*When using SILworX version 3.38, 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 HIMax controller.*

5.1 Restrictions with Respect to the FBD Editor

- If input connectors are cut and pasted, the logics is "destroyed".
Cutting and pasting an input connector causes it to be reset to its default name and the corresponding output connector is lost.
Workaround: Use copy&paste instead of cut&paste, then delete the original input connector. [HE11563]
- SILworX terminates after copy and paste
SILworX terminates after copying and pasting a logical network, if all following conditions are fulfilled:
 - The network contains a value field.
 - For the value field, an OLT field and an assigned comment exist.Another action such as moving the variable leads to the termination.
[HE17261]

5.2 Restrictions with Respect to Online View and Test

- Reconnecting the diagnostic view causes a module login, instead of a system login
If the diagnostic view is opened during a system login and the connection is closed, the module login is offered when attempting to re-establish the connection.
Workaround: Establish the connections to the system either using the Hardware Online View or the Control Panel.
If the module login was opened, close all views of the module: The diagnostic and the module view. [HE11926]
- The value of array elements cannot be displayed during the online test
SILworX cannot display the value of individual array elements during the online test.
Workaround: Use the Force Editor to display the value of array elements.
[HE14990]
- Selection of multiple modules or remote I/Os in the Online View of the Hardware Editor
The selection of multiple modules or remote I/Os (multiselection) does not function properly. Attempting to open the context menu for several selected modules or remote I/Os results in only one context menu of an arbitrary object opening. Operational errors can thus occur easily, particularly since all remote I/Os are selected when the Online View is opened.
Workaround: After opening the Online View, make a point of clicking the object to be edited. [HE16851]

- Erroneous representation of a not loaded (copied) project during online test
If a copy of the project loaded in the controller is used for the online test, objects may be displayed with errors. Examples:

- Wrong colors of lines representing digital values (powerflow).
- Online test fields are empty.

Furthermore, warning messages may be displayed in these cases such as "The displayed variable names can be old since they refer to a resource configuration with identical CRC, but different serial number as the loaded configuration."

This occurs if a copy (e.g., safety copy) of the project loaded in the controller is used for the online test instead of the project itself.

Workaround: If available, use the original project for the online test. [HE17649]

- Force state of a partially visible variable is not updated
The force state of a variable is not updated, if the variable is only partially visible.

The force dialog box also does not update the force switch state.

Workaround: Move the selected view such that the variable, the value field or the corresponding OLT field becomes completely visible. [HE17872]

5.3 Restrictions with Respect to OPC Configuration

- More than four OPC servers can be configured
It is possible to configure more than four OPC servers for alarms and events, and to compile the project successfully. Four of those OPC servers can safely connect to the resource, the remaining, however, have connection problems.
Workaround: Make sure (manually) that not more than four OPC servers are configured. [HE14543]

- Structured global variable and OPC server
In the following cases, the Force Editor can display an incorrect variable path for structured global variables exchanged with the OPC server:
 - After renaming the structured global variables.
 - When referencing other global variables.

[HE16768]

5.4 Restrictions with Respect to Communication

- Structures in the Modbus slave
Global variables with structures as their data type cannot be transferred to the Modbus slave since their elements are not assigned proper offsets.
Workaround: Transfer the elements to the Modbus slave individually. [HE16360]
- PROFINET: ARUIDs after copying a controller
After copying a controller, it can happen that the ARUIDs are no longer unique in the PROFINET configuration.
Workaround: After copying the controller, delete the devices from the copy and then copy them on an individual basis. [HE16453]
- Use of structured global variables in the safe**ethernet** Editor
The safe**ethernet** Editor only adopts the element name from an inserted element of a structured global variable, such that the reference to the variable is no

longer evident. It is not possible to insert an additional element of the same name. [HE16460]

- Import of variable in the safe**ethernet** Editor
The safeethernet Editor import function for variables only functions properly if the editor does not contain any variables.
Workaround: Only import variables if the safeethernet Editor does not contain any variable. [HE16407]
- Wrong bit addressing when importing Modbus registers to the slave
When importing Modbus registers, SILworX assigns bit addresses that are 8 bits too high to all data types except BOOL.
Workaround: Specify addresses in the CSV file to be imported with the format "Register.Bit" or, alternatively, create *New Offsets*. [HE15056]

5.5 Restrictions with Respect to Structure Data Types

- Elements of structure variables used as index
It is not possible to use elements of variables with structure data type as array index. [HE16159]
- 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]
- Assignments to structure variables and changes (e.g., renaming elements)
Changes to a structure data type, e.g., renaming its elements, cause the deletion of assignments to variables of this data type within the hardware or in the protocols. [HE15095]
- Initial values of structures
The following example demonstrates how the initial values of structures can behave inconsistently:
Define a data type Strukt_1 with the elements A and B, both of type BOOL. Assign the initial value TRUE to the element A. Define a global variable of type Strukt_1.
In the described scenario, SILworX and the target system behave as follows:
 - During run time, element A of the global variable correctly assumes the initial value TRUE.
 - The Global Variable Editor does not display the initial value.
 - The compiler generates a message, warning that A has no initial value. [HE16884]
- Changing the sequence number of structure elements
If the sequence number of structure elements is changed after assigning global variables, the initial values used for structure elements during the code generation are incorrect. [HE17032]

5.6 Restrictions for Projects with H1Max Controllers

- SILworX terminates after ungrouping redundant I/O modules
SILworX terminates during a save operation in the Hardware Editor, if the following conditions are met:
 - The Online View is opened.
 - An I/O module that is member in a redundancy group or the entire redundancy group was deleted.

Workaround: Close the Online View prior to saving. [HE17330]

- Deleting base plates destroys redundancy groups of I/O modules
If base plates with inserted modules are deleted, SILworX does not check whether the inserted modules form redundancy groups with modules on other base plates. As a result, the project contains deleted redundancy groups and is therefore corrupted.
Workaround: Prior to deleting base plates, check whether the associated modules form redundancy groups. If this is the case, ungroup the modules. [HE17020]
- After deleting base plates, the function for creating new base plates does not function correctly.
If multiple base plates are not selected consecutively and are deleted, no new base plate can be created in the affected project file. Workaround: Only delete base plate on an individual basis. [HE17337]

5.7 Further Restrictions

- 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. [HE14244]
- Version comparator provides wrong information during module comparison
When comparing modules, the version comparator identifies the X-DI 32 02 digital input module channels as "AI Channel" since internally, this module runs in analog. [HE14494]
- The Hardware Editor cannot copy any base plates
The Hardware Editor can neither copy base plates individually nor as a group. Workaround: Create new base plates and copy the modules on an individual basis. [HE15972]
- It takes too long to open the Documentation Editor
When opening the Documentation Editor, SILworX first prepares the documentation without offering the user any sign of progress and only then opens the editor. For large projects, this waiting time can last approximately 30 minutes. During this period, the PC load is extremely high and SILworX cannot be used. [HE16047]
- Offline simulation not possible for projects converted from version 1.20
The FDB Editor does not open when attempting to perform an offline simulation on projects converted from version 1.20.
Workaround: Intermediate conversion to version 2.36, 2.46 or 2.52. [HE16255]

SILworX Programming Tool V.3.x

- CSV files cannot be opened in Excel and imported into SILworX simultaneously
It is not possible to import a CSV file into SILworX if it is already open in Microsoft Excel. [HE16134]
- PES password length during SILworX conversion from version 3.x to version 2.x
SILworX version 3.x accepts PES passwords with more than 32 characters. When converting the project to version 3.x, such a password remains valid, even in the controller. Since in SILworX version 3.x, the input box for the password only allows 31 characters, it is no longer possible to open the project with the corresponding user account or log in to the controller.
Workaround: Prior to converting, reduce the password to ≤ 31 characters. If required, repeat the conversion. [HE16838]
- Representation of system variables with offline simulation
In the offline simulation, the value of the program's digital system variables is represented in the wrong color.
Workaround: Connect the system variable to a value field. After this connection, SILworX displays correct colors. [HE17033]
- Retain attribute of timer function blocks is not safe
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]
- It is possible to access functions and function blocks of other resources
After moving a function or function block type from one resource to another, the function or function block type can still be used in connection with the original resource, although the resource is out of scope.
This may result in errors, e.g., in the following case:
 - A function block type is defined in a configuration library.
 - A function block of this type is used in a resource A.
 - The function block type is moved to a resource B (using cut and paste).Resource A continues to reference the original function block type. After moving the function block type, a newly created type that is valid for resource B and has the same name will not be taken into account when editing or generating code.
Workaround: Only move function block types using the copy, paste and delete functions. [HE17533]
- It is possible to access global variables of other resources
SILworX does not check the scope of global variables that are used in function blocks as VAR_EXTERNALS.
This may result in errors, e.g., in the following case:
 - A global variable is defined in a resource A and is used in a function block within the resource.
 - The global variable is moved to a resource B (using the cut and paste functions of the global variable container).SILworX continues to compile the original resource A correctly although the global variable is no longer defined.

Workaround: Use the copy, paste and delete functions to move the global variable container. [HE17784]

- No test of successive use of selection and simultaneous nodes
Using selection and simultaneous nodes within the same hierarchy causes deadlocks. SILworX makes no reference to this problem. Workaround: Do not use selection and simultaneous nodes within the same hierarchy. [HE17716]
- Code is not generated for looping SFC networks
SILworX does not generate the code for steps that have the input and output connected to the same transition and therefore forming a loop. [HE17743]
- The dialog box for safety parameters always displays TRUE instead of the current value
In the dialog box for setting the safety parameters of a user program, the *Start Allowed* parameter is always set to TRUE and ignores the value used in the controller. This behavior may have the following consequences:
 - If the parameter has the value FALSE and one tries to set it to TRUE, SILworX displays the message "Change at least one safety parameter...". Workaround: Change an additional parameter and, if necessary, set it back to the previous value.
 - When changing another parameter, SILworX sets the *Start Allowed* parameter back to TRUE, if it is set to FALSE in the controller! Workaround: When changing parameters, always set the *Start Allowed* parameter to the desired value. [HE17746]

5.8 Basic Restrictions

The restrictions described in this chapter are fundamental in their nature. They relate to properties of the resources that SILworX requires to run, e.g., the PC operating system. For this reason, it is unlikely that these restrictions will be lifted in the foreseeable future.

- Base points of analog values are read as REAL
SILworX reads the values specified for base points of an analog value (at 4 mA and 20 mA) as REAL. 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]
- Logic operations of BOOL variables having values that do not originate from safety-related communication, can provide results that differ from those expected.
The cause is that the coding of BOOL values used in the external system deviates from the coding used in the HiMax.
Two workarounds are possible:
 - The external system only transmits 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].

- Is it impossible to save changes in an Editor?
After specific changes 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 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]
- 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]
- 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]
- Use of hardlocks
Licensing SILworX using hardlocks (U3 USB sticks) is managed differently among the various operating systems:
 - With Windows XP, hardlocks can only be used to license SILworX to users with administrator rights, but not for power or standard users.
 - With Windows Vista, hardlocks can be used to license SILworX to all types users.Workaround for Windows XP: Use softlock licenses. [HE17056, Restriction]
- DIV_TIME with REAL typecasting 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]
- Handling of VAR_INPUT variables in connection with user-defined function blocks
With user-defined function block, SILworX handles VAR_INPUT variables differently, depending on how the inputs are wired:
 - If the inputs are wired with variables of a default data type, SILworX transfers the value of the variable to a copy within the function block (call by value). Changes to the original variables have no effects on this copy.
 - If the inputs are wired with variables of a user-defined data type, SILworX transfers a reference to the variable (call by reference).

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 are effective immediately when the corresponding VAR_INPUT variable is read in the function block. As such, the value can change while executing the function block. [HE17740]

6 Migration from one Previous Version to Version 3.38

Observe the following procedure to migrate from version 2.36 or below to version 3.38:

- Generate all resources prior to conversion. This allows potential deviations after the conversion to be detected during generation.
- Prior to converting, save the project, e.g., on a data medium.
- Open the project in version 3.38 and convert it.
- Since the conversion is extensive, check the project integrity after completing the conversion.
- Generate the code in version 3.38 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.

Notes on Conversion:

- The conversion is performed for versions 2.46 and V.2.52 such that the generated configurations remain identical. The exceptions are described in Chapter 4.1. The changes from version 2.36 to version 2.46 are described in the release notes for version 2.46.
- Depending on the project size, the conversion can take several hours.
- If one of the resources contained in the project to be converted has a user management scheme, the conversion adds a PADT user management scheme to the project and creates for each user of one resources a PADT user and a PADT user group with all rights.
If this is not required, all PES user management schemes must be deleted from the project prior to converting. This has, however, the effect that no reload can be performed after the next code generation.