Content

1 New SILworX Version

- Version **3.30** 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 3.30, its restrictions, and its improvements compared to the previous version:

- Chapter 2 and 3 contain the new functions and improvements.
- Chapter 4 specifies the current restrictions of version 3.30.
- Chapter 5 describes the migration procedure from the previous version.

2 Remarks about Version 3.30

Refer to the release notes for versions 2.36 and 2.46 for more information on the extensions and improvements of version 3.30 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 3.30 is identical with that specified in the release notes for versions 2.36 and 2.46.

2.1 Support of New HIMax V.3.30 Features

SILworX supports the following additional HIMax version 3.6 features:

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

2.2 Multitasking

In a HIMax controller with operating system version 3.30 and higher, 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

2.3 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.

2.4 Support of 3 GB Memory

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

Version 3.30 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

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.

2.5 Modified Handling of I/O Module Redundancy

Beginning with this version, 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.

2.6 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 Corrected Problems

This chapter lists problems within previous versions that have been resolved in this version 3.30.
The first paragraph lists problems that may cause the CRC of converted projects to change after correction.

3.1 Improvements Potentially Resulting in a Changed CRC

If the problems of previous versions specified in this paragraph are eliminated in version 3.30, 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 the previous version. In version 3.30, the compiler rejects this as an error and requires that it be 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 the previous version, 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 3.30 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. The change can be loaded by performing a reload. [HE15168, HE15170]

- Non-redundant I/O channel can be nevertheless redundantly assigned.
  In the previous version, 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 3.30, 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 the previous version, the CRC of user-defined data types did not include the initial value. In version 3.30, 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 3.30 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]

3.2 Improvements of FBD Editor

- Retain property with function block instances.
  In the previous version, 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. 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 3.30, the Retain property can still be changed, but it does no longer affect VAR_OUT. [HE14182]

- Processing sequence view refresh
  In the previous version, the processing sequence was displayed without changes even though the logic had been modified using the Function Block Editor. Version 3.30 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 priority of all transitions must either be set manually or automatically.
  - These priorities must be different.

  The previous version did not verify these conditions during compilation whereas version 3.30 does. [HE13911]

- Value of internal value fields
  In the previous version, internal value fields of function blocks did not adopt the specified value, but instead assumed the initial value from the type definition. In version 3.30, they adopt the specified value. The conversion to version 3.30 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 the previous version, directly connecting an output to an input created an invisible feedback loop, if SILworX was not able to find any suitable routing. For instance, 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 3.30 displays the connection line in the foreground. [HE15365]

3.3 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 the previous version:
  - 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 does not disappear even if data was reduced to the permitted amount.

  The problem does no longer exist in version 3.30. [HE12899]
- Deleting a P2P-connected signal with user-defined data types
  The previous version terminated when deleting a global variable with the following properties:
  - User-defined data type
  - Data exchange with other resources via safe**ethernet**.

  The problem does no longer exist in version 3.30. [HE14271]
- Import of exported proxy resources
  The previous version 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 does no longer exist in version 3.30. [HE15226]
- Adding CUT (ComUserTask) for each resource
  The previous version allowed the user to only add one CUT to the resource protocols even if a CUT for each communication module may exist.
  The problem does no longer exist in version 3.30. [HE15624]
- Settings for safe**ethernet** of an OPC server set
  In the previous version, if the safe**ethernet** parameters "Event Priority" and "Condition Value Priority" were modified when configuring the OPC server, the configuration was no longer consistent.
  The problem does no longer exist in version 3.30. [HE14664]
- Code generation for OPC server
  In the previous version, the code generation for the OPC server terminated if more than one view was used.
  The problem does no longer exist in version 3.30. [HE15331]
- Global variables in the Object Panel
  In the previous version, 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. These variables were deleted from the project and needed to be created newly, if necessary.
  The problem does no longer exist in version 3.30. [HE14575]

3.4 Improvements of Version Comparator

- Program information tab displayed in the version comparator
  In the previous version, the version comparator displayed implausible data in the Program information tab.
  Version 3.30 displays plausible data and takes multitasking into account. [HE14533]
- List of embedded data types in the version comparator
  In the previous version, the list of "embedded" data types, i.e., data types only used with array or structure elements, was missing.
  The problem does no longer exist in version 3.30. [HE16248]
- New elements in the version comparator represented as changed
  The previous SILworX version only marked elements (modules, POUs) that were changed in comparison to the previous project version, but it did not mark new elements.
  The problem does no longer exist in version 3.30. [HE14685]

3.5 Further Improvements

- Changing the module name of a redundant module
  With the previous version, changing the name of a module redundantly connected caused SILworX to loop endlessly. This does no longer occur with version 3.30. [HE14738]
- Compiling remote I/Os
  The compiler of the previous version terminated when compiling 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 type were contained in the parameters for the remote I/O.
  - At least two of such variables used the same sub-element.

  The problem does no longer exist in version 3.30. [HE15607]
- Memory requirements while printing documents
  After printing documents, the previous version could terminate SILworX with the error message "out of memory" when the code generation was started.
  Version 3.30 does no longer terminate.
  Note: Refer to the previous section "Support of 3 GB Memory". [HE15653, HE15676]
- Code generation with SOE modules with I/O events
  In the previous version, the code generator was aborted with an error message if the project contained SOE modules in which I/O events were configured.
  The problem does no longer exist in version 3.30. [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 version:
  - No license despite the existence of valid licensing in Windows XP.
    Only demo mode was possible.

  - The SILworX log book log file was not written to in Windows XP.
  - SILworX could not write on 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 does no longer exist in version 3.30. [HE 14880]

- Performance
  Compared to the previous version, the performance of version 3.30 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, the previous version 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 3.30 sets unwritten local and global variables to their initial values.
  Exception: Version 3.30 does not set the elements of a global variable of Structure or Array data types to their initial values if one of the elements was written to in the user program. [HE12058]

- Partial writing to structured variables
  The previous version 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 3.30, 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, the previous version was unable to access the individual elements of the variable. The elements had to be specified as text.

In version 3.30, 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 the previous version, no PDF file could be attached as online help for a user-defined POU.
  In version 3.30, a PDF file can be used as online help for user-defined POUs . [HE14246]
- Use of global variables as index
  In the previous version, 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 3.30 recognizes that the use of a global variable as an index represents a read access. [HE14353]
- "Go to" removed from the cross-reference list of the Global Variable Editor
  In the previous version, the "Go to..." function in the editor's cross-reference list did not work with global variables used in a protocol.
  The problem does no longer exist in version 3.30. [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 the previous version.
  The language dependency was eliminated from version 3.30. [HE15036]

4 Restrictions of the Version 3.30

When using SILworX, version 3.30, 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.

4.1 Restrictions with Respect to the FBD Editor

- Logic operations of BOOL variables having values that do not result 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].
- If input connectors are copied 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]

4.2 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 Hardware Editor's Online View
  The selection of multiple modules or remote I/Os (multiselection) does not operate correctly. Attempting to open the context menu for multiple selected modules or remote I/Os results in the context menu of only one of these objects opening. Operating errors can thus occur easily, particularly since all remote I/Os are selected after the Online View is opened.

Workaround: After opening the Online View, make a point of clicking the object to be edited. [HE16851]

4.3 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]

4.4 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: ARUUIDs after copying a controller
  After copying a controller, it can happen that the ARUUIDs 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 safe**ethernet** 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]

4.5 Further Restrictions

- 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]

- 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 digital X-DI 32 02 input module channels as "AI Channel" since internally, this module runs in analog. [HE14494]

- 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 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]

- 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 individually. [HE15972]

- 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 variable.
  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]

- 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]
- Assignments to structure variables are lost after 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]
- 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 logics. [HE16288, Restriction]
- 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]
- 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 main or standard users.
  - With Windows Vista, hardlocks can be used to license SILworX to all types users.

  Workaround for Windows XP: Use softlock licenses.

5 Migration from the Previous Version to Version 3.30

Observe the following procedure to migrate from version 2.36 or higher to version 3.30:

- 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 3.30 and convert it.
- Since the conversion is extensive, check the project integrity after completing the conversion.
- Generate the code in version 3.30 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 3.1. 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 may 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 every user of every resource a PADT user and a PADT user group with all rights.
  If this is not desired, 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.