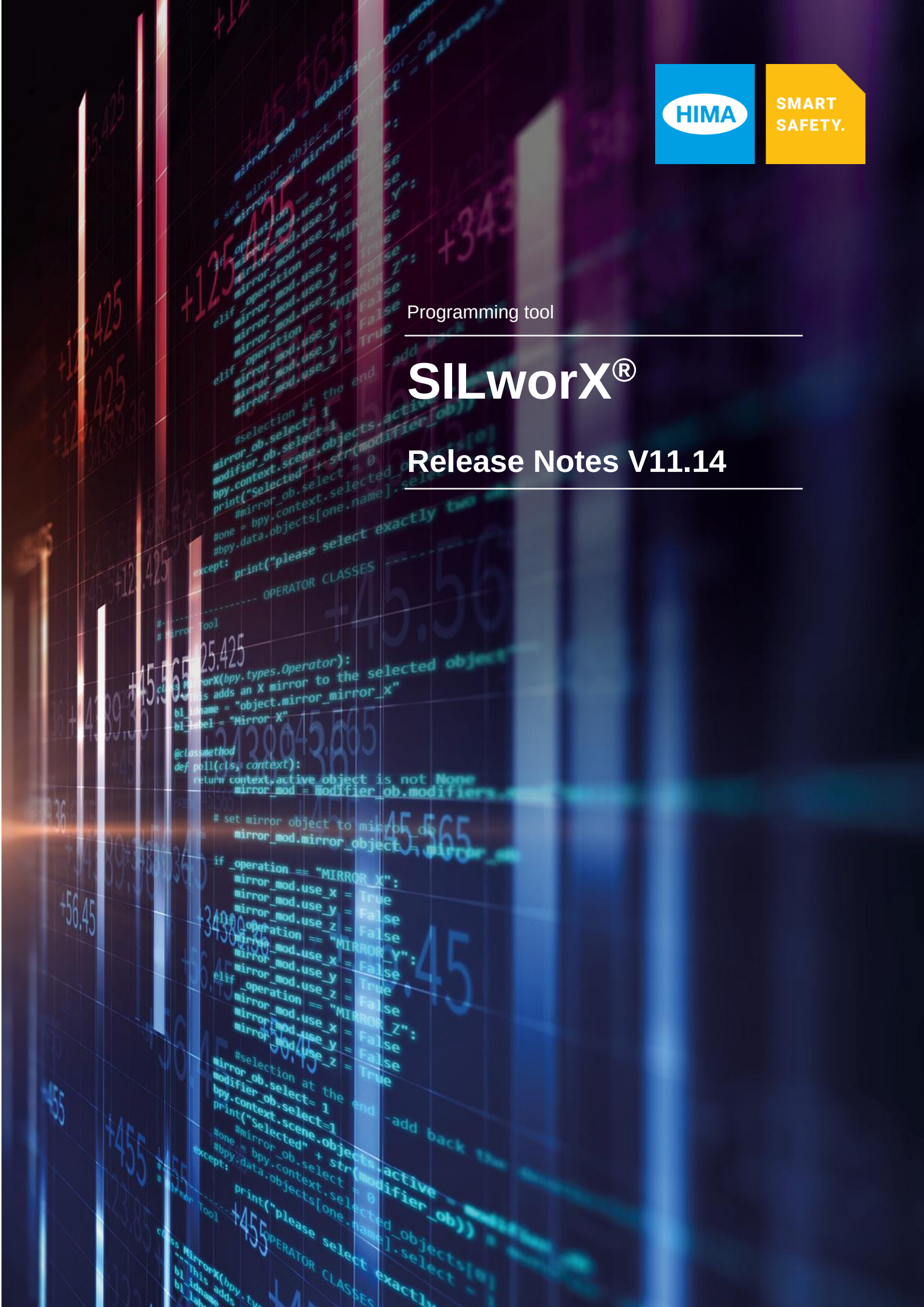

HIMA

SMART SAFETY.

Programming tool

SILworX®

Release Notes V11.14

Contact

HIMA Paul Hildebrandt GmbH

P.O. Box 1261

68777 Brühl, Germany

Phone: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com

Document designation	Description
HI 801 510 D, Rev. 11.00 (1920)	German original document
HI 801 519 E, Rev. 11.00.00 (1924)	English translation of the German original document

Table of Contents

1 SILworX V11.14

This document describes the improvements, new functions and restrictions of SILworX V11.14 compared to the previous versions.

1.1 Compatibilities

This chapter describes the compatibility of the individual software versions with the hardware versions and Windows operating systems.

1.1.1 PADT Operating System

SILworX as of V11 is a 64-bit application and can only be started on Windows 64-bit operating systems.

- Windows 10 (all the variants)

1.1.2 Supported HIMA System Families

SILworX V11.14 can be used for the following HIMA system families:

- HIMax
- HIMatrix F systems
- HIQuad X

1.1.3 Not Supported HIMA System Families

SILworX versions as of V9.36 no longer support the following controllers:

- HIMatrix F10 PCI 03
- HIMatrix F20 01
- HIMatrix F30 01
- HIMatrix F31 02
- HIMatrix F31 03
- HIMatrix F35 01
- HIMatrix F60 01

These controllers may no longer be used in conjunction SILworX versions as of V9.36 (exclusion of liability). This applies particularly to code generation, download and online services.
Projects including the specified controllers may be opened with SILworX for update or upgrade purposes (replacement by successor products). Connections to the specified controllers are no longer permitted with SILworX versions as of V9.36.

1.1.4 Compatibility with Previous Projects

SILworX V11.14 can convert and edit projects that were created with a previous version. Generating code for an unchanged project does not cause the CRC to change.

1.1.5 Compatibility with PADT Hardware

HIMA recommends using up-to-date computer hardware. In particular with very large projects, old PCs may require long processing times and thus be inappropriate for this task. Therefore, state-of-the-art computers should be used whenever possible. Enhanced hardware features such as computing power (number of CPU cores) and memory space (RAM) result in improved performance.

1.1.6 Use of Hardlocks

The following points must be taken into account if SILworX is licensed on Windows 10 using hardlocks (USB sticks):

- Administrator rights are required to perform the installation.
- User privileges are sufficient for operation.

2 New Functions

- The minimum configuration version was extended by V11 to support new functionality in the X-HART module and MultiForcing.

- Extension of the product documentation:
  In the variable cross-references, the page numbers of the usages are displayed in a new column. This makes it quick and easy to find the different usages of variables in a printout of the project documentation.

- Extension of the product documentation:
  Default setting or user setting profiles can be configured to define the required print selection more efficiently.

- The channel table of the X-HART 32 01 module includes the two new columns *HART: Allow Write Commands [BOOL]* and *HART: Allow Device-Specific Commands [BOOL]*. This enables users to set the parameters of the permitted commands for the entire module or for each individual channel.

  Reload when changing from module parameter settings to channel parameter settings is possible.

  To set the parameters on a per channel basis, a new operating system must be loaded into the X-HART 32 01 module. In this case, SILworX generates a new io4cpu.config, io4io.config and KE version. This results in a CRC change. Read commands are always permitted.

  The downward compatibility is ensured thanks to the extension of the minimum configuration version to V11.

- Within the scope of V11, four points have been implemented to optimize the Smart Safety Test.
  1. Import and export of a complete test plan in CSV format
  2. Export of the report in CSV format
  3. New test step: Checkpoint
  4. Check value extended by further operators

- Modbus slave V2:
  The Modbus slave V2 master contains the new parameter *Number of TCP Connections*. Users can configure up to 20 TCP connections to a Modbus master. The online view displays the number of TCP connections to a master object that are currently open.

  A reload is possible, but results in a reload warning if the number of TCP connections is reduced.

  The downward compatibility is ensured thanks to the extension of the minimum configuration version to V11.

- The PADT supports control of the MultiForcing feature on the PES, including its online change through the Control Panel, possible using a corresponding enabling switch of the resource, as well as a corresponding system variable for controlling the feature via the application logic.

- Several users can simultaneously log in to the PES with MultiForcing permission. Each of these users has concurrent write forcing permissions.

- The Smart Safety Test displays how many forcing-capable sessions exist at the start of system testing.

- In the online view of the PES, the status bar includes a field indicating the number of sessions with forcing permissions.

3 Improvements

3.1 Project Configuration

- The *SILworX V10b* value has been removed for the resource's *Minimum Configuration Version*.
  If a V10b project has the resource's minimum configuration version set to *SILworX V10b* and is converted, the resource's minimum configuration version becomes *SILworX V10*.

3.2 System Configuration

- The behavior of the version comparator has changed in such a way that the detail view can only be opened if the configuration file is available in both versions. The only exceptions are the comparisons for ls.config and ke.config.

- The names of properties in the program's Properties dialog box and in the Schedule Editor (editor at resource level) have been standardized.

  The following list presents the new vs. the old names:
  Program ID vs. *ID*
  Program's Maximum Number of CPU Cycles vs. *Maximum Number of Cycles*
  Max. Duration for Each Cycle [µs] vs. *Maximum Cycle Duration [µs]*
  Watchdog Time [ms] (calculated) vs. *Watchdog Time Program*

- Users are now already informed during the reload code generation that the system ID cannot be changed by performing a reload.

- The PADT now allows online testing for POUs (OLT) that were edited and saved in the PES after the configuration's download or reload. Previously, this was rejected with a corresponding error message, even if a new code generation showed that the configuration CRC and the AddOn CRC matched the state loaded in the PES. Thanks to the PADT improvement, the OLT is possible again after such a new code generation with no new download or reload.

- The reload code generation reported an error if an invalid data type change was made. The error message, however, only specified the name of the variable and it has now been extended by the indication of the path.

- The Force Editor now has a cross-reference view to jump to the variable.
  The designation *Forced Variables* in the Force Editor was change to *Individual Force Switch Set*.
  The Watchpage Editor can be opened with Read-Only access and the preset variables can be forced.

3.3 Smart Safety Test

- Users can now already recognize from the MDI window title whether they are in DEMO mode. The DEMO mode indicates that the corresponding license for the Smart Safety Test is missing.

- When users enter a test element name, they are supported by the automatic removal of the leading spaces.

- After the test plan name has changed, users can now use the structure tree to draw focus on an already open test plan.

3.4 Dialog Boxes

- The *Save As* dialog box, which can be opened in the archiving dialog box, displays all the files with any extension (*.*) on request of the user. By default, the file extensions allowed in the context are displayed. For the archiving dialog box, these files have the extension .A3. Additionally, it is no longer necessary to indicate the file path and file name.

3.5 Function Block Editors

- SILworX no longer terminates when a user-defined data type is copied from one variable to another within an ST Editor and the data type selection box of the target variable is in editing mode.

- Users can jump to the function block usage in the cross-reference view of the cross-reference dialog box by double-clicking it. Previously, this was only possible by right-clicking the function block usage and selecting the corresponding context menu option.

- An error was fixed which caused a crash during the code generation if a POU instance within an ST function block or an FBD logic referred to the name of an FBD typical in the structure tree. This is now detected and reported during validation and code generation.

3.6 Protocols

- The minimum value recommended for *Max. Duration of Configuration Connections* is calculated based on a new formula. Since this new formula is very conservative, the warning was downgraded to an info message. Additionally, the default value was increased from 12 to 20 ms so that no warning is issued for new resources.

- Messages about errors in safe**ethernet** transport variables that displayed incomprehensible internal names have been removed or changed. Incomprehensible names are no longer displayed.

- The operating mode of a connection between CPU and COM in the controller cannot be changed by performing a reload.
  SILworX now rejects such reload code generation. This may occur, for example, when upgrading from Modbus slave V1 to Modbus slave V2 with the same protocol instance name.

- In Modbus slave V2, superfluous error messages were issued during the code generation if the *Explicit Check of Redundant Master* parameter was activated in the masters and the redundant masters were explicitly named. This has been fixed.

3.7 User Management

- The table "PES User Management" includes new options to change the user data:

  1. A CSV import which may be used to change any number of access permissions quickly and differently.
  2. The option to mark several PES user permissions and edit them all at once. This also allows several user permissions to be changed concurrently. This method should be preferred if several user permissions must be subject to the same change.

3.8 General

- General enhancement of code generation performance. SILworX was restructured to a 64-bit application and now supports the use of multicore processors.

- According to the specification, SILworX must not support the creation of an OTS resource in the DEMO project with no valid license (DEMO license).

4 Restrictions

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

4.1 FBD Editor

- Effect:
  Empty pages in the logic section of the FBD Editor cannot always be deleted.

  Condition:
  The *Delete Empty Page* context menu option is not active if the following conditions occur simultaneously:
  - A line extends over two or more adjacent sides of the empty page.
  - The line does not cross the empty page.

  Therefore, the empty page cannot be deleted.

  Workaround:
  None.

- Effect:
  In the FBD Editor, text in existing page comments is no longer displayed after inserting new pages.

  Condition:
  The following conditions must occur simultaneously:
  - A page comment is located next to a second page with at least one logic element.
  - On the second page, the action *Insert Empty Pages -> Insert Column* or *Insert Empty Pages -> Insert Row* is executed such that the new pages are inserted between the page with the page comment and the page with the element.
  The text of the existing page comment is temporarily no longer displayed.

  Workaround:
  The text reappears if the comment is moved or the editor is closed and then reopened.

- Effect:
  In the FBD Editor, page information is not correctly positioned after importing a project from ELOP II.

  Condition:
  The following conditions must occur simultaneously:
  An associated comment or OLT field is located on an empty page without further logic elements, while the main element is located on a different page.

  Workaround:
  Associated comment or OLT fields should be located on the same page as their main elements.

- Effect:
  Conflict icon for variables remains visible, in spite of fixed conflict.

  Condition:
  In the following cases, the conflict icon remains visible although the invalid action was canceled and the valid value displayed:
  - Invalid name is entered for a variable.
  - An existing sequence number is assigned to an interface variable.

  Workaround:

Start verification or update process.

- Effect:
  Information on global variables used as VAR_EXTERNAL is not displayed:
  If global variables with Struct or Array data types are used as VAR_EXTERNAL, the FBD Editor does not display the information entered in the columns *Initial Value*, *Description*, *Additional Comment*, and *Technical Unit* for the sub-elements.

  Condition:
  Create a global variable with *Struct* data type and enter a description in a *Struct* element. Now use this global variable as a value field in a program. Then view the created VAR_EXTERNAL in the Local Variables tab.

  Workaround:
  View the attribute properties of the VAR_EXTERNAL in the corresponding global variable.

4.2 Structured Text

- Effect:
  2700 consecutive comment lines are not possible in the ST Editor.

  Condition:
  SILworX terminates when commenting out 2700 consecutive lines in the ST Editor.

  Workaround:
  Partition long comments, e.g., by grouping 1000 lines to one comment.

4.3 Hardware

- Effect:
  The Detail View button in the Hardware Editor is not active.

  Condition:
  Start the Hardware Editor online view and open a module's detail view: The detail view button is not active.

  Workaround:
  Not required, as the detail view contains a Close button.

- Effect:
  During code generation, SILworX V6 and higher no longer stores the licenses sorted by entry order, but by name. This may result in a changed CRC when converting projects from previous versions.

  Condition:
  Enter the license names in non-alphabetical order in V5 and generate the code. Then generate the code in V6.

  Workaround:
  Use suitable names, ask for HIMA technical support.

- Effect:
  The global error statistics are only displayed in the master and not in the remote I/Os. The system variables for the error statistics as well as for *Forcing*, *CPU Autostart Enable*, *CPU Start Enable*, *CPU Main Enable*, *ReadOnlyInRun*, *Start Cycle* are not activated in remote I/Os.

  Condition:

Connect the system variables to global variables and simulate the corresponding errors.

Workaround:
Use the error statistics in the master.

4.4 User Program

- Effect:
  The following behavior of the EXPT function in PES does not comply with the IEEE-754 standard.
  1.0 ** NaN := 1.0 expected: NaN
  EXPT.ENO := TRUE expected: FALSE

  NaN ** 0.0 := 1.0 expected: NaN
  EXPT.ENO := TRUE expected: FALSE

  EXPT in OTS and in the offline simulation behaves in compliance with IEEE-754.

  Condition:
  See above.

  Workaround:
  If ENO is required, trap or avoid a NaN on both inputs.

- Effect:
  The DIV_TIME function from the standard library improperly sets the ENO error output to FALSE and therefore reports an error under the following conditions:
  - The IN2 input (divisor) is of type REAL.
  - The value of IN2 is +/-INF.

  Condition:
  Use DIV_TIME with EN/ENO and enter +/- INF as the divisor (INF is the result of 1.0 / 0.0, for example)

  Workaround:
  Ignore ENO in this case.

- Effect:
  During the offline simulation and OTS, the EXPT function provides the result NaN instead of 1.0, if IN1 = 1.0 is used for the basis and IN2 =-INF is entered for the exponent.

  Condition:
  Call up EXPT with IN1 = 1.0 and IN2 = -INF (or another large negative number) and view the result in the offline simulation or OTS.

  Workaround:
  If this special case is relevant for the application, this has to be programmed accordingly in the user logic.

- Effect:
  A POU is processed in accordance with the following sequence: first the sequences, afterwards the SFC actions, and then the FBD logic. As a result, the input values of SFC transitions and SFC actions that are described in the FBD logic always originate from the previous cycle. The specific evaluation of the input values, however, reveals small differences:
  During the FBD processing, the input value of an SFC transition is written to and retained in the SFC transition memory and only processed in the sequence during the next cycle. After a cold start, this has the effect that sequences generally do not move on to the next step

before the second cycle.
The input value of an SFC action is read from the source during the processing of the SFC action. If this is a function, the initial value is read since functions are initialized at the beginning of POU processing and are only processed after the SFC actions.

Condition:
See above.

Workaround:
SFC transition:
When programming sequences, users must take into account that an SFC transition is performed in the second cycle at the earliest.
SFC action:
To use a function result as input value for an SFC action, a variable must be connected between function output and SFC action input.

- Effect:
  Sequences of a branch ending with SFC steps result in a deadlock. Sequences of a simultaneous branch ending with transitions, result in several active steps outside the simultaneous branch.

  Condition:
  See above.

  Workaround:
  Users must take suitable measures to ensure that such faulty sequences are not used.

- Effect:
  Access to an array element with an index outside the range of values result in accessing an element of the array based on a defined and high-performance procedure, to avoid random access to memory areas.

  Condition:
  See above.
  Workaround:
  Using suitable programming, users must ensure that array elements are only accessed through indexes within the value range of the array.

- Effect:
  Various elements of a structure variable cannot be written from different sources.
  The user program and the hardware or communication cannot write to two different elements of the same structure variable.

  Condition:
  See above.

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

- Effect:
  Elements of variables of a user-defined data type cannot be used as array index.

  Condition:
  See above.

  Workaround:
  Copy the value of the required variable to a simple variable and use this as index.

- Effect:
  When floating point numbers are converted from the decimal format, which is entered by the user, to the binary format, which corresponds to the data type (REAL, LREAL), SILworX V11 obtains a different result for some entries than SILworX V10. The reason is that Qt library has been using a more modern conversion algorithm since V5.7, which is supposed to operate faster and more accurately.

  The new results are used in every configuration generated with SILworX as of V11 (program, ke.config, etc.) and may also become effective with it, if applicable.

  The version comparison between a configuration generated up to V10 and a configuration generated as of V11 indicates a CRC change for each file or POU affected by it, plus:
  - For an affected value field, the POU detail view reports a change in the associated instance or assignment.
  - For an affected local variable, the POU detail view reports a change in the initial value CRC of the POU.
  - For an affected global variable, no message is issued (in the detail view of ke.config).

  For a force value, the *Edit Force Data* dialog box immediately displays the result of the conversion of the decimal entry to binary format and back to decimal format, so that users can view it before sending.

  So far, only one concrete case has been observed: SILworX V10 converts the literal 2.2250738585072012e-308 into the LREAL value with the bit pattern 16#000F_FFFF_FFFF_FFFF, which corresponds in SILworX to the decimal format 2.2250738585072009e-308. SILworX V11 converts the literal into the bit pattern 16#0010_0000_0000_0000 or in decimal format 2.2250738585072014e-308.
  The two results only differ by one unit in the last place (ulp). The exact literal value deviates approx. 0.37 ulp from the new result and approx. 0.63 ulp from the old result. For the decimal conversion within certain value ranges, which also include this literal, the IEC 559:1989 standard allows the result to slightly deviate from the optimal approximation, so that both results appear permissible (0.63 ulp < (0.37 ulp + 0.47 ulp)).

  Condition:
  For the case considered: Enter the literal 2.2250738585072012e-308 into a value field or enter it as the initial value of a local or a used global variable with LREAL data type. Connect the output of the value field to another value field or a POU instance appropriately.
  Generate code for such a structure in both SILworX V10 and V11. In SILworX V11, perform a version comparison between the V10 configuration and the V11 configuration.
  Either load the respective configuration into the PES or start the offline simulation in SILworX V10/V11 and observe the effective values in the Force Editor.

  As online alternative, enter the same literal as the force value of a variable and observe the displayed result.

  Workaround:
  If each bit is important, enter a REAL or LREAL literal in such a way that it comes as close as possible to the real number assigned to the required bit pattern[1]. For an unambiguous conversion, up to 9 significant decimal places are required for REAL and up to 17 for LREAL (total number before + after the decimal point, without consideration of the exponent).

  [1] The bit pattern of a binary floating point number is mathematically assigned to a very specific real number. At the same time, the floating point number is a proxy for all real numbers in an interval around the number assigned to it (in many cases, but not always, symmetrically).

4.5 Version Comparator

- Effect:
  The detail view of a POU in the version comparator incorrectly shows a change for a POU

instance if the two following points apply:
1. The comparison base was created with a version prior to V4.116.
2. The name of the called-up POU type contains umlauts.

Condition:
See above.

Workaround:
Only use spaces and characters from the following list for function block names:
- 0 1 2 3 4 5 6 7 8 9
- A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- a b c d e f g h i j k l m n o p q r s t u v w x y z
- $ % & ( ) * + - / : ; < = > ? \ ^ _ ` { | }

- Effect:
  When comparing a configuration generated with a SILworX prior to V9 and a configuration generated with SILworX V9 or higher, the following message no longer appears: "The order of the variable and instance declaration has changed." 1) For all function blocks and functions.
  2) For programs that use at least one of the following standard functions (in the configuration generated with V9 or higher): ADD, SUB, MUL, DIV, MOD, MOVE, AND, OR, XOR, NOT, SHL, SHR, all ATO... function blocks, ADD_TIME, SUB_TIME, MAX, MIN, SEL, MUX, GT, LT, GE, LE, EQ, NE, PACK.

  Condition:
  Create a resource with two configurations (first config loaded or imported, second config generated) in a SILworX version prior to V9. A comparison of these versions (correctly) notifies a change in the declaration order in a POU.
  The project state has to match the generated configuration.
  Convert the project to SILworX V9 and generate the code.
  The version comparison will not show the declaration order message, even though the respective POU displays the same CRC difference as before.

  Workaround:
  If the project state matching the comparison base (e.g., the loaded configuration) is still available:
  1. Convert a copy of this old project state to SILworX V9 or the required new version. Generate the code there and export the result via the start dialog box of the version comparison.

- 2. Open the other project to be used for the version comparison in the new SILworX version, import the previously exported configuration into the start dialog box of the version comparison and use it as a comparison base.

4.6 Code Generation

- Effect:
  Conflict resulting from changing the constant attribute for global variables after their use:
  A conflict occurs during code generation, if a global variable is used as VAR_EXTERNAL and is set from Constant to Changeable or vice versa, when a value is assigned to this VAR_EXTERNAL and the global variable is constant.

  Condition:
  Use the global variable as VAR_EXTERNAL in the logic and change the constant state in the global variable.

  Workaround:
  Delete the global variable at all positions where it is used so that VAR_EXTERNAL disappears. Then insert it again at all positions.

4.7 Reload

- Effect:
  1. An incomprehensible error message may appear if *Abort* was selected when assigning the comparison configuration.
  2. When generating reload codes for several resources and importing the configuration file, the SILworX user guidance only resumes the reload code generation once the configuration file has been confirmed.

  Condition:
  Start the reload code generation for a configuration with multiple resources that include a backup or import configuration and abort in the selection dialog box.

  Workaround:
  Not available.

4.8 Protocols

- Effect:
  Create a safeethernet connection with two resources (Res1, ID=1 and Res2, ID=2) in a project prior to V6 and set the connection to *V6 and higher* in a current version. This may lead to Res2 becoming the timing master.

  Condition:
  From the user's perspective, it is not possible forecast who will be the timing master when converting to *V6 and higher*.

  Workaround:
  Set the timing master explicitly.

4.9 Project

- Effect:
  Projects can get lost on the network drive during Windows internal synchronization of network drives. They may then only be available locally.

Condition:
Open, edit, and close a project in a synchronized folder. Afterwards, the project is removed from the network drive.

Workaround:
Do not open projects that are being synchronized and where there is a network connection. Opening the project leads to the network version being deleted by Windows.
If this issue occurs, the local copy can be copied back onto the network drive.

- Effect:
  If a combination of a non-breaking hyphen and a space is used in the program name, SILworX can terminate.

  Condition:
  Create a program P 1 under the resource in the structure tree. Then create a program *P* 1 under the same resource, using Alt-0173 to create a non-breaking hyphen in the name before the space. SILworX then terminates.

  Workaround:
  None.

- Effect:
  In SILworX V4, delete actions could cause objects that could no longer be edited to remain in the database. These objects did not affect the rest of the project but were reported during the project integrity check.

  Condition:
  Projects created in SILworX V4 and V5 that contain such "residual" objects most likely cannot be converted to SILworX V6 and V7. The likelihood is particularly high if the projects contain user-defined data types.

  Workaround:
  Remove the objects found during the integrity check prior to converting the project. The simplest procedure is described below and must be performed in the previous SILworX version:
  1) Archive all child nodes of the project that are positioned in the structure tree under the project, except for *Programming and Debugging Tool*.
  2) Create a new project in the previous SILworX version.
  3) Delete the *Configuration* node in the new project.
  4) In the new project, restore the configuration archived in step 1 and, if existing, additional child nodes of the project.

  The project just created should be convertible to the current SILworX version.

- Effect:
  Error message due to destinations that cannot be resolved after project conversion to V10. References to objects that were deleted prior to V10 reappear after the conversion and must be removed manually. This could be a deleted processor module or communication module, for example, which was used in a safe**ethernet** interface channel. In V10, this channel will show a *?* reference to the module that is no longer available. Otherwise, a path to the referenced object is displayed.

  Condition:
  Use a version prior to V10 (e.g., V9.36.0) to create a new project with two resources and add a processor module and a communication module to each. Create a safeethernet connection between the resources and use both modules. Then delete the communication module or the processor module.
  The reference to the channel seems to have disappeared. Now convert to V10. The reference reappears (with a *?* as system ID, as the target cannot be resolved).

  Workaround:

Manually correct the reference or set it to *None*. For a safeethernet interface: Manually deactivate the second channel.

- Effect:
  If a SILworX crash occurs after a project archive has been restored and the restored project was open, the project recovery offered when opening SILworX cannot be performed successfully. An error message indicating that the project was created with a demo license is issued instead.

  Condition:
  Restore a project archive. Terminate SILworX (e.g., using the Task-Manager). The next time you open SILworX, the dialog to recover the project appears. The recovery, however, cannot be performed since an error message indicating that the project was created with a demo license is issued.

  Workaround:
  The best way is to first close a restored project and then open it again.

4.10 Project History

- Effect:
  When an English project created with SILworX V2 is imported, SILworX does not properly interpret the date in the project history. Example: 1/11/2013 is interpreted as November 1, 2013 instead of January 11, 2013. 1/13/2013 is interpreted as an invalid date and results in the default value January 1, 2000.

  Condition:
  Create a project in an English language version of SILworX V2 (or lower).
  Then open this project with V8.34. The messages are read and added to the project history as described above.

  Workaround:
  None.

4.11 Documentation

- Effect:
  Cross-references of structure or array elements do not appear in print.

  Condition:
  Use a structure element or array element and check the cross-reference in the documentation.

  Workaround:
  None.

4.12 User Management

- Effect:
  When restoring a user management archive created with V9, a default user contained in the archive is ignored. The user must explicitly log in with user name and password.

  Condition:
  - Create a new project with user management.
  - Define the default user.
  - Archive the user management.
  - Remove the existing user management in the project.

- Restore the archived user management.

Workaround:
None.

5 Special Points

When using SILworX, the described characteristics must be observed.

- Effect:
  In the Hardware Editor, the scaling settings for an analog value are read as REAL. SILworX reads the values specified for the vertices 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 vertex values.

  Condition:
  Using extremely small or extremely large vertex values can impair process value accuracy.

  Workaround:
  Process raw values in the user program.

- Effect:
  Logic operations of BOOL variables having values that originate from third-party systems can provide results that differ from those expected.
  Condition:
  The cause is that the coding of BOOL values used in the third-party system deviates from the coding used in the HIMA system.

  Workaround:
  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

- Effect:
  The cycle times can strongly vary during calculations with variables of data types REAL or LREAL, particularly when using trigonometric functions.
  Condition:
  See above.

  Workaround:
  To measure the watchdog time, the cycle time must be determined under realistic conditions. For further details, refer to the safety manual and the chapter on accurately determining the watchdog time.

- Effect:
  The value of user program's 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.

  Condition:
  See above.

  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 variable and connect this variable to an OLT field. Forcing is only possible if the system variable is connected with a variable.

- Effect:
  Value changes for VAR_INPUT variables in user-defined function blocks

In user-defined function blocks, 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, the value of the variable is transferred to a copy within the function block (call by value).
- If the inputs are connected to variables of a user-defined data type, a reference to the variable is transferred to the function block (call by reference).

Condition:
This behavior may result in errors if all the following conditions are met:
- The source of a VAR_INPUT variable is a VAR_EXTERNAL variable.
- The same source of the VAR_INPUT variable is simultaneously used in the called function block as VAR_EXTERNAL variable.

If the value of the VAR_EXTERNAL variable is changed in the function block, the subsequent reading of the corresponding VAR_INPUT variable in the function block results in the following actions:
- For a user-defined data type, the current values are read.
- For an elementary data type, the previous values, which were valid at the beginning of the function block instance processing, are read.

Workaround:
VAR_EXTERNAL variables should not be used simultaneously as the source of a VAR_INPUT variable for instances of this POU.

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

  Condition:
  -

  Workaround:
  Use Windows to display the online help content and print out the individual topics.

- Effect:
  It cannot be ensured that key terms in the export or import files (.CSV, .XML) do not change between SILworX versions. If this occurs, SILworX imports the corresponding data as default values and issues an error message.

  Condition:
  Example: The data type for the English language setting was denoted *Data Type* in versions prior to V5.xx, and *Data type* in V5.xx and higher. When an export file is imported from a version prior to V5.xx, SILworX creates all the variables with the default data type BOOL.

  Workaround:
  Adjust the corresponding key words in the file to be imported.

- Effect:
  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.

  Condition:
  Hardware login, open the detail view for the module.
  Open the diagnostics for this module in a second window.
  Then close the connection.

  After a lost connection, the diagnostics only offers the module login, the detail view offers a system login.

Workaround:
Once the module login dialog box for the diagnostics has been opened, all online views of this module (diagnostics and module online view) must be closed and then reopened so they can be read again via the system.

6 Upgrading from a Previous Version

Project data from previous versions can still be used in V11.64.

No CRC changes occur as long as the minimum configuration version setting remains unchanged for a resource. SILworX ensures compatibility of the CRCs, provided that no changes occur or no new features are used.

Observe the following procedure to upgrade from V2.36 and higher to V11.14:

- A backup should be performed for the project.
- Generate code for all resources prior to conversion. An export and import of the configuration in the version comparator allows potential deviations after the conversion to be detected.
- Open the project in V11.14 and convert it.
- Since the conversion is extensive, check the project integrity after completing the conversion.
- Generate the code in V11.14 to detect potential errors and check if CRCs have changed. A version comparison with the imported configuration will detect this.
- Remove detected errors and re-generate the code to detect CRC changes.
- If no CRC changes are detected, the migration was completed successfully.
- If CRC changes are detected, verify whether they can be accepted.
- If the changes can be accepted, the migration is successfully completed.
- If they cannot be accepted, continue to work with corresponding previous version.

Conversion notes:

- The procedure to convert versions prior to V2.36 is described in the release notes to V2.36.
- For very large projects, the conversion can take several hours.

6.1 References

- SILworX online help
- SILworX first steps manual, HI 801 103 E
- Communication manual, HI 801 101 E
- HIPRO-S V2 manual, HI 800 723 E
- ISOfast manual, HI 801 465 E
- Modbus V2 manual, HI 801 475 E
- X-OPC server manual, HI 801 480 E

MANUAL
Release Notes

HI 801 519 E

For further information, please contact:

HIMA Paul Hildebrandt GmbH
Albert-Bassermann-Str. 28
68782 Brühl, Germany

Phone: +49 6202 709-0
Fax: +49 6202 709-107
E-mail: info@hima.com

Learn more about **HIMA** solutions online:

www.hima.com/en

www.hima.com