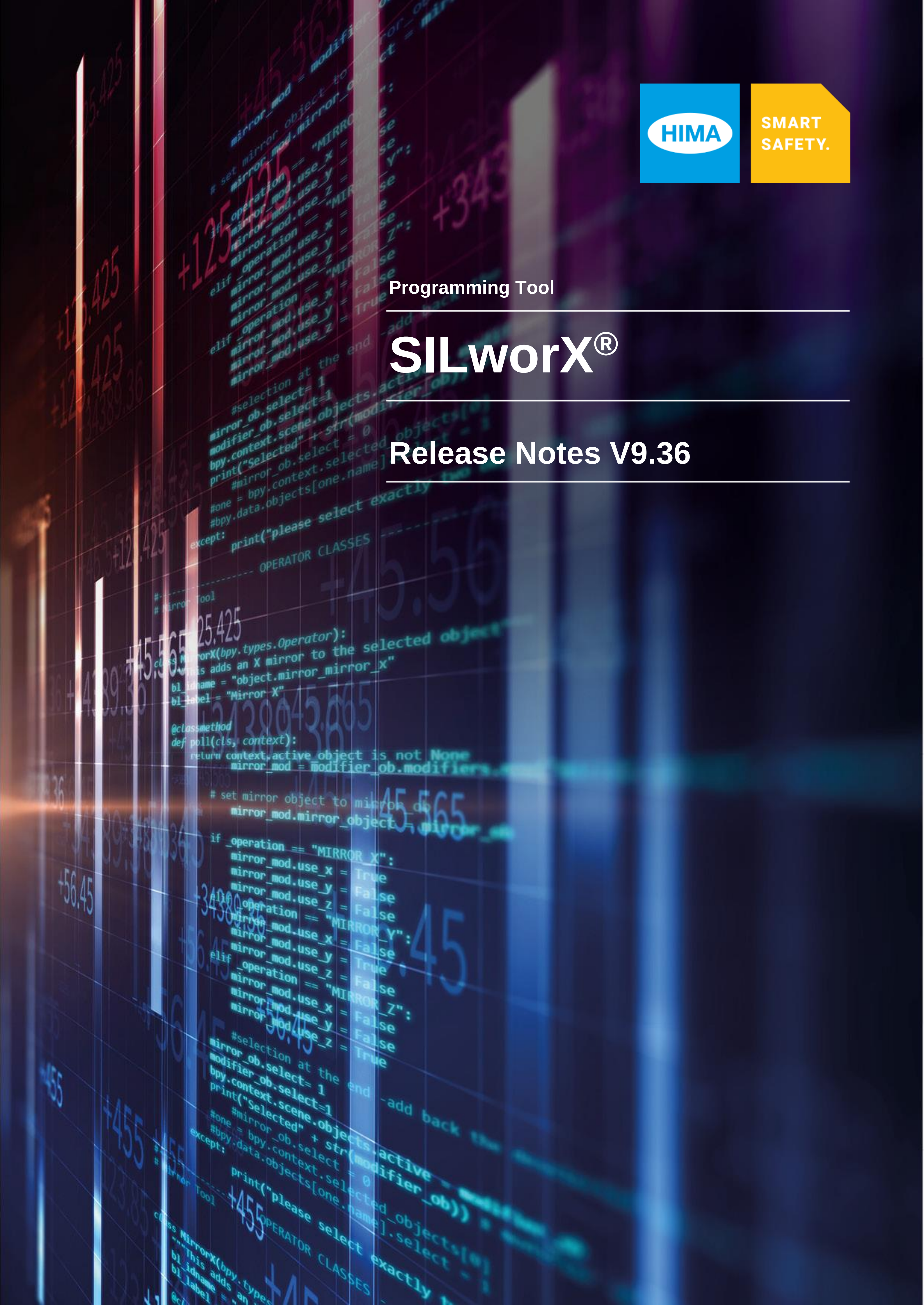

Programming Tool

SILworX®

Release Notes V9.36

All of the technical specifications and information in this manual were prepared with great care and effective control measures were employed for their compilation. For questions, please contact HIMA directly. HIMA appreciates any suggestion on which information should be included in the manual.

Equipment subject to change without notice. HIMA also reserves the right to modify the written material without prior notice.

All the current manuals can be obtained upon request by sending an e-mail to: documentation@hima.com.

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 801 471 D, Rev. 9.01 (2008)	German original document
HI 801 500 E, Rev. 9.01.00 (2008)	English translation of the German original document

Table of Contents

1 SILworX V9.36

This document describes the improvements and new functions of version 9.36 compared to the previous versions.

1.1 Compatibility with the PC in use

SILworX V9.36 can be used in PCs with the following operating systems:

- Microsoft® Windows 7 Professional/Ultimate, 64-bit
- Microsoft® Windows 8.1 Professional, 64-bit
- Microsoft® Windows 10 Professional, 64-bit
- Microsoft® Windows Server 2008 R2, 64-bit
- Microsoft® Windows Server 2012 R2, 64-bit

The minimum requirements for the computer used to run SILworX are specified on the corresponding HIMA DVD.

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 and memory space, result in improved performance.

1.2 Compatibility with the Operating Systems of the Controllers

Version 9.36 can be used for the following HIMA system families:

- HIMax with operating systems V2…V9
- HIMatrix up to F*03
  - CPU operating system V7…V8
  - COM operating system V12…V13
  - Remote I/O operating system V7…V8
- HIMatrix F*03
  - CPU operating system V8…V13
  - COM operating system V13…V18
- HIMatrix M45
  - CPU operating system V10…V12
  - COM operating system V15…V17
- PFF-HM31
  - CPU operating system V8…V12
  - COM operating system V13…V16

1.2.1 No longer supported HIMA controllers

As of SILworX V9.36, the following controllers are no longer supported:

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

SILworX versions as of V9.36 no longer support these controllers (exclusion of liability). This applies to code generation, download and online services.

This means that projects in SILworX may be opened with the specified controllers, e.g., 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.

1.3 Compatibility with Existing Projects

Version 9.36 can convert and edit projects that were created with a previous version. When generating code for the unchanged project, the CRC is maintained except for cases in which code is generated for several resources, see Chapter 1.8.1, Point 2.

1.4 Use of Hardlocks

The following points must be taken into account if SILworX under Window is licensed using hardlocks (USB sticks).

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

1.5 New Functions

1.5.1 Support for Modbus Slave V2

SILworX V9.36 supports the Modbus slave V2 communication protocol for HIMax and HIMatrix. The Modbus slave V2 protocol is not based on a new Modbus specification, but is an enhanced HIMA variant concerning the internal processing of the protocol data on HIMA controllers.

The Modbus slave V2 protocol features the following new functions:

- Reload without cold reload of the communication module.
- Write requests from several Modbus sources and user programs to a single global variable are possible.
- Definition of the authorized Modbus masters with corresponding data view assignment. Unauthorized Modbus masters are rejected!
- Compatibility with HIQuad Modbus slave.
- The settings for the transfer of non-safe protocol data between CPU and COM modules are specified in the configuration file *sys/cpcnsip.config* of the version comparator.

1.5.2 Multi Control Panel

The Multi Control Panel serves as a common platform to simultaneously manage up to 150 resources:

- Status overview (connection state, access type, system ID, states, configuration comparison, etc.).
- Control operations such as Start, Stop, Load,
- The *Go to...*function

1.6 Improvements

This chapter describes the improvements of V9.36 compared to versions prior to V9.36.

1.6.1 Support for X-OPC Server redundancy

For HIMax and HIMatrix, SILworX V9.36 supports the X-OPC redundancy improvements:

- Hierarchy event definitions.
- New X-OPC system variable and online value.
- Cookies and new parameters.

1.6.2 Tables in SILworX

Copy actions from and to external programs via the clipboard. Copy&Paste actions from and to SILworX tables is also possible for newer Excel versions, which no longer support the Symbolic-Link-Format, and for LibreOffice.

1.6.3 FBD Editor

- In the page list of the FBD Editor, the pages are sorted first in ascending order, from the greatest negative number to the greatest positive number, which corresponds to the execution order. Sorting can also be reversed by clicking the table header of the Page Position column.
- An assigned comment field can be placed on a free space within the page. The comment field automatically adopts the details of the subordinated page (page name, page description and drawing number).
- The user can remove all the pages of a column X, unless one of the pages contains elements. In a dialog box, the user can choose if the pages to the left of the column X to be deleted should be moved to the right or the pages to the right of column X should be moved to the left. Moving cannot be applied to columns and rows of empty frames.
- The user can remove all the page of a row Y, unless one of the pages contains elements. In a dialog box, the user can choose if the pages above the row Y to be deleted should be moved downwards or the pages below row Y should be moved upwards.
- It is now possible to add an empty page within the logic. A dialog box appears where the user can indicate the direction in which the pages should be moved.
- The user can add a row of empty pages at row position Y. In a dialog box, the user can choose if the existing pages should be moved up or down.
- The user can add a column of empty pages at column position X. In a dialog box, the user can choose if the existing pages should be moved to the left or to the right.

1.6.4 Reload information

- Given, reload-specific warnings are generated and displayed to the user at different positions in SILworX. These warnings are intended to alert the user to potential problems during reload. The warnings are displayed during reload code generation, in the version comparison during the actual reload process.
- Replace the load number with the internal CRC during the reload. The load number of the configuration running in the controller does no longer have to comply with the load number of the new reload configuration. The compatibility of the reload configuration to be loaded is ensured by a new CRC (AddOnCRC).

1.6.5 User management

The PADT and PES User Management have been grouped to a structure tree element and are now child element of the SILworX project. User groups, user accounts and access permissions are defined in the following tab of the user management:

- PADT User Management: defines the access mode for the SILworX project with password protection.
- PES User Management: defines the access mode for the controller (resource) with password protection. The user groups created in the PADT User Management are automatically adopted in the PES User Management. The access modes to the corresponding controller can be allocated to the user groups with password protection as done so far.

  Data migration from previous SILworX versions is ensured. The requirement for data migration is that a PADT User Management (in addition to the PES User Management) exists in the source project. In particular, projects, which should be directly converted from V2 and contain at least a PES User Management, are no longer converted directly. Such projects must be first converted to a version between V4 and V8 (which automatically creates a PADT User Management). The resulting converted project can be then opened with V9.

1.6.6 New parameters in the resource and OPC Server Set

Using the new mode *Optimized Use of Com. Time Slice,* shorter response times can be achieved for (safe**ethernet**, OPC/HIPROS-2/PADT/RIONC) communication via the process module.

Caution:

This mode can affect the temporal utilization of *Max.Com. Time Slice ASYNC [ms]* and the system parameter *Max. Duration of Configuration Connections [ms]* such that these two times can be subject to more demands (e.g., during reload) [HE27939].

1.6.7 Changeable start value for new offsets in the protocol variable editor

In the protocol variable editor, the "new offsets" function has a dialog box for entering a different start value.

1.6.8 Criticality of two messages for Modbus Slave V1 reduced from Error to Tolerant Error

The criticality of the two messages was reduced from Error to Tolerant Error During validation, the PADT creates a tolerant error for each variable.

The first message appears during validation, if the Redundancy mode switch is set to **OFF** and a global variable is assigned to one of the following system variables:

- System variables of the set
- System variables of the second instance
- System variables from the first instance
  - Response Timeout
  - Average Buffer Fill Level for Requests
  - Maximum Buffer Fill Level for Requests
  - Discarded Requests

The second message appears during validation of the slave instance, if the following conditions are met:

1. Behavior on connection loss is identical to initial data at connection loss.
2. The master request timeout of the instance is 0.
3. If the requirements for a Modbus configuration V2.3 and higher are met, the PADT must report a tolerant error.

1.7 Fixes

This chapter describes problems in versions prior to V9.36 that have been resolved in V9.36.

1.7.1 General

1.7.1.1 SILworX termination due to license loss fixed

If the license is lost, SILworX no longer terminates if other dialog boxes are open.

1.7.1.2 Multi resource code generation no longer closes automatically after an error

The dialog box for multi resource code generation no longer closes automatically if a warning or error message is issued for at least one of the resources.

1.7.2 Hardware Editor

1.7.2.1 Creating a redundancy group

When creating a redundancy group, the Hardware Editor displays all slots [HE27811].

1.7.2.2 A message appears when the name of a redundancy group is assigned twice

SILworX issues an error message if the name assigned to a redundancy group in the Hardware Editor already exists. Additionally, a logbook entry is created. The redundancy group receives the default name [HE27149].

1.7.2.3 Check of event definition names

The event definition now checks if names include characters to be replaced or are in excess of the total name length limits. A warning is issued if this happens [HE25067, HE25207].

1.7.2.4 Revised titles in the German A&E Editor

Some event table titles for alarms and events were in English in the German SILworX user interface [HE20842].

1.7.2.5 Term changed in the module online view

To reflect the correct names used in the system variables of the system, the German term *Ventilatorzustand* was renamed *Lüfterzustand* in the module online view. In addition to this term correction, other textual assignments such as *Ok, nicht verfügbar* and *Relay-Zustand* were harmonized [HE24640].

1.7.2.6 Check of redundancy group names sharpened in accordance with flexible names

If names not complying with the new requirements are used in a redundancy group, a message is issued during validation. The user must correct the names. The change has no effects on the CRCs [HE28652].

1.7.2.7 The code generation accepts again up to 63 characters for A&E event source names

In SILworX V8.34, the A&E event source names were limited to 31 characters. Up to 63 characters can be used again for A&E event source names.

[HE28792].

1.7.2.8 SILworX termination due to differing system IDs fixed

When comparing the I/O configuration files of HIMax I/O modules, SILworX terminated if the system IDs of the compared configuration was not identical. This is now displayed as a normal difference [HE28964].

1.7.2.9 Line structure in the *Maximum System Bus Latency* parameter

In the *Maximum System Bus Latency* parameter, the value zero (0) corresponds to the line structure. In the resource properties, this value can be entered directly or set with radio buttons in a dialog box.

1.7.2.1 Text adjusted for the *Max. Duration of Configuration Connections* warning

The text adjusted for the *Max. Duration of Configuration Connections* warning provides information about a time reserve to compensate for the value calculated with the old procedure [HE29156].

1.7.2.2 SILworX no longer terminates when creating new hardware

When creating new hardware, SILworX no longer terminates if the clipboard contains hardware objects [HE27975].

1.7.2.3 Mode switch position DEFECT

Mode switch position is now set to DEFECT, BOOTSTOP no longer applies. The online values can be found in the online detail view of the HIMax processor module.

1.7.2.4 Harmonized I/O modules names

Naming of I/O module was not consistent. The KE names of HIMatrix and M45 I/O modules have been harmonized to the HIMax naming procedure. The user can view them in the Force Editor and in the version comparison (*ke.config file)*

1.7.3 Protocols

1.7.3.1 SILworX no longer terminates when generating code for a resource with PROFIBUS

SILworX terminated during code generation if the resource contained a PROFIBUS configuration with no PROFIBUS function block directory.
SILworX no longer terminates, and a message appears instead [HE27616].

1.7.3.2 Updating of X-OPC online view corrected

Updating of X-OPC online view was interrupted after a namespace type. Updating was fixed [HE28753].

1.7.3.3 Support of unexpected data types for EU initial and EU final values

All unexpected data types possible for EU initial and EU final values of the global variables contained in the X-OPC configuration are now properly written to the configuration. Data types TIME, BYTE, WORD, DWORD and LWORD are now reliably supported and not decimal numbers are represented as 0.0 [HE29075].

1.7.3.4 Value set checks of packed/unpacked BOOL variables corrected

In the PROFIBUS DP slave of SILworX V8, the value set check was too sharpened. The self packing and unpacking of BOOL variables on the communication module was not taken into account.. This check was removed from SILworX V9 [HE29292].

1.7.3.5 Number of TCP connections increased to 20 for Modbus (V1)

In Modbus slave V1 and Modbus master, the default value of the *Maximum Number of TCP Connections* parameter was increased from 3 to 20.

1.7.3.6 Two default values adjusted in HIPRO-S V2

The default values for Receive Timeout and Production Rate were adjusted in ELOP II (HIQuad).

- Receive Timeout was increased from 1000 ms to 6000 ms.
- Production Rate was increased from 50 ms to 400 ms.

1.7.3.7 The Save function in the ISOfast and HIPRO-S V2 editors corrected

After using the Save function in the ISOfast and HIPRO-S V2 editors, all changes performed are completely saved.

1.7.4 ISOfast

1.7.4.1 SILworX no longer terminates if the configuration is wrong

SILworX no longer terminates in the following cases:

- A 0-byte slave configuration file is imported.
- The optional *URL* element is not indicated in the VendorInfo of an IDD device set.

1.7.4.2 Values are reset when assigning an IDD to a module

The read-only properties from an ISOfast IDD are reset when assigning an IDD to a module The property values, which are set by the user and cannot accept default values from the IDD, are retained. The property values, which are set by the user and can accept default values from the IDD, are overwritten with default values.

1.7.4.3 *Slave Configuration Length [Bytes]* displayed correctly

The ISOfast *Slave Configuration Length [Bytes]* is displayed correctly. 0 is displayed when assigning a new device description that does not allow slave configuration files.

1.7.4.4 Numeric values in the ISOfast online pane and no error messages in the logbook

The ISOfast Control system variable is displayed in the ISOfast online pane as hex number and no longer as textual value.

No error issued is in the logbook if the value is not equal to 0x80, 0x00, 0x10 or 0x11.

1.7.4.5 Validation no longer terminates

Validation no longer terminates after the first error, but continues to run. This allows other errors to be reported.

1.7.4.6 Other fixes

- Validation advice messages are now added to the logbook as expandable entries.
- The *Go to…* function in the version comparison now jumps to the corresponding editor.
- If the IDD reference exists, the default value for new selections is *None*.

1.7.5 FBD Editor

1.7.5.1 SILworX no longer terminates when validating function blocks containing conflicts

If the associated function blocks had conflicts, SILworX terminated during validation e.g., when opening the FBD Editor, validating the structure tree and code generation.

Tolerant behavior if a function block output no longer exists, but one of the function block uses accessed via interface access has a reference to that output. This also applies if the entire function block no longer exists and is referenced through interface [HE28503].

1.7.6 ST Editor (Structured Text Editor)

1.7.6.1 No more error messages in connection with drag&drop actions in the ST Editor

Dragging global variables from the Global Variables tab of the ST Editor and dropping them onto its text area is no longer rejected with an error message. Not even if a variable with that name already exists as VAR_EXTERNAL [HE27817].

1.7.7 Logic

1.7.7.1 Behavior of the time variable in the X-OTS fixed

The time variable in the X-OTS (Operator Training System) accepted an invalid value. In user programs generated with *Code Generation Compatibility* ≥ SILworX V7, the timers in X-OTS now behave correctly after a snapshot restoration [HE27929].

1.7.7.2 REAL calculations corrected in OLS and X-OTS

REAL calculations in the OLS (Offline Simulation) and X-OTS (Operator Training System) were corrected as follows:

- The EXPT function correctly returns +INF instead of -INF if the base -3.40282346e + 38 of type REAL is exponentiated by 2147483647 of type DINT [HE15388].
- ADD correctly returns NaN, if one of the two operators is +INF or -INF [HE20249].
- When executing the ADD and MUL functions with three or more inputs of type REAL, OLS and X-OTS now calculate the results with the same accuracy as the controllers.
  The changes result in a new CRCs for the X-OTS targets [HE28632].

1.7.7.3 SILworX no longer terminates when creating the documentation

SILworX terminated when opening the detail view of the `.ldb` file in the version comparison if a program name contained a slash (/). Under this condition, SILworX also terminated when attempting to create the documentation if this contained data from the version comparison [HE29107].

1.7.7.4 Changes to instances of POU types are now displayed correctly

Changes to instances of POU types with // or /* are now displayed correctly in the version comparison If an instance type name contained the character string //, the subsequent characters were not taken into account during a version comparison related to the instance. The same applied for characters in the type name between /* and */ or to the end of the type name. This means that, provided that the new type name only differed from the old one in those parts, the version comparison was likely to show no instance change, although the POU type invoked by the instance was replaced or renamed.
Compatibility with SILworX versions prior to V9:
Both version comparisons were created with a version < V9: The version comparison does not display any changes in the cases previously mentioned.
A version was created with SILworX < V9, the other with SILworX ≥ V9: The version comparison

always displays a change for the instance with the type name complying with the conditions mentioned before [HE29119].

1.7.7.5 Message about declaration sequence is missing in the version comparison

With respect to the declaration order of variables and instances within a POU, the version comparison ignored the following elements:

- All the instance of the following function blocks: 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.
- The own ENO variable of the POU.

A change in the declaration order was therefore not reported if it could not be recognized without taking those elements into account.

This has been fixed. The shifted sequence is now correctly displayed in these cases; however, only if both comparison versions were generated with SILworX V9 and higher [HE29140].

1.7.7.6 Display problems fixed in the FBD and ST comparator

Literals and array variables with index access that are used in assignments are now correctly displayed in the FBD version comparison.

Multi-line structured text statements are now correctly displayed in the version comparison.

1.7.8 Online

1.7.8.1 SILworX no longer terminates if online tools are used

SILworX no longer terminates in the following cases:

- When closing the last online tool while the configuration download is running.
- When changing the system ID in the *Search via MAC* dialog box of the *System Login* and subsequently using the *module login*.
- When opening a CPU detail view (slot 1), which hides a connection to an OPC Server.

1.7.8.2 SRS of the system bus now displayed correctly in the Search via MAC dialog box

If the *Search via MAC* dialog box is changed in HIMax system bus modules, the SRS is now correctly displayed.

1.7.8.3 Module names correctly displayed when updating the operating system

Module names are now correctly displayed in the logbook when updating the operating system.

1.7.8.4 Changing the network parameters via MAC address

Changing the network parameters via MAC address (login dialog box) now works for gateways with 0.0.0.0 for all remote I/Os with firmware prior to V8. Additionally, the gateway can also be set in this case.

1.7.8.5 Defined the order of the information in the Control Panel

The information provided in the structure tree of the Control Panel is now sorted in accordance with the requirements and no longer alphabetically. In this way, undesirable positioning of data within the Control Panel is avoided.

1.7.9 System configuration

1.7.9.1 Optimized error message when reload is rejected

The error message was optimized so that the user can better understand why the reload was rejected [HE28780].

1.7.9.2 Renaming global variables of Array data type corrected

Child elements of global variables receive a correct name if the data type is an array and the name was changed by search and replace [HE27851].

1.7.9.3 Program size displayed in the Control Panel and version comparison

The user program size displayed in the Control Panel could differ from that displayed in the version comparison. Both the Control Panel and the version comparison now display the memory actually used [HE25706].

1.7.9.4 SILworX terminated when attempting to edit global force data for multiple graphic objects

When calling Edit Global Force Data, SILworX could terminate if multiple graphic objects connected to the same variable or the same structure variable element were selected.

Example:

- A POU contains a variable with a value field.
- In the online test, the text area and the input pin of the value field are selected.
- In the context menu, the *Edit Global Force Data* option is clicked.

[HE28489].

1.7.9.5 Unexpected detection of inconsistencies in user-defined data types fixed

By converting the existing SILworX projects to V9, inconsistencies that may have existed when using user-defined data types were detected. This could manifest itself in corresponding error messages concerning those data types or the associated variables. This behavior was fixed.

1.8 Restrictions

When using SILworX versions V9.36, 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.

1.8.1 Restrictions when converting from previous versions

Observe the following restrictions when converting projects from previous SILworX versions.

- All SILworX versions: CRC change in X-OPC configuration.

  Generating a new code for X-OPC or X-OTS in a project converted from a previous version causes the CRC of the opc.conf file to change.
- All SILworX versions: Code generation of several resources connected via safe**ethernet**

  Generating code for several resources that are connected via safe**ethernet** and belong to a project converted from a previous version, may cause the CRC to change. Generating code for a single resource does not cause the CRC to change:
- SILworX V5: Licenses are sorted by names, which may cause the CRC to change

  During code generation, SILworX V6.x 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.

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

1.8.2 Sequential function chart: No indication of deadlocks

Combined use of selection and simultaneous nodes causes deadlocks, i.e., undefined states in which either all steps or no steps are active. SILworX does not display any advice message for the user [HE17716].

1.8.3 Converting a safe**ethernet** connection from prior to V6 to V6 and higher

When converting from a version prior to V6 to V6 and higher, observe that the timing master and its behavior may be changed. For further details on the timing master, refer to the SILworX communication manual (HI 801 101 E) V6.01 and higher.

Workaround: Set the timing master explicitly [HE25666].

1.8.4 Use of cross-references within ST editors

When cross-references of functions are displayed, only the usages within function blocks and user program appear, but not the use within ST editors [HE26451].

1.8.5 2700 consecutive comment lines are not possible in the ST Editor.

SILworX terminates when commenting out 2700 consecutive lines in the Structured Text Editor.

Workaround: Partition long comments, e.g., by grouping 1000 lines to one comment [HE27464].

1.8.6 Empty pages cannot always be deleted

The **Delete Empty Page** context menu option is not active under the following conditions:

- 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.

1.8.7 The number of instances of function blocks restricts the program's reloadability

If the user program has a very nested structure, the maximum number of operations necessary to perform a reload may be exceeded.

Only resources with 21845 or less instances can be reloaded. Depending on its structure, a user program may not be capable of reload in connection with a significant lower number.

Possible workaround:

- Avoid using POUs and data types that are excessively structured.
- In POUs with many instances, the problem can be worked around by changing the type of the variables from VAR to VAR_OUTPUT [HE26889].

1.8.8 Deleting and re-inserting many objects during a load procedure

The number of objects that can simultaneously be added to a resource during a load procedure (download or reload) is limited to the maximum number of objects within the resource. During a load procedure, it is only possible to add as many objects as previously allowed. Deleting objects in the same load procedure does not increase the number of insertable objects.

Insertable objects are, e.g., programs, safe**ethernet** connections, hardware modules, remote I/Os or protocols.

Moving objects, e.g., modules, to another rack is the same as deleting them and inserting the new ones.

Workaround: Delete and insert the new objects in separate load procedures [HE25955].

1.8.9 SILworX may not always be started under Windows 8

Under unspecified circumstances, it may happen that SILworX will not start under Windows 8.

Workaround: Reboot the computer. In this case, an error message also recommends rebooting the PC.

1.8.10 EXPT function for large negative exponents

During the offline simulation and X-OTS, the EXPT function provides the result NaN (not a number) instead of 1, if 1.0 is used for the basis (IN1) and a large negative number or –∞ is entered for the exponent (IN2).

Workaround: If this particular case is relevant for the application, the function must be handled in the user program [HE14526].

1.8.11 Windows synchronization deletes the project file from the network drive

The following sequence could cause a project file to be unintentionally deleted:

- The project file is located in a directory on a network drive.
- The Windows synchronization is running on the client PC.
- The project file is being edited with SILworX on the client PC.
- A synchronization process is started.
- The user stops editing the project file and exits SILworX.

Cause: When a project is being edited, SILworX saves the project to a temporary file. When the project is closed, SILworX deletes the previous project file and renames the temporary file. In the process, the Windows synchronization on the server may delete the previous project file, but not rename the temporary file.

Workaround: Only perform synchronization after closing the project in SILworX [HE25231].

1.8.12 SILworX terminates if large array variables are used in protocols

Attempting to adopt an array variable with more than 32 768 elements into a communication protocol causes SILworX to terminate.

Workaround: Partition large array variables into several smaller parts [HE24258].

1.8.13 SILworX V4 project integrity

In SILworX V4, deletion actions could cause objects to remain in the database, but be no longer editable. These objects did not affect the rest of the project, but they were reported during the project integrity check.

Projects that contain such "residual" objects are very likely not to be converted to SILworX V6 and higher. 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:

a Archive all the child nodes of the project that are positioned in the structure tree under the project, except for Programming and Debugging Tool.

b Create a new project in the previous SILworX version.

c Delete the *Configuration* node in the new project.

d In the new project, restore the configuration archived in point a) and, if existing, additional child nodes of the project.

The project just created should be convertible to the current SILworX version [HE25994].

1.8.14 SILworX projects in directories with names not complying with the range of values

The use of SILworX is restricted in that the user cannot store project files in directories for which the path name contains characters that do not comply with the value range of the Windows code page.

Condition:

Also in this case, the behavior of SILworX probably depends on the Windows code page. On a "German" Windows computer, the error can be reproduced as follows:

Store the existing SILworX project file in a directory for which the path name does not comply with the Latin1 encoding scheme, e.g., it contains a dotless ı. Start SILworX. Select *Open Project.* A dialog box appears: click **...** to open a file selection dialog box. Navigate to the project file and select the project file. Confirm the action. Check the file path displayed by the *Open project* dialog box. (The name has been reduced to the Latin1 encoding scheme.) Now confirm the setting in the *Open Project* dialog box. SILworX now issues an error message informing that the file with the name reduced to Latin1 could not be found.

Workaround:

Do not save any SILworX projects with names that do not match the Windows code page, in directories [HE29103].

1.8.15 Invalid date if the project history is imported from an English V2 project

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 1st November

2013 instead of the 11th January 2013. 1/13/2013 is interpreted as invalid date and results in the default value 1st January 2000 [HE28418].

1.8.16 A project becomes unusable due to name errors

If a non-breaking space and space are used in combination in the program name, SILworX can terminate [HE29072].

1.8.17 Path names with more than 32 767 characters cause SILworX to terminate

SILworX terminates if objects with names containing more than 32 767 characters are processed. For example, the path name is composed of the following elements:

Project name + configuration name + resource name + library name + … + function block name

Very long names can result from nested libraries if the individual libraries have long names.

Workaround:

a When possible, avoid long path names.

b If required, shorten the names of objects located at the beginning of the path.

c If necessary, delete the lowest library. In doing so, SILworX terminates again, but the library is deleted once the project has been restored [HE26815].

1.8.18 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.

Workaround:
Delete the global variable at all positions where it is used so that VAR_EXTERNAL disappears. Then insert it again at all positions [HE24487].

1.8.19 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 [HE19688].

1.8.20 Conflict icon remains visible, in spite of fixed conflict

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 [HE24339].

1.8.21 User program

1.8.21.1 The following message no longer appears: *order of the variable and instance declaration has changed*.

The version comparison no longer displays a declaration sequence change in the POU. This happens when a configuration generated with SILworX < V9 is compared with a configuration generated with SILworX ≥ V9, and at least one of the following standard functions is used in the configuration generated with ≥ V9:

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.

Workaround:
If the project state matching the comparison basis (e.g., the loaded configuration) is still available (e.g., if backup files (*.A3) automatically created are loaded):

- Convert a copy of this old project state to SILworX V9, generate code and export the result via the start dialog box of the version comparison.
- In SILworX V9, open the new project, in which the version comparison should be performed. Import the configuration exported during step 1 into the start dialog box of the version comparison and use it as a comparison basis (tick the imported and generated configuration). Start the version comparison [HE29141].

1.8.21.2 Behavior of the EXPT function

The behavior of the EXPT function on the PES target does not comply with the IEEE-754 standard.

- 1.0 exponentiated by NaN
  Output = 1.0 expected: NaN
  ENO = TRUE expected: FALSE
- NaN exponentiated by 0.0
  Output = 1.0 expected: NaN
  ENO = TRUE expected: FALSE

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

Workaround: If ENO is required, trap or avoid a NaN on the input [HE29121].

1.8.21.3 The version comparison displays changes, but no CRC changes

For used function blocks, the version comparison displays changes, but no CRC changes.

This is the case if function blocks with umlauts were used after a project conversion from SILworX version prior to V4.116.

Workaround:

Only spaces and characters from the following list may be used 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
- $ % & ( ) * + - / : ; < = > ? \ ^ _ ` { | }

[HE28833].

1.8.21.4 Different handling for inputs of SFC elements

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 calculated in the FBD logic always originate from the previous cycle. The evaluation of the input values, however, reveals small differences. These differences and the connected impact are explained in the following table:

SFC Element	Evaluation of the inputs	Effect	Workaround
SFC Transition	During the FBD processing, the input value of an SFC transition is stored in the SFC transition memory. Thus, a step sequence does not move before the second cycle, even if the SFC transition contains the value TRUE.	If a field with the value TRUE is used as the SFC transition input after the initial state, the SFC transition is only switched in the second cycle.	When programming sequences, users must take into account that an SFC transition is performed in the second cycle at the earliest.
SFC Action	The input value of an SFC action is read from the source during the processing of the SFC action. The value FALSE is read whenever the source is a function since functions are initialized at the beginning of the POU processing and are only processed after the SFC actions.	If a function output is used as input of an SFC action, the SFC action input is always FALSE, even if the function output has the value TRUE.	To use a function result as input value for an SFC action, a variable must be connected between function output and SFC action input.

Table 1: Behavior of SFC Transitions and SFC Actions Resulting from the Processing Sequence

[HE28370].

1.8.21.5 Force messages with no reference to target object

Force messages in the logbook provide no details on the objects they refer to. The user cannot discern from which user program forcing was started [HE25923].

1.8.21.6 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:

Create different structure variables containing elements:

- to which the user program writes.
- to which hardware or communications write [HE15700].

1.8.21.7 Elements of structure variables used as index

Elements of variables with struct data type cannot be used as array index [HE16159].

1.8.21.8 Invalid array index addresses a random array element

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.

Using suitable programming, users must ensure that array elements are only accessed through indexes within the value range of the array [HE25075].

1.8.21.9 Use of specific function blocks in HIMatrix devices prior to F*03

For HIMatrix devices prior to F*03 (e.g., F30 01, F31 02, F35 01, F35 012 and F60), avoid using the mentioned standard function blocks in the following cases:

- If EN and ENO are visible.
- If the function blocks and data types are among those mentioned in Table 2.

Function block type	Data type
ADD	ULINT/LINT
SUB	LINT
ADD_TIME	TIME
SUB_TIME	TIME

Table 2: Not Allowed Combinations

[HE29128].

1.8.21.10 DIV_TIME issues a divisor error to ENO.

The DIV_TIME function from the standard library improperly sets the ENO 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 [HE15199].

1.8.21.11 MUL provides wrong results

The MUL function block provides faulty values if the following conditions occur simultaneously:

- Use of one HIMatrix prior to F*03
- Use of the LREAL data type.
- One input has the value +/-∞, the other input the value NaN (not a number).
  In this case the result is -∞, and not NaN as specified [HE21924].

1.8.21.12 Copying obsolete online values

In the Force Editor and other force tables, online values can be copied to the clipboard. If values that were not located in the visible window are copied, they may be obsolete.

Workaround: Sorting the table after the process value provides the current values. Afterwards, the values may be sorted in accordance with the desired criteria, and may be copied once sorting has finished (indicated by the mouse pointer shape). Select and copy only the table parts that are located in the visible area [HE23314].

1.8.21.13 SILworX terminates when forcing is performed too fast

When forcing is performed too fast, SILworX terminates if the following conditions are met:

- In online mode, multiple variables are forced too fast in the user program.
- In the Force dialog box, the *Automatically close the dialog upon success* option is ticked.
- The next variable from the table is double-clicked before the dialog box for the variable, which was forced last, is closed.

Workaround: Only force further variables if the force dialog box of the previous forcing process is closed.

1.9 Special Points

When using SILworX, the described characteristics must be observed.

1.9.1 General

1.9.1.1 Scaling settings for an analog value in the Hardware Editor

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 [HE16388].

1.9.1.2 Logic operations of BOOL variables from third-party systems

Logic operations of BOOL variables having values that originate from third-party systems can provide results that differ from those expected.

The cause is that the coding of BOOL values used in the third-party system deviates from the coding used in the HIMA system.

Two workarounds are possible:

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

[HE13042].

1.9.1.3 Impossible to save certain changes in a SILworX editor

After specific changes are made in an editor, the following message appears when attempting to save: *Impossible to save changes.* After confirming the message, however, the changes are saved.

If the SILworX editor is then closed and re-opened, the following message appears: *The required data is being processed*.

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

Workaround: Avoid exchanging names. If required, restart SILworX [HE11613].

1.9.1.4 Variations of the cycle time during LREAL calculations.

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

1.9.1.5 Module login offered by SILworX when the connection is interrupted

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

1.9.1.6 Online help associated with a POU not printable

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

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

1.9.1.7 System variable values during the online test and offline simulation

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.

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 possible because the system variable in the program is connected with a variable. This variable can be forced [HE15396].

1.9.1.8 Import of export files from a previous version

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.

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 [HE21691].

1.9.1.9 Misleading indication of the force status for local forcing in connection with HIMatrix F*01x devices

For HIMatrix F*03 devices (such as F30 01, F31 02, F35 01, F35 012 or F60 CPU 01), the parameters indicating the status of local forcing (located above the force table) are displayed with regular values as if the information was actually available. In particular, these parameters are Force State, Forced Variables, Remaining Force Duration and Force Time Reaction [HE23021].

1.9.1.10 Inconsistent state due to links to objects that have already been deleted

If two editors are used simultaneously, the following sequence may lead to an inconsistent project state:

a In the first editor, a linkable object is deleted, but the action is not saved.
b In the second editor, a link to the deleted, but still visible object is created.
c The changes are saved in the second editor.
d The changes are saved in the first editor.

This causes the second editor to have a link to an object that no longer exists! The project state is not consistent and can cause the editors to terminate when they are opened.

Workaround:

If the editor can be opened, the link can be set to a valid object. Otherwise, the entire parent object must be deleted and created again. The parent object can be a POU or a hardware part [HE27882].

1.9.2 Function blocks and functions

1.9.2.1 Changed value for VAR_INPUT variables of 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).

This behavior may result in errors if all the following conditions are met:

- The VAR_INPUT variable is a global variable.
- The VAR_INPUT variable is additionally used in the called function block as VAR_EXTERNAL.

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: Do not simultaneously use VAR_INPUT and VAR_EXTERNAL for transferring the value of identical global variables [HE17740].

1.10 Upgrading from a Previous Version

Project data from previous versions can still be used in V9.36.

No CRC changes occur as long as the minimum configuration version setting remains unchanged for a resource and none of the cases described in Chapter 1.8.1 has occurred. 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 V9.36:

- Generate code for all resources prior to conversion. This allows potential deviations after the conversion to be detected during generation.
- Prior to converting the project, save it, e.g., on a removable medium.
- Open the project in V9.36 and convert it.
- Since the conversion is extensive, check the project integrity after completing the conversion.
- Generate the code in V9.36 to detect potential errors and check if CRCs have changed.
- 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.

2 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

Appendix

Index of Tables

Release Notes

HI 801 500 E

For further information, please contact:

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

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

🌐 www.hima.com

HIMA SMART SAFETY.

www.hima.com