

The HIMA logo consists of a blue square containing a white oval with the word "HIMA" in blue capital letters.

HIMA

The SMART SAFETY logo is a yellow rectangle with the words "SMART SAFETY." in blue capital letters.

SMART
SAFETY.

Programming Tool

SILworX[®]

Release Notes V10.64

All of the HIMA products mentioned in this manual are trademark protected. Unless noted otherwise, this also applies to other manufacturers and their respective products referred to herein.

HIQuad®, HIQuad X®, HIMax®, HIMatrix®, SILworX®, XMR®, HICore® and FlexSILon® are registered trademarks of HIMA Paul Hildebrandt GmbH.

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.

© Copyright 2020, HIMA Paul Hildebrandt GmbH
All rights reserved.

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 801 501 D, Rev. 10.03 (2008)	German original document
HI 801 502 E, Rev. 10.03.00 (2008)	English translation of the German original document

Table of Contents

1	SILworX V10.64	5
1.1	Compatibilities	5
1.1.1	Compatibility with the PES operating system	5
1.1.2	No longer supported HIMA controllers	5
1.1.3	Compatibility with existing projects	5
1.1.4	Compatibility with the PC in use	5
1.1.4.1	Use of hardlocks	5
2	New Functions	6
3	Improvements	7
3.1	FBD Editor	7
3.2	Structured Text	7
3.3	Watchpage Editor	8
3.4	Smart Safety Test	8
3.5	Hardware	8
3.6	Online	8
3.7	Archiving and Restoration	8
4	Restrictions	9
4.1	FBD Editor	9
4.2	Structured Text	11
4.3	Cross-References	12
4.4	File Selection Dialog Boxes	12
4.5	Hardware	13
4.6	User Program	14
4.7	Version Comparator	17
4.8	Code Generation	18
4.9	Reload	19
4.10	Protocols	20
4.11	Project	22
4.12	Project History	23
4.13	Documentation	23
4.14	User Management	24
5	Special Points	25
6	Upgrading from a Previous Version	29
6.1	References	29

1 **SILworX V10.64**

This document describes the improvements, new functions and restrictions of SILworX V10.64 compared to the previous versions.

1.1 **Compatibilities**

1.1.1 Compatibility with the PES operating system

SILworX V10.64 can be used for the following HIMA system families:

- HIMax
- HIMatrix F systems
- HIQuad X

1.1.2 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.

1.1.3 Compatibility with existing projects

Version V10.64 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.

1.1.4 Compatibility with the PC in use

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.

1.1.4.1 Use of hardlocks

The following points must be taken into account if SILworX is licensed on Windows 7, 8, 8.1, and 10 using hardlocks (USB sticks):

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

2 New Functions

- The parameter settings of the X-MIO 7/6 01 module have been redesigned and are now more user-friendly. The parameters (limit values) in the Hardware Editor can now be edited and are expressed in 1/min and 1/min², instead of mHz and mHz/s. The resulting frequency, however, is still displayed in mHz and mHz/s.

SILworX now automatically calculates the scaling factor via the new parameter Gear Wheel Teeth.

Additionally, two new parameters for the X-MIO 7/6 01 are available in the Hardware Editor.

* The parameter *Max. Allowed Speed Deviation [1/min]* is used to define the maximum speed deviation allowed between the three input signals.

* The parameter *Gradient Monitoring* is used to activate or deactivate monitoring of the acceleration (*Max. Increase Positive/Max. Increase Negative*).

No downward compatibility is available for the X-MIO 7/6 01! SILworX generates a new io4io.config version for the X-MIO 7/6 01. This results in a CRC change.

No reload is possible.

To be able to use the new functions, the appropriate operating system version must be loaded in the X-MIO 7/6 01. When converting previous SILworX projects including an X-MIO 7/6 01 module, a warning appears. A subsequent code generation generates an error message since the maximum allowed speed deviation is set to 0 during conversion.

3 Improvements

3.1 FBD Editor

- Renaming function blocks could lead to conflicts in the logic's inputs and outputs of any function block instances that had the previous name of the function block. This error has been removed.
- The issue that a system crash occurred in certain circumstances when saving within the FBD Editors has been fixed. If an FBD logic part with connections was rejected during a copy action and the copy action was therefore aborted, it could happen that a subsequent saving action within the FBD Editor caused the system to crash.
- Deleting an OLT field, which is connected to a variable or block instance at a connection point, in the FBD Editor no longer causes an incorrect message to be issued during code generation if another connection exists at the connection point.

3.2 Structured Text

- The FOR loop ends properly when the control variable is outside the range of values of its data type.

The CRC of the POU changes when generating code against previous SILworX versions if the control variable may be outside the range of values of its data type. The change is displayed in the version comparator.

When processing the FOR loop, the actual final value of the control variable may be outside its value range so that an overflow occurs at runtime. In such a case, the FOR loop will not be ended properly.

If the final value may be outside the range of values, SILworX will generate corrected code for the FOR loop. This therefore changes the CRC of the POU when generating code against previous SILworX versions. The change is displayed in the version comparator.

If the data type of the control variable is LINT, the code correction and CRC change may also occur when it would not be necessary.

3.3 Watchpage Editor

- Watchpages: The use of drag&drop in connection with global and local variables is no longer subject to restrictions. In practice, variables with a name containing a special number suffix according to the following pattern can be used: "*_0#*" where *#* represents one or more digits and any number of leading zeros can be used, e.g., *Program.Pou1.Var_01*, *Global Variable_002*.

3.4 Smart Safety Test

- If a tolerance value greater than the set point is set for an unsigned global variable, this tolerance value is now processed correctly. Supposing a set point 10 with tolerance value 11, the lower limit of the tolerance range was not properly calculated in SILworX V10.58.0. This caused the status 'Not OK' to be displayed during the test run, although the actual value was within the required limits.

3.5 Hardware

- After adding or deleting a remote I/O, the user can now open the detail view of the corresponding configuration file. Furthermore, the documentation no longer crashes as soon as the version comparison is included.
- If the watchdog time or process safety time are set online or in the project, a warning message informing that the relation $3 * \text{watchdog time} \leq \text{process safety time}$ has been violated, now also appears for HIQuad X.
- Temperature threshold 1 messages of the F-PWR 01 power supply are now displayed as 'Info', irrespective of whether the user has deactivated warnings for the base rack.
- Minor bugs have been fixed in the diagnostic overview for HIQuad systems:
 1. The default refresh rate has been increased.
 2. The table's sorting order still applies after a refresh.
 3. Messages related to already deleted modules are removed.
 4. Temperature threshold 1 messages are displayed as 'Info'.
 5. Projects with English language settings show English message texts.

3.6 Online

- For long-running online commands, such as firmware update, in conjunction with large systems such as a HIMax system with many racks, an error message may appear after a long time and/or at least one of the command dialog boxes may get stuck and no longer be closed. This error has been removed.

3.7 Archiving and Restoration

- Archives created in a version prior to V10 that have an invalid file name (e.g., containing /) cannot be restored in V10. This error has been removed.
- SILworX terminated with an error message when trying to restore an archive containing objects for which no license is available.

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:

No workaround exists for this issue.

- 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.

[HE 27520 t.koller]
- **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.
- **Effect:**
Deleting a previously used POU and creating or inserting a new Typical using the name of the deleted POU leads to a system crash during code generation with this unsupported type.

Condition:
Create a POU *B* function block and use it in a program. Save and close all. Delete the POU *B* and create a Typical function block *B* there. Go where *B* is used and call up the update. The errors are rectified and the Typical is called up.

Workaround:
The change can be undone by using the correct type of function block.. The correct type of instance can be checked by double-clicking the instance. The POU called up must not have the Typical type, which can be recognized by a small icon next to the name.
- **Effect:**
Simultaneously changing the data type and name of a global variable causes the reference to be lost when referencing child elements of this global variable.

Condition:
Requirements: Array_1[1...5], Array_2[1...8], global variable of type Array_1. Reference to

GV[1].

Action: Open the Global Variable Editor öffnen. Change the data type of the global variable to Array_2. Change the name of the global variable. Save within the editor.

Result: The reference still includes the previous name GV[1], but this can no longer be found.

Workaround:

Either save between changing the data type and the name, or - if it is already too late - manually adjust the reference.

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.

- Effect:

Copying a user-defined data type from one variable to another within an ST Editor leads to a system crash if the data type selection box of the target variable is in editing mode.

Condition:

Open an ST function or an ST function block and create 2 variables (any variable type).

Assign a user-defined data type to the first variable, then select this data type in the selection box and select *Copy* in the context menu. Then go to the data type for the second variable and press *Enter* to switch the selection box to editing mode. Insert the previously copied data type while in this mode.

Workaround:

Copy a selected, user-defined data type from one variable to another within an ST function block without switching the data type selection box of the target variable to editing mode, i.e., without pressing the Enter key or double-clicking it.

4.3 Cross-References

- **Effect:**
If variable cross-references are used in HIPRO-S, no structure information is displayed; the field remains empty.

Condition:
Use a variable in a HIPRO-S connection and view the cross-references in the Global Variable Editor: The *Structure Info* column remains empty.

Workaround:
None.
- **Effect:**
Cross-references are shown in the language that was used for saving them. The current language setting in SILworX is irrelevant

Condition:
Create cross-references that contain language-dependent parts. Then open the project with a SILworX instance in another language.
The cross-references still contain the texts in the previous language.
Example: Names of system variables in the Hardware Editor are shown in the cross-reference of the global variables, but if they are linked in a German SILworX version, they will still remain in German in the English SILworX version.
- **Workaround:**
 - a) Ignore the issue.
 - b) Save the editor where the link exists. To be able to save the editor, make a modification in this editor. This regenerates the cross-reference information.

4.4 File Selection Dialog Boxes

- **Effect:**
After an invalid file selection, the hardware XML import dialog box remains open and only responds to the Cancel button.

Condition:
Select an incorrect or invalid file path in the hardware XML import dialog box and start the import.

Workaround:
Terminate the dialog box with Cancel and restart the import.

- **Effect:**
The restore dialog box shows only the path for the last restored archive files. The list of the recently created archives is missing.

Condition:

Open or create a project. Create one or more archives for the project. In the restore dialog box, open the selection box under archive files: This is empty; no other archive files are listed.

Workaround:

Select the archive file by browsing to the archive path.

4.5 Hardware

- **Effect:**
The detail view toggle button in the Hardware Editor is not active. The view can be closed using the Close button.

Condition:

Start the Hardware Editor online view and open a module's detail view: The detail view toggle 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 RIOs.
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 RIOs.

Condition:

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

Workaround:

Use the error statistics in the master.

4.6 User Program

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

$\text{NaN} ** 0.0 := 1.0$ 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.

- Auswirkung:
The MUL function block provides faulty values if the following conditions occur simultaneously:
 - HIMatrix standard resource (e.g., F30 01, F35 01, F60 01)
 - Data type LREALInput IN1 has the value $\pm\infty$, input IN2 has the value NaN (not a number)
 - In this case, the result is $-\infty$, and not NaN as specified.
 - In this case, ENO provides the correct result, i.e., False.

Condition:

See Effect.

Workaround:

Ignore the result in this specific case.

- **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.
- [HE 15700 s.braun]
- **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.

4.7 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.

4.8 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.

4.9 Reload

- Effect:

A reload code generation leads to a reload warning if the partner's download info cannot be detected:

safeethernet reload sequence started. A dual configuration has been created. No download configuration is available for "[partner name]" to align the safeethernet signatures.

This also occurs if the reload code generation was started without prior changes.

Condition:

Establish the safeethernet connection to the proxy resource. Perform the reload code generation on the real controller without any changes.

Workaround:

Not available.

- 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.

4.10 Protocols

- Effect:

Create a safe**ethernet** 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.

- Effect:

A fault is reported during verification if the maximum permissible data volume of a PROFIBUS DP slave is exceeded.

Condition:

The process data for a PROFIBUS DP slave must exceed the maximum Input data volume > 192 bytes (for HIQuad H41X or 51X > 244 bytes).

Output data volume > 240 bytes (for HIQuad H41X or 51X > 244 bytes).

Perform the verification or code generation to create the message.

Workaround:

Observe the limits.

- Effect:

SILworX does not reject the reload code generation when a COM protocol is switched from Classic to Write Request. The controller behavior is further monitored in HE 31070.

Condition:

Perform a reload in a COM, e.g., to upgrade the Modbus slave from V1 to V2, but keep the name of the protocol.

This results in a reload transition that SILworX must reject.

In general, this applies if a reload is performed to convert a COM protocol from a Classic (Modbus slave V1, PROFIBUS slave, Modbus master, etc.) to a Write Request protocol (currently Modbus slave V2 and future protocols).

Workaround:

Rename the protocol. In doing so, the protocol obtains a new identity and the reload can be performed as a cold reload.

- Effect:
During code generation, unauthorized messages of this type are generated: *Uncritical error: Redundant master 'Master2' not found in the redundant slave 'Modbus-Slave_1'.*

This message is not displayed during the actual validation.

Thus, mutual master references by name are not possible.

The functionality is only intended as a validation aid for the user. It does not affect redundancy or configuration.

Condition:

- 1) Create a Modbus slave V2 with two slaves.
- 2) Give the masters different names.
- 3) In each master, activate the explicit check of redundant masters and name the redundant masters.

This does not result in a message during validation of the Modbus slave set, but during code generation.

Workaround:

For validation purposes, it is sufficient to reference only one master to another. Therefore, remove the reference from the problem reported unauthorizedly. This will not result in any validation loss.

4.11 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 CPU or COM module, for example, which was used in a safeethernet 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 CPU and a COM to each. Create a safeethernet connection between the resources and use both modules. Then delete the COM or CPU 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.

4.12 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.

4.13 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.

[HE17921 t.koller]

4.14 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.

- Effect:

PES accesses of user groups with one (or more) quotation marks (") in their name do not work in the usual way:

When logging on to a controller, the password of the user group for this controller is not automatically suggested. Instead, the name of the group without quotation marks (") and the corresponding password could be entered manually. To do so, however, the password must be known.

When renaming the group, the link may be lost altogether, and the access permission is no longer displayed. However, it is retained and taken into account during code generation.

Condition:

Create a user group with at list a quotation mark (") in the name.

Create a user account within this group. Create a PES resource access for this user group.

Generate code and load it.

Log in with the user account and try to connect to the controller.

The login data are automatically preset. To connect to the controller, it is possible to manually enter the user group name without all quotation marks (") and the password.

When renaming the user group, the PES access is no longer displayed, but is still available.

This means that it is still possible to log in to the controller with the old name of the user group.

Workaround:

Do not use names with quotation marks (") for user groups.

If an access including quotation marks (") has already been created, rename the user group by simply removing the quotation marks ("). In doing so, the accesses are deleted when the user group is removed.

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 module's detail view.
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.

- Effect:
For HIMatrix devices prior to F*03 (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, but they have no function. In particular, these parameters are *Force State*, *Forced Variables*, *Remaining Force Duration*, and *Force Time Reaction*.

Condition:

-

Workaround:

Refer to the online help for further details.

6 Upgrading from a Previous Version

Project data from previous versions can still be used in V10.64.

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 V10.64:

- 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 V10.64 and convert it.
- Since the conversion is extensive, check the project integrity after completing the conversion.
- Generate the code in V10.64 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

Release Notes

HI 801 502 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