

Programmierwerkzeug

# SILworX<sup>®</sup>

## Release-Notes V10.58

Alle in diesem Handbuch genannten HIMA Produkte sind mit dem Warenzeichen geschützt. Dies gilt ebenfalls, soweit nicht anders vermerkt, für weitere genannte Hersteller und deren Produkte.

HIQuad<sup>®</sup>, HIMax<sup>®</sup>, HIMatrix<sup>®</sup>, HIQuad X<sup>®</sup>, SILworX<sup>®</sup>, XMR<sup>®</sup>, HICore<sup>®</sup> und FlexSILon<sup>®</sup> sind eingetragene Warenzeichen der HIMA Paul Hildebrandt GmbH.

Alle technischen Angaben und Hinweise in diesem Handbuch wurden mit größter Sorgfalt erarbeitet und unter Einschaltung wirksamer Kontrollmaßnahmen zusammengestellt. Bei Fragen bitte direkt an HIMA wenden. Für Anregungen, z. B. welche Informationen noch in das Handbuch aufgenommen werden sollen, ist HIMA dankbar.

Technische Änderungen vorbehalten. Ferner behält sich HIMA vor, Aktualisierungen des schriftlichen Materials ohne vorherige Ankündigungen vorzunehmen.

Alle aktuellen Handbücher können über die E-Mail-Adresse [documentation@hima.com](mailto:documentation@hima.com) angefragt werden.

© Copyright 2019, HIMA Paul Hildebrandt GmbH

Alle Rechte vorbehalten.

## Kontakt

HIMA Paul Hildebrandt GmbH

Postfach 1261

68777 Brühl

Tel.: +49 6202 709-0

Fax: +49 6202 709-107

E-Mail: [info@hima.com](mailto:info@hima.com)

Revisions-index	Änderungen	Art der Änderung	
		technisch	redaktionell
10.00	Erstausgabe des neu erstellten SILworX V10 Release-Notes Dokuments	X	X
10.01	Kompatibilitäten aktualisiert	X	X

## Inhaltsverzeichnis

<b>1</b>	<b>SILworX V10.58</b>	<b>5</b>
<b>1.1</b>	<b>Kompatibilitäten</b>	<b>5</b>
1.1.1	Kompatibilität zum Betriebssystem des PES	5
1.1.2	Kompatibilität zu bestehenden Projekten	5
1.1.3	Kompatibilität zum verwendeten Rechner	5
1.1.3.1	Benutzung von Hardlocks	5
<b>2</b>	<b>Neue Funktionen</b>	<b>6</b>
2.1	HIQuad X	6
2.2	Smart Safety Test	6
2.3	Diagnoseübersicht	6
<b>3</b>	<b>Verbesserungen der Version</b>	<b>7</b>
3.1	Anwenderprogramm	7
3.2	Querverweise	7
3.3	Strukturbaum	7
3.4	Ressourceeigenschaften	7
3.5	FBS-Editor	7
3.6	Forcen	8
3.7	SILworX Datenbankoptimierung	8
3.8	Hardware	9
3.9	Protokolle	9
3.10	System Konfiguration	9
<b>4</b>	<b>Einschränkungen</b>	<b>11</b>
4.1	FBS-Editor	11
4.2	ST-Editor	13
4.3	Querverweise	15
4.4	Dateiauswahldialoge	16
4.5	Hardware	16
4.6	Anwenderprogramm	17
4.7	Versionsvergleicher	20
4.8	Codegenerierung	21
4.9	Forcen	21
4.10	Reload	22
4.11	Smart Safety Test	22
4.12	Protokolle	23
4.13	Projekt	23
4.14	Projekthistorie	25
4.15	Dokumentation	25
4.16	Benutzerverwaltung	26
4.17	Englisches SILworX	26
<b>5</b>	<b>Besonderheiten</b>	<b>27</b>

<b>6</b>	<b>Upgrade von einer Vorversion</b>	<b>30</b>
6.1	Referenzen	30

# 1 SILworX V10.58

Dieses Dokument beschreibt die Verbesserungen und neuen Funktionen von SILworX V10.58 gegenüber Vorversionen.

## 1.1 Kompatibilitäten

### 1.1.1 Kompatibilität zum Betriebssystem des PES

SILworX V10.58 ist für folgende HIMA Systemfamilien einsetzbar:

- HIQuad X

### 1.1.2 Kompatibilität zu bestehenden Projekten

Die Version V10.58 kann Projekte konvertieren und bearbeiten, die mit einer Vorversion erstellt wurden. Bei einer Codegenerierung des ungeänderten Projekts bleibt der Konfigurations-CRC erhalten.

### 1.1.3 Kompatibilität zum verwendeten Rechner

Die Mindestanforderungen an den für den Betrieb von SILworX verwendeten Rechner sind auf der jeweiligen HIMA DVD angegeben.

Speziell bei sehr großen Projekten können ältere Rechner möglicherweise lange Verarbeitungszeiten aufweisen und dadurch ungeeignet sein. Die Rechner-Hardware sollte daher möglichst dem Stand der Technik entsprechen. Bessere Hardware-Eigenschaften wie Rechenleistung und Speicherausbau führen zu verbesserter Performance.

#### 1.1.3.1 Benutzung von Hardlocks

Erfolgt die Lizenzierung von SILworX unter Windows 7, 8, 8.1 und 10 mit Hilfe von Hardlocks (USB-Sticks), ist folgendes zu beachten:

- Für die Installation sind Administrator-Rechte nötig.
- Für den Betrieb sind Benutzer-Rechte ausreichend.

## **2 Neue Funktionen**

### **2.1 HIQuad X**

- Unterstützung der HIQuad X (H41X und H51X) in SILworX.

### **2.2 Smart Safety Test**

- Mit dem Smart Safety Test kann der Anwender Testpläne für seine Logik erstellen und automatisiert ausführen. Der Smart Safety Test kann als Simulation in SILworX oder auf einer Steuerung erfolgen. Nach jedem Test kann bei Bedarf ein Testreport im PDF-Format erstellt werden.

### **2.3 Diagnoseübersicht**

- Die aktuelle Version enthält in der Hardware-Onlineansicht für HIQuad X-Systeme einen neuen Eintrag "Diagnoseübersicht". Sie bietet dem Anwender eine Tabelle mit Meldungen, die ihn zur Laufzeit über verschiedene Zustände seines Systems informieren. Diese Informationen, Warnungen oder Fehlermeldungen werden auf der Basis von vordefinierten Regeln erzeugt, welche die Zustandsinformationen des Systems oder von E/A-Modulen zu anwenderfreundlichen und praxisrelevanten Meldungen verarbeiten.

## 3 Verbesserungen der Version

- Dieses Kapitel beschreibt die SILworX-Verbesserungen von V10.58 gegenüber Versionen vor V10.58.

### 3.1 Anwenderprogramm

- Für C++-POE wird nach der Fehlerbeseitigung für X-CPU 31 der Stackbedarf korrekt verwendet.

Falls bei einer C++-POE ein unterschiedlicher Stackbedarf für X-CPU 01 und X-CPU 31 eingetragen ist, wird sich die Stackgröße des Anwenderprogramms und damit der CRC des Anwenderprogramms ändern. Der Versionsvergleich zeigt an, dass die Datengröße des Anwenderprogramms geändert ist. Um wieder das gleiche Anwenderprogramm wie mit der vorherigen SILworX-Version zu generieren, übernimmt der Anwender für die X-CPU 31 den Stackbedarf der X-CPU 01.

### 3.2 Querverweise

- Es werden ab V10 auch Querverweise angezeigt, die sich auf die Verwendung von benutzerdefinierten Funktionen in ST-Quelltexten (Funktionen und Funktionsbausteine) beziehen.
- Die Querverweisanzeige für Bausteine und Funktionen wurde verbessert. Sie enthält ab V10 für jede Verwendung einen eigenen Eintrag. Zusätzlich gibt die Spalte Info die grafische Position innerhalb der FBS-Logik an oder die Spalten-/Zeilen-Nummer bei Verwendung innerhalb von ST. Für jede einzelne Verwendung kann ein 'Gehe Zu' aufgerufen werden, dabei öffnet sich der Zieleditor (FBS oder ST) und selektiert die Verwendung.

### 3.3 Strukturbaum

- Die Dokumentation ist über das Kontextmenü selektierter Objekte im Projektbaum abrufbar, jedoch nicht mehr im Menü "Projekt". Die Optionen "Dokumentation" im Menü "Extras" und in der Aktionsleiste sind unverändert verfügbar.

### 3.4 Ressourceeigenschaften

- Die Bezeichnung für den Default bei Systembuslatenz heißt jetzt 'System-Standardwerte' und nicht mehr 'Linienstruktur'

### 3.5 FBS-Editor

- Im FBS-Editor lassen sich Logikelemente wie z. B. Variablen direkt neben einem Kommentarfeld oder einem zugeordneten Blattkommentarfeld platzieren. Bisher wurde ein Mindestabstand von 1 Grid zu einem Pin gefordert.
- Der FBS-Editor zeigt beim Bewegen von Elementen in großen Zoomstufen keine verwirrenden gelben Artefakte mehr an.

### 3.6 Forcen

- Der Force-Dialog reagiert robuster auf schnell nacheinander ausgeführtes Forcen von Variablen. Es treten dabei keine Abstürze mehr auf.

### 3.7 SILworX Datenbankoptimierung

- Die Performance von SILworX konnte in vielen Bereichen aufgrund des Umbaus der internen Struktur zum Teil massiv verbessert werden. Insbesondere beim Öffnen von Editoren, und beim Kopieren größerer Datenmengen ist die Beschleunigung spürbar. Auch das Öffnen und Schließen von Projekten geht nun schneller.
- Der Menüpunkt Reorganisation entfällt ersatzlos.  
Die Funktionalität wird beim Schließen des Projektes automatisch durchgeführt.
- Verweise auf andere Objekte werden in SILworX nun per Text aufgelöst. Es ergeben sich einige Verhaltensänderungen:
  - Es kann zu anderen Fehlermeldungen kommen.
  - Referenzen verbinden entfällt. Um einen unverbundenen Verweis zu verbinden reicht es aus, ein Objekt mit dem entsprechenden Zielnamen zu erstellen.
  - Bei Umbenennungen von Objekten werden alle Verweise, die auf diese Objekte verweisen, angepasst.
  - Wird der Name eines Objektes geändert, so werden dessen Querverweise erst dann wieder angezeigt, wenn der Editor, in dem die Namensänderung stattfand, gespeichert wird.
  - Gibt es mehrere Ziele im Sichtbarkeitsbereich, die den gleichen Zielnamen haben, so wird ein (für den Anwender) zufälliges, aber konstantes dieser Ziele ausgewählt.
  - Wird in Funktionsbausteinen eine Untervariable einer Globalen Variable (mit benutzerdefiniertem Datentyp) verwendet, so wird diese Verwendung außerhalb dieses Funktionsbausteins auch bei der Hauptvariable als Querverweis angezeigt.

Achtung: Existieren mehrere Objekte mit dem gleichen Namen, so führt ein Umbenennen eines Objektes dazu, dass die Verknüpfungen von diesem Objekt "mitgenommen" werden. Das Objekt, das seinen Namen behalten hat, verliert damit seine Verknüpfungen.

Lessons learned:

Es wird empfohlen keine Umbenennungen von Objekten durchzuführen solange der Objektname nicht eindeutig ist.

- Da *Referenzen* verbinden entfällt, werden auch beim Wiederherstellen, Verschieben oder Kopieren im Strukturbaum keine Meldungen mehr generiert, dass Referenzen nicht verbunden werden können.
- Brach man die Erstellung der Dokumentation ab, so konnte das - vorwiegend bei großen Projekten - zu einer Terminierung von SILworX führen. Die Ursache konnte identifiziert und beseitigt werden.
- In V9 konnten Projektarchive älterer Versionen, die eine Benutzerverwaltung enthielten, nicht konvertiert werden. Solche Projektarchive mussten in der alten Version wiederhergestellt und konnten dann als Projekt geöffnet werden. Dieser Zwischenschritt ist nicht mehr notwendig.
- Die interne Datenablage in SILworX wurde umgestellt. Dadurch können Daten innerhalb eines Editors nun ihre Namen tauschen.



### 3.8 Hardware

- Die Bezeichnungen für die Feldbus-Schnittstellen der COM-Baugruppen entsprechen den Aufdrucken auf den Gehäusen: FB1, FB2, FB3.
- Die Bezeichnungen für die Feldbus-Schnittstellen der COM-Baugruppen beinhalten nicht mehr die SRS der Baugruppen.
- Wenn der Anwender einen Destport konfiguriert aber keine Source, bekommt er jetzt bei Validierung und Codegenerierung einen Warnhinweis, dass ohne Definition einer Source kein Mirroring stattfindet.
- Der Versionsvergleich zeigt bei der HI Max X-AI 16 51 die Änderungen der Prozesswerte 4mA und 20mA und nicht wie bisher Skal-Faktor + Offset.
- Im Versionsvergleich für ke.config werden HIMatrix-IO-Module jetzt mit Modulname + SRS bezeichnet anstatt wie bisher mit 'IO' + SRS.
- Bei der Codegenerierung von OPC-Server-Sets wird die Bedingung "Sollzykluszeit < Watchdog-Zeit - 1000 ms" jetzt geprüft.

### 3.9 Protokolle

- Die Datenmengenprüfung für PROFIBUS DP ist wieder aktiv. Sie war in V9 vorübergehend deaktiviert worden.
- Verweise auf Server-Partner in der safe**ethernet**-Verbindungsübersicht von Ressourcen bleiben auch nach dem Löschen von Servern bestehen. Anwender erhalten Verifikationsmeldungen über ungültige Verweise.  
Abhilfe: Wenn die Ressource wiedergestellt wird, ist nichts weiter zu tun. Ansonsten muss der Verweis auf die Ressource in den safe**ethernet**-Einstellung auf "keine" gesetzt werden.
- Die System ID wird bei nicht auflösbaren IP-Links durch ein Fragezeichen ersetzt, z.B. "?0.5 (192.160.0.1:6010)".
- Wenn eine FB-Schnittstelle öfter als erlaubt benutzt wird, werden in der Fehlermeldung die betreffenden Protokolle explizit aufgeführt, die die FB-Schnittstelle benutzen, aufgezählt. Im Falle des Modbus-Slaves erscheint dort jetzt das Modbus-Slave-Set anstelle des Modbus-Slaves.

### 3.10 System Konfiguration

- Die Beschriftung ist für die Auswahl der Reloadbasis im Codegendialog verbessert. Der Titel zeigt den entsprechenden Kontext an um die Ressource zuordnen zu können.
- Die Fehlermeldung die bei einer Reload-Codegenerierung wegen einer Datentypänderung erscheint, bietet ein "Gehe zu" in den entsprechenden Editor an.
- Der Anwender wird bei der Reload-Codegen über die Anzahl der benötigten und der maximal erlaubten Transferoperationen für einen Reload informiert. Durch diese Information kann der Anwender besser einschätzen, wie nahe er der Grenze der möglichen Transferoperationen gekommen ist um zukünftige Aktionen besser zu planen.
- Wenn ein Download/Reload stattfand, während der Force-Editor oder ein anderes Online-Tool mit Force-Funktionen geöffnet war, gab das PADT in früheren Versionen meist folgende Fehlermeldung aus:

"Die globale Force-Daten-Abfrage wurde abgebrochen: <x>" ("The global force data query was aborted: <x>")

Die PADT-Fehlermeldung wurde entfernt.

Anmerkung: Während des Download/Reload werden die angezeigten Force-Daten der Variablen nicht aktualisiert. Dies ist an abgegrauten Werten im Force-Editor usw., leeren OLT-Feldern und schwarzen Powerflow-Linien zu erkennen.

## 4 Einschränkungen

Beim Einsatz von SILworX sind die folgenden Einschränkungen zu beachten. Bei Beachtung der folgenden Hinweise haben die Einschränkungen keine Auswirkungen auf die Sicherheit und die Verfügbarkeit des Codes, der für eine Steuerung generiert wird.

### 4.1 FBS-Editor

- **Auswirkung:**

Leere Seiten im Logikbereich des FBS-Editors sind nicht immer löschtbar

Bedingung:

Der Menüpunkt Leere Seite löschen im Kontextmenü ist nicht aktiv, wenn folgende Bedingungen zusammentreffen:

- Eine Linie verläuft über zwei oder mehrere Nachbarseiten der leeren Seite
- Die Linie verläuft nicht durch die leere Seite

Die leere Seite kann somit nicht gelöscht werden.

Abhilfe:

Es existiert keine Abhilfe für das Problem

- **Auswirkung:**

Auf dem FBS-Editor wird der Text bestehender Seitenkommentare nach dem Einfügen neuer Seiten nicht angezeigt.

Bedingung:

Folgende Bedingungen müssen dabei zusammentreffen:

- Ein Seitenkommentar befindet sich neben einer zweiten Seite mit mindestens einem Logikelement.
- Auf der zweiten Seite wird die Aktion "Leere Blätter einfügen" -> "Spalte oder Zeile einfügen" so ausgeführt, dass die neuen Seiten zwischen der Seite mit dem Blattkommentar und der Seite mit dem Element eingefügt werden.

Der Text des bestehenden Seitenkommentars wird temporär nicht mehr angezeigt.

Abhilfe:

Der Text erscheint wieder, wenn der Kommentar bewegt oder der Editor geschlossen und neu geöffnet wird.

- **Auswirkung:**

Auf dem FBS-Editor werden Seiteninformationen nach einem Projektimport aus ELOP II nicht korrekt platziert.

Bedingung:

Folgende Bedingungen müssen dabei zusammentreffen:

Ein zugeordnetes Kommentar- oder OLT-Feld liegt auf einer leeren Seite ohne weitere Logikelemente, während sich das Hauptelement auf einer anderen Seite befindet.

Abhilfe:

Zugeordnete Kommentar- oder OLT-Felder sollten sich auf der gleichen Seite wie ihre Hauptelemente befinden.

- **Auswirkung:**

Konflikt-Icon bei Variablen bleibt auch nach Behebung des Konflikts sichtbar.

Bedingung:

In folgenden Fällen bleibt das Konflikt-Icon auch nach Ablehnung einer ungültigen Aktion und Anzeige des korrekten Werts sichtbar:

- Eingabe eines ungültigen Namens an einer Variablen
- Vergabe einer bereits existierenden Reihenfolge-Nummer an einer Interface-Variablen

Abhilfe:  
Verifizieren oder Aktualisieren.

- **Auswirkung:**  
Anzeige von globalen Variablen, die als VAR\_EXTERNAL verwendet werden:  
Werden globale Variable mit Datentyp Struct oder Array als VAR\_EXTERNAL verwendet, zeigt der FBS-Editor für die Unterelemente die Einträge in den Spalten Initialwert, Beschreibung, Zusatzkommentar und technische Einheit nicht an

**Bedingung:**  
Global Variable vom Typ Struct erstellen und in einem Struct-Element eine Beschreibung eintragen. Nun diese GV in einem Programm als Wertfeld verwenden. Anschließend die erzeugte VAR\_EXTERNAL in der Lokale Variablen-Lasche anschauen.

**Abhilfe:**  
Attributeigenschaften der VAR\_EXTERNAL in der entsprechenden Globalen Variable anschauen.

- **Auswirkung:**  
Durch das Löschen einer verwendeten POE und Neuanlegen/Einfügen eines Typicals mit dem Namen der gelöschten POE kommt es in der Codegenerierung bei diesem nicht unterstützten Typ zu einen Absturz.

**Bedingung:**  
Einen FB POE "B" anlegen und diesen in einem Programm verwenden. Alles speichern und schließen. Die POE "B" löschen und dort ein Typical Baustein "B" anlegen. An der Verwendung von "B" die Aktualisierung aufrufen. Es werden die Fehler bereinigt und das Typical wird aufgerufen.

**Abhilfe:**  
Die Änderung kann rückgängig gemacht werden, in dem der richtige Typ von Baustein verwendet wird. Der richtige Typ der Instanz kann durch einen Doppelklick auf die Instanz kontrolliert werden. Die dann aufgerufene POE darf nicht vom Typ Typical sein, zu erkennen am kleinen Icon neben dem Namen.

- **Auswirkung:**  
Absturz beim Speichern eines FBS-Editors, wenn ein Logikteil mit Verbindungen während eines Kopiervorgangs beim Platzieren abgelehnt wird und die Kopier-Aktion abgebrochen wird. Dies bemerkt man nicht sofort nach dem fehlgeschlagenen Einfügen der Kopie sondern erst beim Speichern des Editors. Achtung Datenverlust.

**Bedingung:**  
Logik mit Verbindungen kopieren (egal wie) dann so Einfügen, das der Kopiervorgang abgelehnt wird. Beim nächsten Speichern des Editors kann es mit hoher Wahrscheinlichkeit zum Absturz kommen.

**Abhilfe:**  
Kopierte Logik so einfügen, dass das Einfügen nicht abgelehnt wird.

**Lessons learned:**  
Während des Arbeitens im FBS-Editor regelmäßig speichern um einen möglichen Datenverlust zu minimieren

- **Auswirkung:**  
Das Löschen eines OLT-Feldes im Bausteineditor, welches an einem Anschlusspunkt einer Variable oder Bausteininstanz verbunden ist, führt zu einer fehlerhaften Meldung bei der Codegenerierung, wenn am Anschlusspunkt zusätzlich eine Verbindung existiert.

- Bedingung:
  1. Zwei Variablen im Bausteineditor verbinden.
  2. Ein zugeordnetes OLT-Feld am ausgehenden Anschlusspunkt der Variablen erzeugen.
  3. Das zugeordnete OLT-Feld wieder löschen
  4. Speichern des Editors.
  5. Codegenerierung starten.

Abhilfe:

Löschen der Verbindungslinie, welche am Anschlusspunkt der Variable des gelöschten OLT-Felds abgeht, und anschließendes erneutes Verbinden.  
GeheZu der Fehlermeldung führt zum betroffenen Objekt.

- Auswirkung:

Durch das Umbenennen von Bausteinen entstehen in der Logik Konflikte für alle Ein-/ und Ausgänge, die den gleichen Namen haben wie der Baustein vor der Umbenennung. Beim Auflösen dieser Konflikte werden Linien zu diesen Ein-/ und Ausgängen gelöscht.

Bedingung:

Baustein anlegen und öffnen. Anschließend im Editor eine Ausgangsvariable mit dem Namen "OUT" anlegen.

Einen weiteren Baustein mit dem Namen "OUT" anlegen.

Den zuerst angelegten Baustein im Programm verwenden.

Im nächsten Schritt den als Zweites angelegten Baustein von "OUT" zu "Baustein\_2" umbenennen.

Programm erneut öffnen oder Verifikation starten. Es wird gemeldet, dass der Verweis des Instanzenausgangs auf "Baustein\_2" nicht aufgelöst werden kann.

Abhilfe:

Bausteininterface, welches gleich heißt wie benutzerdefinierte Bausteine umbenennen, Konflikte in der Logik aktualisieren.

## 4.2 ST-Editor

- Auswirkung:

Im ST-Editor sind 2700 aufeinanderfolgende Kommentarzeilen nicht möglich.

Bedingung:

Beim Auskommentieren mit 2700 aufeinanderfolgende Zeilen im Structured-Text-Editor terminiert SILworX.

Abhilfe:

Längere Kommentare unterteilen, z. B. jeweils 1000 Zeilen zu einen Kommentar zusammenfassen.

- **Auswirkung:**  
Es kommt zu einem Absturz, wenn man innerhalb eines ST-Bausteineditors einen benutzerdefinierten Datentyp von einer Variablen zu einer anderen kopiert und sich die Auswahlbox des Datentyps der Zielvariablen im Editiermodus befindet.

**Bedingung:**

Eine ST-Funktion oder einen ST-Funktionsbaustein öffnen und 2 Variablen anlegen (Variablentyp spielt keine Rolle). Der einen Variablen einen benutzerdefinierten Datentyp zuweisen, dann diesen Datentyp in der Auswahlbox selektieren und im Kontextmenü "Kopieren" auswählen. Anschließend auf den Datentyp der zweiten Variablen ein "Enter" ausführen, sodass die Auswahlbox in den Editiermodus wechselt. In diesem Zustand muss der zuvor kopierte Datentyp eingefügt werden.

**Abhilfe:**

Einen selektierten benutzerdefinierten Datentyp einer Variablen innerhalb eines ST-Bausteins zu einer anderen Variablen kopieren, ohne die Datentyp-Auswahlbox der Zielvariablen in den Editiermodus zu versetzen, d.h. ohne die Enter-Taste zu betätigen oder ein Doppelklick auszuführen.

### 4.3 Querverweise

- **Auswirkung:**  
Es wird für Variablen-Querverweise keine StrukturInfo angezeigt, wenn diese in HIPRO-S benutzt werden, das Feld bleibt leer.

**Bedingung:**

Variable in HIPRO-S-Verbindung benutzen und im GV-Editor die Querverweise anschauen: StrukturInfo bleibt leer.

**Abhilfe:**

keine Abhilfe

- **Auswirkung:**  
**Anzeigefehler bei den Querverweisen:**  
Die Inhalte der Spalten Info, Strukturinfo und Verwendung eines Querverweises können veraltet sein, wenn sie Informationen enthalten, die außerhalb des zugehörigen Editors verändert werden. Dies betrifft z. B. die System-ID bei Hardware-Querverweisen oder den Namen eines Strukturbaumelements.

Die Spalte "Strukturpfad" der Querverweisanzeige von Globale Variablen zeigt den korrekten Namen und Pfad an. Auch der GeheZu-Sprung ist von dem Anzeige-Problem nicht betroffen.

**Bedingung:**

Am Beispiel einer Verwendung einer Globalen Variablen in einem Funktionsbaustein:  
Einen neuen FBS-Baustein anlegen.

Eine Globale Variable anlegen und im FBS-Baustein verwenden.

Danach den FBS-Baustein umbenennen oder kopieren und einfügen.

Im Globale Variablen-Editor die Querverweise der Globalen Variable prüfen. Die Spalte "Info" zeigt den alten Namen des FBs an, wohingegen der "Strukturpfad" in der letzten Spalte korrekt ist

Am Beispiel einer Verwendung einer Globalen Variablen in der Hardware:

Eine Globale Variable einer Systemvariable innerhalb eines Hardware-Racks zuweisen und speichern.

Danach die System-ID der Ressource ändern und den Querverweis der Globalen Variable im Globalen Variablen-Editor anschauen.

**Abhilfe:**

Den Editor, in dem der Verweis angelegt wurde, nach dem Umbenennen eines Strukturbaumelementes öffnen, eine beliebige Änderung machen (beispielsweise einen Baustein minimal verschieben) und speichern. Bei dieser Aktion wird die Querverweisinformation neu erstellt und zeigt anschließend den korrekten Inhalte in der Spalte "Info". Gleiches gilt auch für die Verwendung innerhalb des Hardware-Editors.

- **Auswirkung:**  
Querverweise werden in der Sprachversion angezeigt, in der der Editor, in dem der Verweis stattfindet, gespeichert wurde, nicht in der Sprache, die aktuell in SILworX eingestellt ist.

**Bedingung:**

Querverweise erzeugen, die sprachabhängige Teile enthalten. Dann das Projekt mit einer SILworX-Instanz mit einer anderen Sprachversion öffnen.

Die Querverweise werden nach wie vor die Texte in der vorherigen Sprachversion enthalten.

Beispiel: Die Namen der Systemvariablen im Hardware-Editor werden im Querverweis der Globalen Variablen angezeigt, bleiben aber, wenn in einem deutschen SILworX verknüpft, auch in einem englischen SILworX in der deutschen Version.

**Abhilfe:**

a) Ignorieren

b) Den Editor, an dem die Verknüpfung existiert (z.B. durch Gehe zu auf dem Querverweis erreichbar) speichern (dazu muss zuvor eine (beliebig kleine) Änderung in diesem Editor vorgenommen worden sein). Dabei werden die Querverweisinformationen neu (und damit nun in der aktuellen Sprache) erstellt.

## 4.4 Dateiauswahldialoge

- **Auswirkung:**  
Der Dialog des Hardware-XML-Imports bleibt nach fehlerhafter Dateiauswahl offen und reagiert nur noch auf den Abbrechen-Button

**Bedingung:**

Im Hardware-XML-Import-Dialog einen falschen oder ungültigen Dateipfad auswählen und den Import starten.

**Abhilfe:**

Dialog mit 'Abbrechen' beenden und Import erneut starten.

- **Auswirkung:**  
Im Wiederherstellungsdialog wird nur der Pfad der zuletzt wiederhergestellten Archivdateien angezeigt. Es fehlt die Liste der zuletzt angelegten Archive.

**Bedingung:**

Ein Projekt öffnen oder anlegen. Davon ein Archiv oder mehrere davon erstellen. Im Wiederherstellungsdialog die Auswahlbox unter Archivdatei öffnen: Diese ist leer, es werden keine anderen Archivdateien gelistet.

**Abhilfe:**

Archivdatei manuell suchen.

## 4.5 Hardware

- **Auswirkung:**  
Im Hardware-Editor ist in der Detailansicht-Toggle-Button nicht aktiv. Die Ansicht kann über den Schließen-Button geschlossen werden.

**Bedingung:**

Hardware-Onlineansicht starten und die Detailansicht einer Baugruppe öffnen: der Detailview-Toggle-Button ist nicht aktiv.

**Abhilfe:**

Nicht notwendig, da ein Schließen-Button in der Detailansicht existiert.



- **Auswirkung:**  
Ab SILworX V6 sortiert SILworX bei der Codegenerierung die Lizenzen nach Namen, nicht mehr nach der Reihenfolge des Eintragens. Dadurch kann es bei der Projektkonvertierung aus früheren Versionen zu CRC-Änderungen kommen.

**Bedingung:**  
Lizenznamen in nicht alphabetischer Reihenfolge in V5 eintragen und Code generieren.  
Dann in V6 Code generieren.

**Abhilfe:**  
Geeignete Namensvergabe, HIMA-Support hinzuziehen.

## 4.6 Anwenderprogramm

- **Auswirkung:**  
Folgendes Verhalten der Funktion EXPT weicht auf den PES-Systemen von der IEEE-754-Norm ab.  
 $1.0 ** NaN := 1.0$  erwartet: NaN  
EXPT.ENO := TRUE erwartet: FALSE

$NaN ** 0.0 := 1.0$  erwartet: NaN  
EXPT.ENO := TRUE erwartet: FALSE

Mit OTS und Offline-Simulation funktioniert EXPT gemäß IEEE-754.

**Bedingung:**  
s.o.

**Abhilfe:**  
Bei ENO-Bedarf den Wert NaN an beiden Eingängen programmtechnisch abfangen / vermeiden.

- **Auswirkung:**  
Der Funktionsbaustein MUL liefert falsche Ergebnisse, wenn folgende Bedingungen zusammentreffen:
  - HIMatrix Standard-Ressource (z.B. F30 01, F35 01, F60 01)
  - Datentyp LREALAm Eingang IN1 liegt der Wert  $\pm\infty$ , am Eingang IN2 liegt nan (not a number) an
  - In diesem Fall ist das Ergebnis  $-\infty$ , und nicht nan wie spezifiziert.
  - Der ENO liefert in diesem Fall korrekt das Ergebnis False.

**Bedingung:**  
s. Auswirkung

**Abhilfe:**  
Ergebnis in diesem speziellen Fall ignorieren

- **Auswirkung:**  
Die Funktion DIV\_TIME aus der Standardbibliothek setzt fälschlich den Fehlerausgang ENO auf FALSE und meldet somit einen Fehler unter folgenden Bedingungen:
  - Der Eingang IN2 (Divisor) ist vom Typ REAL.
  - Der Wert von IN2 ist +/-INF.

**Bedingung:**

DIV\_TIME mit EN/ENO verwenden und als Divisor +/- INF anlegen. (INF ist z.B. das Ergebnis aus 1.0 / 0.0)

**Abhilfe:**

ENO in diesem Fall ignorieren

**Auswirkung:**

Die Funktion EXPT liefert bei Offline-Simulation und OTS für die Basis IN1 = 1.0 und Exponent IN2 = -INF den Wert NaN statt 1.0.

**Bedingung:**

EXPT mit IN1 = 1.0 und IN2 = -INF (oder andere große negative Zahl aufrufen) und Ergebnis in der Offline-Simulation oder OTS ansehen.

**Abhilfe:**

Wenn der Sonderfall für die Applikation relevant ist, muss dies in der Anwenderlogik entsprechend ausprogrammiert werden.

- **Auswirkung:**  
Bei der Abarbeitung einer POE werden zuerst die Schrittketten bearbeitet, danach die AS-Aktionen und dann die FBS-Logik. Das führt dazu, dass die Eingangswerte von AS-Transitionen und AS-Aktionen, die in der FBS-Logik beschrieben werden, immer aus dem vorherigen Zyklus stammen. Bei der konkreten Auswertung der Eingangswerte bestehen allerdings kleine Unterschiede:  
Der Eingangswert einer AS-Transition wird während der FBS-Bearbeitung in den Speicher der AS-Transition geschrieben und dort gemerkt und erst im nächsten Zyklus in der Schrittkette bearbeitet. Dies führt dazu, dass nach einem Kaltstart Schrittketten generell erst ab dem zweiten Zyklus weitergeschaltet werden.  
Der Eingangswert einer AS-Aktion wird bei der Abarbeitung der AS-Aktion aus der Quelle gelesen. Falls dies eine Funktion ist, wird der Initialwert gelesen, da Funktionen zu Beginn der POE-Ausführung initialisiert werden und erst nach den AS-Aktionen ausgeführt werden.

**Bedingung:**

s. oben

**Abhilfe:**

**AS-Transition:**

Bei der Programmierung der Schrittkette muss berücksichtigt werden, dass frühestens im zweiten Zyklus eine AS-Transition durchgeführt wird.

**AS-Aktion:**

Um ein Funktionsergebnis als Eingangswert einer AS-Aktion zu verwenden, muss zwischen Ausgang der Funktion und Eingang der AS-Aktion eine Variable geschaltet werden.

- **Auswirkung:**  
Wenn die Ketten einer Auswahlverzweigung mit AS-Schritten enden, führt dies zu einem Deadlock. Wenn die Ketten einer Simultanverzweigung mit Transitionen enden, führt dies zu mehreren aktiven Schritten außerhalb der Simultanverzweigung.

**Bedingung:**  
s. oben

**Abhilfe:**  
Der Anwender muss durch geeignete Maßnahmen dafür sorgen, dass derart fehlerhafte Schrittketten nicht verwendet werden.

- **Auswirkung:**  
Beim Zugriff auf ein Array-Element mit einem Index außerhalb des Wertebereiches des Arrays wird nach einem definierten und performanten Verfahren auf ein Element innerhalb des Arrays zugegriffen, um willkürliche Zugriffe auf Speicherbereiche zu vermeiden.

**Bedingung:**  
s. oben

**Abhilfe:**  
Der Anwender muss durch geeignete Programmierung dafür sorgen, dass nur mit Indizes innerhalb des Wertebereichs des Arrays auf Array-Elemente zugegriffen wird.

- **Auswirkung:**  
Verschiedene Elemente einer Struktur-Variablen nicht von unterschiedlichen Quellen beschreibbar  
Es ist nicht möglich, dass sowohl das Anwenderprogramm als auch die Hardware oder die Kommunikation zwei unterschiedliche Elemente derselben Struktur-Variablen beschreiben.

**Bedingung:**  
s. oben

**Abhilfe:**  
Unterschiedliche Struktur-Variablen für Elemente, in die das Anwenderprogramm schreibt, und für Elemente, in die Hardware oder Kommunikation schreiben.

- **Auswirkung:**  
Es ist nicht möglich, Elemente von Variablen eines benutzerdefinierten Datentyps als Index eines Arrays zu benutzen.

**Bedingung:**  
s. oben

**Abhilfe:**  
Wert der gewünschten Variablen in einfache Variable kopieren und diese als Index verwenden.

## 4.7 Versionsvergleich

- **Auswirkung:**  
Die Detail-Ansicht einer POE im Versionsvergleich zeigt unberechtigt eine Änderung für eine POE-Instanz an, wenn
  1. die Vergleichsbasis mit einer Version vor V4.116 erzeugt wurde und
  2. der aufgerufene POE-Typ Umlaute im Namen enthält.

**Bedingung:**  
s. oben

**Abhilfe:**  
In Bausteinennamen nur Leerzeichen und Zeichen aus der folgenden Liste verwenden

- 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
- \$ % & ( ) \* + - / : ; < = > ? \ ^ \_ ` { | }

- **Auswirkung:**  
Der Vergleich zwischen einer mit SILworX vor V9 generierten und einer mit SILworX ab V9 generierten Konfiguration zeigt die Meldung "Die Reihenfolge der Variablen- und Instanzendeklaration hat sich geändert." nicht mehr an
  - 1) Für alle Funktionen und Funktionsbausteine
  - 2) Für Programme die (in der ab V9 generierten Konfiguration) mindestens eine der folgenden Standardfunktionen verwenden: ADD, SUB, MUL, DIV, MOD, MOVE, AND, OR, XOR, NOT, SHL, SHR, alle Ato...-Bausteine, ADD\_TIME, SUB\_TIME, MAX, MIN, SEL, MUX, GT, LT, GE, LE, EQ, NE, PACK.

**Bedingung:**  
In SILworX vor V9 eine Ressource mit zwei Konfigurationen (1. geladen oder importiert, 2. generiert) erstellen, deren Versionsvergleich (zu Recht) eine Änderung der Deklarationsreihenfolge in einer POE meldet.  
Der Projektstand muss dem der generierten Konfiguration entsprechen.  
Das Projekt nach SILworX V9 konvertieren und Code generieren.  
Im Versionsvergleich fehlt die Reihenfolge-Meldung, obwohl die betreffende POE den gleichen CRC-Unterschied aufweist wie zuvor.

**Abhilfe:**  
Wenn der zur Vergleichsbasis (z. B. zur geladenen Konfiguration) passende Projektstand noch verfügbar ist
 

1. Eine Kopie dieses alten Projektstandes nach SILworX V9 bzw. in die gewünschte, neue Version konvertieren. Dort Code generieren und das Ergebnis über den Startdialog des Versionsvergleichs exportieren.
2. Das andere Projekt, in dem der Versionsvergleich stattfinden soll, in der neuen SILworX-Version öffnen, die eben exportierte Konfiguration im Startdialog des Versionsvergleichs importieren und als Vergleichsbasis verwenden.

## 4.8 Codegenerierung

- **Auswirkung:**  
Konflikt durch Änderung des Konstant-Attributs globaler Variablen nach Verwendung:  
Wird eine globale Variable als VAR\_EXTERNAL verwendet und danach von konstant auf änderbar gesetzt oder umgekehrt, führt dies bei der Codegenerierung zum Konflikt, wenn diese VAR\_EXTERNAL beschrieben wird und die globale Variable konstant ist.

**Bedingung:**

GV als VAR\_EXTERNAL in der Logik verwenden und den Konst-Zustand in der GV ändern.

**Abhilfe:**

Globale Variable an allen Verwendungsstellen entfernen, so dass VAR\_EXTERNAL verschwindet. Anschließend an allen Stellen neu einfügen

## 4.9 Forcen

- **Auswirkung:**  
Die Anzeige des Force-Status im Bearbeiten-Dialog von Force-Daten ist falsch, wenn mehr als ein Programm vorhanden ist und die Aktion nicht auf das Programm abzielt, das im ersten Tab (von links) im Force-Editor angezeigt wird. D.h., es wird im Force-Daten-Bearbeiten-Dialog für Programme, die im Force-Editor in den Tabs 2 - 32 angeordnet sind, - einheitlich falsch - der Force-Status des Programms des 1. Tabs von links präsentiert. Das ist besonders ungünstig, wenn das Force-Daten-Bearbeiten nicht im Force-Editor stattfindet, sondern im FBS-Editor-Online, da dort der Force-Status nur in diesem Force-Daten-Bearbeiten-Dialog angezeigt wird und nicht im Hintergrund oberhalb der Variablentabelle zusätzlich der korrekt dargestellte Force-Status ersichtlich ist.

**Bedingung:**

Es existieren mindestens zwei Programm-Tabs im Force-Editor.

Der Force-Status im Bearbeiten-Dialog von Force-Daten wird nur für das erste Tab von links für lokales Forcen korrekt angezeigt, egal ob der Bearbeiten-Dialog im Force-Editor geöffnet wurde oder im FBS-Editor-Online. Beim Bearbeiten von Force-Daten für alle weiteren Programm-Tabs wird fälschlicherweise und für den Anwender irreführend immer der Force-Status des Programms aus dem ersten Tab im Bearbeiten-Dialog angezeigt.

**Abhilfe:**

Lokale Variablen im Force-Editor / in der Watchpage bearbeiten, da dort der korrekte Force-Status des jeweiligen Programms noch über die Force-Tabelle im Hintergrund sichtbar ist, was im FBS-Editor-Online nicht der Fall ist.

- **Auswirkung:**  
Es können keine Globalen/Lokalen Variablen mit einem speziellen Ziffernsuffix im Namen nach folgendem Muster in der Watchpage neu verwendet werden: "\_0#" wobei die # eine oder mehrere Ziffern darstellt, z.B.: "Programm.Poe1.Var\_01", "Globale Variable\_02". D.h., beim Drag'n'drop derartiger Variablen in die Watchpagetabelle, erhalten die neuen Elemente geänderte Namen. Z.B. wird aus "Globale Variable\_02" --> "Globale Variable\_2".  
Der Fehler hat sowohl eine sichere als auch eine unsichere - weil irreführende - Auswirkung.  
**Sichere Auswirkung:**  
Es kann mit dem fälschlicherweise geänderten Namen keine Force-Variable adressiert werden.  
**Unsichere Auswirkung:**  
Es kann mit dem fälschlicherweise geänderten Namen eine unbeabsichtigte Force-Variable adressiert werden.

**Bedingung:**

Es existieren mit Verwendung innerhalb Ressourcenkonfiguration / Anwendungslogik Globale / Lokale Variablen mit folgendem Suffix (Ziffernsuffix) im Bezeichner: "\_0#", wobei die # eine oder mehrere Ziffern darstellt und die "0" auch mehrere Nullen umfassen kann.  
Neu angelegte Watchpage für Globales/Lokales Forcen Online/Offline öffnen.

Per D&D Variablen mit Ziffernsuffix in die Watchpage-Tabelle ziehen und fallen lassen. Die daraufhin erzeugten Watchpage-Variablen haben abweichende Namen und adressieren somit abweichende Force-Variablen oder nicht existierende (Zustand: Unverbunden).

Abhilfe:

Der Anwender kann den Namen der neu erzeugten Watchpage-Variable in der Watchpage-Tabelle korrigieren.

#### 4.10 Reload

- Auswirkung:

In einer Reload-Codegenerierung kommt es zu einer Reload-Warnung, wenn die Download-Info des Partners nicht zu finden ist:

"Safeethernet-Reload-Sequenz gestartet. Es wurde eine Dualkonfiguration angelegt. Es ist keine Download-Konfiguration für [Partner-Name] zum Abgleich der safeethernet-Signaturen vorhanden."

Das geschieht auch dann, wenn die Reload-Codegenerierung ohne vorherige Änderungen gestartet wurde.

Bedingung:

Safeethernet-Verbindung mit Proxy erstellen und laden. Reload-Codegenerierung auf der realen Steuerung ohne irgendwelche Änderungen.

Abhilfe:

Nicht vorhanden.

#### 4.11 Smart Safety Test

- Auswirkung:

Unter den genannten Bedingungen kommt es bei der Berechnung der Untergrenze des Toleranzbereichs (Sollwert - Toleranzwert) als ULINT zum Unterlauf. Die tatsächlich wirksame Untergrenze lautet dann  $((2 \text{ hoch } 64) + (\text{Sollwert} - \text{Toleranzwert}))$ .

In dem in der Praxis sehr wahrscheinlichen Fall, dass diese wirksame Untergrenze größer ist als die Obergrenze des Toleranzbereichs (die das PADT korrekt in Sättigungsarithmetik als ULINT berechnet) liefert der Vergleich unabhängig vom Istwert immer den Status NOT\_OK.

Bedingung:

Im Testplan in einem CHECK\_VALUE-Testschritt für eine globale Variable mit Datentyp USINT, UINT, UDINT oder ULINT einen Sollwert mit Toleranzwert eingeben, wobei Toleranzwert > Sollwert, so dass  $(\text{Sollwert} - \text{Toleranzwert}) < 0$ .

Beispiel: Datentyp USINT, Sollwert-Eigenschaft "12 [14]":  $(12 - 14) = -2$  ist kleiner als 0.

Abhilfe:

Sollwert und Toleranzwert wie folgt verändern:

Neuer Sollwert =  $(\text{alter Sollwert} + \text{alter Toleranzwert}) / 2$ , nötigenfalls auf- oder abgerundet, wie es besser für den Testfall passt.

Neuer Toleranzwert = neuer Sollwert.

Die neue Untergrenze ist dann 0, die neue Obergrenze liegt im Bereich alte Obergrenze  $\pm 1$ .

## 4.12 Protokolle

- **Auswirkung:**  
In einem Projekt vor V6 eine safeethernet-Verbindung mit zwei Ressourcen (Res1, ID=1 und Res2, ID=2) erzeugen und in einer aktuellen Version die Verbindung auf "Ab V6" einstellen. Das kann zur Folge haben, dass Res2 zum Timing Master ernannt wird.

**Bedingung:**

Aus Anwendersicht ist keine Vorsage möglich, wer beim Übergang zu "Ab V6" zum Timing Master ernannt wird.

**Abhilfe:**

Timing Master explizit festlegen.

- **Auswirkung:**  
Beim Versuch, eine Array-Variable mit mehr als 32 768 Elementen in ein Kommunikationsprotokoll zu übernehmen, terminiert SILworX.

**Bedingung:**

Globale Variable mit 32800 Unterelementen anlegen und diese in einem Protokoll als Prozessvariable anlegen.

**Abhilfe:**

Große Array-Variable in mehrere kleinere aufteilen.

- **Auswirkung:**  
Bei Überschreitung der zulässigen Maximaldatenmengen von PROFIBUS DP Slave wird bei der Verifikation ein Fehler gemeldet.

**Bedingung:**

Die Prozessdaten für einen PROFIBUS DP Slave müssen die erlaubte Maximaldatenmengen überschreiten:

Eingangsdatengröße > 192 Bytes (bei HIQuad H41X bzw. 51X > 244 Bytes)

Ausgangsdatengröße > 240 Bytes (bei HIQuad H41X bzw. 51X > 244 Bytes)

Verifikation bzw. Codegenerierung durchführen um die Meldung zu erzeugen.

**Abhilfe:**

Grenzen einhalten

## 4.13 Projekt

- **Auswirkung:**  
Projekte können bei der Windowsinternen Synchronisierung mit Netzlaufwerken auf dem Netzlaufwerk verloren gehen. Sie sind dann unter Umständen nur noch lokal verfügbar.

**Bedingung:**

Ein Projekt in einem synchronisierten Ordner öffnen, bearbeiten und schließen.

Anschließend ist das Projekt vom Netzlaufwerk gelöscht.

**Abhilfe:**

Projekte nicht öffnen, wenn sie synchronisiert werden, und eine Netzwerkverbindung besteht. Durch das Öffnen des Projektes wird die Netzversion von Windows gelöscht. Sollte es dennoch dazu gekommen sein, kann die lokale Kopie wieder auf das Netzlaufwerk kopiert werden.

- **Auswirkung:**  
Werden in Kombination ein geschütztes Trennzeichen und Leerzeichen im Programmname verwendet, kann SILworX terminieren.

**Bedingung:**

Im Strukturbaum unter der Ressource ein Programm "P 1" anlegen. Unter derselben Ressource dann ein Programm "P- 1" anlegen, wobei bei diesem Namen vor dem Leerzeichen ein durch Alt-0173 eingebaues geschütztes Trennzeichen steht. Dann stürzt SILworX ab.

**Abhilfe:**

Keine Abhilfe

- **Auswirkung:**  
In SILworX V4 konnte es bei Löschaktionen dazu kommen, dass in der Datenbank Objekte übrig blieben, die nicht mehr bearbeitet werden konnten. Diese Objekte hatten keinen Einfluss mehr auf das übrige Projekt, wurden aber bei Prüfe Projektintegrität gemeldet.

**Bedingung:**

Projekte der SILworX Versionen 4 und 5, die solche "übriggebliebenen" Objekte enthalten, sind mit großer Wahrscheinlichkeit nicht in die Versionen V6 und V7 konvertierbar. Die Wahrscheinlichkeit ist besonders groß, wenn die Projekte benutzerdefinierte Datentypen enthalten.

**Abhilfe:**

Die in der Integritätsprüfung gefundenen Objekte sind vor der Konvertierung zu entfernen. Die einfachste Vorgehensweise hierzu ist nachfolgend beschrieben und ist in der alten SILworX Version durchzuführen:

- 1) Alle Knoten im Strukturbaum, die sich unterhalb des Projekt-Knotens im Projekt befinden, archivieren, mit Ausnahme des Knotens Programmiergerät.
- 2) In der alten SILworX Version ein neues Projekt erstellen.
- 3) Im neuen Projekt den Knoten Konfiguration löschen.
- 4) Die unter Punkt 1 archivierte Konfiguration und, falls vorhanden, weitere Knoten im neuen Projekt unter dem Projektknoten wiederherstellen.

Das so erzeugte Projekt sollte in die aktuelle SILworX Version konvertierbar sein.

- **Auswirkung:**  
Fehlermeldung aufgrund nicht auflösbarer Ziele nach Projekt-Konvertierung auf V10  
Verweise auf Objekte, die in einer vor-V10-Version gelöscht wurden, werden nach der Konvertierung wieder sichtbar und müssen manuell entfernt werden. Das kann z.B. eine gelöschte CPU- oder COM-Baugruppe sein, die in einem safeethernet-Interface Kanal genutzt wurde. Dieser Kanal zeigt in V10 den Verweis "?" auf die nicht mehr verfügbare Baugruppe. In anderen Fällen wird ein Pfad auf das verwiesene Objekt angezeigt.

**Bedingung:**

Mit einer Version vor V10 (beispielsweise V9.36.0) ein neues Projekt mit zwei Ressource anlegen und jeweils eine CPU und eine COM einfügen. Zwischen den Ressourcen eine Safeethernet-Verbindung erzeugen und beide Baugruppen verwenden. Dann COM oder CPU löschen.

Der Verweis auf den Kanal scheint verschwunden zu sein. Nun nach V10 konvertieren. Verweis ist wieder da (mit einer "?" als System-ID, da das Ziel nicht auflösbar ist).

**Abhilfe:**

Den Verweis von Hand korrigieren bzw. auf "Keine" setzen. Im Fall eines safeethernet-Interface: den zweiten Kanal manuell deaktivieren.



- **Auswirkung:**  
Archive aus SILworX Versionen vor V10 können in SILworX V10 nicht wiederhergestellt werden, wenn das archivierte Objekt einen Namen hat, der kein gültiger Dateiname in Windows darstellt, insbesondere also wenn er ungültige Zeichen enthält ( \ : \* ? " < > )

**Bedingung:**

In einer SILworX Version vor V10 ein Strukturbaumobjekt erzeugen, dessen Name kein gültiger Dateiname ist (a:b oder Test/01). Das Objekt archivieren. Es wird ein Archiv angelegt. Der vorgeschlagene Name dieses Archivs ersetzt die ungültigen Zeichen durch einen Unterstrich.

Der Versuch, dieses Archiv wiederherzustellen schlägt in V10 fehl.

**Abhilfe:**

Archiv in der Originalversion (oder in V9) öffnen, Objekt umbenennen und erneut archivieren oder

Archiv in Originalversion wiederherstellen und übergeordnetes Objekt archivieren. Dann kann das übergeordnete Objekt in V10 wiederhergestellt werden, und das gewünschte Objekt kopiert oder verschoben werden.

## 4.14 Projekthistorie

- **Auswirkung:**  
Beim Import eines mit SILworX V2 erstellten englischen Projekts interpretiert SILworX das Datum in der Projekthistorie falsch. Z. B. wird 1/11/2013 statt dem 11. Jan 2013 als 1. November 2013 interpretiert. 1/13/2013 wird als ungültiges Datum interpretiert und führt zum Ergebnis 1. Jan 2000 (Standardwert).

**Bedingung:**

SILworX V2 (oder kleinere Version) Projekt mit englischer Sprachversion erzeugen. Anschließend dieses Projekt mit V8.34 öffnen. Die Meldungen werden in der oben angegebenen Weise in die Projekthistorie eingelesen.

**Abhilfe:**

keine Abhilfe

## 4.15 Dokumentation

- **Auswirkung:**  
Querverweise von Struktur- oder Array-Elementen werden nicht ausgedruckt.

**Bedingung:**

Strukturelement oder Array-Element verwenden und Querverweis in Dokumentation ansehen.

**Abhilfe:**

keine Abhilfe

## 4.16 Benutzerverwaltung

- **Auswirkung:**  
Beim Wiederherstellen eines Benutzerverwaltungs-Archivs, welches mit V9 erstellt wurde, wird ein enthaltener Standard-Benutzer nicht berücksichtigt. Der Anwender muss sich dabei mit Benutzernamen und Passwort explizit anmelden

**Bedingung:**

- Neues Projekt mit Benutzerverwaltung anlegen.
- Standardbenutzer festlegen.
- Benutzerverwaltung archivieren.
- vorhandene Benutzerverwaltung im Projekt entfernen.
- archivierte Benutzerverwaltung wiederherstellen.

**Abhilfe:**

keine Abhilfe

## 4.17 Englisch des SILworX

- **Auswirkung:**  
Die Diagnoseübersicht für HIQuad X-Systeme zeigt einige Meldungen auch bei eingestellter Sprachauswahl "Englisch" nur in deutscher Sprache an.

**Bedingung:**

Nach Sprachauswahl "Englisch" die Diagnoseübersicht eines HIQuad X-Systems anschauen: einige Meldungen erscheinen trotzdem in deutscher Sprache.

**Abhilfe:**

Keine Abhilfe

## 5 Besonderheiten

Bei der Benutzung von SILworX sind die beschriebenen Besonderheiten zu beachten.

- **Auswirkung:**

Im HW-Editor werden Skalierungseinstellungen eines Analogwerts als REAL gelesen. SILworX liest die eingegebenen Werte für die Stützpunkte eines Analogwerts (bei 4 mA und 20 mA) als REAL. Die Weiterverarbeitung erfolgt dagegen als LREAL. Auch im Anwenderprogramm kann LREAL benutzt werden. Die Einschränkung hat aber nur bei sehr großen oder sehr kleinen Stützpunkt-Werten Auswirkungen.

**Bedingung:**

Durch die Verwendung von extrem kleinen oder großen Stützpunkt-Werten kann die Genauigkeit des Prozesswertes schlechter werden.

**Abhilfe:**

Rohwerte im Anwendungsprogramm verarbeiten.

- **Auswirkung:**

Logische Verknüpfungen von BOOL-Variablen, deren Werte von Fremdsystemen stammen, können andere als die erwarteten Ergebnisse liefern

**Bedingung:**

Die Ursache ist die Codierung der BOOL-Werte im externen System, die von der im HIMA System abweicht.

**Abhilfe:**

Es gibt zwei Möglichkeiten der Abhilfe:

- das externe System liefert definiert nur die Werte 0 für FALSE und 1 für TRUE
  - ins Anwenderprogramm wird für alle entsprechenden BOOL-Variablen eine Korrekturschaltung eingefügt, die den Wert auf 0 bzw. 1 normiert:
- unnormierte Variable -> Baustein AtoByte -> Baustein AtoBOOL -> normierte Variable

- **Auswirkung:**

Bei Berechnungen mit Variablen vom Datentyp REAL oder LREAL können die Zykluszeiten insbesondere bei Verwendung von trigonometrischen Funktionen extrem schwanken

**Bedingung:**

s.o.

**Abhilfe:**

Für die Bemessung der Watchdog-Zeit ist es notwendig, die Zykluszeit unter realistischen Bedingungen zu bestimmen. Siehe Sicherheitshandbuch: Kapitel: „Genaue Bestimmung der Watchdogzeit.“

- **Auswirkung:**  
Online-Test und Offline-Simulation zeigen den Wert von Systemvariablen des Anwenderprogramms nicht an:
  - OLT-Feld ist leer
  - Der Wert von digitalen Systemvariablen ist nicht durch die Farbe der entsprechenden Linien dargestellt
  - Die Spalte Prozess-Wert im Register System-Variablen der Objektauswahl ist leer
  - Der Force-Editor enthält keine Systemvariablen

Bedingung:

s. o.

Abhilfe:

Die meisten Informationen sind an anderer Stelle, z. B. im Control Panel, ersichtlich. Zur Anzeige im

OLT die Systemvariable mit einer Variablen verbinden und an die Variable ein OLT-Feld anschließen.

Forcen ist nur möglich, wenn die Systemvariable mit einer Variablen verbunden ist.

- **Auswirkung:**  
Wertänderungen von VAR\_INPUT-Variablen benutzerdefinierter Funktionsbausteine SILworX behandelt VAR\_INPUT-Variablen bei benutzerdefinierten Funktionsbausteinen je nach Beschaltung der Eingänge unterschiedlich:
  - Bei Beschaltung der Eingänge mit Variablen von Standard-Datentypen wird der Wert der Variablen an eine baustein-lokale Kopie übergeben (Call by value).
  - Bei Beschaltung der Eingänge mit Variablen von benutzerdefinierten Datentypen wird die Referenz auf die Variable übergeben (Call by reference).

Bedingung:

Dieses Verhalten kann zu Fehlern führen wenn alle folgenden Bedingungen zutreffen:

- Die Quelle einer VAR\_INPUT-Variable ist eine VAR\_EXTERNAL-Variable.
- Die selbe Quelle der VAR\_INPUT-Variable wird im aufgerufenen Funktionsbaustein gleichzeitig als VAR\_EXTERNAL-Variable verwendet.

Wird der Wert der VAR\_EXTERNAL-Variablen im Funktionsbaustein verändert, hat anschließendes Lesen der entsprechenden VAR\_INPUT-Variablen im Funktionsbaustein folgende Auswirkungen:

- Bei benutzerdefiniertem Datentyp werden die aktuellen Werte gelesen.
- Bei elementarem Datentyp werden die alten Werte gelesen, die bei Beginn der Abarbeitung der Bausteininstanz gültig waren.

Abhilfe:

VAR\_EXTERNAL-Variablen sollten nicht gleichzeitig als Quelle einer VAR\_INPUT von Instanzen dieser POE verwendet werden.

- **Auswirkung:**  
Das Dokumentenmanagement kann den Inhalt der Online-Hilfe einer benutzerdefinierten POE nicht ausdrucken.

Bedingung:

-

Abhilfe:

Anzeigen der Online-Hilfe und Ausdrucken der einzelnen Themen aus Windows.

- **Auswirkung:**  
Es kann nicht ausgeschlossen werden, dass sich zwischen SILworX Versionen Schlüsselbegriffe in den Export-/Import-Dateien (.CSV, .XML) ändern. Dadurch importiert SILworX die entsprechenden Daten als Standardwert und gibt eine Fehlermeldung aus.

Bedingung:

Beispiel: In Versionen vor V5.xx ist der Datentyp für die Spracheinstellung English als «Data Type» gekennzeichnet, ab V5.xx als «Data type». Beim Import einer Export-Datei einer Version vor V5.xx legt SILworX alle Variablen mit dem Standard-Datentyp BOOL an.

Abhilfe:

In den zu importierenden Dateien die entsprechenden Schlüsselwörter anpassen

- **Auswirkung:**

Wird bei bestehendem System-Login die Diagnoseansicht geöffnet und die Verbindung getrennt, dann bietet SILworX beim Versuch, die Verbindung neu aufzubauen, das Modul-Login an

Bedingung:

Hardware-Login, Detailansicht zu Baugruppe öffnen.

Diagnose zu dieser Baugruppe in zweitem Fenster öffnen.

Anschließend Verbindung-Trennen.

Diagnose bietet nach Verbindungsverlust nur den Modul-Login an,  
Detailansicht bietet System-Login an.

Abhilfe:

Wenn Modul-Login-Dialog für Diagnose einmal geöffnet wurde, müssen alle Online-Ansichten dieses Moduls (Diagnose und Modul-Online-Ansicht) geschlossen und dann wieder geöffnet werden, damit sie wieder über das System ausgelesen werden können.

- **Auswirkung:**

Für HlMatrix Geräte vor F\*03 (z. B. F30 01, F31 02, F35 01, F35 012 und F60 CPU 01) wird der Force-Status in Force-Ansichten für lokales Forcen (Oberhalb der Force-Tabelle) mit regulären Werten angezeigt, als wären die Informationen tatsächlich verfügbar, sie sind aber ohne Funktion. Im Einzelnen sind dies: Force-Zustand, Geforcte Variablen, Verbleibende Force-Dauer und Force-Timeout-Reaktion

Bedingung:

-

Abhilfe:

Verhalten ist in Online Hilfe beschrieben

## 6 Upgrade von einer Vorversion

Projektdateien aus vorherigen Versionen können in V10.58 weiterverwendet werden.

Dabei werden keine CRC-Änderungen auftreten, solange die Versionseinstellung minimale Konfigurationsversion einer Ressource nicht verändert wird. SILworX hält die CRCs kompatibel, sofern keine Änderungen erfolgen bzw. keine neuen Features verwendet werden.

Das Upgrade von einer Version ab V2.36 auf V10.58 ist folgendermaßen durchzuführen:

- Es sollte eine Sicherung des Projektes durchgeführt werden.
- Vor der Konvertierung für alle Ressourcen Code generieren. Durch den Export und Import der Konfiguration im Versionsvergleich lassen sich eventuelle Abweichungen bei der Konvertierung feststellen.
- Projekt in V10.58 öffnen und konvertieren.
- Da die Konvertierung sehr umfangreich ist, nach der Konvertierung die Projektintegrität prüfen.
- In V10.58 eine Codegenerierung durchführen, um festzustellen, ob Fehler auftreten und/oder sich CRCs ändern. Im Versionsvergleich kann dies mit der importierten Konfiguration festgestellt werden.
- Erkannte Fehler beseitigen und erneut Code generieren, um CRC-Änderungen festzustellen.
- Liegen keine CRC-Änderungen vor, ist der Übergang erfolgreich abgeschlossen.
- Liegen CRC-Änderungen vor, prüfen, ob diese akzeptabel sind.
- Sind die Änderungen akzeptabel, ist der Übergang erfolgreich abgeschlossen.
- Falls diese inakzeptabel sind, mit der entsprechenden Vorversion weiterarbeiten.

Hinweise zur Konvertierung:

- Die Konvertierung von Versionen vor V2.36 ist in den Release-Notes zu V2.36 beschrieben.
- Die Konvertierung kann bei sehr großen Projekten bis zu mehreren Stunden dauern.

### 6.1 Referenzen

- SILworX Online-Hilfe
- SILworX Erste-Schritte-Handbuch, HI 801 102 D
- Kommunikationshandbuch, HI 801 100 D
- HIPRO-S V2 Handbuch, HI 800 722 D
- ISOfast Handbuch, HI 801 464 D
- Modbus V2 Handbuch, HI 801 474 D
- X-OPC Server Handbuch, HI 801 479 D

## Release-Notes

---

HI 801 498 D

Für weitere Informationen kontaktieren Sie:

**HIMA Paul Hildebrandt GmbH**

Albert-Bassermann-Str. 28

68782 Brühl, Deutschland

Telefon: +49 6202 709-0

Fax +49 6202 709-107

E-Mail: [info@hima.com](mailto:info@hima.com)

Erfahren Sie online mehr über **HIMA** Lösungen:



[www.hima.com/de](http://www.hima.com/de)



[www.hima.com](http://www.hima.com)