Versionsvergleich

Für mit SILworX programmierbare Steuerungen

SAFETY
NONSTOP

HIMA

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.

HIMax®, HIMatrix®, SILworX®, XMR® und FlexSILon® 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.

Weitere Informationen sind in der Dokumentation auf der HIMA DVD und auf unserer Webseite unter http://www.hima.de und http://www.hima.com zu finden.

© Copyright 2017, HIMA Paul Hildebrandt GmbH

Alle Rechte vorbehalten.

Kontakt

HIMA Adresse:

HIMA Paul Hildebrandt GmbH

Postfach 1261

68777 Brühl

Tel.: +49 6202 709-0

Fax: +49 6202 709-107

E-Mail: info@hima.com

Revision	Änderung	Art der Änderung	
		Technisch	Redaktionell
V2.00	Anpassung an SILworX V7	x	x
V2.02	Redaktionelle Änderungen		x

Inhaltsverzeichnis

1 Einleitung **7**
1.1 Zielgruppe und erforderliche Kompetenz 7
1.2 Anwendung des Versionsvergleichs 7
2 Darstellungskonventionen **8**
2.1 Sicherheitshinweise 8
2.2 Gebrauchshinweise 8
3 Prinzip **9**
4 Vorbereitung **10**
4.1 Empfehlungen zur Programmierung 10
4.2 Versionsvergleich vorbereiten 10
4.3 Geladene Ressourcekonfiguration online vergleichen 11
4.4 Ressourcekonfiguration exportieren 12
4.5 Konfigurationen für Vergleich auswählen 13
4.5.1 Ressourcekonfiguration importieren 14
5 Anzeigen des Versionsvergleichs **17**
5.1 CRC-Vergleich 17
5.1.1 Kennzeichnen von hinzugefügten, gelöschten und geänderten Funktionsgruppen 18
5.2 Inhalt und Bedeutung der wichtigsten Dateien 18
5.2.1 Hardware, Module 19
5.2.2 CPU-Konfiguration und Systemdaten 20
5.2.3 COM-Konfiguration und Protokolle 22
5.2.4 Logikdaten 23
5.2.5 PGS-Daten (Konfigurationsverbindungen, Benutzermanagement) 25
5.2.6 Erforderliche Betriebssystem-Version eines Objekts 26
6 Detailauswertung **27**
6.1 Hardware-Änderungen (im Hardware-Editor) 27
6.1.1 E/A-Module: io4io.config 27
6.1.2 Systemebene (CPU): io4cpu.config 28
6.1.3 Neues Modul hinzufügen 29

6.2 Logikänderungen (Logic-Solver) 30
6.2.1 Wertfeld am Eingang eines Funktionsbausteins ändern 30
6.2.2 Neues Objekt in Logik einfügen 31
6.2.3 Ausgang einer Funktion invertieren 32
6.2.4 POE in der Logik löschen 33
6.2.5 Netzwerk verschieben (Abarbeitungsreihenfolge ändern) 34
6.2.6 Lokale Variable ändern (Neu, löschen, Initialwert) 37
6.2.7 Neues Netzwerk erstellen 37
6.2.8 Funktionsbaustein-Instanz umbenennen 38
6.2.9 Globale Variable neu zuweisen 39
6.2.10 Neue Variablenzuweisung einfügen 39
6.2.11 Variable umbenennen 41
6.2.12 Spezielle Änderungen der Datei Programm.ldb 43
6.2.13 Änderungen in der Structured-Text-Logik 44
6.3 Änderung der Zuordnung von globalen Variablen 46
6.3.1 Globale Variable einem anderen Hardware-Eingang zuordnen 46
6.3.2 Neuer Initialwert für eine globale Variable 47
6.4 Trennung sicherer und nicht sicherer Logik 48
6.5 Speicherübersicht für Code und Daten 50
6.5.1 Speicherübersicht am Beispiel HIMax 50
6.6 Änderungen in safeethernet Kommunikation 51
6.6.1 Neue Variable zu bestehender Verbindung hinzufügen 51
6.6.2 safeethernetParameter Änderungen 52
7 Ausdruck der Vergleichsinformationen 53
7.1 Ausdruck vorbereiten 54
7.2 Referenzierung auf Projektarchive 55
7.3 Erzeugen einer Dokumentation (Ausdruck) 56
7.3.1 Dokumentation starten 56
7.3.2 Deckblatt ausfüllen 57
7.3.3 Objekte (Seiten) zum Drucken auswählen 59

7.3.3.1 Übersicht der wichtigsten Dokumente des Versionsvergleichs 60
7.4 Dokumentation geänderter Objekte 61
7.4.1 Auswahl zu druckender Seiten 61
7.4.2 Dokumentation von Hardware-Änderungen 63
7.4.3 Dokumentation geänderter Variablenzuordnungen 64
7.4.4 Dokumentation geänderter safeethernet Einstellungen 65

1 Einleitung

Dieses Handbuch beschreibt die Anwendung des SILworX Versionsvergleich für die programmierbaren, sicherheitsgerichteten Steuerungen der Systemfamilien HIMax, HIMatrix und HIMatrix M45.

Bedingt durch den unterschiedlichen Hardware-Aufbau ergeben sich vor allem in der Dateistruktur der Ressourcekonfiguration und bei der Anzeige von Hardware-Änderungen Unterschiede. Sofern nicht ausdrücklich anders erwähnt, gelten daher alle Aussagen für alle oben genannten Systemfamilien.

1.1 Zielgruppe und erforderliche Kompetenz

Dieses Dokument wendet sich an Planer, Projekteure und Programmierer von Automatisierungsanlagen sowie an Personen, die zur Inbetriebnahme, zum Betrieb und zur Wartung der Geräte und Systeme berechtigt sind. Vorausgesetzt werden spezielle Kenntnisse auf dem Gebiet der sicherheitsgerichteten Automatisierungssysteme.

Die Teilnahme an einer SILworX Systemschulung ist zum Verständnis der Inhalte dieses Dokumentes dringend empfohlen.

Jedes Fachpersonal (Planung, Montage, Inbetriebnahme) muss über die Risiken und deren mögliche Folgen unterrichtet sein, die im Falle einer Manipulation von einem sicherheitsgerichteten Automatisierungssystem ausgehen können.

Der Anlagenbetreiber ist für die Qualifikation und Sicherheitseinweisung des Bedien- und Wartungspersonals verantwortlich.

1.2 Anwendung des Versionsvergleichs

Das HIMax Sicherheitshandbuch fordert ausdrücklich die Anwendung des Versionsvergleichs:

> **!** Der sichere Versionsvergleich ist zur Prüfung von Programmänderungen vor dem Laden in die Steuerung einzusetzen!

Die Regelwerke der funktionalen Sicherheit (z. B. IEC 61508, IEC 61511) fordern bei geplanten Änderung in einem Sicherheitstechnischen System die Anwendung entsprechender normkonformer Modifikationsprozeduren.

Wesentlicher Bestandteil dieser Modifikationsprozeduren ist die Bewertung der Auswirkungen von geplanten Änderungen auf die funktionale Sicherheit und die nachfolgende Re-Validierung (Test).

Der Versionsvergleich stellt dem Anwender die dafür erforderlichen Informationen technisch detailliert und in ausreichender Aussagekraft zur Verfügung.

> **i** Der Versionsvergleich informiert nicht nur über sicherheitsrelevante Änderungen sondern auch über verfügbarkeitsrelevante Änderungen! Er hilft damit auch, unnötige Anlagenabschaltungen durch das Erkennen von nicht beabsichtigten Änderungen zu vermeiden.

Fazit:

Der Versionsvergleich ist zwingend vor jedem Ladevorgang in eine sicherheitsgerichtete Steuerung anzuwenden! Darüber hinaus ist der Versionsvergleich ein nützliches Werkzeug während der Design- und Implementierungsphase der Anwenderapplikation, z. B. durch den Vergleich unterschiedlicher Projektstände.

2 Darstellungskonventionen

Zur besseren Lesbarkeit und zur Verdeutlichung gelten in diesem Dokument folgende Schreibweisen:

Format	Beschreibung
Fett	Hervorhebung wichtiger Textteile. Diese sind Bezeichnungen von Schaltflächen, Menüpunkten und Registern in SILworX, welche angeklickt werden können.
Kursiv	Parameter, Systemvariablen und Referenzen auf andere Textstellen.
`Courier`	Wörtliche Benutzereingaben, oder Anzeigen, welche mit dem abgedruckten Wert exakt identisch sind.
RUN	Bezeichnungen von Betriebszuständen in Großbuchstaben.
Kapitel 1.2.3	Querverweise auf andere Kapitel. Sie sind als Hyperlinks ausgeführt. Klicken Sie auf den Hyperlink, um im Dokument zur referenzierten Stelle zu springen.

Sicherheits- und Gebrauchshinweise sind - wie nachfolgend beschrieben - besonders gekennzeichnet.

2.1 Sicherheitshinweise

Um sich als Anwender einem möglichst geringen Risiko auszusetzen, sind die Sicherheitshinweise unbedingt zu befolgen.

Die Sicherheitshinweise werden wie folgt dargestellt:

- Signalwort: Warnung, Vorsicht
- Art und Quelle des Risikos
- Folgen bei Nichtbeachtung
- Vermeidung des Risikos

SIGNALWORT
Art und Quelle des Risikos!
Folgen bei Nichtbeachtung
Vermeidung des Risikos

Bedeutung der Signalworte

- **Warnung:** Bei Missachtung droht schwere Körperverletzung bis Tod.
- **Vorsicht:** Bei Missachtung droht leichte Körperverletzung.

2.2 Gebrauchshinweise

Zusatzinformationen sind wie folgt aufgebaut:

An dieser Stelle steht der Text der Zusatzinformation.

3 Prinzip

Bei der Codegenerierung werden durch SILworX verschiedene Dateien erzeugt. Dieser Datensatz wird als *die Ressourcekonfiguration* bezeichnet. Beim Download oder Reload wird immer die komplette Ressourcekonfiguration in die Ressource geladen.

Beim Versionsvergleich werden verschiedene Ressourcekonfigurationen miteinander verglichen und die Unterschiede zwischen den einzelnen Dateien angezeigt. Das Ergebnis hat SIL-3-Qualität und beruht auf den Dateien, welche den ausführbaren Code beschreiben.

Im Wesentlichen gibt es drei Typen von Ressourcekonfigurationen:

1. Die erzeugte Ressourcekonfiguration ist das Ergebnis der letzten Codegenerierung (→ Codegenerator).
2. Die geladene Ressourcekonfiguration ist die durch einen Download (→ Download) oder Reload in die Steuerung übertragene Ressourcekonfiguration.
3. Eine unbekannte Ressourcekonfiguration stellt einen beliebigen Stand einer Ressource dar, der exportiert und gesichert wurde (→ IM).

4 Vorbereitung

4.1 Empfehlungen zur Programmierung

Bei der Programmierung sicherheitstechnisch relevanter Logik sollten die Folgen zukünftiger Änderungen frühzeitig bedacht werden. Damit die Ergebnisse des Versionsvergleichs möglichst einfach zu interpretieren sind, werden folgende Maßnahmen empfohlen:

- Strukturierte Programmierung und an den Prozess angepasste Aufteilung der Logik in einzelne Programme und Funktionsbausteine.
- Individuelle und an den Prozess angepasste Namen für Instanzen (verwendete Bausteine).
- Die Namen für Logikseiten sollten aussagekräftig und eindeutig sein.
- Konnektoren sollten nicht über eine größere Anzahl von Logikseiten verwendet werden. Dadurch entstehen weit verzweigte Netzwerke, was die Übersichtlichkeit, insbesondere beim Versionsvergleich, vermindert.

4.2 Versionsvergleich vorbereiten

Die im vorliegenden Dokument beschriebene Qualität des Versionsvergleichs setzt voraus, dass für das zu vergleichende Projekt bereits eine Codegenerierung mit mindestens SILworX V4 durchgeführt wurde.

Bevor geplante Änderungen ausgeführt werden, sollte eine Kopie des Projekts erstellt werden, so dass man letztlich ein Projekt [ALT] ohne Änderungen und ein Projekt [NEU] mit Änderungen erhält.

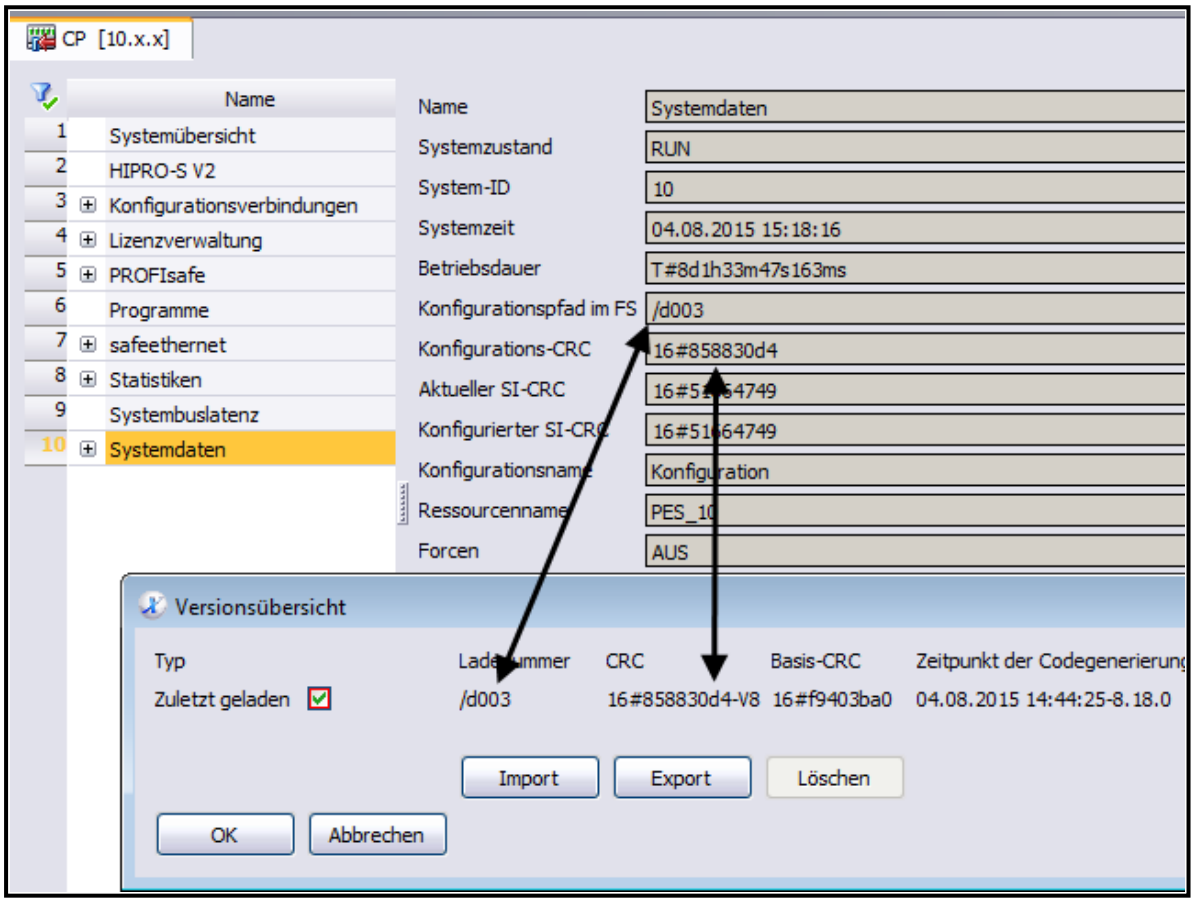
Projekt [ALT] entspricht der abgenommen und für den sicherheitsgerichteten Betrieb zugelassenen Version und wird durch einen eindeutigen CRC beschrieben. Projekt [ALT] muss vollständig dokumentiert sein.

Ziel des Versionsvergleichs ist, die Änderungen zwischen der in einer Steuerung geladenen (importierten) Ressourcekonfiguration von Projekt [ALT] und der modifizierten (code-generierten) Ressourcekonfiguration von Projekt [NEU] aufzuzeigen.

4.3 Geladene Ressourcekonfiguration online vergleichen

Es muss geprüft werden, ob die im SILworX Projekt gespeicherte *geladene Ressourcekonfiguration* der tatsächlich in der Steuerung geladenen Ressourcekonfiguration entspricht.

- Öffnen Sie das Control Panel (CP) der Ressource, für welche Sie den Vergleich durchführen wollen.
- Klicken Sie auf **Systemdaten**.
- Klicken Sie im Menü **Online, Versionsvergleich**. Der Dialog *Versionsübersicht* öffnet sich.



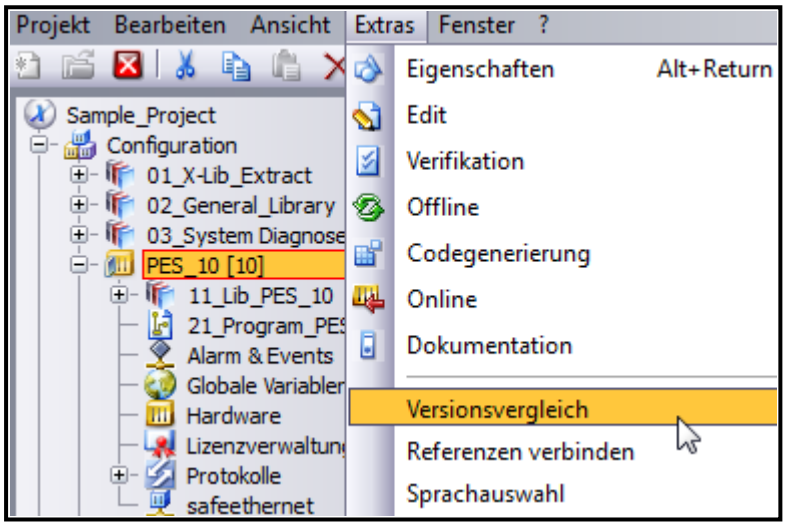
- Vergleichen Sie *Ladenummer* und *CRC*. Diese müssen mit den angezeigten Werten im Control Panel übereinstimmen.

4.4 Ressourcekonfiguration exportieren

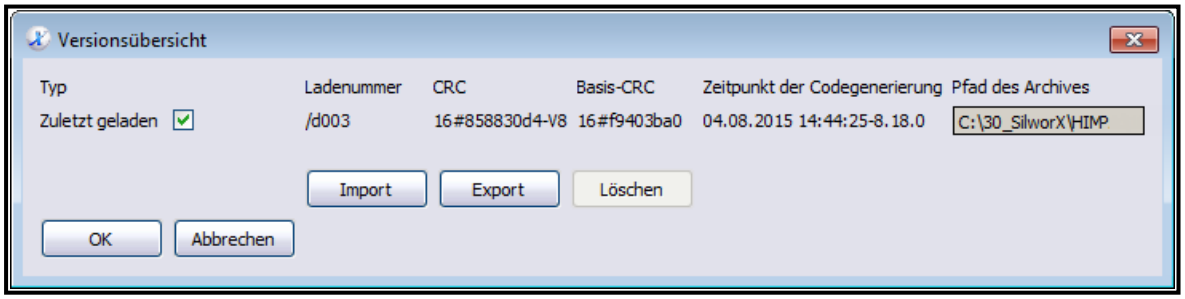
Aus dem als Vergleichsbasis gewählten Projekt [ALT] wird die zuletzt geladene Ressourcekonfiguration exportiert. Führen Sie dazu die folgenden Schritte aus:

- Markieren Sie im Strukturbaum die Ressource, für welche Sie den Versionsvergleich durchführen wollen.

- Klicken Sie im Menü auf **Extras, Versionsvergleich.** Der Dialog *Versionsübersicht* öffnet sich.



- Aktivieren Sie im Dialog *Versionsübersicht* die Option **Zuletzt geladen**. Dadurch wird die zuletzt in die Ressource geladene Version exportiert.



- Klicken Sie auf **Export**. Der Dialog *Archivieren* öffnet sich. SILworX erzeugt automatisch einen *Archivnamen* mit den wichtigsten Daten.

Beispiel:

PES_10_WGL_4_D_1_DL_0xf9403ba0_V3_28_03_2011_15_53_58

PES_10_	WGL_4_D_1_	DL_	0xf9403ba0_V3_	28_03_2011_15_53_58
Ressourcename	Projektname	DL = Download - = geladene Datei. Identisch für Reload und Download.	CRC der Ressourcekonfiguration	Datum und Uhrzeit der Codegenerierung

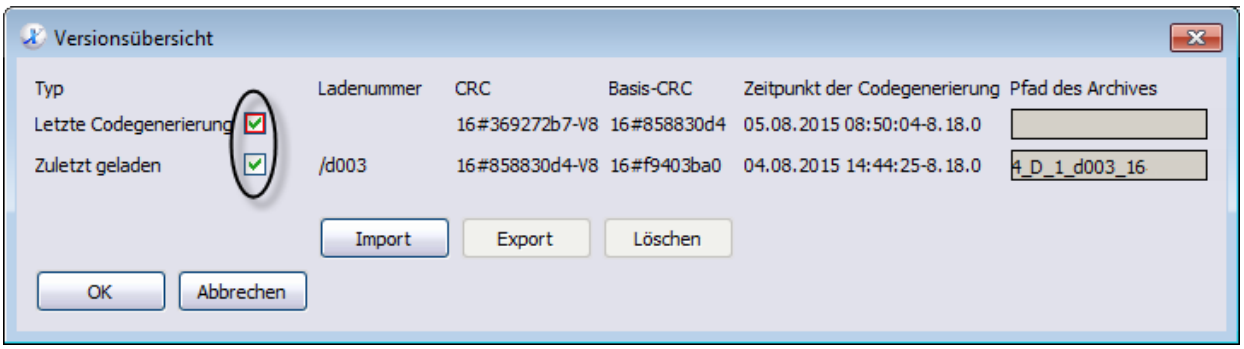
Passen Sie bei Bedarf das *Archivverzeichnis* an. Falls gewünscht können Sie zusätzlich einen Kommentar eingeben. Klicken Sie abschließend auf **OK**, um das Archiv zu speichern.

4.5 Konfigurationen für Vergleich auswählen

Ein Codevergleich kann zu jedem Zeitpunkt durchgeführt werden.

Entsprechend den Regeln der Funktionalen Sicherheit und des HIMA Sicherheitshandbuches muss bereits direkt nach der Codegenerierung und noch vor dem Laden einer Änderung die neu generierte Konfiguration (CG) mit der zuletzt geladenen Konfiguration (DL) verglichen werden. Beachten Sie dazu die entsprechenden Kapitel in den HIMax und HIMatrix Sicherheitshandbüchern.

Das folgende Bild zeigt beispielhaft eine solche Situation.



Folgende Konfigurationen können untereinander verglichen werden:

- Importierte Konfiguration(en) → IM
- Zuletzt geladene Konfiguration → DL
- Zuletzt generierte Konfiguration → CG

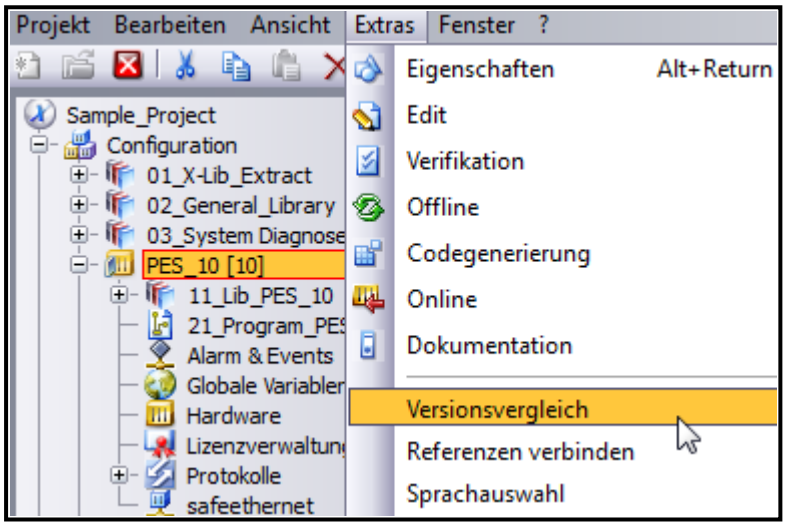
Der endgültige Nachweis aller Änderungen erfolgt erst nach erfolgreich ausgeführten Tests. Dann wird die zuletzt geladene Konfiguration (DL) mit der importierten, ursprünglichen Konfiguration (IM) verglichen.

4.5.1 Ressourcekonfiguration importieren

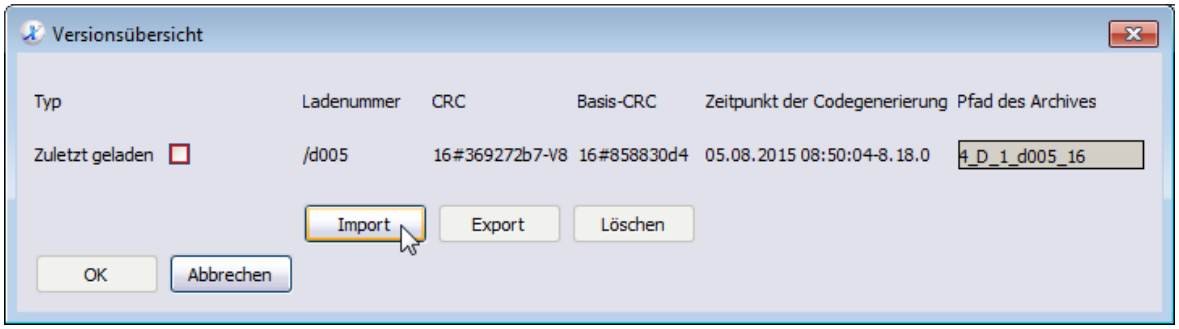
- Stellen Sie sicher, dass eine geeignete Ressourcekonfiguration exportiert wurde (siehe*Ressourcekonfiguration exportieren*.

- Markieren Sie im Strukturbaum die Ressource, für welche Sie den Versionsvergleich durchführen wollen.

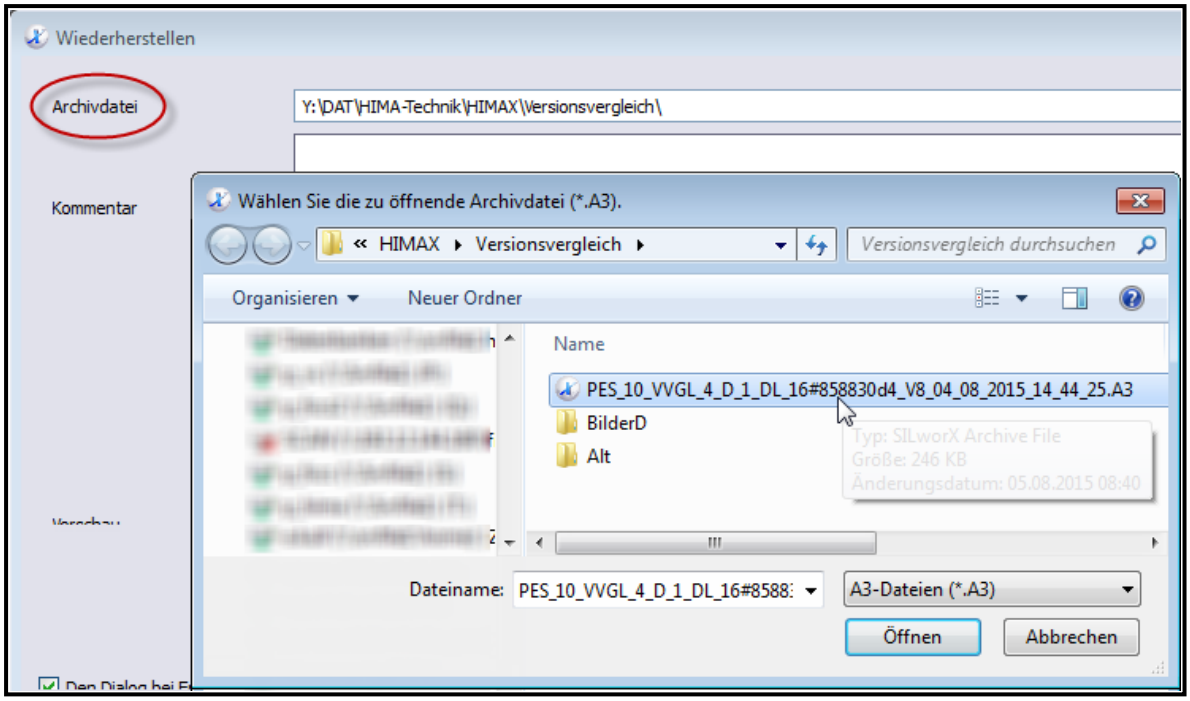
- Klicken Sie im Menü auf **Extras, Versionsvergleich.** Der Dialog *Versionsübersicht* öffnet sich.



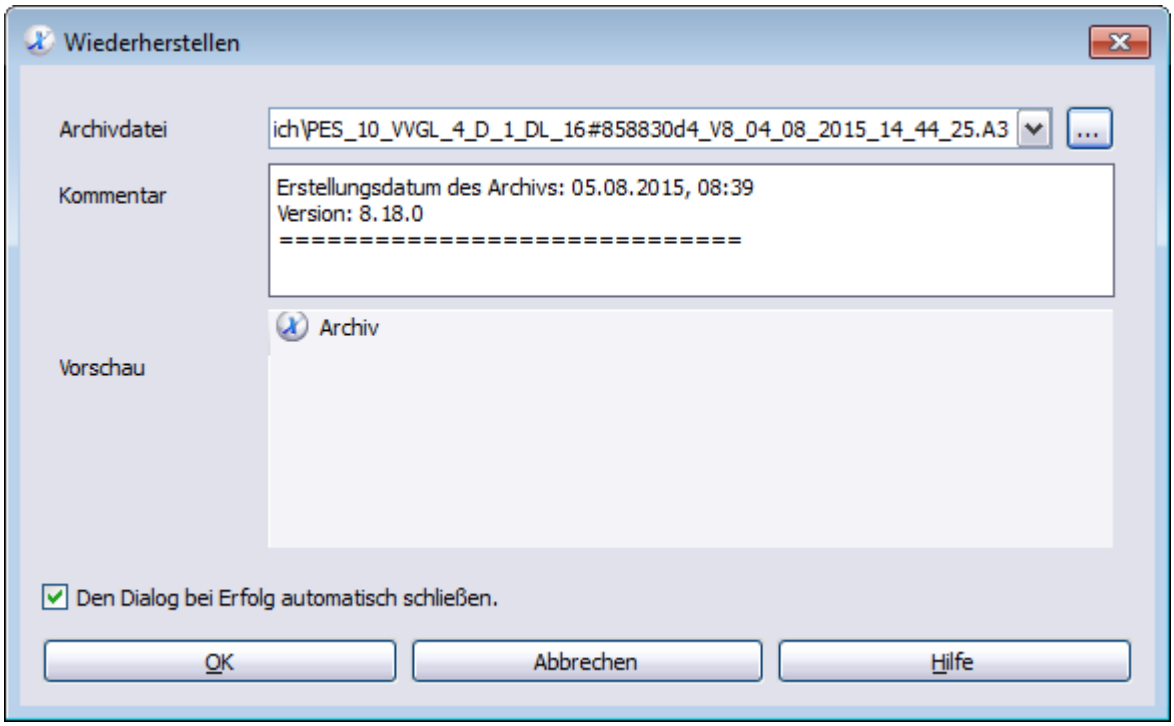
- Klicken Sie auf die Schaltfläche **Import**. Der Dialog *Wiederherstellen* öffnet sich.



- Öffnen Sie die Dropdown-Liste von *Archivdatei* und wählen Sie ein Archiv aus.
  Sollte sich das gewünschte Archiv nicht in der Dropdown-Liste befinden, klicken Sie auf die Schaltfläche rechts neben dem Textfeld und wählen Sie die Datei über den Windows-Dialog, wie im folgenden Bild dargestellt.

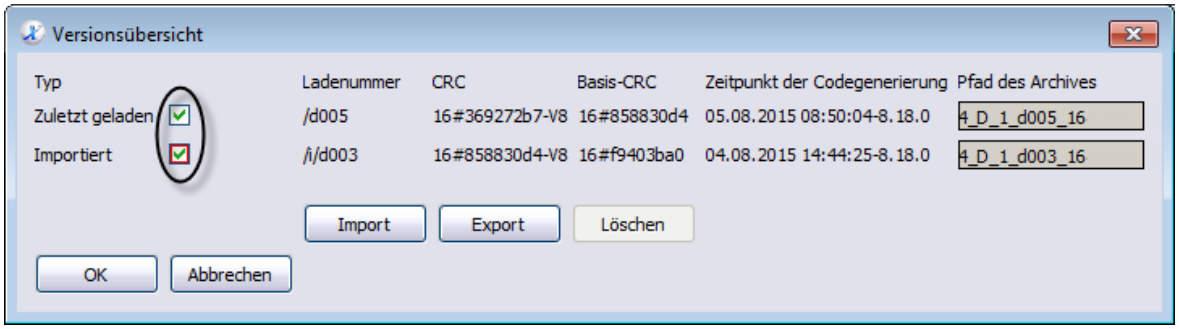


Beachten Sie die Informationen in den Feldern *Kommentar* und *Vorschau*. Sie können das Archiv anhand dieser Angaben identifizieren.



Klicken Sie auf **OK**, um das Archiv wieder herzustellen. Die erfolgreiche Wiederherstellung wird angezeigt und der Dialog *Versionsübersicht* öffnet sich.

Wählen Sie die Versionen, welche Sie vergleichen möchten. Klicken Sie dazu in die Checkboxen rechts von *Zuletzt geladen* und *Importiert*.



Klicken Sie auf **OK**, um den Versionsvergleich zu starten. Anschließend wird das Ergebnis in tabellarischer Form angezeigt.

5 Anzeigen des Versionsvergleichs

5.1 CRC-Vergleich

Der Versionsvergleich erfolgt anhand der vom Codegenerator erstellten Prüfsummen (CRCs) für die unterschiedlichen Funktionsgruppen des Projekts. Die Funktionsgruppen sind hierarchisch gegliedert und für jede Funktionsgruppe existiert mindestens eine Konfigurationsdatei.

Die Konfigurationsdateien werden in einer Liste angezeigt und bei Änderungen farblich gekennzeichnet.

Rot Die Funktionsgruppe, die durch diese Konfigurationsdatei beschrieben ist, wurde geändert.

Gelb Die Funktionsgruppe, die durch diese Konfigurationsdatei beschrieben ist, wurde neu hinzugefügt oder gelöscht.

Die erste Zeile */root.config* entspricht der übergeordneten Codeversion, wie sie auch im Logbuch, oder im Control Panel in den Systemdaten angezeigt wird. Die übergeordnete Codeversion kombiniert die Codeversionen aller Funktionsgruppen. Durch Klicken auf das (+)-Zeichen links in der Zeile, können die untergeordneten Elemente angezeigt werden.

Eine detaillierte Prüfung der Ergebnisse des Codevergleichs ist nur erforderlich, wenn sich die übergeordnete Codeversion ändert.

Versionsvergleich: IM <- CG PES_10 [10]

	Name	Beschreibung	CRC IM	Versi IM	CRC CG	Version CG	Vergleich CRC's
1	/root.config	Hauptdatei der Konfiguration	16#f9403ba0	V3	16#42c4394f	V3	-
2	/0000.01/root.config	Hauptdatei - Systembusmodul	16#239441f4	V3	16#239441f4	V3	ok
3	/0000.02/root.config	Hauptdatei - Systembusmodul	16#104e7fd2	V3	16#104e7fd2	V3	ok
4	/0000.03/root.config	Hauptdatei - CPU	16#acf41d80	V3	16#acf41d80	V3	ok
5	/0000.04/root.config	Hauptdatei - CPU	16#abce2685	V3	16#abce2685	V3	ok
6	/0000.05/root.config	Hauptdatei - Kommunikationsmodul	16#d2396bda	V3	16#d2396bda	V3	ok
7	/0000.06/root.config	Hauptdatei - E/A	16#17d23555	V3	16#17d23555	V3	ok
8	/0000.07/root.config	Hauptdatei - E/A	16#7847c4e6	V3	16#0fdd5505	V3	-
9	/0000.07/io4io.config	E/A	16#d66d29d9	V2	16#dfa1df94	V2	-
10	/0000.07/iot.config	Spannungsversorgung und Temperatu...	16#209c5b8f	V3	16#209c5b8f	V3	ok
11	/0000.08/root.config	Hauptdatei - E/A	16#f1539d17	V3	16#f1539d17	V3	ok
12	/0000.09/root.config	Hauptdatei - E/A	16#bfda79c8	V3	16#bfda79c8	V3	ok
13	/0000.10/root.config	Hauptdatei - E/A	16#fa7051b0	V3	16#fa7051b0	V3	ok
14	/sys/root.config	Hauptdatei - System	16#3cafce69	V3	16#7d39fa1b	V3	-
15	/sys/bgp.config	System Modul	16#d1d4ad81	V2	16#008ddceb	V2	-
16	/sys/cpc.config	Systemprotokolle Basis	16#ced2a001	V2	16#ced2a001	V2	ok
17	/sys/cpcnsip.config	Standardprotokoll	16#64ecf542	V3	16#64ecf542	V3	ok
18	/sys/cpcsip.config	Sicherheitsgerichtetes Protokoll	16#9855ad19	V2	16#9855ad19	V2	ok
19	/sys/cpu.config	Systemdaten	16#03161291	V3	16#03161291	V3	ok
20	/sys/io4cpu.config	System IO	16#5e571427	V3	16#5e571427	V3	ok
21	/sys/ke.config	Datenlayout- und Transport	16#dac785e2	V2	16#dac785e2	V2	ok
22	/sys/lm.config	Lizenz	16#889b1742	V2	16#889b1742	V2	ok
23	/sys/ls.config	Logic-Solver-Konfiguration	16#f8d40ff3	V3	16#f8d40ff3	V3	ok
24	/sys/ls/01_Programm01.cor	Programm-Parameter	16#ef6c15b4	V3	16#ef6c15b4	V3	ok
25	/sys/ls/01_Programm01.ldb	Programm Binärdatei	16#222df24b	V2	16#222df24b	V2	ok

In Ausnahmefällen kann eine Änderung in einer Funktionsgruppe ohne funktionale Auswirkung auf den generierten Code bleiben, wie z. B. das Umbenennen einer Eingangsvariable (siehe *Variable umbenennen*).

Wenn sich die Codeversion nicht ändert, ist keine weitere Funktionsprüfung erforderlich.

Achtung! Gefahr von ungewollten Anlagenabschaltungen!
Umbenennen von Variablen, Programmnamen, FB-Namen, FB-Instanzen etc. führen bei einem Reload zur Neuinitialisierung der genannten Objekte. Gespeicherte Werte und Zustände gehen verloren!

5.1.1 Kennzeichnen von hinzugefügten, gelöschten und geänderten Funktionsgruppen

Die nachfolgende Abbildung zeigt hinzugefügte und gelöschte Funktionsgruppen (gelb), sowie geänderte Funktionsgruppen (rot).

35	Modul-Diag_PES_10	Funktionsbaustein-Typ	16#19d11128	V2	16#19d11128	V2	ok
36	NEUER FB	Funktionsbaustein-Typ	16#00000000		16#dbf3b08e	V2	--
37	Schrittkette-PES_10	Funktionsbaustein-Typ	16#fb5dc124	V2	16#00000000		--
38	SystemTime	Funktionsbaustein-Typ	16#b404049e	V2	16#b404049e	V2	ok
39	Systemüberwachung-PES_10	Funktionsbaustein-Typ	16#579d399c	V2	16#579d399c	V2	ok
40	/sys/ls/01_Programm01_force.config	Force-Daten der Applikation	16#8322f556	V2	16#b8d1ba79	V2	-
41	/sys/ls/01_Programm01_retain.config	Retain-Daten der Applikation	16#b9ae59cc	V2	16#ddaa51f5	V2	-
42	/sys/pgs.config	Konfigurationsverbindungen	16#bf8d0bc3	V2	16#f07e8dfb	V4	-

5.2 Inhalt und Bedeutung der wichtigsten Dateien

Dieses Kapitel stellt lediglich die für den Versionsvergleich zu Grunde liegende Dateistruktur vor. Die genaue Auswertung eventuell angezeigter Änderungen wird in *Detailauswertung* erklärt.

Versionsvergleich: IM <- DL PES_10 [10]

	Name	Beschreibung	CRC IM	Version IM	CRC DL	Version DL	Vergleich der CRCs
1	/root.config	Hauptdatei der Konfiguration	16#31f01e89	V5	16#495704e7	V5	-
2	/0000.01/root.config	Hauptdatei - Systembusmodul	16#b52cf39c	V3	16#b52cf39c	V3	ok
3	/0000.02/root.config	Hauptdatei - Systembusmodul	16#40ce99e0	V3	16#40ce99e0	V3	ok
4	/0000.03/root.config	Hauptdatei - CPU	16#f1df5d79	V3	16#f1df5d79	V3	ok
5	/0000.04/root.config	Hauptdatei - CPU	16#9c6d4236	V3	16#9c6d4236	V3	ok

Die Informationen werden wie folgt angezeigt:

Spalte	Beschreibung
Name	Position des Moduls im Format Rack.Slot, gefolgt vom Namen der Konfigurationsdatei.
Beschreibung	Kurzbeschreibung der Konfigurationsdatei.
CRC IM	Prüfsumme der importierten Konfigurationsdatei.
Version IM	Mindestens erforderliche Betriebssystem-Version für das Modul (importiert), siehe *Erforderliche Betriebssystem-Version eines Objekts*.
CRC CG	Prüfsumme der vom Codegenerator erzeugten Konfigurationsdatei.
Version CG	Wie Version IM, jedoch für die vom Codegenerator erzeugten Konfigurationsdatei.

Spalte	Beschreibung
Version DL	Mindestens erforderliche Betriebssystemversion zur Ausführung der Konfigurationsdatei auf dem entsprechenden Modul.
Vergleich der CRC	OK: Keine Änderung -: Änderung

5.2.1 Hardware, Module

Für jedes Modul gibt es eine Datei mit den zugehörigen Konfigurationsdaten. Wenn Sie die Konfiguration eines Moduls ändern (z. B. IP-Einstellungen, Skalierungswerte, Leitungsüberwachung, Aktivierung von Kanälen usw.), ändert sich auch der Inhalt der Datei.

⊞ /0000.03/root.config	Hauptdatei - CPU	16#acf41d80	V3
⊟ /0000.04/root.config	Hauptdatei - CPU	16#abce2685	V3
/0000.04/ethsw.config	Ethernet-Switch	16#db7bc26f	V3
/0000.04/hh.config	HIMA-HIMA-Kommunikation-Konfiguration	16#18b411e8	V2
/0000.04/iot.config	Spannungsversorgung und Temperaturüberwachung	16#209c5b8f	V3
/0000.04/net.config	Netzwerkeinstellung	16#b808ec69	V2
⊞ /0000.05/root.config	Hauptdatei - Kommunikationsmodul	16#d2396bda	V3
⊞ /0000.06/root.config	Hauptdatei - E/A	16#17d23555	V3
⊟ /0000.07/root.config	Hauptdatei - E/A	16#7847c4e6	V3
/0000.07/io4io.config	E/A	16#d66d29d9	V2
/0000.07/iot.config	Spannungsversorgung und Temperaturüberwachung	16#209c5b8f	V3
⊞ /0000.08/root.config	Hauptdatei - E/A	16#f1539d17	V3

Konfigurationsdatei	Beschreibung
/0000.04/root.config	Hauptdatei des CPU-Moduls in Rack 0, Slot 4. Diese Konfigurationsdatei referenziert auf untergeordnete Konfigurationsdateien und ändert sich immer, wenn sich eine untergeordnete Konfigurationsdatei ändert.
/0000.04/ethsw.config	Eigenschaften der Ethernet-Switches des CPU-Moduls. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Einstellungen der CPU im Hardware-Editor in den Registern *Ethernet-Switch*, *VLAN* und *Port Mirroring*.
/0000.04/hh.config:	Eigenschaften der safe**ethernet** Kommunikation des CPU-Moduls. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Einstellungen im safe**ethernet** Editor. Typische Änderungen sind neue oder gelöschte Verbindungen.
/0000.04/iot.config	Spannungsversorgung (einfach oder redundant) und Temperaturüberwachung des CPU-Moduls. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Eigenschaften des Racks im Hardware-Editor.

Konfigurationsdatei	Beschreibung
/0000.04/net.config	Netzwerkeinstellungen des CPU-Moduls. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Einstellungen der CPU im Hardware-Editor in den Registern *Modul* und *Routings*. Typische Änderungen sind Umstellungen der IP-Adresse.
/0000.05/root.config	Hauptdatei des Kommunikationsmoduls in Rack 0, Slot 5.
/0000.06/root.config	Hauptdatei des E/A-Moduls in Rack 0, Slot 6.
/0000.07/root.config	Hauptdatei des E/A-Moduls in Rack 0, Slot 7. Rot: Der Versionsvergleich hat eine Änderung zwischen importierter und generierter Konfigurationsdatei festgestellt.
/0000.07/io4io.config	Konfigurationsdatei des E/A-Moduls. Rot: Der Versionsvergleich hat eine Änderung zwischen importierter und generierter Konfigurationsdatei festgestellt. Bei Änderungen ist eine Detailansicht verfügbar.
/0000.07/iot.config	Spannungsversorgung (einfach oder redundant) und Temperaturüberwachung des E/A-Moduls. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Eigenschaften des Racks im Hardware-Editor.

5.2.2 CPU-Konfiguration und Systemdaten

Zentrale und übergeordnete Daten des CPU-Moduls werden in der Konfigurationsdatei *sys/root.config* zusammengefasst.

/sys/root.config	Hauptdatei - System	16#bf430d31	V6	ok
/sys/bgp.config	System Modul	16#0fdde3b9	V4	ok
/sys/cpc.config	Systemprotokolle Basis	16#e72bf406	V2	ok
/sys/cpcnsip.config	Standardprotokoll	16#203dfbe7	V4	ok
/sys/cpcsip.config	Sicherheitsgerichtetes Protokoll	16#5d23543b	V6	ok
/sys/cpu.config	Systemdaten	16#9623c1c4	V3	ok
/sys/io4cpu.config	System IO	16#81c4bf29	V5	ok
/sys/ke.config	Datenlayout und Transport	16#1cba6c28	V5	ok
/sys/lm.config	Lizenz	16#889b1742	V2	ok

Konfigurationsdatei	Beschreibung
/sys/root.config	Hauptdatei des CPU-Moduls. Diese Konfigurationsdatei referenziert auf untergeordnete Konfigurationsdateien und ändert sich immer, wenn sich eine untergeordnete Konfigurationsdatei ändert.
/sys/bgp.config	Die Modulkonfiguration beschreibt alle Moduldaten, z. B. die Zuordnung der Module zu den Steckplätzen (Slots). Diese Datei ändert sich fast immer, wenn sich Moduldaten ändern (siehe *Hardware, Module*). Keine Detailansicht verfügbar.

Konfigurationsdatei	Beschreibung
	Wenn Änderungen angezeigt werden prüfen Sie den Bereich *Hardware*, *Module*.
/sys/cpc.config	Anzahl der Protokolle, Kommunikationszeitscheibe ASYNC, SYNC. Keine Detailansicht verfügbar. Diese Datei ändert sich, wenn sich z. B. die Anzahl existierender Protokolle oder der Parameter *Max. Kom-Zeitscheibe Async* verändert hat. Änderungen werden detailliert in anderen, ebenfalls geänderten Dateien angezeigt, z. B. *cpu.config*.
/sys/cpcnsip.config	Allgemeine Regeln der Protokolle zum Datentransport vom COM-Modul zum CPU-Modul. Keine Detailansicht verfügbar. Diese Datei ändert sich, wenn grundlegende Eigenschaften von auf dem COM-Modul existierenden, nicht sicheren Protokollen (z. B. Modbus) verändert wurden. Änderungen werden auch in anderen, ebenfalls geänderten Dateien angezeigt, z. B. *ke.config*.
/sys/cpcsip.config	safe**ethernet** Parameter, Eigenschaften von safe**ethernet** Verbindungen. Bei Änderungen sind Detailinformationen verfügbar, siehe *Neue Variable zu bestehender Verbindung hinzufügen* und *Änderungen in safeethernet Kommunikation*
/sys/cpu.config	Einstellungen der Ressource, wie z. B. erlaubte Aktionen, Sicherheitszeit, Watchdogzeit usw. Bei Änderungen sind Detailinformationen verfügbar.
/sys/io4cpu.config	Redundanzinformationen der E/A-Module, Skalierung von Analogwerten und Zählereingängen. Bei Änderungen sind Detailinformationen verfügbar, siehe *Systemebene (CPU): io4cpu.config*
ke.config	Konfigurationsdatei für die Zuordnung (Verwendung) globaler Variablen zur Hardware, zu Protokollen, zu POEs usw. (*ke* = Kommunikations-Endpunkt = globale Variable). Bei Änderungen sind Detailinformationen verfügbar, siehe *Globale Variable einem anderen Hardware-Eingang zuordnen* und *Neuer Initialwert für eine globale Variable*.
lm.config	Konfigurationsdatei für die Lizenzverwaltung. Keine Detailansicht verfügbar. Diese Datei ändert sich, wenn die Anzahl oder die Bezeichnung von Lizenzen geändert wurden. Diese Änderungen sind grundsätzlich nicht sicherheitsrelevant. Es ist möglich, dass sich die Datei ändert, obwohl keine Lizenzen verändert wurden. Die Ursache dafür liegt in geänderten internen Sortierkriterien ab SILworX V6 bei mehreren vorhandenen Lizenzen. Solche Änderungen können ignoriert werden.

5.2.3 COM-Konfiguration und Protokolle

Die Daten für ein COM-Modul (Protokolle, Schnittstellen etc.) werden in einzelnen Konfigurationsdateien unterhalb der Hauptdatei *root.config* des COM-Moduls gespeichert.

Beispiel

/0000.05/root.config	Hauptdatei - Kommunikationsmodul	16#d628c334	V4	ok
/0000.05/cpcnsip.config	Standardprotokoll	16#b83413c1	V4	ok
/0000.05/ethsw.config	Ethernet-Switch	16#016c5e67	V3	ok
/0000.05/iot.config	Spannungsversorgung und Temperatu...	16#209c5b8f	V3	ok
/0000.05/ke.config	COM-Datenlayout und Transport	16#a5a951ff	V2	ok
/0000.05/modbus.config	Modbus-Slave	16#afee3c2d	V3	ok
/0000.05/net.config	Netzwerkeinstellung	16#588b9fc4	V2	ok

Konfigurationsdatei	Beschreibung
/0000.05/root.config	Hauptdatei des COM-Moduls. Diese Konfigurationsdatei referenziert auf untergeordnete Konfigurationsdateien und ändert sich immer, wenn sich eine untergeordnete Konfigurationsdatei ändert.
/0000.05/cpcnsip.config	Allgemeine Regeln der Protokolle zum Datentransport vom COM-Modul zum CPU-Modul, z. B. Verhalten bei Verbindungsverlust. Keine Detailansicht verfügbar. Diese Datei ändert sich, wenn grundlegende Eigenschaften von auf dem COM-Modul existierenden, nicht sicheren Protokollen (z. B. Modbus) verändert wurden. Änderungen werden auch in anderen, ebenfalls geänderten Dateien angezeigt, z. B. *ke.config*.
/0000.05/ethsw.config	Eigenschaften der Ethernet-Switches des COM-Moduls. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Einstellungen der COM im Hardware-Editor in den Registern *Ethernet-Switch*, *VLAN* und *Port Mirroring*.
/0000.05/iot.config	Spannungsversorgung (einfach oder redundant) und Temperaturüberwachung des COM-Moduls. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Eingenschaften des Racks im Hardware-Editor.
/0000.05/ke.config	Konfigurationsdatei für das Lesen und Schreiben globaler Variablen in Protokollen (*ke* = Kommunikationsendpunkt = globale Variable). Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden stehen weitere Informationen in der Datei *ke.konfig* auf Systemebene zur Verfügung, siehe *Globale Variable einem anderen Hardware-Eingang zuordnen* und *Neuer Initialwert für eine globale Variable*.
/0000.05/modbus.config	Konfigurationsdatei der Eigenschaften des Modbus-Protokolls. Keine Detailansicht verfügbar. Diese Datei ändert sich, wenn grundlegende Eigenschaften des Modbus-Protokolls verändert wurden. Prüfen Sie in diesem Fall auch auf weitere geänderte

Konfigurationsdatei	Beschreibung
	Dateien, z. B. *ke.config*.
/0000.05/net.config	Konfigurationsdatei der Netzwerkeinstellungen für dieses COM-Modul. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Einstellungen des COM-Moduls im Hardware-Editor in den Registern *Modul* und *Routings*. Typische Änderungen sind Umstellungen der IP-Adresse.

5.2.4 Logikdaten

Die Prüfsummen der angezeigten Bausteine (POEs) sind sogenannte Source-Code-CRCs. Ändert sich der ausführbare Code eines Bausteins, ändert sich auch immer die Binärdatei (.ldb) des Programms, in dem der Baustein verwendet wird.

Nicht alle Änderungen in Bausteinen haben Einfluss auf den ausführbaren Code. z. B. das Ändern des Namens einer lokalen Variable. Der Codevergleich erkennt zwar eine Änderung des Source-Codes (die Zeile wird rot), aber die Binärdatei ändert sich nicht. In diesem Fall sind keine coderelevanten Auswirkungen der Änderung zu erwarten und es ist keine Nachprüfung erforderlich.

Achtung! Gefahr von ungewollten Anlagenabschaltungen!
Umbennen von Variablen, Programmnamen, FB-Namen, FB-Instanzen etc. führen bei einem Reload zur Neuinitialisierung der genannten Objekte. Gespeicherte Werte und Zustände gehen verloren!

Beispiel

27	/sys/ls.config	Logic-Solver-Konfiguration	16#f8d40ff3	V3
28	/sys/ls/01_Programm01.config	Programm-Parameter	16#ef6c15b4	V3
29	/sys/ls/01_Programm01.ldb	Programm Binärdatei	16#222df24b	V2
30	01_Programm01	Programm	16#cf62d361	V2
31	1oo2_R	Funktionsbaustein-Typ	16#665f1b8c	V2
32	2oo3	Funktionsbaustein-Typ	16#0c925672	V2
33	2von3B	Funktionsbaustein-Typ	16#0c1081a2	V2
34	Average	Funktionsbaustein-Typ	16#830d0f29	V2
35	BLINK	Funktionsbaustein-Typ	16#7c7eda91	V2
36	BUFFER	Array	16#f0d16020	V2
37	Diag_Modul	Funktionsbaustein-Typ	16#3c68356e	V2
38	Globale Variablen	Globale Variablen	16#c286681f	V2
39	LIMH_R	Funktionsbaustein-Typ	16#939cbd0b	V2
40	LIML_R	Funktionsbaustein-Typ	16#882b701d	V2
41	Modul-Diag_PES_10	Funktionsbaustein-Typ	16#19d11128	V2
42	Schrittkette-PES_10	Funktionsbaustein-Typ	16#fb5dc124	V2
43	SystemTime	Funktionsbaustein-Typ	16#b404049e	V2
44	Systemüberwachung-PES_10	Funktionsbaustein-Typ	16#579d399c	V2
45	/sys/ls/01_Programm01_force.config	Force-Daten der Applikation	16#8322f556	V2
46	/sys/ls/01_Programm01_retain.config	Retain-Daten der Applikation	16#b9ae59cc	V2
47	/sys/pgs.config	Konfigurationsverbindungen	16#bf8d0bc3	V2

Konfigurationsdatei, Objektdaten	Beschreibung
/sys/ls.config	Hauptdatei der Logik (Logic-Solver). Diese Konfigurationsdatei referenziert auf untergeordnete Konfigurationsdateien und ändert sich immer, wenn sich eine untergeordnete Konfigurationsdatei (z. B. Multitasking Eigenschaften) ändert.
/sys/ls/Programm.config	Eigenschaften des Programms, Multitasking-Einstellungen, erlaubte Aktionen usw. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Eigenschaften des Programms.
/sys/ls/Programm.ldb	Die Binärdatei (Loadable) ist der ausführbare Code der gesamten Logik und ändert sich immer bei Logikänderungen. Ab SILworX V7 sind Detailinformationen verfügbar, siehe *Logikänderungen (Logic-Solver)*.
01_Programm01 (= Name eines Programms)	CRC des Programms (als POE). Bei Änderungen sind Detailinformationen verfügbar, siehe *Logikänderungen (Logic-Solver)*.
1oo2_R (= Name eines Funktionsbausteins)	CRC des Funktionsbausteins (als POE). Bei Änderungen sind Detailinformationen verfügbar, siehe *Logikänderungen (Logic-Solver)*.
Buffer (= Name eines Datentyps)	CRC eines benutzerdefinierten Datentyps. Keine Detailansicht verfügbar. Wenn Änderungen angezeigt werden prüfen Sie die Eigenschaften des genannten Datentyps.
Globale Variablen	Eigenschaften der Verwendung von globalen Variablen in Bausteinen, z. B. Datentyp, Reihenfolge (Sortierung) usw. Beeinflusst die Änderung einer dieser Eigenschaften den ausführbaren Code, ändert sich auch die Binärdatei (.ldb). In diesem Fall gibt es weitere Informationen in der Detailansicht des betroffenen Bausteins. Falls zutreffend werden auch Änderungen in der *ke.config* angezeigt. Dort gibt es ebenfalls Detailinformationen.
/sys/ls/force.config	Zusatzinformationen zur Unterstützung des Forcens in der Logik. Ändert sich möglicherweise ebenfalls, wenn sich die Verwendung einer globalen Variable in der Logik ändert, siehe *Globale Variable einem anderen Hardware-Eingang zuordnen* und *Neuer Initialwert für eine globale Variable* Falls zutreffend werden auch Änderungen in der *ke.config* angezeigt. Dort gibt es ebenfalls Detailinformationen.
/sys/ls/retain.config	Retain-Informationen der in der Logik verwendeten globalen Variablen. Ändert sich möglicherweise ebenfalls, wenn sich die Verwendung einer globalen Variable in der Logik ändert, siehe *Globale Variable einem anderen Hardware-Eingang zuordnen* und *Neuer Initialwert für eine globale Variable*. Falls zutreffend werden auch Änderungen in der *ke.config* angezeigt. Dort gibt es ebenfalls Detailinformationen.

5.2.5 PGS-Daten (Konfigurationsverbindungen, Benutzermanagement)

Die Daten für das Baugruppenprotokoll (BGP), die Remote-I/O-Verbindungen und das Benutzermanagement werden in der Konfigurationsdatei *pgs.config* gespeichert.

35	Modul-Diag_PES_10	Funktionsbaustein-Typ	16#19d11128	V2
36	Schrittkette-PES_10	Funktionsbaustein-Typ	16#fb5dc124	V2
37	SystemTime	Funktionsbaustein-Typ	16#b404049e	V2
38	Systemüberwachung-P...	Funktionsbaustein-Typ	16#579d399c	V2
39	/sys/ls/01_Programm01_force.c...	Force-Daten der Applikation	16#8322f556	V2
40	/sys/ls/01_Programm01_retain.c...	Retain-Daten der Applikation	16#b9ae59cc	V2
41	/sys/pgs.config	Konfigurationsverbindungen	16#bf8d0bc3	V2

Konfigurationsdatei	Beschreibung
/sys/pgs.config	Daten zu den Konfigurationsverbindungen, z. B. *Max.Dauer Konfigurationsverbindungen*, *PES Benutzermangement*. Keine Detailansicht verfügbar. Eine angezeigte Änderung kann folgende Ursachen haben: • Die Einstellung *Max.Dauer Konfigurationsverbindungen* (Resourceeigenschaft) wurde verändert. • Die Einstellungen des PES Benutzermanagements (wenn vorhanden) wurden verändert. • Die Einstellungen von Remote-I/O-Verbindungen (wenn vorhanden) wurden verändert.

5.2.6 Erforderliche Betriebssystem-Version eines Objekts

Für jede Konfigurationsdatei wird neben der Prüfsumme auch die Betriebssystem-Version angezeigt. Die erforderliche Betriebssystem-Version eines Moduls ist abhängig von den verwendeten Funktionen. So kann beispielsweise die *Max. Dauer Konfigurationsverbindungen* erst ab Betriebssystem-Version V4 geändert werden, siehe Beispiel unten.

Bereits nach der Codegenerierung wird im Logbuch neben dem CRC auch die höchste irgendwo benötigte Betriebssystem-Version ausgegeben.

Info	Codegenerierung beendet. Warnungen: 0, Fehler: 0, CRC: 16#8183e67a-V7.

Im Versionsvergleich können Sie der Zeile */root.config, Hauptdatei der Konfiguration*, ebenfalls die höchste irgendwo benötigte Betriebssystem-Version entnehmen.

> Eine Konfiguration kann nur dann in die Steuerung geladen werden, wenn alle verwendeten Module mindestens über die vom Codegenerator ermittelte Betriebssystem-Versionen verfügen. Module mit unpassenden Betriebssystem-Versionen lehnen die Konfiguration als ungültig ab.

Bei allen Dateien, die nicht nach SRS (System-Rack-Slot, z. B. /0000.03/ = Rack0, Slot3) einem einzelnen Modul zugeordnet werden können, wird diese Funktion von der CPU ausgeführt. In diesem Fall muss die CPU über die geforderte Betriebssystem-Version verfügen.

Klicken Sie in der hierarchischen Auflistung der Konfigurationsdateien auf das (+)-Zeichen rechts neben der Zeilennummer, um die untergeordneten Elemente anzuzeigen.

Versionsvergleich: CG PES_10 [10]

	Name	Beschreibung	CRC CG	Version CG	Vergleich CRCs
1	/root.config	Hauptdatei der Konfiguration	16#8183e67a	V7	ok
2	/0000.01/root.config	Hauptdatei - Systembusmodul	16#b52cf39c	V3	ok
3	/0000.02/root.config	Hauptdatei - Systembusmodul	16#40ce99e0	V3	ok
4	/0000.03/root.config	Hauptdatei - CPU	16#f1df5d79	V3	ok
5	/0000.04/root.config	Hauptdatei - CPU	16#9c6d4236	V3	ok
6	/0000.05/root.config	Hauptdatei - Kommunikationsmodul	16#3a3b0adc	V3	ok
7	/0000.06/root.config	Hauptdatei - E/A	16#248607c0	V3	ok
8	/0000.07/root.config	Hauptdatei - E/A	16#28566e08	V3	ok
9	/0000.08/root.config	Hauptdatei - E/A	16#34bd9da7	V3	ok
10	/0000.09/root.config	Hauptdatei - E/A	16#205dcdb0	V3	ok
11	/0000.10/root.config	Hauptdatei - E/A	16#cd0ec990	V3	ok
12	/sys/root.config	Hauptdatei - System	16#2c95ee51	V7	ok
13	/sys/bgp.config	System Modul	16#c1a3c686	V4	ok
14	/sys/cpc.config	Systemprotokolle Basis	16#da613dbe	V2	ok
15	/sys/cpu.config	Systemdaten	16#23eb0339	V7	ok
16	/sys/io4cpu.config	System IO	16#81c4bf29	V5	ok
17	/sys/ke.config	Datenlayout und Transport	16#e75e75d5	V5	ok

6 Detailauswertung

6.1 Hardware-Änderungen (im Hardware-Editor)

6.1.1 E/A-Module: io4io.config

Die Konfigurationsdatei eines E/A-Moduls *io4io.config* ändert sich, wenn sich die Konfigurationsdaten des E/A-Moduls ändern. Dazu gehören:

- Änderungen im Register *Modul*, z. B. Störaustastung.
- Änderungen im Register *IO-Submodul*, z. B. Aktivierung einer Speisung.
- Änderungen von fest eingestellten Werten, z. B. *LB Limit* oder *Skalierungswerte*.

> Beachten Sie, dass sich die Konfigurationsdaten nicht zwangsläufig ändern, wenn Sie Eigenschaften eines Moduls (z. B. Kanalwert) einer globalen Variable zuordnen.

Durch das Ändern, Hinzufügen und Löschen einer globalen Variable ändert sich bei folgenden Modulen die Konfigurationsdatei, da implizit die Information *Kanal verwendet* verwaltet wird:

• X-DI 32 02 • X-DI 32 05	• X-AI 32 01 • X-AI 32 02	• X-CI 24 01

Beispiel

7	/0000.06/root.config	Hauptdatei - E/A	16#17d23555	V3
8	/0000.07/root.config	Hauptdatei - E/A	16#7847c4e6	V3
9	/0000.07/io4io.config	E/A	16#d66d29d9	V2
10	/0000.07/iot.config	Spannungsversorgung und Temperatu...	16#209c5b8f	V3
11	/0000.06/root.config	Hauptdatei - E/A	16#f1539d17	V3

Durch Doppelklicken in die Zeile der Konfigurationsdatei *io4io.config* (Zeile 9 im Bild oben) wird die Detailansicht geöffnet.

	Steckplatz	Kanal	Einstellung	Version IM	Version CG	
1	10.0.7	1	LS-Limit	65500	80000	Geändert
2	10.0.7	7	Kanal verwendet	nein	ja	Geändert

In der Detailansicht werden folgende Informationen angezeigt:

Spalte	Beschreibung
Steckplatz	Steckplatz des E/A-Moduls im Format System.Rack.Slot.
Kanal	Betroffener Kanal des E/A-Moduls.
Einstellung	Bedeutung des Parameters oder der Funktion.
Version IM	Wert in der importierten Version der Konfigurationsdatei.
Version CG	Wert in der vom Codegenerator erzeugten Version der Konfigurationsdatei.

6.1.2 Systemebene (CPU): io4cpu.config

Es gibt Änderungen, bei denen sich zusätzlich zur Konfigurationsdatei io4io.config auch die Konfigurationsdatei *io4cpu.config* ändert. Dies ist beispielsweise der Fall, wenn die Skalierungswerte für analoge Eingangsmodule geändert werden.

Da das CPU-Modul die E/A-Module bei der Berechnung der Skalierung unterstützt, wirken sich Änderungen an Skalierungswerten sowohl in der *io4io.config*, als auch in der *io4cpu.config* aus (siehe nachfolgende Abbildung).

Beispiel

13	/0000.10/root.config	Hauptdatei - E/A	16#fa7051b0	V3	16#cc56a43d	V3	-
14	/0000.10/io4io.config	E/A	16#b602116e	V2	16#0f7b4b73	V2	-
15	/0000.10/iot.config	Spannungsversorgung und Temperatu...	16#209c5b8f	V3	16#209c5b8f	V3	ok
16	/sys/root.config	Hauptdatei - System	16#e937d550	V4	16#cac13212	V4	-
17	/sys/bgp.config	System Modul	16#1072a538	V2	16#07b4fb56	V2	-
18	/sys/cpc.config	Systemprotokolle Basis	16#ced2a001	V2	16#ced2a001	V2	ok
19	/sys/cpcnsip.config	Standardprotokoll	16#64ecf542	V3	16#64ecf542	V3	ok
20	/sys/cpcsip.config	Sicherheitsgerichtetes Protokoll	16#9855ad19	V2	16#9855ad19	V2	ok
21	/sys/cpu.config	Systemdaten	16#03161291	V3	16#03161291	V3	ok
22	/sys/io4cpu.config	System IO	16#5e571427	V3	16#16be65ea	V3	-
23	/sys/ke.config	Datenlayout- und Transport	16#65017dc0	V2	16#65017dc0	V2	ok

Durch Doppelklicken in die Zeile der Konfigurationsdatei *io4cpu.config* (Zeile 22 im Bild oben) wird die Detailansicht geöffnet.

In der nachfolgenden Abbildung sind die beiden Module in den Steckplätzen 9 und 10 redundant, so dass beide Module von der Änderung betroffen sind.

Versionsvergleich: IM <- CG PES_10 [10] /sys/io4cpu.config

Schließen

	Steckplatz	Kanal	Einstellung	Version IM	Version CG	
1	10.0.9	1	4 mA	100.0	110.00000000000364	Geändert
2	10.0.9	1	20 mA	100000.0	100000.00000000001	Geändert
3	10.0.10	1	4 mA	100.0	110.00000000000364	Geändert
4	10.0.10	1	20 mA	100000.0	100000.00000000001	Geändert

Bei der Darstellung von REAL-Zahlen (siehe Spalte *Version CG*) können Nachkommastellen vorkommen, die aus dem internen Aufbau mit Mantisse und Exponent resultieren. Die Nachkommastellen können ignoriert werden!
Auch wenn nur ein Parameter geändert wurde, werden meist beide Stützpunkte *4 mA* und *20 mA* durch die zugrunde liegende mathematische Funktion beeinflusst.

Die nachfolgende Abbildung zeigt ein Modul X-DI 32 01 in einer Redundanzgruppe. Da die Redundanzauswertung im CPU-Modul erfolgt, hat dies ebenfalls Auswirkungen auf die *io4cpu.config*.

Versionsvergleich: IM <- CG PES_10 [10] /sys/io4cpu.config

Schließen

	Steckplatz	Kanal	Einstellung	Version IM	Version CG	
1	10.0.6/2.6		Redundanz-Gruppe			Hinzugefügt
2	10.2.6		Modul		X-DI 32 01	Hinzugefügt

6.1.3 Neues Modul hinzufügen

Wird einem System ein neues Modul hinzugefügt, wird die Konfigurationsdatei des neuen Moduls im Versionsvergleich gelb hinterlegt.

Zeile 12 der nachfolgenden Abbildung ist zu entnehmen, dass ein neues Modul in Rack 2, Steckplatz 5 hinzugefügt wurde. Dies hat auch Auswirkungen auf die Modulverwaltung */sys/bgp.config*. Die Konfiguration des Moduls wird in den Dateien */sys/io4cpu.config* und */sys/ke*.config gespeichert.

Das Modul muss mit einer Betriebssystem-Version ab V3 ausgestattet sein, da das Rack für Temperaturüberwachung konfiguriert wurde (Zeile 14).

11	/0000.10/root.config	Hauptdatei - E/A	16#fa7051b0	V3	16#fa7051b0	V3	ok
12	/0002.05/root.config	Hauptdatei - E/A	16#00000000		16#f080ff59	V3	--
13	/0002.05/io4io.config	E/A	16#00000000		16#f03c1e0f	V2	--
14	/0002.05/iot.config	Spannungsversorgung und Tem..	16#00000000		16#209c5b8f	V3	--
15	/sys/root.config	Hauptdatei - System	16#e937d550	V4	16#2a159ea3	V4	-
16	/sys/bgp.config	System Modul	16#1072a538	V2	16#4f4ce4b9	V2	-
17	/sys/cpc.config	Systemprotokolle Basis	16#ced2a001	V2	16#ced2a001	V2	ok
18	/sys/cpcnsip.config	Standardprotokoll	16#64ecf542	V3	16#64ecf542	V3	ok
19	/sys/cpcsip.config	Sicherheitsgerichtetes Protokoll	16#9855ad19	V2	16#9855ad19	V2	ok
20	/sys/cpu.config	Systemdaten	16#03161291	V3	16#03161291	V3	ok
21	/sys/io4cpu.config	System IO	16#5e571427	V3	16#8b1a9e1c	V3	-
22	/sys/ke.config	Datenlayout- und Transport	16#65017dc0	V2	16#34bb4dad	V2	-
23	/sys/lm.config	Lizenz	16#889b1742	V2	16#889b1742	V2	ok

Durch Doppelklicken in die Zeile */sys/io4cpu.config* (Zeile 21 im Bild oben) werden weitere Details zum hinzugefügten Modul angezeigt.

Versionsvergleich: IM <- CG PES_10 [10] /sys/io4cpu.config

Schließen

	Steckplatz	Kanal	Einstellung	Version IM	Version CG	
1	10.2.5		Modul		X-DI 32 01	Hinzugefügt

6.2 Logikänderungen (Logic-Solver)

Bei Änderungen in der Logik ändert sich mindestens eine Konfigurationsdatei unterhalb der Logic-Solver-Konfiguration */sys/ls.config*. Außerdem hat eine Änderung Auswirkungen auf die Programm-Binärdatei *Programm.ldb*.

Bei allen POE wird der Quellcode angezeigt. Der Quellcode wird bei der Codegenerierung in einen ausführbaren Code gewandelt. Nur wenn sich der ausführbare Code ändert (= funktionale Änderung), ändert sich auch die Programm-Binärdatei.

/sys/ls.config	Logic-Solver-Konfiguration	16#52a3f499	V3	16#3cc4b9f2	V3	-
/sys/ls/21_Program_PES_10.config	Programm-Parameter	16#ed8b2329	V3	16#7813f844	V3	-
/sys/ls/21_Program_PES_10.ldb	Programm Binärdatei	16#90ba99c0	V2	16#bf165f3b	V2	-
2oo3	Funktionsbaustein-Typ FBS	16#0c925672	V2	16#0c925672	V2	ok
2oo3B	Funktionsbaustein-Typ FBS	16#2a457e09	V2	16#2a457e09	V2	ok
21_Program_PES_10	Programm	16#10a20dca	V2	16#10a20dca	V2	ok
Average	Funktionsbaustein-Typ FBS	16#020731ca	V2	16#020731ca	V2	ok

Durch Doppelklicken in die Zeile einer geänderten POE öffnet sich die Detailansicht. In mindestens einem Register werden Änderungen angezeigt. Prüfen Sie alle Register auf Änderungen! Erklärungen zur Auswertung finden Sie in den nachfolgenden Beispielen.

Versionsvergleich: DL <- CG PES_10 [10] BLINK

Schließen

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

Name	Typ	Position DL	Position CG	Abarbeitungsreihenfolge DL	Abarbeitungsreihenfolge CG	Änderung
Diese Ansicht ist leer.						

6.2.1 Wertfeld am Eingang eines Funktionsbausteins ändern

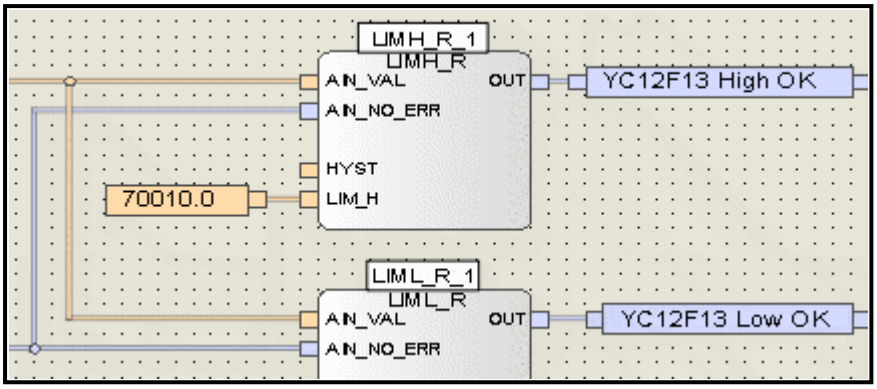
In der Abbildung unten weist die Anmerkung *Geändert* im Register *POE Änderungen* darauf hin, dass sich Eingangsinformationen der Funktionsbaustein-Instanz *LIMH_R_1* geändert haben. Die *Position* ist die linke obere Ecke der Funktionsbaustein-Instanz auf dem Arbeitsblatt.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarbeitung IM	Abarbeitung CG	
1	LIMH_R_1	Instanz	Blatt: 0/1, Pos.: 100/13	Blatt: 0/1, Pos.: 100/13	15	15	Geändert

Durch Doppelklicken in die Zeile *LIMH_R_1* wird der FBS-Editor geöffnet und die Logik auf die geänderte POE zentriert.

Details können durch einen direkten Vergleich der Logikseite (IM) mit der Logikseite (CG) erkannt werden, z. B durch Auswertung einer entsprechenden POE-Dokumentation. Im Beispiel unten wurde das Wertfeld auf `70010.0` geändert.



6.2.2 Neues Objekt in Logik einfügen

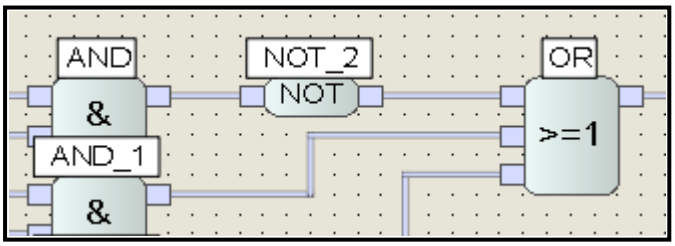
Im Register *POE Änderungen* weisen die leeren Spalten *Position IM* und *Abarbeitungsreihenfolge IM* darauf hin, dass das Element *NOT_2* in der importierten Konfiguration nicht existierte. In der zuletzt generierten Konfiguration sind die Informationen für *Position CG* und *Abarbeitungsreihenfolge CG* vorhanden. Zusätzlich wird die POE als *Neu* gekennzeichnet.

Für das Element *OR* haben sich die Eingangsinformationen geändert.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarbeitung IM	Abarbeitung CG	
1	NOT_2	Instanz	Blatt: -/-, Pos.: -/-	Blatt: 0/0, Pos.: 46/10	-	3	Neu
2	OR	Instanz	Blatt: 0/0, Pos.: 57/10	Blatt: 0/0, Pos.: 57/10	5	6	Geändert

Durch Doppelklicken in die Zeile *NOT_2* wird der Programmeditor geöffnet und die Logik auf die geänderte POE zentriert.



In der Logik wurde die POE-Instanz *NOT_2* zusätzlich eingefügt.

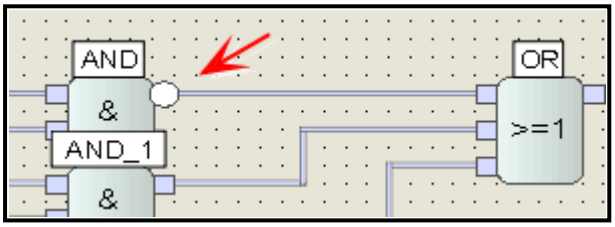
6.2.3 Ausgang einer Funktion invertieren

Im Register *POE Änderungen* weist die Anmerkung *Geändert* darauf hin, dass sich an einem der Eingänge des Elements *OR* Logikänderungen ergeben haben.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarbeitung IM	Abarbeitung CG	
1	OR	Instanz	Blatt: 0/0, Pos.: 57/10	Blatt: 0/0, Pos.: 57/10	5	5	Geändert

Durch Doppelklicken in die Zeile *OR* wird der FBS-Editor geöffnet und die Logik auf die geänderte POE zentriert.



Das Signal des ersten Eingangs der OR-Instanz wurde invertiert.

> Angezeigte Änderungen beziehen sich immer nur auf die Eingangsverarbeitung von Instanzen von Funktionen und Funktionsbausteinen.
> Die Funktion AND wurde hingegen nicht geändert, da die Invertierung nach der Ausführung des AND durchgeführt wird.

6.2.4 POE in der Logik löschen

Im Register *POE Änderungen* weisen die leeren Spalten *Position CG* und *Abarbeitungsreihenfolge CG* darauf hin, dass die POE-Instanzen *GE_1* und *TON* in der zuletzt generierten Konfiguration des Projekts nicht existieren. In der importierten Konfiguration sind die Informationen für *Position IM* und *Abarbeitungsreihenfolge IM* vorhanden. Zusätzlich werden die POE-Instanzen als *Gelöscht* gekennzeichnet.

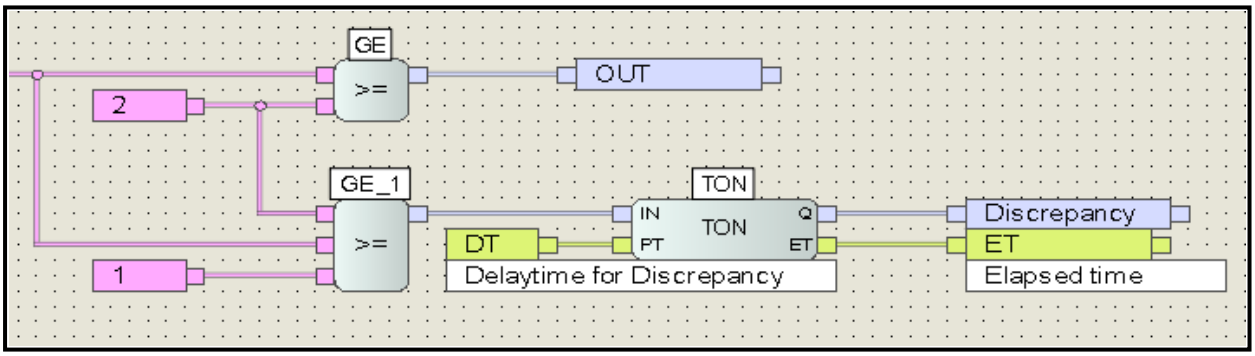
POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarbeitung IM	Abarbeitung CG	
1	GE_1	Instanz	Blatt: 0/0, Pos.: 82/22	Blatt: -/-, Pos.: -/-	5	-	Gelöscht
2	TON	Instanz	Blatt: 0/0, Pos.: 98/22	Blatt: -/-, Pos.: -/-	6	-	Gelöscht

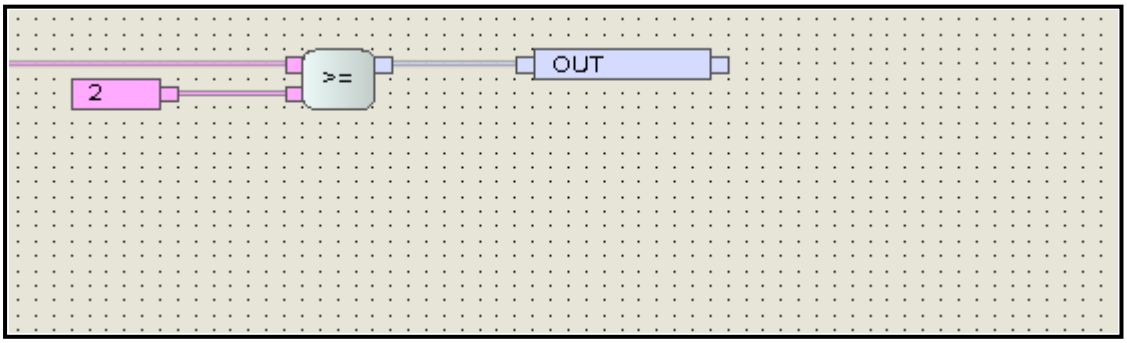
Durch Doppelklicken in eine der Zeilen wird der FBS-Editor geöffnet.

> Die Logik kann nicht auf gelöschte POE-Instanzen zentriert werden. Benutzen Sie die angezeigten Positionen, um die Stellen zu finden, an denen Logikelemente gelöscht wurden.

Teil der Logik der importierten Konfiguration:



Teil der modifizierten Logik der zuletzt generierten Konfiguration:



6.2.5 Netzwerk verschieben (Abarbeitungsreihenfolge ändern)

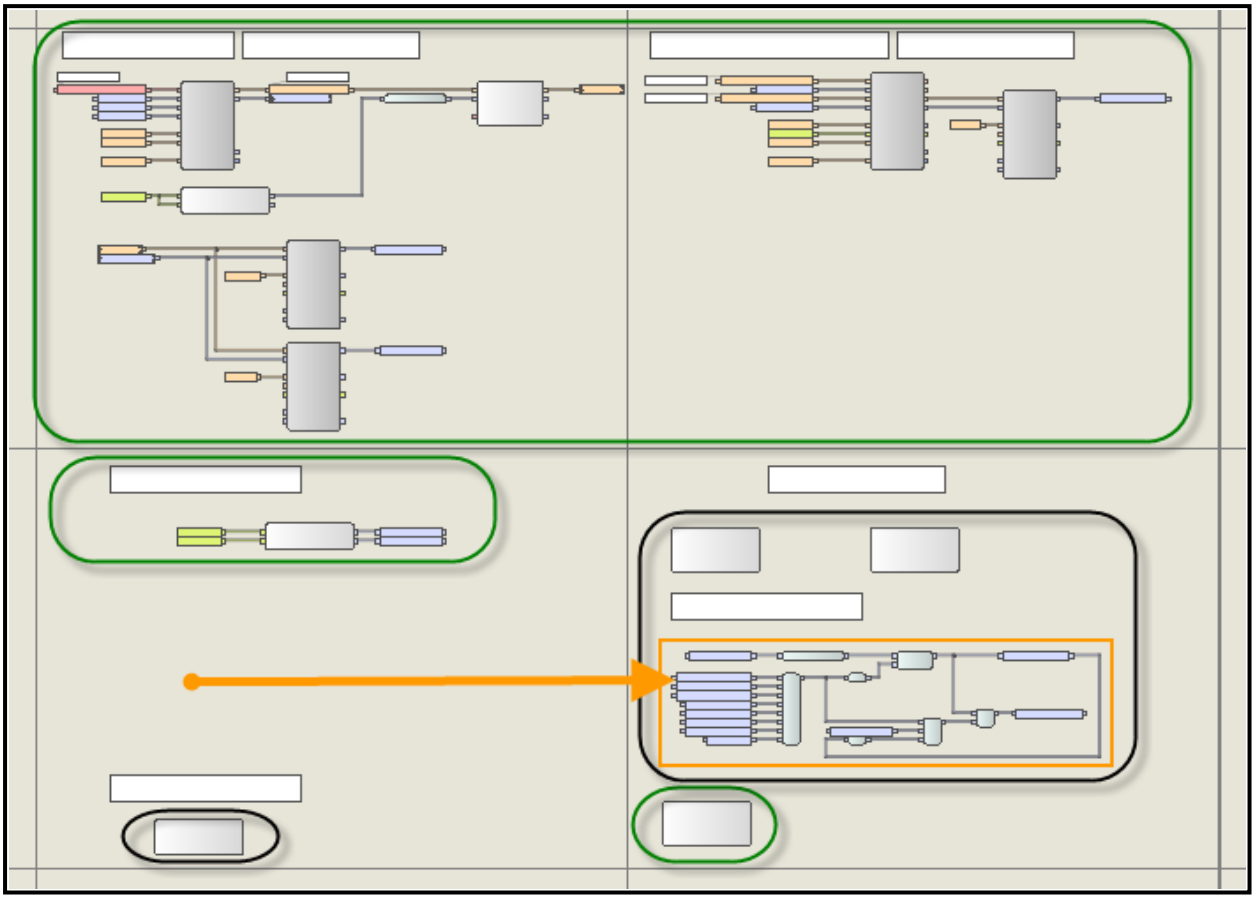
Im Register *POE Abarbeitungsreihenfolge* weisen die Informationen in den Spalten *Abarbeitungsreihenfolge IM* und *Abarbeitungsreihenfolge CG* darauf hin, dass POEs ihre Abarbeitungsreihenfolge geändert haben. Alle anderen Instanzen haben ihre Position nicht verändert.

18	Average	Instanz	18	18
19	YC12F13 High	Instanz	19	19
20	YC12F13 Low	Instanz	20	20
21	X_1oo2_R_1	Instanz	21	21
22	X_LimH	Instanz	22	22
23	BLINK	Instanz	23	23
24	Step_Sequence_PES_10	Instanz	Verschoben von 31	24
25	GOTO1	Instanz	Verschoben von 32	25
26	GOTO2	Instanz	Verschoben von 33	26
27	R_TRIG	Instanz	24	27
28	AND_3	Instanz	25	28
29	NOT	Instanz	26	29
30	RS	Instanz	27	30
31	NOT_1	Instanz	28	31
32	AND_4	Instanz	29	32
33	OR_1	Instanz	30	33
34	Step_Sequence_PES_10	Instanz	31	Verschoben nach 24
35	GOTO1	Instanz	32	Verschoben nach 25
36	GOTO2	Instanz	33	Verschoben nach 26
37	GOTO3	Instanz	34	34

SILworX versucht, die Anzahl der als *Verschoben* gekennzeichneten Instanzen möglichst klein zu halten. Daraus resultiert, dass die als verschoben angezeigten Instanzen nicht unbedingt diejenigen sind, die tatsächlich bewegt wurden.

- Für die im Bild oben grün markierten Objekte hat sich die Abarbeitungsreihenfolge nicht geändert.
- Für die im Bild oben schwarz markierten Objekte hat sich die Abarbeitungsreihenfolge geändert.
- Die tatsächlich bewegten Objekte sind orange markiert.

Tatsächlich verschobene Logik für das obige Beispiel



Die Konsequenzen der veränderten Abarbeitungsreihenfolgen müssen individuell geprüft werden. Insbesondere muss auf geänderte Schreibe-/Lese-Sequenzen von Variablen geachtet werden.
Eine veränderte Abarbeitungsreihenfolge kann dazu führen, dass der neue Wert einer beschrieben Variable beim Lesen der Variable erst im folgenden Zyklus erkannt wird. Dies kann Auswirkungen auf die Reaktionszeiten der SIF (Safety Instrumented Function) haben.

Für die als *Verschoben* angezeigten Instanzen muss die Untersuchung für alle diejenigen Variablen durchgeführt werden, welche von anderen Logikteilen gelesen oder geschrieben werden. Nutzen Sie dafür die *Querverweise* in SILworX.

Die Untersuchung darf sich auf den veränderten Bereich beschränken (im Beispiel unten schwarz markiert). Achten Sie insbesondere auf Logikverbindungen. Die Funktionen müssen komplett geprüft werden.

Variablen dieser Logik mit ausschließlichen Querverweisen zur Hardware (I/O, System) oder Kommunikation (Modbus, safe**ethernet** etc.) ändern ihr Verhalten durch eine andere Abarbeitungsreihenfolge nicht.

Die als *Verschoben* angezeigten Instanzen umschließen den zu untersuchenden Bereich wie einen Rahmen, wie die nachfolgenden Beispiele zeigen.

12	2oo3B CH789	Instanz	12	12
13	2oo3	Instanz	13	13
14	OR_2	Instanz	14	14
15	BLINK	Instanz	Verschoben von 23	15
16	X_Hx_AI	Instanz	15	16
17	BLINK_1	Instanz	16	17
18	R_TRIG_1	Instanz	17	18
19	Average	Instanz	18	19
20	YC12F13 High	Instanz	19	20
21	YC12F13 Low	Instanz	20	21
22	X_1oo2_R_1	Instanz	21	22
23	X_LimH	Instanz	22	23
24	BLINK	Instanz	23	Verschoben nach 15
25	R_TRIG	Instanz	24	24
26	AND_3	Instanz	25	25
27	NOT	Instanz	26	26

10	TON	Instanz	10	10
11	2oo3B_CH456	Instanz	11	11
12	2oo3B CH789	Instanz	12	12
13	2oo3	Instanz	13	13
14	OR_2	Instanz	14	14
15	X_1oo2_R_1	Instanz	Verschoben von 21	15
16	X_LimH	Instanz	Verschoben von 22	16
17	X_Hx_AI	Instanz	15	17
18	BLINK_1	Instanz	16	18
19	R_TRIG_1	Instanz	17	19
20	Average	Instanz	18	20
21	YC12F13 High	Instanz	19	21
22	YC12F13 Low	Instanz	20	22
23	X_1oo2_R_1	Instanz	21	Verschoben nach 15
24	X_LimH	Instanz	22	Verschoben nach 16
25	BLINK	Instanz	23	23
26	R_TRIG	Instanz	24	24
27	AND_3	Instanz	25	25

6.2.6 Lokale Variable ändern (Neu, löschen, Initialwert)

Im Bild unten weisen im Register *Lokale Variablen* die Informationen in der Spalte *Änderung* auf folgende Modifikationen hin:

- Die Variable *NEUE VAR* wurde hinzugefügt.
- Die Initialwerte der Variablen *SW-Tol-Zeit* und *SW-Tol-Zeit-C* wurden geändert (siehe Spalten *Wert IM* und *Wert CG*).
- Die Variable *Var_1* wurde gelöscht.

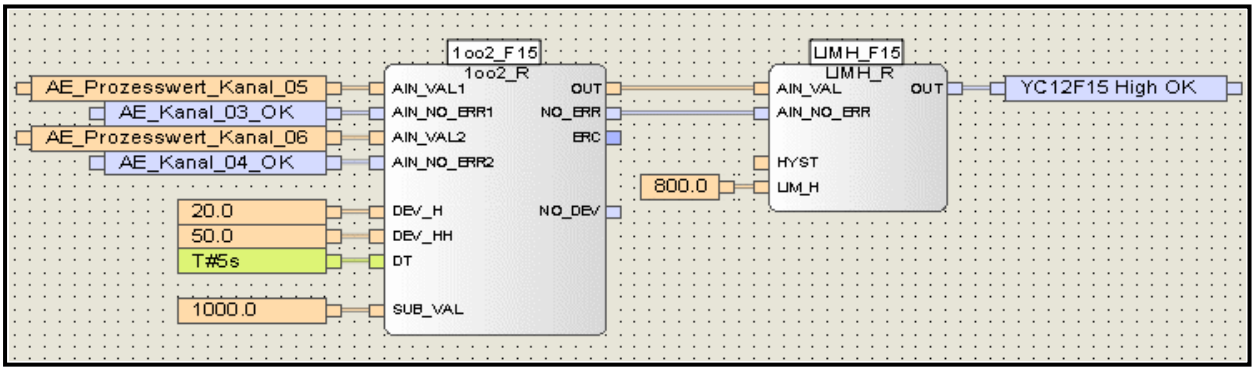
Durch Doppelklicken in eine Zeile wird der FBS-Editor geöffnet und die Logik auf die geänderte Variable zentriert (gilt nicht für gelöschte Variablen).

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Eigenschaft	Wert IM	Wert CG	Änderung
1	NEUE VAR				Neu
2	SW-Tol-Zeit	Initialwert	T#10s	T#12s	Geändert
3	SW-Tol-Zeit-C	Initialwert	T#10s	T#12s	Geändert
4	Var_1				Gelöscht

6.2.7 Neues Netzwerk erstellen

Eine bestehende Logik wird um folgende Programmierung ergänzt und danach neu kompiliert (CG):



Im Register *POE Änderungen* weisen die leeren Spalten *Position IM* und *Abarbeitungsreihenfolge IM* darauf hin, dass die POE-Instanzen *1002_F15* und *LIMH_15* in der importierten Konfiguration nicht existieren. In der neuen, zuletzt generierten Konfiguration sind die Informationen für *Position CG* und *Abarbeitungsreihenfolge CG* vorhanden. Zusätzlich werden die POE-Instanzen als *Neu* gekennzeichnet.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarbeitung IM	Abarbeitung CG	
1	1oo2_F15	Instanz	Blatt: -/-, Pos.: -/-	Blatt: 0/2, Pos.: 61/17	-	28	Neu
2	LIMH_F15	Instanz	Blatt: -/-, Pos.: -/-	Blatt: 0/2, Pos.: 98/17	-	29	Neu

Im Register *Lokale Variablen* weisen die Informationen in der Spalte *Änderung* darauf hin, dass drei lokale Variablen neu erstellt wurden.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Eigenschaft	Wert IM	Wert CG	Änderung
1	AE_Prozesswert_Kanal_05				Neu
2	AE_Prozesswert_Kanal_06				Neu
3	YC12F15 High OK				Neu

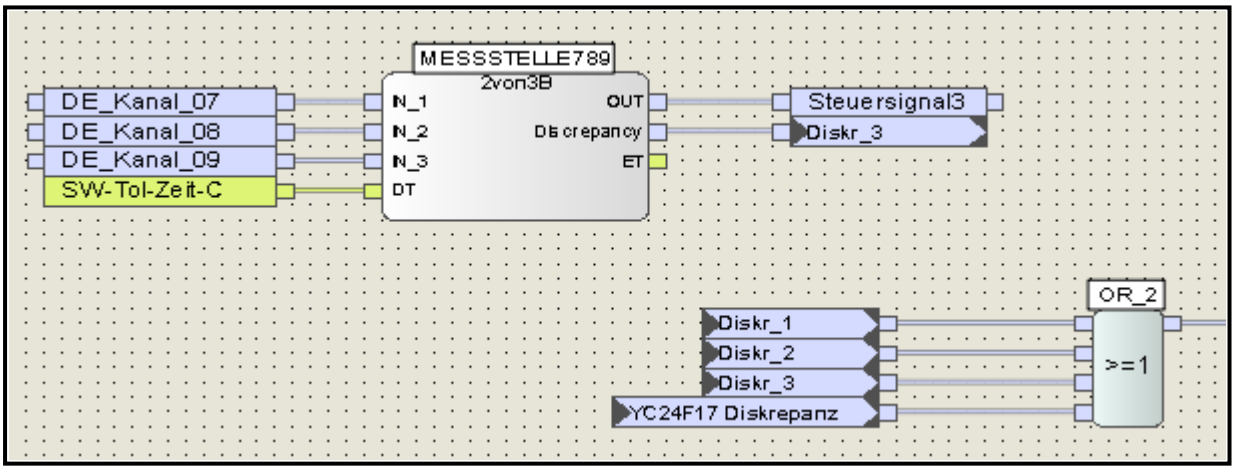
Durch Doppelklicken in eine der Zeilen wird der FBS-Editor geöffnet und die Logik auf die gewählte POE zentriert.

6.2.8 Funktionsbaustein-Instanz umbenennen

Das Umbenennen einer Instanz wird wie das Löschen und Einfügen einer neuen Instanz behandelt. Der alte Instanzname wird gelöscht und der neue Instanzname wird hinzugefügt.

Achtung! Gefahr von ungewollten Anlagenabschaltungen!
Das Umbennen von Funktionsbaustein-Instanzen führt bei einem Reload zur Neuinitialisierung aller internen Daten des Funktionsbausteins. Gespeicherte Werte und Zustände gehen verloren!

Die POE-Instanz *2von3B_1* wurde umbenannt in *MESSSTELLE789*.



Im Register *POE Änderungen* weisen die leeren Spalten *Position IM* und *Position CG*, sowie die zusätzlichen Informationen *Gelöscht/Neu* auf die durchgeführten Änderungen hin.

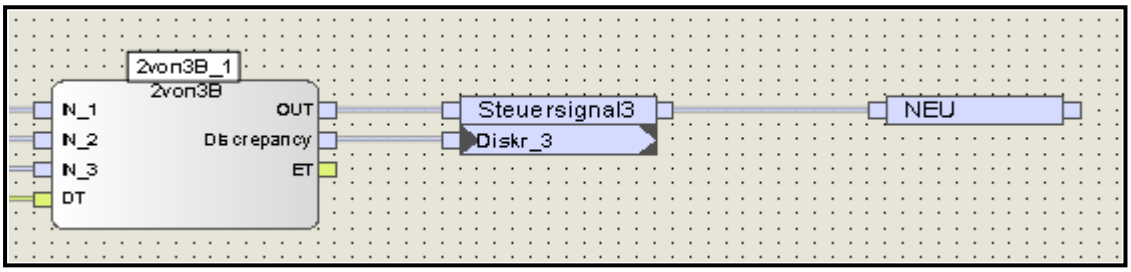
POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarbeitung IM	Abarbeitung CG	
1	2von3B_1	Instanz	Blatt: 0/0, Pos.: 47/67	Blatt: -/-, Pos.: -/-	9	-	Gelöscht
2	MESSSTELLE789	Instanz	Blatt: -/-, Pos.: -/-	Blatt: 0/0, Pos.: 47/67	-	9	Neu
3	OR_2	Instanz	Blatt: 0/0, Pos.: 101/83	Blatt: 0/0, Pos.: 101/83	11	11	Geändert

- Die POE-Instanz *2von3B_1* wurde gelöscht.
- Eine neue POE-Instanz *MESSSTELLE789* wurde erstellt.
- Die neue Instanz befindet sich an der gleichen Position wie die gelöschte Instanz und die Abarbeitungsreihenfolge ist identisch.
- Die Instanz *OR_2* wird als *Geändert* klassifiziert, da der Konnektor *Diskr_3* mit der neuen POE-Instanz *MESSSTELLE789* verbunden ist.

6.2.9 Globale Variable neu zuweisen

In der zuletzt generierten Konfiguration wurde die globale Variable *NEU* eingefügt und mit der Logik wie gezeigt verbunden.



Im Register *POE Änderungen* wird die POE-Instanz *2von3B_1* als *Geändert* gekennzeichnet. Da Variablen keine Instanznamen haben, wird die Instanz angezeigt, welche die Variable beschreibt.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarbeitung IM	Abarbeitung CG	
1	2von3B_1	Instanz	Blatt: 0/0, Pos.: 47/67	Blatt: 0/0, Pos.: 47/67	9	9	Geändert

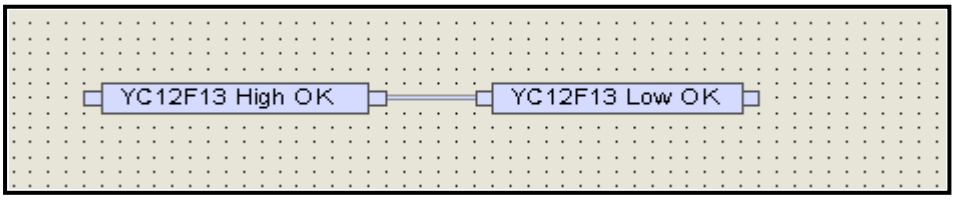
Im Register *Lokale Variablen* wird die Variable *NEU* als *Neu* gekennzeichnet.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Eigenschaft	Wert IM	Wert CG	Änderung
1	NEU				Neu

6.2.10 Neue Variablenzuweisung einfügen

In der zuletzt generierten Konfiguration wurde die unten dargestellte Änderung durchgeführt. Die Variable *YC12F13 Low OK* wurde eingefügt, welcher der Wert der Variable *YC12F13 High OK* zugewiesen wurde.



- Im Register *POE Änderungen* wird der Typ der Änderung als *Zuweisung* klassifiziert.
- In der Spalte *Name* ist erkennbar, in welcher Richtung die Zuweisung erfolgt.
- Die leeren Spalten *Position IM* und *Abarbeitungsreihenfolge IM* weisen darauf hin, dass die Zuweisung in der importierten Konfiguration (IM) nicht existierte.
- In der neuen, zuletzt generierten Konfiguration (CG) sind die Informationen für *Position CG* und *Abarbeitungsreihenfolge CG* vorhanden. Zusätzlich wird die Zuweisung als *Neu* gekennzeichnet.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarbeitung IM	Abarbeitung CG	
1	YC12F13 High OK => YC12F13 Low OK	Zuweisung	Blatt: -/-, Pos.: -/-	Blatt: 1/0, Pos.: 8/86	-	12	Neu

Im Register *POE Abarbeitungsreihenfolge* ist erkennbar, dass die Zuweisung zu einer Veränderung der Abarbeitungsreihenfolge führt.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

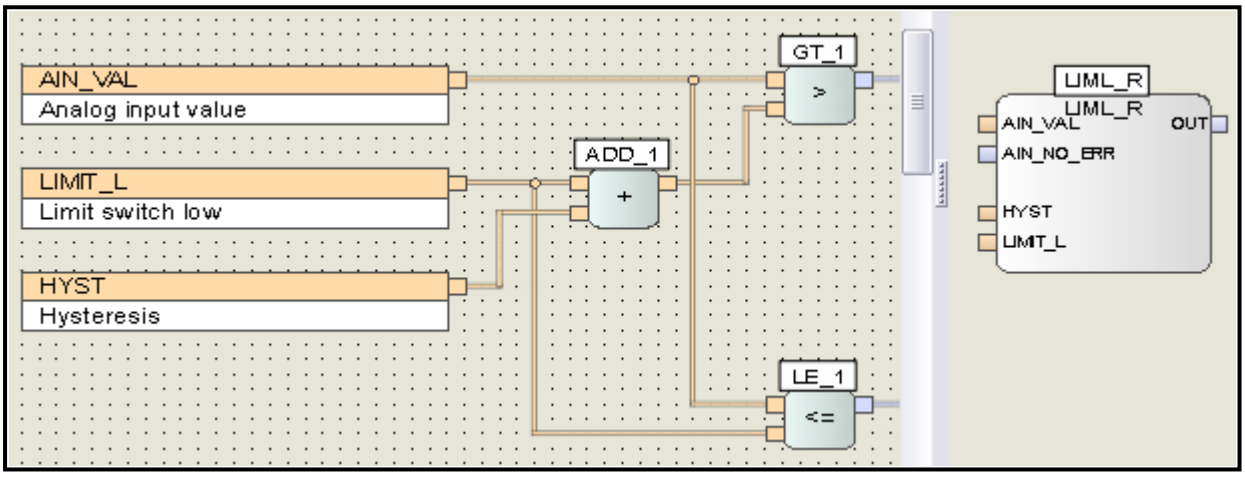
	Sort	Name	Typ	Abarbeitungsreihenfolge IM	Abarbeitungsreihenfolge CG
4	3	AND_1	Instanz	3	3
5	4	AND_2	Instanz	4	4
6	5	OR	Instanz	5	5
7	6	XOR	Instanz	6	6
8	7	TON	Instanz	7	7
9	8	2von3B	Instanz	8	8
10	9	2von3B_1	Instanz	9	9
11	10	2oo3	Instanz	10	10
12	11	OR_2	Instanz	11	11
13	12	YC12F13 High OK => YC12F13 Low OK	Zuweisung	-	12
14	13	BLINK	Instanz	12	13
15	14	R_TRIG_1	Instanz	13	14

6.2.11 Variable umbenennen

Das Umbenennen einer Variable vom Typ VAR_INPUT wird wie das Löschen und Einfügen einer neuen Variable behandelt. Die alte Variable wird gelöscht und die neue Variable wird hinzugefügt.

Achtung! Gefahr von ungewollten Anlagenabschaltungen!
Das Umbenennen bestehender Variablen führt bei einem Reload zur Neuinitialisierung dieser Variablen. Gespeicherte Werte gehen verloren!

Im folgenden Beispiel wurde die Eingangsvariable *LIM_L* in *LIMIT_L* umbenannt.



Das Umbenennen der Eingangsvariable führt nicht zu einer funktionalen Änderung der Logik. Daher ist die neu generierte Binärdatei */sys/ls/01_Programm01.ldb* identisch mit der importierten Version. Beide Versionen haben identische Prüfsummen. Daher ist keine funktionale Nachprüfung erforderlich.

Durch das Umbenennen der Eingangsvariable ändert sich jedoch die Prüfsumme des Funktionsbaustein-Typs LIML_R, was im Versionsvergleich angezeigt wird.

Nicht alle Änderungen im Quellcode resultieren in funktionalen Auswirkungen. Das Ändern des Namens einer VAR Input oder VAR Output resultiert zwar in einer veränderten Prüfsumme (CRC) des Funktionsbausteins und die entsprechende Zeile wird im Versionsvergleich rot hinterlegt, aber die Programm-Binärdatei ändert sich nicht. Dies bedeutet, dass keine funktionalen Änderungen erfolgt sind und keine weitere Prüfung erforderlich ist.

/sys/ls.config	Logic-Solver-Konfiguration	16#13071f67	V3	16#13071f67	V3	ok
/sys/ls/01_Programm01.config	Programm-Parameter	16#ee65ff57	V3	16#ee65ff57	V3	ok
/sys/ls/01_Programm01.ldb	Programm Binärdatei	16#b46e7734	V2	16#b46e7734	V2	ok
01_Programm01	Programm	16#293da5cc	V2	16#293da5cc	V2	ok
1oo2_R	Funktionsbaustein-Typ	16#665f1b8c	V2	16#665f1b8c	V2	ok
2oo3	Funktionsbaustein-Typ	16#0c925672	V2	16#0c925672	V2	ok
2von3B	Funktionsbaustein-Typ	16#0c1081a2	V2	16#0c1081a2	V2	ok
Average	Funktionsbaustein-Typ	16#830d0f29	V2	16#830d0f29	V2	ok
BLINK	Funktionsbaustein-Typ	16#7c7eda91	V2	16#7c7eda91	V2	ok
BUFFER	Array	16#f0d16020	V2	16#f0d16020	V2	ok
Diag_Modul	Funktionsbaustein-Typ	16#3c68356e	V2	16#3c68356e	V2	ok
Globale Variablen	Globale Variablen	16#c286681f	V2	16#c286681f	V2	ok
LIMH_R	Funktionsbaustein-Typ	16#939cbd0b	V2	16#939cbd0b	V2	ok
LIML_R	Funktionsbaustein-Typ	16#882b701d	V2	16#74f12d83	V2	-
Modul-Diag_PES_10	Funktionsbaustein-Typ	16#19d11128	V2	16#19d11128	V2	ok

Durch Doppelklicken in die Zeile *LIML_R* werden weitere Details zu dieser POE angezeigt.

Im Register *Lokale Variablen* können Sie erkennen, dass die Variable *LIM_L* in *LIMIT_L* umbenannt wurde.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Eigenschaft	Wert IM	Wert CG	Änderung
1	LIM_L				Gelöscht
2	LIMIT_L				Neu

Im Register *POE Änderungen* werden die im Funktionsbaustein-Typ *LIML_R* verwendeten Instanzen *ADD_1* und *LE_1* ebenfalls als *Geändert* klassifiziert, da beide Instanzen mit der Eingangsvariable *LIMIT_L* verbunden sind.

POE Änderungen | POE Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarbeitung IM	Abarbeitung CG	
1	ADD_1	Instanz	Blatt: 0/0, Pos.: 41/23	Blatt: 0/0, Pos.: 41/23	0	0	Geändert
2	LE_1	Instanz	Blatt: 0/0, Pos.: 64/38	Blatt: 0/0, Pos.: 64/38	2	2	Geändert

6.2.12 Spezielle Änderungen der Datei Programm.ldb

Ab SILworX V7 gibt es eine Detailansicht für die Datei *Programm.ldb*. Hier werden Änderungen angezeigt, die nicht unmittelbar aus POE-Änderungen resultieren, z. B. Veränderung der Stack-Größe, oder geänderte Retain-Timer-Behandlung.

Diese Änderungen müssen durch Tests der betroffenen Objekte verifiziert werden.

/sys/ls.config	Logic-Solver-Konfiguration	16#52a3f499	V3	16#94f2e678	V3	-
/sys/ls/21_Program_PES_10.config	Programm-Parameter	16#ed8b2329	V3	16#6cec3893	V3	-
/sys/ls/21_Program_PES_10.ldb	Programm Binärdatei	16#90ba99c0	V2	16#ff6e129f	V2	-
2oo3	Funktionsbaustein-Typ FBS	16#0c925672	V2	16#0c925672	V2	ok

- Wenn in den Programmeigenschaften der Parameter *Codegenerierung Kompatibilität* von ≤ V3 auf ≥ V4 umgestellt wird, verändert sich die Stack-Berechnung für benutzerdefinierte Datentypen (= Fehlerkorrektur).
- Wenn in den Programmeigenschaften der Parameter *Codegenerierung Kompatibilität* von < V7 auf ≥ V7 umgestellt wird, verändert sich die Retain-Timer-Behandlung (= Fehlerkorrektur).

> HIMA empfiehlt grundsätzlich bei bestehenden Projekten die *Codegenerierung Kompatibilität* nicht ohne Notwendigkeit umzustellen. Damit bleibt der CRC der Datei *Programm.ldb* unverändert und aufwändige Nachtests können vermieden werden. Bei Fragen hierzu kontaktieren Sie den HIMA Customer Support.

Beispiel

Die folgende Meldung in der Detailansicht der *Programm.ldb* entsteht durch die Umstellung der *Codegenerierung Kompatibilität* von V3 auf ≥ V7.

Versionsvergleich: IM <- CG PES_10 [10] /sys/ls/21_Program_PES_10.ldb

	Eigenschaft	Änderung
1	Interne Erweiterung der Programmlogik zur Unterstützung Retain-fähiger Timer/AS-Elemente mit Timerfunktion	Neu
2	Stackbedarf des Anwenderprogramms: IM: 9232 Byte <- CG: 4644 Byte	Geändert

- Nach der Umstellung ist das Programm für die korrekte Bearbeitung von Retain-Timern vorbereitet.
- Wenn Retain-Timer vorhanden und von der Änderung betroffen sind, ändert sich die Datei */sys/ls/"Programm"_retain.config*. Prüfen Sie in diesem Fall das geändete Verhalten der Retain-Timer.
- Die Berechnung des Stack-Bedarfs wurde geändert. Dies erfordert eine komplette Prüfung aller Logikteile des betroffenen Programms.

6.2.13 Änderungen in der Structured-Text-Logik

Ab SILworX V6 können Sie Logik auch in Structured Text (ST) programmieren.

Durch Einschränkungen zum Programmierstandard IEC 61131-3 ist SILworX Structured Text gemäß IEC 61508/IEC 61511 als Programmiersprache mit eingeschränktem Sprachumfang (LVL) für die Programmierung von sicherheitsgerichteteter Logik zugelassen.

Änderungen in einer Structured-Text-POE werden im Versionsvergleich detailliert angezeigt.

/sys/ls/05_Program_ST.ldb	Programm Binärdatei	16#25142913	V2	16#b587335e	V2	-
05_Program_ST	Programm	16#181b6fc9	V2	16#181b6fc9	V2	ok
CYCLE TIME SIMULATOR	Funktionsbaustein-Typ ST	16#96f8dbbb	V2	16#07b85f90	V2	-
Fahrenheit>Celsius	Funktion ST	16#ef614728	V2	16#ef614728	V2	ok

Durch Doppelklicken in eine Zeile werden bei Änderungen (hier: *CYCLE TIME SIMULATOR*) weitere Details zur POE angezeigt.

Beispiel 1: Geänderte Anweisung

ST Änderungen | ST Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Name	Typ	Position IM	Position CG	Abarb	Abarb	Änderung
1	COUNTMAX:=2000.0 -> COUNTMAX:=2010.0	Zuweisung	18,4	18,4	15	15	Geändert
2	2010.0	Ausdruck	-	18,16	-	0	Neu
3	2000.0	Ausdruck	18,16	-	0	-	Gelöscht

In den Spalten *Position IM* und *Position CG* werden die Objekte mit Zeile und Spalte aufgelistet. Die Angabe «18,16» bedeutet Zeile 18, Spalte 16.

Durch Doppelklicken auf die geänderte Zuweisung (Zeile 1 im Bild oben) wird der zugehörige ST-Baustein im Structured-Text-Editor geöffnet und die geänderte Zuweisung wird markiert.

```
13 END_IF;
14
15 CASE LEVEL OF
16 1: COUNTMAX := 500.0;
17 2: COUNTMAX := 1000.0;
18 3: COUNTMAX := 2010.0;
19 4: COUNTMAX := 4000.0;
20 5: COUNTMAX := 8000.0;
21 6: COUNTMAX := 16000.0;
```

Beispiel 2: Geänderte Abarbeitungsreihenfolge

ST Änderungen | ST Abarbeitungsreihenfolge | Lokale Variablen | Sonstige Änderungen

	Sortier	Name	Typ	Abarbeitungsreihenfolge IM	Abarbeitungsreihenfolge CG
1	0	TP(IN:=NOT TICK,PT:=INTERVALL,ET=>ET)	Anweisung	0	0
2	1	TICK:=TP.Q	Zuweisung	1	1
3	2	R_TRIG(CLK:=TP.Q)	Anweisung	2	2
4	3	IF R_TRIG.Q THEN	Anweisung	3	Verschoben nach 6
5	4	IF R_TRIG.Q THEN	Anweisung	Verschoben von 3	6
6	5	LEVEL:=LEVEL+1	Zuweisung	4	Verschoben nach 7
7	6	END_IF	Anweisung	5	5
8	7	IF LEVEL=17 THEN	Anweisung	6	3
9	8	LEVEL:=1	Zuweisung	7	4
10	9	LEVEL:=LEVEL+1	Zuweisung	Verschoben von 4	7
11	10	END_IF	Anweisung	8	8
12	11	CASE LEVEL OF	Anweisung	9	9

Die Abarbeitungsreihenfolgen der importierten und der neu generierten Konfiguration können Sie den Spalten *Abarbeitungsreihenfolge IM* und *Abarbeitungsreihenfolge CG* entnehmen.

Die Abarbeitungsreihenfolge entspricht der Reihenfolge der tatsächlichen Anweisungen ohne Leerzeilen und Kommentare und kann in der Logik nicht dargestellt werden.

```
1
2  TP(IN := NOT TICK , PT := INTERVA
3  (* Alternative: TP(IN := NOT TICK
4  TICK := TP.Q;
5  R_TRIG(CLK :=TP.Q); (* also R_TRI
6
7  IF R_TRIG.Q THEN (* also IF R.TRI
8  LEVEL := LEVEL + 1;
9  END_IF;
10
11 IF LEVEL = 17 THEN
12 LEVEL := 1;
13 END_IF;
14
15 CASE LEVEL OF
16 1: COUNTMAX := 500.0;
17 2: COUNTMAX := 1000.0;
```

```
1
2  TP(IN := NOT TICK , PT := INTERVA
3  (* Alternative: TP(IN := NOT TICK
4  TICK := TP.Q;
5  R_TRIG(CLK :=TP.Q); (* also R_TRI
6
7  IF LEVEL = 17 THEN
8  LEVEL := 1;
9  END_IF;
10
11 IF R_TRIG.Q THEN (* also IF R.TRI
12 LEVEL := LEVEL + 1;
13 END_IF;
14
15 CASE LEVEL OF
16 1: COUNTMAX := 500.0;
17 2: COUNTMAX := 1000.0;
```

6.3 Änderung der Zuordnung von globalen Variablen

6.3.1 Globale Variable einem anderen Hardware-Eingang zuordnen

Das Ändern der Zuordnung von globalen Variablen (neue Quelle, neues Ziel, geänderte Initialwerte) wirkt sich immer auf die Konfigurationsdatei für das Lesen und Schreiben globaler Variablen */sys/ke.config* aus.

/sys/cpc.config	Systemprotokolle Basis	16#ced2a001	V2	16#ced2a001	V2	ok
/sys/cpcnsip.config	Standardprotokoll	16#64ecf542	V3	16#64ecf542	V3	ok
/sys/cpcsip.config	Sicherheitsgerichtetes Protokoll	16#9855ad19	V2	16#9855ad19	V2	ok
/sys/cpu.config	Systemdaten	16#03161291	V3	16#03161291	V3	ok
/sys/io4cpu.config	System IO	16#5e571427	V3	16#5e571427	V3	ok
/sys/ke.config	Datenlayout- und Transport	16#65017dc0	V2	16#b6820685	V2	-
/sys/lm.config	Lizenz	16#889b1742	V2	16#889b1742	V2	ok
/sys/ls.config	Logic-Solver-Konfiguration	16#13071f67	V3	16#13071f67	V3	ok
/sys/pgs.config	Konfigurationsverbindungen	16#f07e8dfb	V4	16#f07e8dfb	V4	ok

Durch Doppelklicken in die Zeile */sys/ke.config* werden weitere Details zu dieser Konfigurationsdatei angezeigt.

Versionsvergleich: IM <- CG PES_10 [10] /sys/ke.config

Schließen

	Globale	Variable	Quelle	Ziel	Art der Änderung
1	DI_Initi	DI Channel10 -> Kanalwert [BOOL]		A0.K6.(10.0.6)	Variable 'DI Channel10 -> Kanalwert [BOOL]' ist nun mit 'DI_Initiator_Sensor_01' verbunden.
2	DI_Initi	DI Channel01 -> Kanalwert [BOOL]		A0.K7.(10.0.7)	Variable 'DI Channel01 -> Kanalwert [BOOL]' ist nicht mehr mit 'DI_Initiator_Sensor_01' verbunden.
3	DI_Initi	DI_Initiator_Sensor_01	A0.K6.(10.0.6)	Alarm & Events	Quelle ist nun: 'A0.K6.(10.0.6)'
4	DI_Initi	DI_Initiator_Sensor_01	A0.K6.(10.0.6)	AP:21_Program_PES_10	Quelle ist nun: 'A0.K6.(10.0.6)'
5	DI_Initi	REGISTER/Register-Out-00_Bit-00	A0.K6.(10.0.6)	Standardprotokoll 'Modbus-Slave-Set_1'	Quelle ist nun: 'A0.K6.(10.0.6)'

Zeile	Beschreibung
1	Für Kanal 10 des Moduls mit der SRS `10.0.6` wurde der Parameter *Kanalwert* neu mit der globalen Variable *DI_Initiator_Sensor_01* verbunden.
2	Für Kanal 01 des Moduls mit der SRS `10.0.7` ist der Parameter *Kanalwert* nicht mehr mit der globalen Variable *DI_Initiator_Sensor_01* verbunden.
3	Die Quelle der als *Alarm & Events* definierten globalen Variable *DI_Initiator_Sensor_01* hat sich geändert und ist nun das Modul mit der `SRS 10.0.6`.
4	Die Quelle der im Anwenderprogramm *21_Program_Pes_10* verwendeten globalen Variable *DI_Initiator_Sensor_01* hat sich geändert und ist nun das Modul mit der SRS `10.0.6`.
5	Die Quelle der vom Standardprotokoll *Modbus-Slave-Set_1* im *Register-Out_Bit-00* übertragenen globalen Variable *DI_Initiator_Sensor_01* hat sich geändert und ist nun das Modul mit der SRS `10.0.6`.

Beispiel: HIMatrix

Versionsvergleich: IM <- CG PES_35 [10] /sys/ke.config

Schließen

	Globale Variable	Variable	Quelle	Ziel	Art der Änderung
1	Globale Variable_5	DO05 Wert [BOOL] ->	Initialisierung	IO:0.2	Variable 'DO05 Wert [BOOL] ->' ist nun mit 'Globale Variable_5' verbunden.
2	Globale Variable_5	Globale Variable_5	Initialisierung		Neue Variable 'Globale Variable_5' in 'Initialisierung'.
3	Globale Variable_6	DI06 -> Wert [BOOL]		IO:0.4	Variable 'DI06 -> Wert [BOOL]' ist nun mit 'Globale Variable_6' verbunden.

Der Versionsvergleich berücksichtigt alle Auswirkungen der Änderung. Der Anwender muss selbst entscheiden, welche dieser Auswirkungen er prüfen muss.

> Auch wenn die Logik nicht geändert wurde, kann sich z. B. das Programm anders verhalten, wenn eine dort verarbeitete Variable eine andere Quelle hat.

6.3.2 Neuer Initialwert für eine globale Variable

Im Globale-Variablen-Editor können Sie den Initialwert einer globalen Variable einstellen oder ändern. Diese Information wird in der Konfigurationsdatei */sys/ke.config* gespeichert. Das Ändern eines oder mehrerer Initialwerte wirkt sich ebenfalls auf die Datei */sys/ke.config* aus.

Durch Doppelklicken in die Zeile */sys/ke.config* werden weitere Details zu den globalen Variablen angezeigt. Der nachfolgenden Abbildung ist zu entnehmen, dass der Initialwert der globalen Variable *AE_Prozesswert_Kanal_03* auf `1010.0` geändert wurde.

Versionsvergleich: IM <- CG PES_10 [10] /sys/ke.config

Schließen

	Globale Variable	Variable	Quelle	Ziel	Art der Änderung
1	AE_Prozesswert_Kanal_03	AI-Kanal03 -> Prozesswert [REAL]	X-AI 32 01-9/10 (10.0.9 / 0.10)		Initialwert ist nun: '1010.0'

6.4 Trennung sicherer und nicht sicherer Logik

SILworX bietet die Möglichkeit, die Aufgaben eines Systems in mehrere Programme aufzuteilen und damit eine Trennung zwischen sicherheitsrelevanter Logik (z. B. ESD-Funktionen) und nicht sicherheitsrelevanter Logik (z. B. Datenaufbereitung für ein Leitsystem) vorzunehmen.

> Die Trennung von sicherer und nicht sicherer Logik in unterschiedliche Programme erleichtert die Darstellung von Änderungen und reduziert den Aufwand bei erforderlichen Nachprüfungen.
> HIMA empfiehlt, diese Trennung auch für die globalen Variablen durchzuführen, z. B. durch eine geeignete Namensgebung. Weitere Hinweise finden Sie auch in der IEC 61511, Teil 1, Kapitel 12.

Durch den Versionsvergleich können Änderungen in der Logik erkannt werden. Wurden Änderungen in der sicherheitsrelevanten Logik vorgenommen, erfordert dies in der Regel eine sicherheitstechnische Nachprüfung.

In der nachfolgenden Abbildung hat der Versionsvergleich eine Änderung in der Binärdatei von */sys/ls/01_Programm01.ldb* ermittelt. Es wurde eine Änderung am Funktionsbaustein-Typ *2von3B* vorgenommen. Diese POE muss geprüft werden.

In der Binärdatei von */sys/ls/02_Programm02* hat der Versionsvergleich hingegen keine Änderung festgestellt.

/sys/ls.config	Logic-Solver-Konfiguration	16#9f04e25f	V3	16#c1df0a55	V3	-
/sys/ls/01_Programm01.config	Programm-Parameter	16#ee65ff57	V3	16#ee65ff57	V3	ok
/sys/ls/01_Programm01.ldb	Programm Binärdatei	16#b46e7734	V2	16#feae836d	V2	-
01_Programm01	Programm	16#293da5cc	V2	16#293da5cc	V2	ok
1oo2_R	Funktionsbaustein-Typ	16#665f1b8c	V2	16#665f1b8c	V2	ok
2oo3	Funktionsbaustein-Typ	16#0c925672	V2	16#0c925672	V2	ok
2von3B	Funktionsbaustein-Typ	16#0c1081a2	V2	16#1e4e7f96	V2	-
Average	Funktionsbaustein-Typ	16#830d0f29	V2	16#830d0f29	V2	ok
BLINK	Funktionsbaustein-Typ	16#7c7eda91	V2	16#7c7eda91	V2	ok
BUFFER	Array	16#f0d16020	V2	16#f0d16020	V2	ok
Diag_Modul	Funktionsbaustein-Typ	16#3c68356e	V2	16#3c68356e	V2	ok
Globale Variablen	Globale Variablen	16#c286681f	V2	16#c286681f	V2	ok
LIMH_R	Funktionsbaustein-Typ	16#939cbd0b	V2	16#939cbd0b	V2	ok
LIML_R	Funktionsbaustein-Typ	16#882b701d	V2	16#882b701d	V2	ok
Modul-Diag_PES_10	Funktionsbaustein-Typ	16#19d11128	V2	16#19d11128	V2	ok
Schrittkette-PES_10	Funktionsbaustein-Typ	16#50d0c1aa	V2	16#50d0c1aa	V2	ok
SystemTime	Funktionsbaustein-Typ	16#b404049e	V2	16#b404049e	V2	ok
Systemüberwachung-PES_10	Funktionsbaustein-Typ	16#579d399c	V2	16#579d399c	V2	ok
/sys/ls/01_Programm01_force.config	Force-Daten der Applikation	16#ec11d429	V2	16#ec11d429	V2	ok
/sys/ls/01_Programm01_retain.config	Retain-Daten der Applikation	16#d69d78b3	V2	16#d69d78b3	V2	ok
/sys/ls/02_Programm02.config	Programm-Parameter	16#0ae4a13a	V3	16#0ae4a13a	V3	ok
/sys/ls/02_Programm02.ldb	Programm Binärdatei	16#ce11e72d	V2	16#ce11e72d	V2	ok
/sys/ls/02_Programm02_force.config	Force-Daten der Applikation	16#aee92083	V2	16#aee92083	V2	ok
/sys/ls/02_Programm02_retain.config	Retain-Daten der Applikation	16#fc73594f	V2	16#fc73594f	V2	ok

Besonderes Augenmerk ist auf Änderungen der Variablenzuordnung (neue Quelle, neues Ziel) von globalen Variablen zu richten. Sie führen nicht notwendigerweise zu einer geänderten Syntax und damit zu Veränderungen der Programm-Binärdatei. Trotzdem können diese Änderungen sicherheitsrelevant sein! Änderungen an globalen Variablen werden in der Detailansicht der Konfigurationsdatei *ke.config* beschrieben, siehe *Globale Variable einem anderen Hardware-Eingang zuordnen*.

Wenn globale Variablen von mehreren Programmen gelesen werden, müssen bei Änderungen der Konfigurationsdatei *ke.config* die Programme individuell geprüft werden.

Mit Hilfe der im Globale-Variablen-Editor angezeigten Querverweise kann festgestellt werden, ob z. B. eine in der Hardware-Zuordnung geänderte Variable vom sicherheitsrelevanten Programm geschrieben wird, oder vom nicht sicherheitsrelevanten Programm. Wird die Variable vom sicherheitsrelevanten Programm lesend verwendet, ist die Änderung immer sicherheitsrelevant!

Änderungen in anderen zentralen Konfigurationsdateien (z. B. Modul-Konfigurationsdateien, Systemeinstellungen usw.) müssen ebenfalls individuell auf Sicherheitsrelevanz geprüft werden. Änderungen in diesen zentralen Konfigurationsdateien haben keinen direkten Bezug zu den Programmen, d. h. die für die Programme vorgenommene Trennung in *sicher/nicht sicher* ist hier irrelevant.

6.5 Speicherübersicht für Code und Daten

Durch Doppelklicken in die Zeile */sys/ls.config* wird die Speicherübersicht geöffnet.

- In einem HIMax System stehen, abhängig vom verwendeten CPU-Modul, entweder 5 MB (für X-CPU 31) oder 10 MB (für X-CPU 01) Speicherplatz für Programmcode und Daten zur Verfügung. Werden in einer Steuerung mehrere Programme ausgeführt, wird der Speicherplatz aufgeteilt.
- In einem HIMatrix System stehen 5 MB Speicherplatz zur Verfügung.

	/sys/lm.config	Lizenz	16#889b1742	V2	16#889b1742	V2	ok
⊟	/sys/ls.config	Logic-Solver-Konfiguration	16#9f04e25f	V3	16#c1df0a55	V3	-
	/sys/ls/01_Programm01.config	Programm-Parameter	16#ee65ff57	V3	16#ee65ff57	V3	ok
⊞	/sys/ls/01_Programm01.ldb	Programm Binärdatei	16#b46e7734	V2	16#feae836d	V2	-

6.5.1 Speicherübersicht am Beispiel HIMax

Die Speicherübersicht zeigt die Auslastung sowohl in Summe, als auch einzeln für jedes Programm. Die Prozentangabenbeziehen sich auf den Gesamtspeicher (in der Abbildung unten: HIMax mit

X-CPU 01).

Versionsvergleich: IM <- CG PES_10 [10] /sys/ls.config

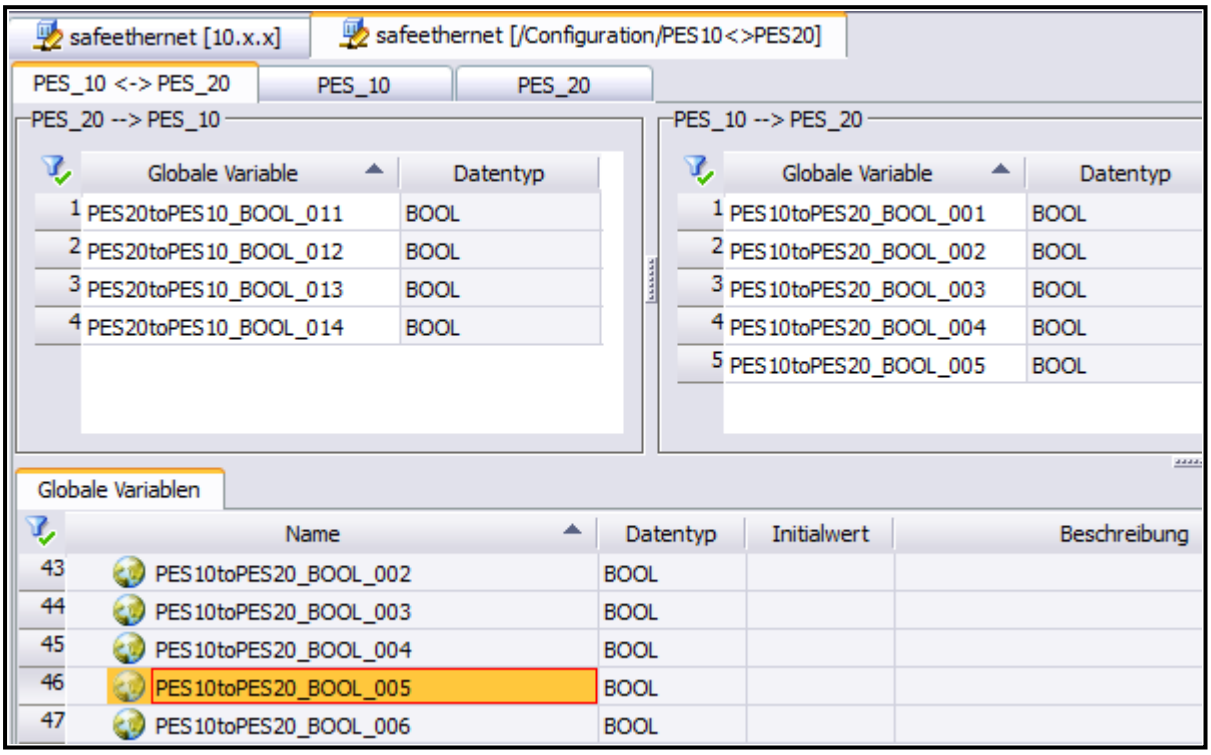
Schließen

	Info	fo der letzten Codegenerierur	Info der Import-Konfiguration	max	
1	⊟ |Summe				
2	Codegröße	116116(1%)	116116(1%)	10481664	-
3	Datengröße	16840(0%)	16840(0%)	10481664	-
4	Größe der Retaindaten	16(0%)	16(0%)	32768	-
5	⊟ 01_Programm01	-	-		-
6	Codegröße	115244(1%)	115244(1%)	10481664	-
7	Datengröße	12608(0%)	12608(0%)	10481664	-
8	Größe der Retaindaten	16(0%)	16(0%)	32768	ok
9	Id	1	1		-
10	⊟ 02_Programm02	-	-		-
11	Codegröße	872(0%)	872(0%)	10481664	ok
12	Datengröße	4232(0%)	4232(0%)	10481664	ok
13	Größe der Retaindaten	0(0%)	0(0%)	32768	ok
14	Id	2	2		-

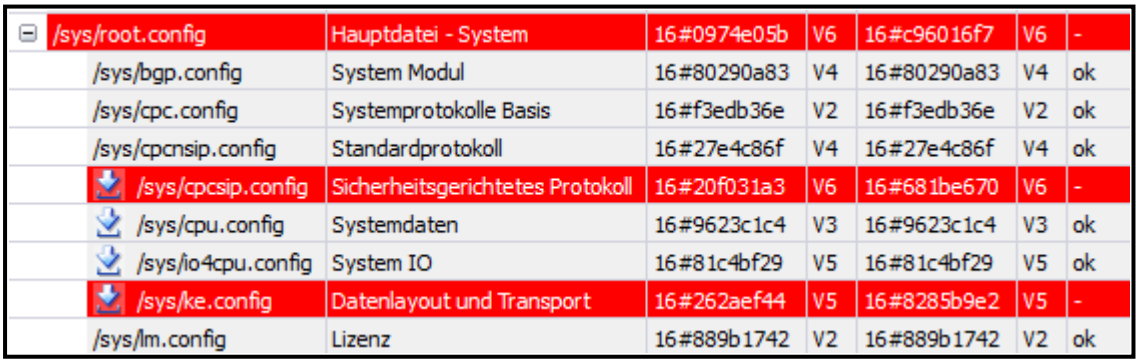
6.6 Änderungen in safeethernet Kommunikation

6.6.1 Neue Variable zu bestehender Verbindung hinzufügen

Im nachfolgenden Beispiel wird einer safe**ethernet** Verbindung die Variable *PES10toPES20_BOOL_005* hinzugefügt.

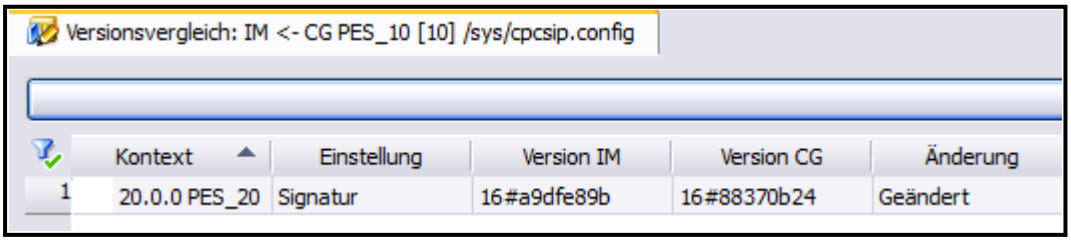


Diese Änderung wird im Versionsvergleich wie folgt dargestellt:



/sys/root.config	Hauptdatei - System	16#0974e05b	V6	16#c96016f7	V6	-
/sys/bgp.config	System Modul	16#80290a83	V4	16#80290a83	V4	ok
/sys/cpc.config	Systemprotokolle Basis	16#f3edb36e	V2	16#f3edb36e	V2	ok
/sys/cpcnsip.config	Standardprotokoll	16#27e4c86f	V4	16#27e4c86f	V4	ok
/sys/cpcsip.config	Sicherheitsgerichtetes Protokoll	16#20f031a3	V6	16#681be670	V6	-
/sys/cpu.config	Systemdaten	16#9623c1c4	V3	16#9623c1c4	V3	ok
/sys/io4cpu.config	System IO	16#81c4bf29	V5	16#81c4bf29	V5	ok
/sys/ke.config	Datenlayout und Transport	16#262aef44	V5	16#8285b9e2	V5	-
/sys/lm.config	Lizenz	16#889b1742	V2	16#889b1742	V2	ok

Durch Doppelklicken in die Zeile */sys/cpcsip.config* wird die Detailansicht geöffnet.



Die Signaturen (CRC) über die sicherheitsgerichteten Daten haben sich geändert! Ursache sind die geänderten Daten, die in der Datei *ke.config* detailliert angezeigt werden.

Versionsvergleich: IM <- CG PES_10 [10] /sys/ke.config

Schließen

	Globale Variable	Variable	Quelle	Ziel	Art der Änderung
1	PES10toPES20_BOOL_005	PES10toPES20_BOOL_005	Initialisierung		Neue Variable 'PES10toPES20_BOOL_005' in 'Initialisierung'.
2	PES10toPES20_BOOL_005	PES10toPES20_BOOL_005	Initialisierung	safeethernet 20.0.0 (Daten)	Neue Variable 'PES10toPES20_BOOL_005' in 'safeethernet 20.0.0 (Daten)'.

Zeile	Beschreibung
1	Die Variable *PES10toPES20_BOOL_005* wird erstmalig benutzt und muss daher initialisiert werden.
2	Die Variable *PES10toPES20_BOOL_005* wird in den Datenbereich der safe**ethernet** Verbindung 20.0.0 geschrieben. Da die safe**ethernet** Verbindung das Ziel ist, wird die Variable zum Partner gesendet. Verbindung `20.0.0` steht für folgende Parameter: • System-ID des Partners: 20 • Rack-ID des Partners: 0 • Verbindungs-ID: 0

6.6.2 safeethernetParameter Änderungen

Im nachfolgenden Beispiel wurde die Receive-Timeout und die Einstellung *Verhalten* geändert. Der Versionsvergleich erkennt Veränderungen in der Datei */sys/cpcsip.config*.

Rsp t	Rcv TMO	Rsnd TMO	Ack TMO	Prod Rate	Speicher	Verhalten	Diag.Eintr.	Prio A&E
500	1200	500	0	0	2	Prozesswert unbegrenzt einfrieren	1	

Durch Doppelklicken in die Zeile */sys/cpcsip.config* wird die Detailansicht geöffnet, in der die geänderten Parameter erkennbar sind.

Versionsvergleich: IM <- CG PES_10 [10] /sys/cpcsip.config

Schließen

	Kontext	Einstellung	Version IM	Version CG	Änderung
1	20.0.0 PES_20	Receive Timeout [ms]	1300	1200	Geändert
2	20.0.0 PES_20	Verhalten bei Verbindungsverlust [ms]	Initialwert verwenden	Prozesswert unbegrenzt einfrieren	Geändert

7 Ausdruck der Vergleichsinformationen

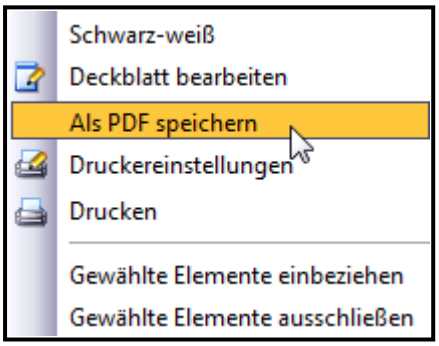
Der Ausdruck der Vergleichsinformationen erfolgt mit Hilfe des Dokumentationseditors und dient als Nachweis der Änderungen. Ein kompletter, schriftlicher Nachweis der Änderungen besteht meistens aus drei Teilen:

1. Ausdruck der Vergleichsinformationen als Ergebnis des Versionsvergleichs.
2. Ausdruck der Logik (Objekte) nach der Änderung.
3. Ausdruck der Logik (Objekte) vor der Änderung.

Bei Änderungen von Parametern in Tabellen (z. B. Hardware) oder Eigenschaftendialogen werden automatisch die Werte vor und nach der Änderung angezeigt. Ein weiterer Ausdruck, der diese Zustände noch einmal dokumentiert, ist daher meistens nicht erforderlich.

Im Gegensatz dazu wird in der Logik nur die Position der Änderung angezeigt. Hier ist es erforderlich die Zustände vor und nach der Änderung auszudrucken!

Die Dokumentation kann in SILworX entweder als Papierausdruck erfolgen, oder als PDF-Datei gespeichert werden.



> Für die Dokumentation wird immer das Ergebnis des zuletzt durchgeführten Versionsvergleichs verwendet. Es ist daher wichtig, dass zuerst der Versionsvergleich durchgeführt wird, bevor der Dokumentationseditor geöffnet wird!

Der Ausdruck enthält grundsätzlich die gleichen Informationen, die auch auf dem Bildschirm angezeigt werden (What You See Is What You Get).

7.1 Ausdruck vorbereiten

Um Änderungen an einem Projekt zu dokumentieren empfiehlt HIMA die unten aufgelisteten Schritte in der genannten Reihenfolge durchzuführen. Zwischen den einzelnen Schritten dürfen keine weiteren Änderungen (offline) vorgenommen werden. Ansonsten würde der Ausdruck auch die ungewollten Änderungen dokumentieren.

1. Die Änderungen offline vorbereiten.
2. Die Codegenerierung durchführen.
3. Die Änderungen entsprechend vorhandener Modifikationsprozeduren genehmigen lassen.
   Wichtig: Die Änderungen werden erst nach erfolgter Genehmigung geladen und getestet. Zur Erstellung des Nachweises für die Genehmigung mit **Schritt 5** fortfahren.
4. Genehmigte Änderungen in die PES laden.

> Entsprechend betrieblicher Modifikationsprozeduren nach den Regelwerken der funktionalen Sicherheit ist eine **Auswirkungsanalyse und Genehmigung der geplanten Änderungen** vor dem Laden in das PES erforderlich. Dazu wird Schritt 4 zunächst ausgelassen und ein dokumentierter Vergleich zwischen der neu generierten Konfiguration und der geladenenVersion (geladen) durchgeführt.

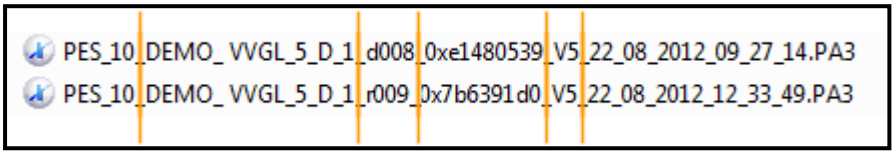
5. Den Versionsvergleich durchführen. Anschließend, wie in den vorangegangenen Kapiteln beschrieben, die Auswertung am Bildschirm vornehmen.
6. Den Dokumentationseditor starten und das Deckblatt bearbeiten, z. B. den CRC eintragen.
7. Die Seiten der Vergleichsdokumentation zum Drucken auswählen.
8. Die geänderten Seiten (Logik, Hardware etc.) auswählen. Weitere Hinweise dazu finden Sie in den nachfolgenden Kapitel.
9. Das Dokument ausdrucken.

> Zum endgültigen Nachweis der durchgeführten Änderungen gegenüber der Genehmigungsbehörde werden die oben genannten Schritte unter Auslassung von Schritt 3 nochmals durchgeführt. Damit wird eine Enddokumentation der durchgeführten, geladenen und getesteten Änderungen erzeugt.

7.2 Referenzierung auf Projektarchive

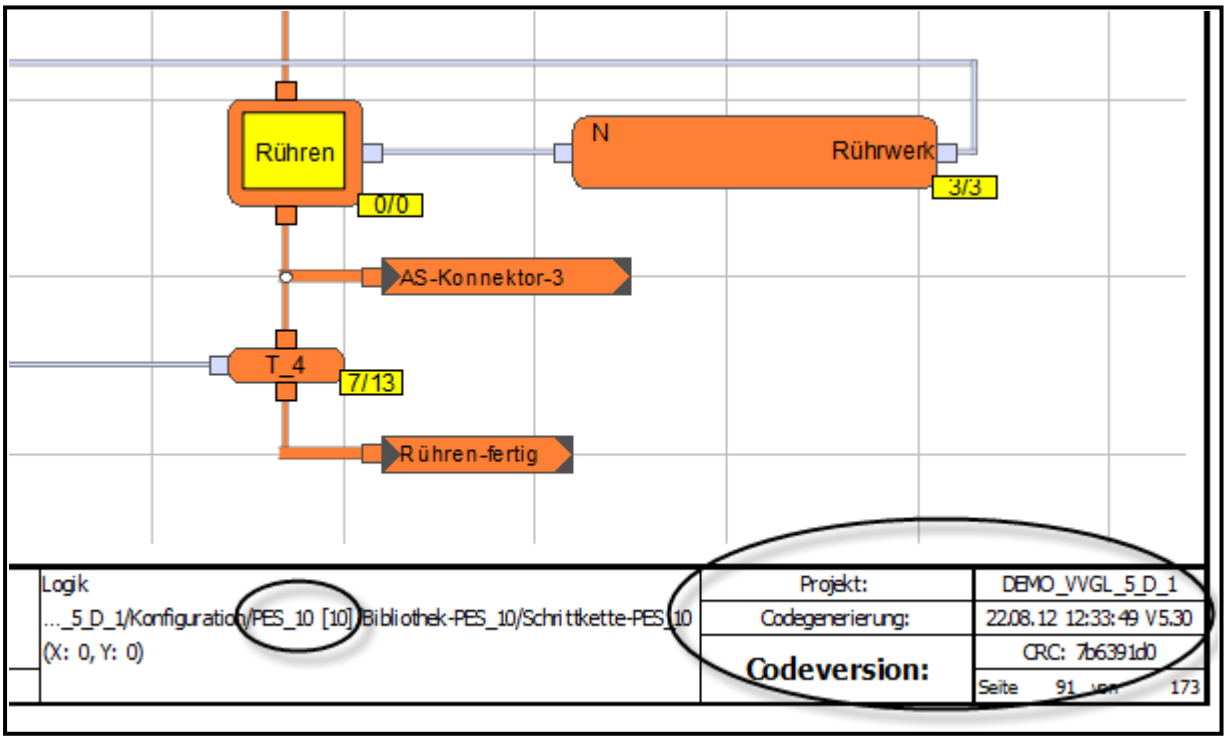
Ab SILworX V5 kann mit jedem Laden einer Steuerung automatisch ein Projektarchiv gespeichert werden. Dieses Projektarchiv enthält die zuletzt geladenen Konfigurationsdateien.

Der Name eines Projektarchivs enthält die folgenden Informationen:



1. Ressourcename
2. Projektname
3. Ladenummer (d = Download, r = Reload)
4. Ressource-CRC
5. SILworX Version
6. Datum und Uhrzeit der Codegenerierung

Der im Projektarchiv gespeicherte Ausgabestand der Konfiguration ist unveränderlich und somit die ideale Referenz zur Dokumentation des Ausdrucks. Wie Sie den Ausgabestand im Ausdruck verwenden, wird in *Deckblatt ausfüllen* erklärt.



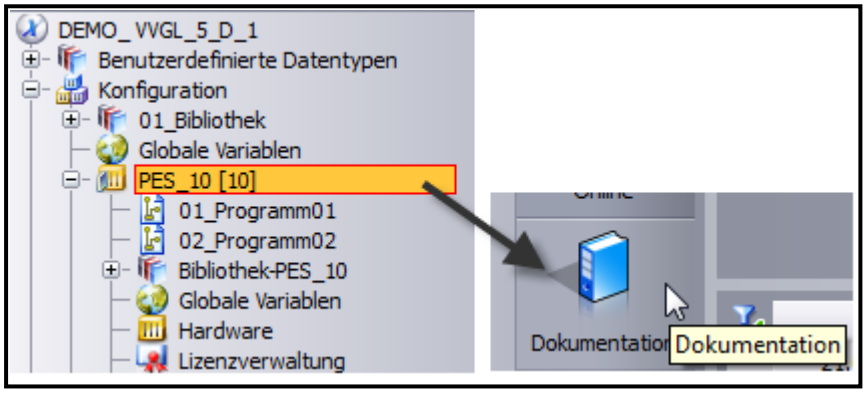
7.3 Erzeugen einer Dokumentation (Ausdruck)

Die folgenden Abschnitte beschreiben in chronologischer Reihenfolge die Schritte, welche für die Erstellung einer Projektdokumentation durchzuführen sind.

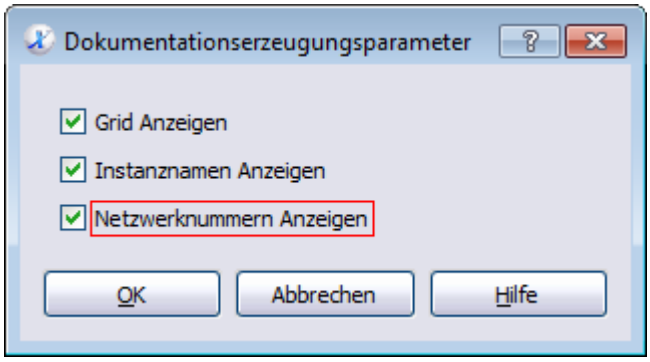
7.3.1 Dokumentation starten

Bevor Sie den Dokumentationseditor starten, wählen Sie im Strukturbaum zuerst die gewünschte Ressource aus. Dadurch werden in der Vorauswahl alle Seiten dieser Ressource zum Drucken ausgewählt.

➤ Wählen Sie im Strukturbaum die gewünschte Ressource und klicken Sie in der Aktionsleiste auf die Schaltfläche **Dokumentation**.

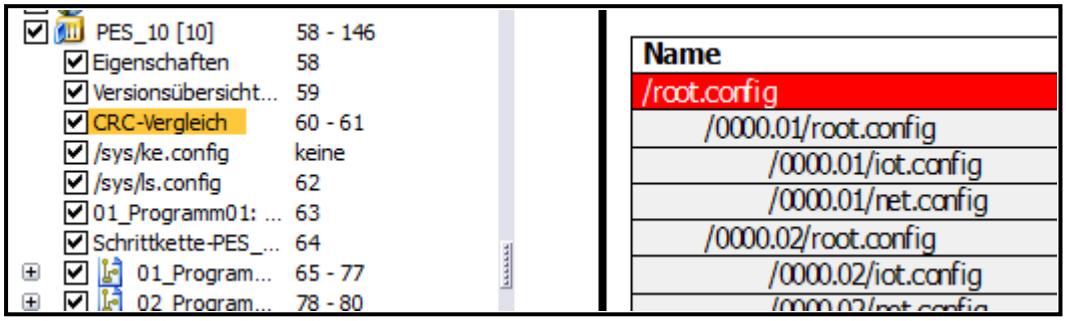


➤ Aktivieren Sie im Dialogfenster *Dokumentationserzeugungsparameter* alle Optionen, wenn der Ausdruck als Änderungsnachweis dienen soll.



7.3.2 Deckblatt ausfüllen

➤ Klappen Sie im Dokumentationseditor die gewünschte Ressource auf und markieren Sie das Element *CRC-Vergleich*.



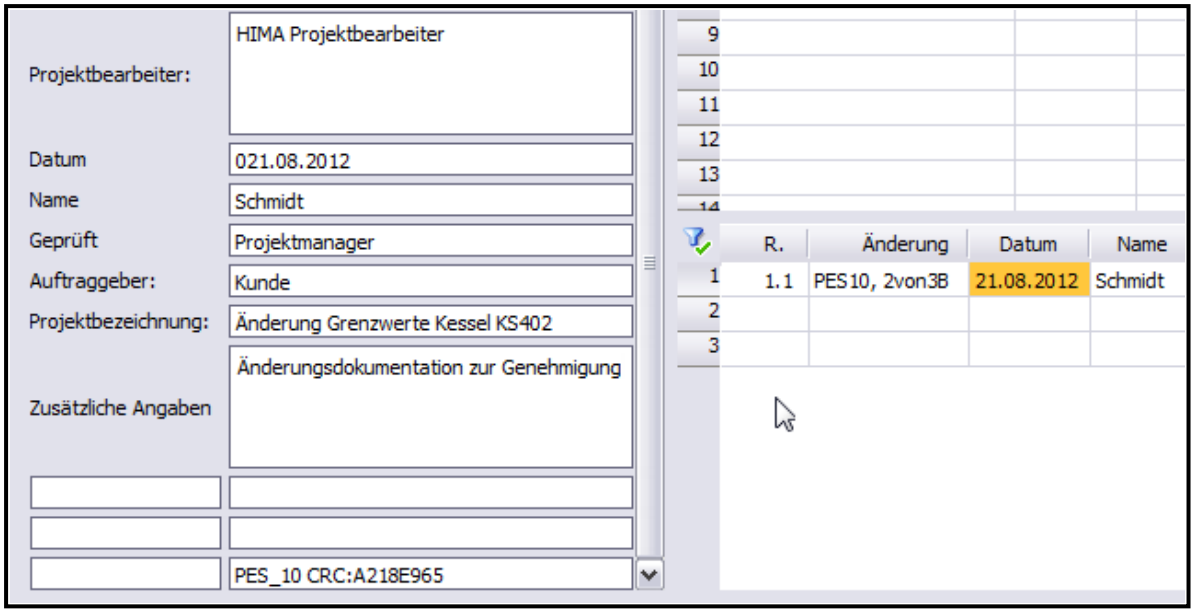
➤ Lesen Sie den aktuellen CRC aus.

Im Beispiel unten soll nach einer neuen Codegenerierung die Änderung (CRC CG) zur Genehmigung vorgelegt werden.

CRC DL	Version DL	CRC CG	Version CG
16#a3ab80d6	V4	16#a218e965	V4
16#239441f4	V3	16#239441f4	V3
16#209c5b8f	V3	16#209c5b8f	V3

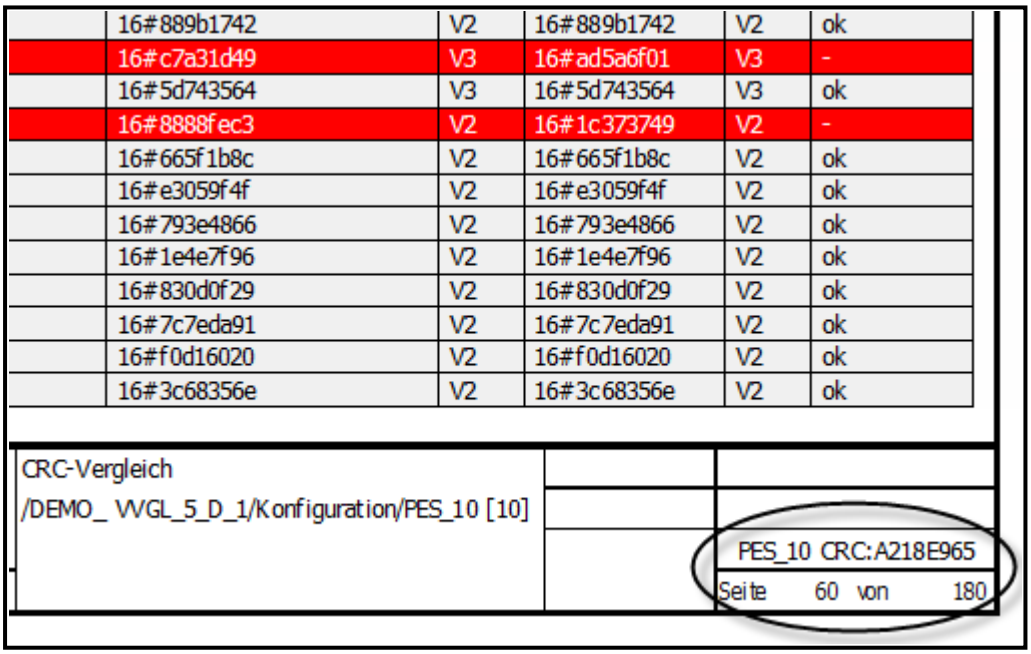
➤ Wählen Sie im Menü **Dokumentation** die Funktion **Deckblatt bearbeiten**. Die Vorlage des Deckblatts öffnet sich.

➤ Tragen Sie im Deckblatt alle Angaben zum Projekt ein, wie z. B. Bearbeiter, Kommentare und CRC.



Speichern Sie das geänderte Deckblatt.

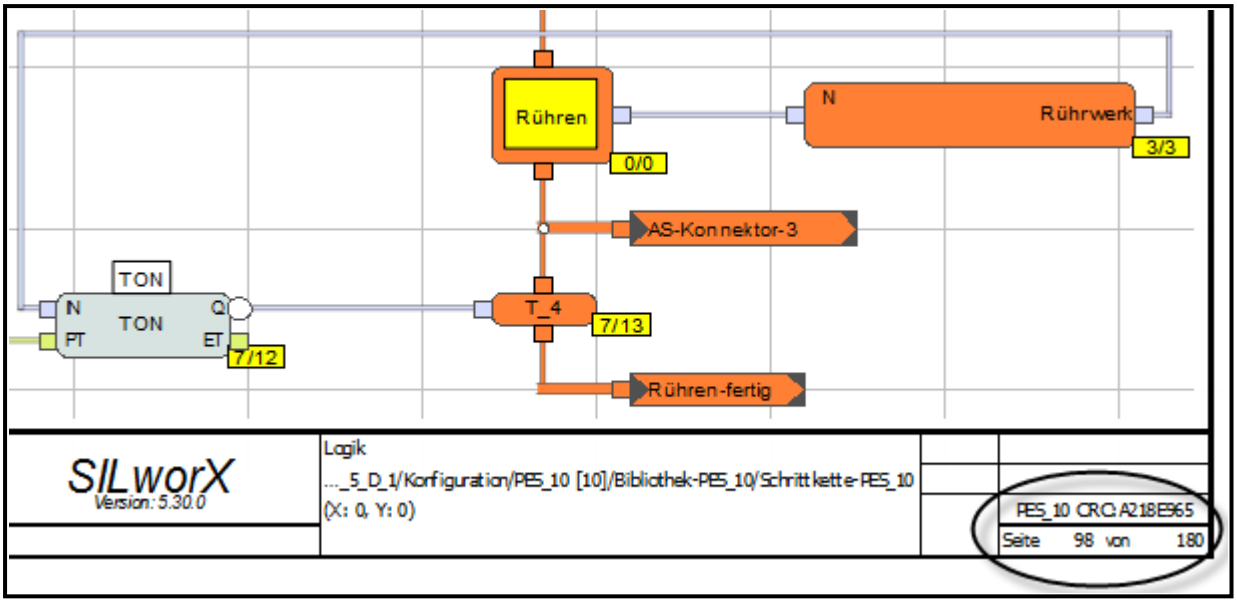
Dieser Bereich erscheint später auf jedem gedruckten Blatt.



16#889b1742	V2	16#889b1742	V2	ok
16#c7a31d49	V3	16#ad5a6f01	V3	-
16#5d743564	V3	16#5d743564	V3	ok
16#8888fec3	V2	16#1c373749	V2	-
16#665f1b8c	V2	16#665f1b8c	V2	ok
16#e3059f4f	V2	16#e3059f4f	V2	ok
16#793e4866	V2	16#793e4866	V2	ok
16#1e4e7f96	V2	16#1e4e7f96	V2	ok
16#830d0f29	V2	16#830d0f29	V2	ok
16#7c7eda91	V2	16#7c7eda91	V2	ok
16#f0d16020	V2	16#f0d16020	V2	ok
16#3c68356e	V2	16#3c68356e	V2	ok

CRC-Vergleich
/DEMO_ VVGL_5_D_1/Konfiguration/PES_10 [10]

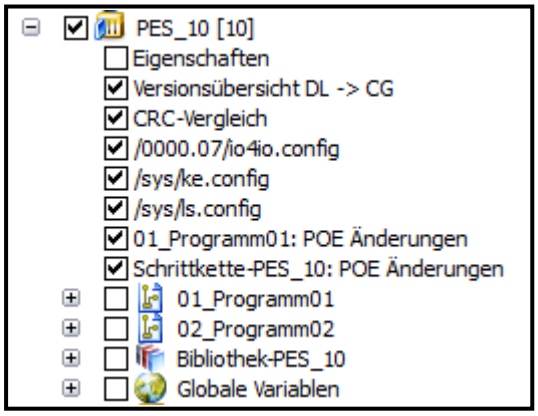
PES_10 CRC:A218E965
Seite 60 von 180



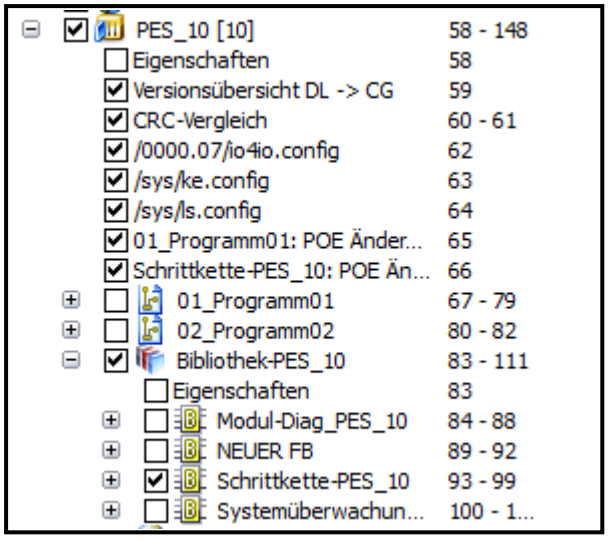
7.3.3 Objekte (Seiten) zum Drucken auswählen

Die Anzahl und Auswahl der zu druckenden Objekte (Seiten) ist vom Ergebnis des Versionsvergleichs abhängig.

➤ Wählen Sie für den Ausdruck immer den kompletten Versionsvergleich.
Das sind normalerweise alle Objekte zwischen *Eigenschaften* und der nächsten Gruppe, im Beispiel unten *01_Programm01*. Die Sortierung der Liste ist abhängig von der alphabetischen Reihenfolge der Objekte.

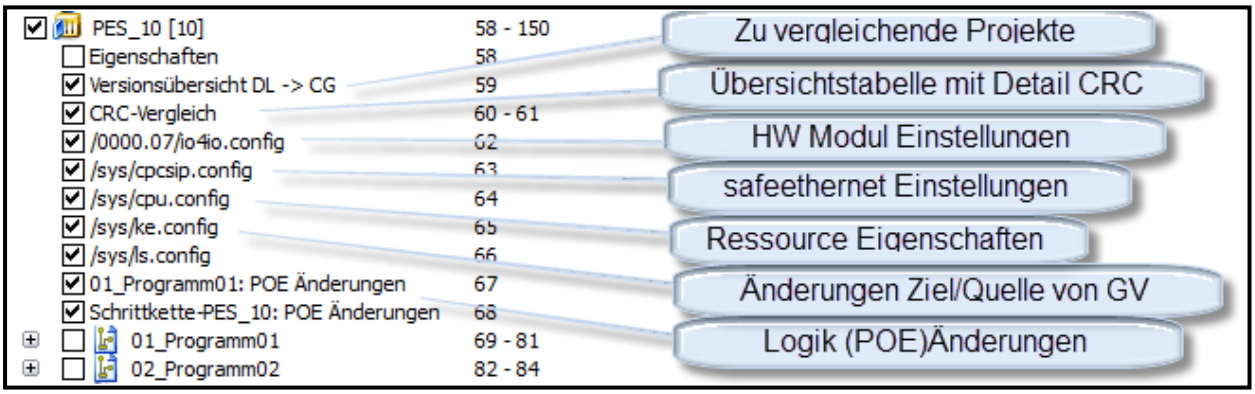


➤ Wählen Sie zusätzlich die geänderten Objekte aus, im Beispiel die POE *Schrittkette-PES_10*.



7.3.3.1 Übersicht der wichtigsten Dokumente des Versionsvergleichs

Das folgende Bild gibt einen Überblick über die wichtigsten Dokumente des Versionsvergleichs.



7.4 Dokumentation geänderter Objekte

Bei Änderungen in der Logik ist zu beachten, dass Funktionsbausteine und Funktionen in übergeordneten Bibliotheken abgelegt sein können. Elemente dieser Bibliotheken können in unterschiedlichen Ressourcen verwendet werden.

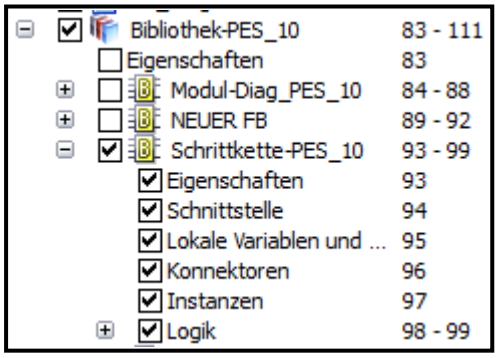
Daher müssen zwei Fälle unterschieden werden:

1. Die Bibliothek ist Bestandteil der Ressource und der geänderte Baustein wird nur in dieser Ressource verwendet. Der geänderte Baustein kann direkt ausgedruckt werden.
2. Die Bibliothek ist Bestandteil der Konfiguration, oder des Projekts und der geänderte Baustein wird in mehreren Ressourcen verwendet. In diesem Fall müssen die Auswirkungen der Änderungen auf alle betroffenen Ressourcen bedacht werden.
   Für den Versionsvergleich wird nur die Version des Bausteins der ausgewählten Ressource berücksichtigt.

> **!** Wenn der ausgedruckte Baustein in einer früheren Version in weiteren Ressourcen geladen ist, für die keine Codegenerierung durchgeführt wurde, kann der Baustein in diesen Ressourcen nicht mehr im Online-Test angezeigt werden.
> Es ist daher zu empfehlen, die kompletten Modifikationsprozeduren inklusive Laden für alle betroffenen Ressourcen durchzuführen.

7.4.1 Auswahl zu druckender Seiten

➤ Bausteine mit geringem Funktionsumfang wählen Sie am einfachsten direkt in der Bibliothek, inklusive aller Eigenschaften.

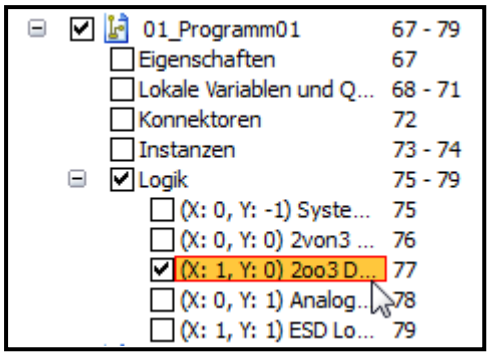


Die Position der Änderungen können Sie dem Protokoll des Versionsvergleichers entnehmen, das ebenfalls ausgedruckt wird.

Name	Typ	Position DL	Position CG
TON.Q => T_4	Zuweisung	Blatt: 0/0, Pos.: 59/84	Blatt: 0/0, Pos.: 59/84

➤ Bei Bausteinen mit größerem Funktionsumfang können Sie auch einzelne Bereiche, z. B. nur bestimmte Logikseiten, auswählen. Dazu müssen die Informationen des Versionsvergleichers genauer analysiert werden.

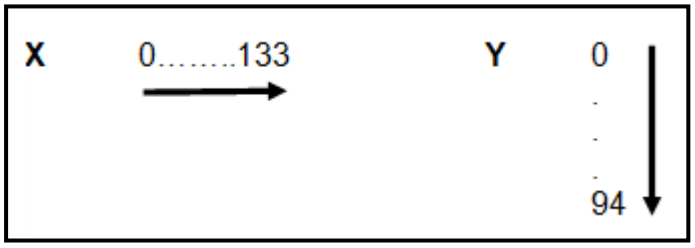
Im Ergebnis wird der Umfang des Ausdrucks geringer, da nur die wirklich relevanten Seiten ausgedruckt werden.



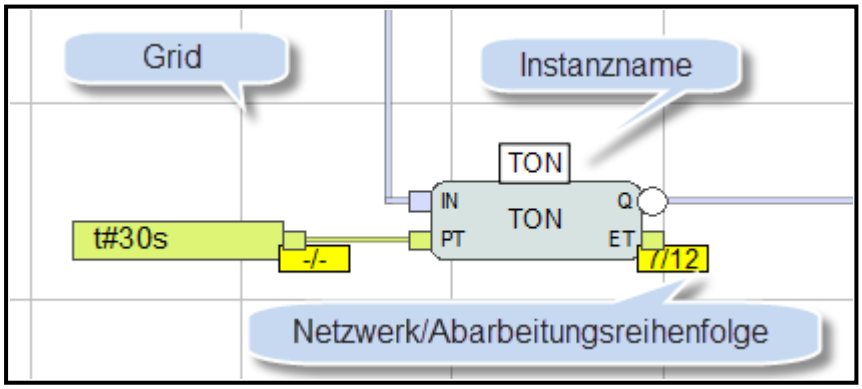
Damit die angezeigte Position im Ausdruck leichter zu finden ist, aktivieren Sie alle Optionen im Dialogfenster *Dokumentationserzeugungsparameter*, wie in*Dokumentation starten* beschrieben.

Die angezeigte Position ist die linke obere Ecke eines Objekts.

- Die X-Koordinate wird blattweise von links nach rechts gezählt.
- Die Y-Koordinate wird blattweise von oben nach unten gezählt.



Das gedruckte Raster (Grid) hat einen Abstand von 10 Einheiten.



7.4.2 Dokumentation von Hardware-Änderungen

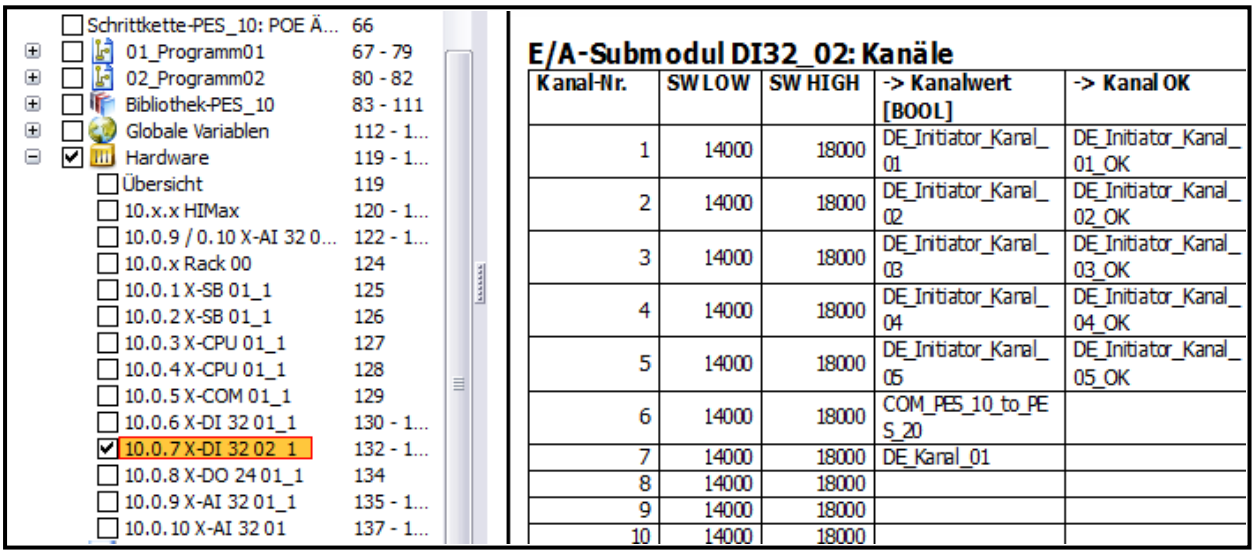
Änderungen von Hardware-Einstellungen, z. B. Modulparameter, werden bereits im Ergebnis des Versionsvergleichs dargestellt (Wert: alt/neu – im Beispiel DL/CG).

Daher ist meistens keine weitere Dokumentation nötig:

Kontext	Einstellung	Version DL	Version CG
10.0.7:1	LS-Limit	80000	65500
10.0.7:7	Kanal verwendet	nein	ja

Sollten diese Informationen nicht ausreichen, können alle Einstellungen zusätzlich so ausgedruckt werden, wie sie im Hardware-Editor erscheinen.

Im Beispiel wurden die Einstellungen für das Modul *X-DI-32 02* mit der *SRS 10.0.7* zum Drucken ausgewählt:



E/A-Submodul DI32_02: Kanäle

Kanal-Nr.	SW LOW	SW HIGH	-> Kanalwert [BOOL]	-> Kanal OK
1	14000	18000	DE_Initiator_Kanal_01	DE_Initiator_Kanal_01_OK
2	14000	18000	DE_Initiator_Kanal_02	DE_Initiator_Kanal_02_OK
3	14000	18000	DE_Initiator_Kanal_03	DE_Initiator_Kanal_03_OK
4	14000	18000	DE_Initiator_Kanal_04	DE_Initiator_Kanal_04_OK
5	14000	18000	DE_Initiator_Kanal_05	DE_Initiator_Kanal_05_OK
6	14000	18000	COM_PES_10_to_PES_20	
7	14000	18000	DE_Kanal_01	
8	14000	18000		
9	14000	18000		
10	14000	18000		

7.4.3 Dokumentation geänderter Variablenzuordnungen

Änderungen von Variablenzuordnungen werden bereits im Ergebnis des Versionsvergleichs dargestellt. Daher ist meistens keine weitere Dokumentation nötig.

Globale Variable	Variable	Quelle	Ziel	Art der Änderung
DE_Kanal_01	DE_Kanal_01	X-DI 32 02_1.(10.0.7)	01_Programm01	Quelle ist nun: 'X-DI 32 02_1.(10.0.7
DE_Kanal_01	DI-Kanal01 -> Kanalwert [BOOL]		X-DI 32 01_1.(10.0.6)	Variable 'DI-Kanal01 -> Kanalwert [
DE_Kanal_01	DI-Kanal07 -> Kanalwert [BOOL]		X-DI 32 02_1.(10.0.7)	Variable 'DI-Kanal07 -> Kanalwert [
DE_Kanal_01	REGISTER/Register-Out-00-Bit-00	X-DI 32 02_1.(10.0.7)	Standardprotokoll 'Modbus-Slave_1'	Quelle ist nun: 'X-DI 32 02_1.(10.0.7

Der Ausdruck der gesamten Liste der globalen Variablen ist nicht zu empfehlen, da diese Liste sehr umfangreich werden kann.

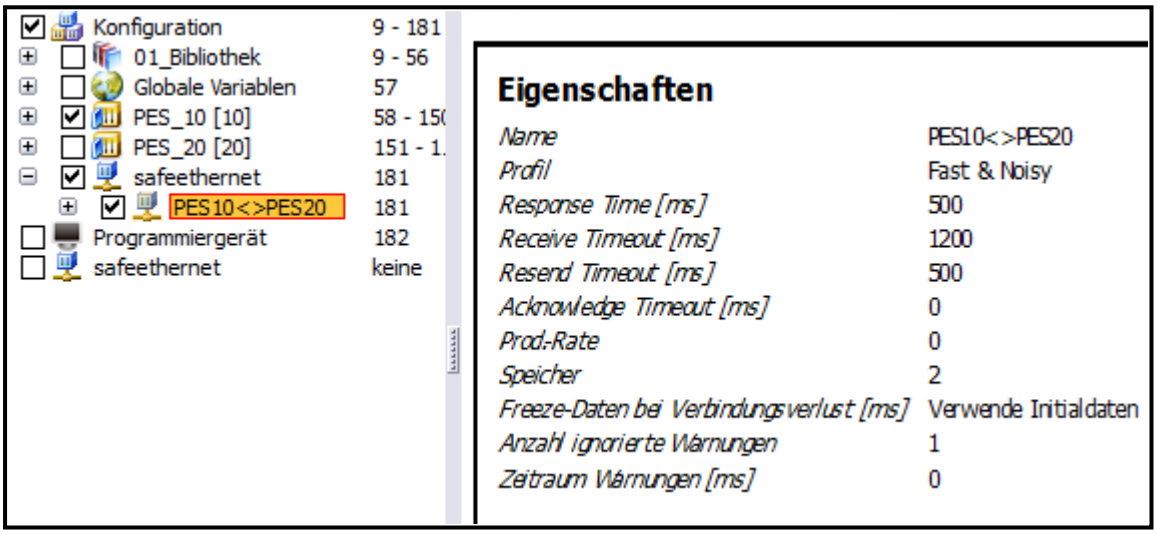
Name	Datentyp	Initialwert	Beschreibung	Zusat
Strukturelement			**Datentyp**	
Verwendung	**Strukturinfo**		**Info**	
DE_Initiator_Kanal_06	BOOL			
DE_Initiator_Kanal_06_OK	BOOL			
DE_Kanal_01	BOOL			
1x lesend	Externe POE		01_Programm01	
schreibend	HW [10.0.7 - 7]		-> Kanalwert [BOOL]	
lesend	Modbus-Slave-Set		Modbus-Slave_1 [Register-Out-00-Bit-00]	
DE_Kanal_02	BOOL			
1x lesend	Externe POE		01_Programm01	
schreibend	HW [10.0.6 - 2]		-> Kanalwert [BOOL]	

7.4.4 Dokumentation geänderter safeethernet Einstellungen

Änderungen von safe**ethernet** Einstellungen werden ab SILworX V5 bereits im Ergebnis des Versionsvergleichs dargestellt (Wert: alt/neu – im Beispiel DL/CG). Daher ist meistens keine weitere Dokumentation nötig.

Kontext	Einstellung	Version DL	Version CG
20.0x PES_20	Max. Receive Timeout	1000	1200
20.0x PES_20	Receive Timeout	1000	1200

Sollten diese Informationen nicht ausreichen, können alle Einstellungen zusätzlich so ausgedruckt werden, wie sie im safe**ethernet** Editor erscheinen.



HI 801 285 D

© 2017 HIMA Paul Hildebrandt GmbH

HIMA Paul Hildebrandt GmbH
Albert-Bassermann-Str. 28 | 68782 Brühl
Telefon 06202 709-0 | Telefax 06202 709-107
info@hima.com | www.hima.de

SAFETY
NONSTOP