Version Comparison

For Controllers Programmable with SILworX

SAFETY
NONSTOP

HIMA

All of the instructions and technical specifications in this manual have been written with great care and effective quality assurance measures have been implemented to ensure their validity. For questions, please contact HIMA directly. HIMA appreciates any suggestion on which information should be included in the manual.

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

For further information, refer to the CD-ROM and our website http://www.hima.de and http://www.hima.com.

Contact

HIMA contact details:

HIMA Paul Hildebrandt GmbH + Co KG
P.O. Box 1261
68777 Brühl, Germany
Phone: +49 6202 709-0
Fax: +49 6202 709-107
E-mail: info@hima.com

Table of Content

1 Principle

During the code generation, SILworX creates various files. This entity is referred to as *resource configuration*. The complete resource configuration is loaded to the resource whenever a download or reload is performed.

During a version comparison, different resource configurations are compared to one another and the differences between the individual files are detected. The result is based on the files that overwrite the executable code and has SIL 3 quality.

Basically, three types of resource configurations exist:

1. The created resource configuration is the result of the last code generation (→ Code generator).
2. The loaded resource configuration is the resource configuration transferred to the controller by performing a download (→ Download) or reload.
3. An unknown resource configuration represents any resource state that was exported and saved (→ IM).

2 Preparations

2.1. Programming Recommendations

When programming safety-relevant logic, the consequences of future changes should be taken into account in early programming stages. HIMA recommends the following measures to ensure that the version comparison results can be interpreted as easily as possible:

- Structured programming and process-specific partitioning of the logic in individual programs and function blocks.
- Individual and process-specific instance names (used function blocks).
- Copying logic objects while pressing the ctrl key generates long instance names if repeated multiple times. It is better to use **Copy** and **Paste**.
- The names assigned to logic pages should be meaningful and unique.
- Connectors should not be used within a large number of logic pages. This causes widely branched networks and thus reduces clarity, in particular during the version comparison.

2.2. Preparing the Version Comparison

A version comparison in the quality described in this document requires that the project code was already generated using *SILworX V4*.

Prior to performing the planned changes, one should create a project copy such that at the end a project [OLD] with no changes and a project [NEW] with changes are available.

Project [OLD] represents the inspected version that was approved for safety-related operation and is identified by a unique CRC. The entire project [OLD] should also be available in a printed form.

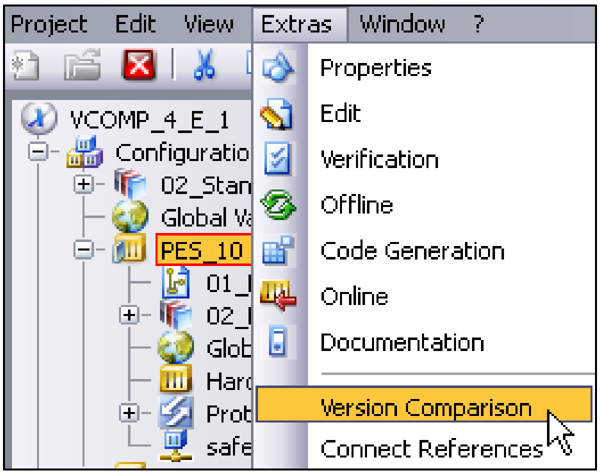
The version comparison has the objective to identify changes between project [OLD] and project [NEW].

To export the resource configuration

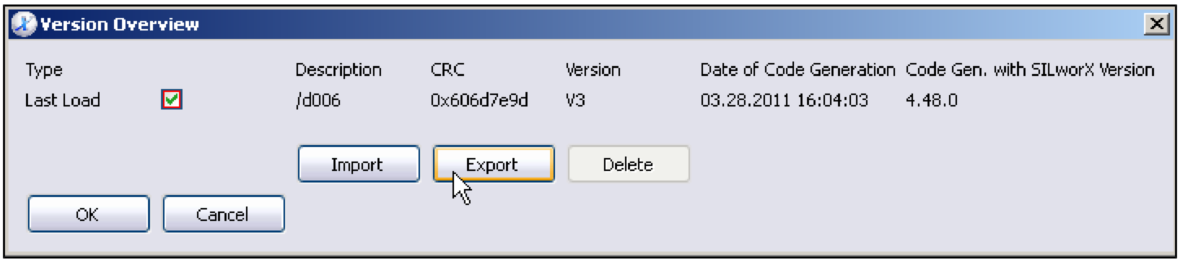
The resource configuration that was loaded last is exported from project [OLD]. Only this file can be used by the version comparator as a reference! Perform the following steps:

- In the structure tree, select the resource for which the version comparison should be performed.

- Click the **Extras,** *Version Comparison*menu functions**.** The *Version Overview* dialog box appears.



- Check the *Last Load*option in the *Version Overview* dialog box. This ensures the export of the last version loaded into the resource.



Type		Description	CRC	Version	Date of Code Generation	Code Gen. with SILworX Version
Last Load	☑	/d006	0x606d7e9d	V3	03.28.2011 16:04:03	4.48.0

- Click **Export**. The *Archive* dialog box appears. SILworX automatically creates an *Archive Name* with all relevant information.

Example: 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
Resource name	Project Name	DL = download = loaded file. Identical for reload and download.	Resource configuration CRC	Date and time of the code generation

☛ If required, adjust the *Archive Directory*. If desired, a comment can be added. Click then **OK** to save the archive.

3 Selecting Configurations for Comparison

A code comparison can be performed at any point in time. Depending on the requirements, the new code (CG) should already be compared to the last loaded version (DL) prior to loading a change.

The following configurations can be compared to one another:

- Imported configuration(s) → IM
- Configuration loaded last→ DL
- Configuration generated last→ CG

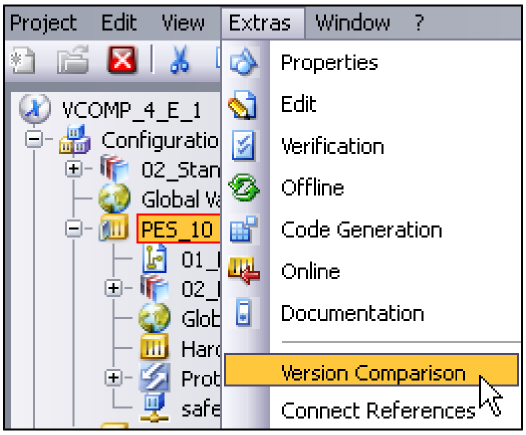
The final proof of the performed changes occurs after the tests have been successfully completed. The last loaded configuration (DL) is compared to the imported, original configuration (IM).

To import the resource configuration

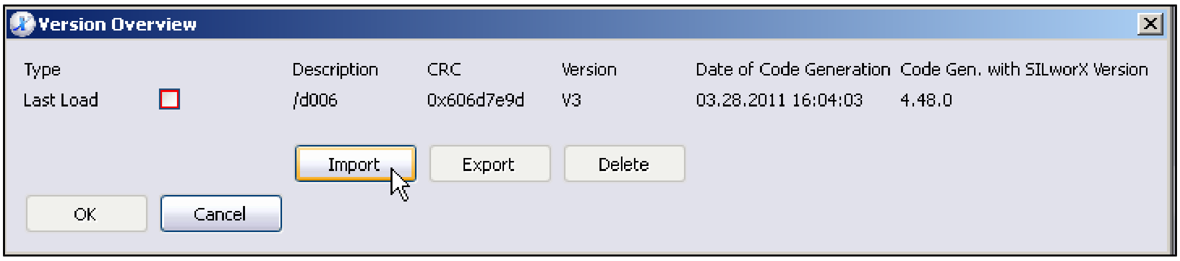
Make sure that a suitable resource configuration has been exported. To export a resource configuration, refer to Chapter Preparing the Version Comparison.

In the structure tree, select the resource for which the version comparison should be performed.

Click the **Extras, Version Comparison** menu function. The *Version Overview* dialog box opens.

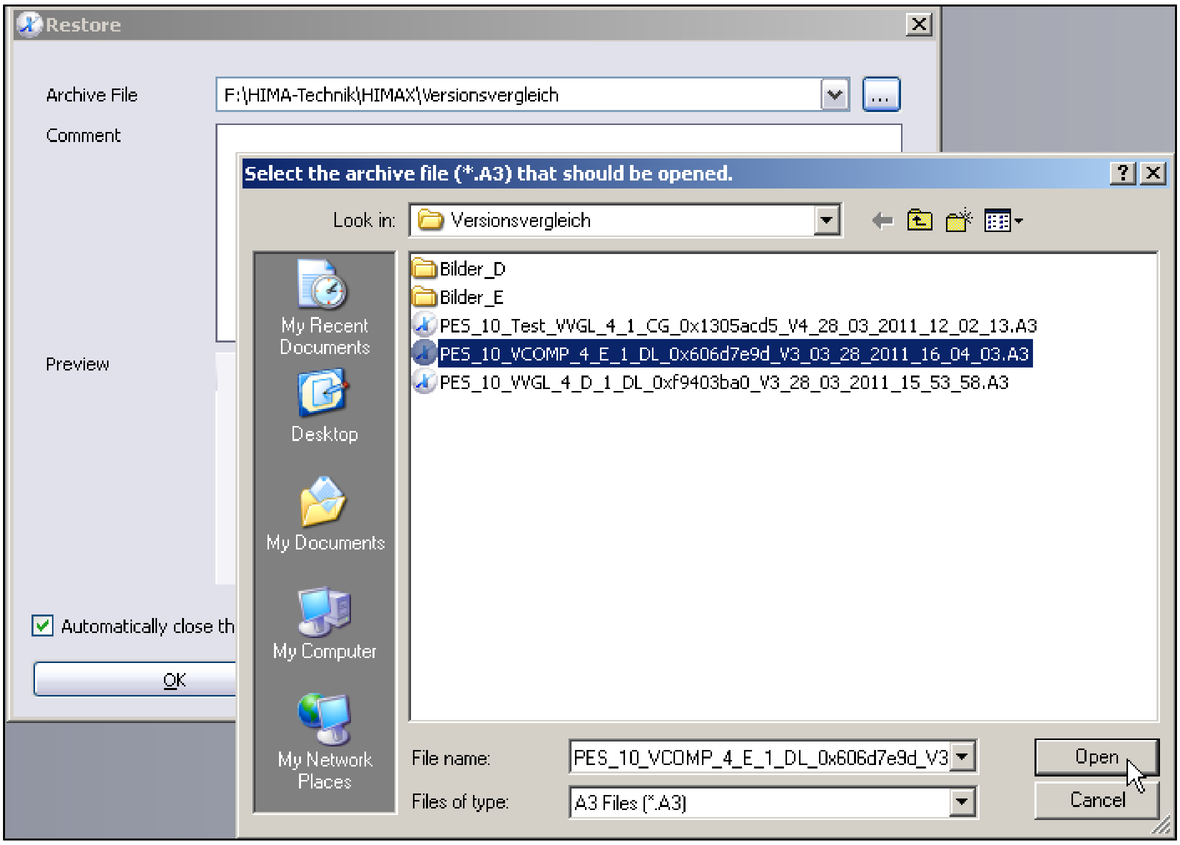


Click **Import**. The *Restore* dialog box appears.



Type		Description	CRC	Version	Date of Code Generation	Code Gen. with SILworX Version
Last Load	□	/d006	0x606d7e9d	V3	03.28.2011 16:04:03	4.48.0

Open the drop-down list of *Archive File* and select an archive.
If the drop-down list does not contain the desired archive, click the button to the right of the text field and select the file via the Windows dialog box.



Archive File: F:\HIMA-Technik\HIMAX\Versionsvergleich

Comment:

☑ Automatically close th

Select the archive file (*.A3) that should be opened.

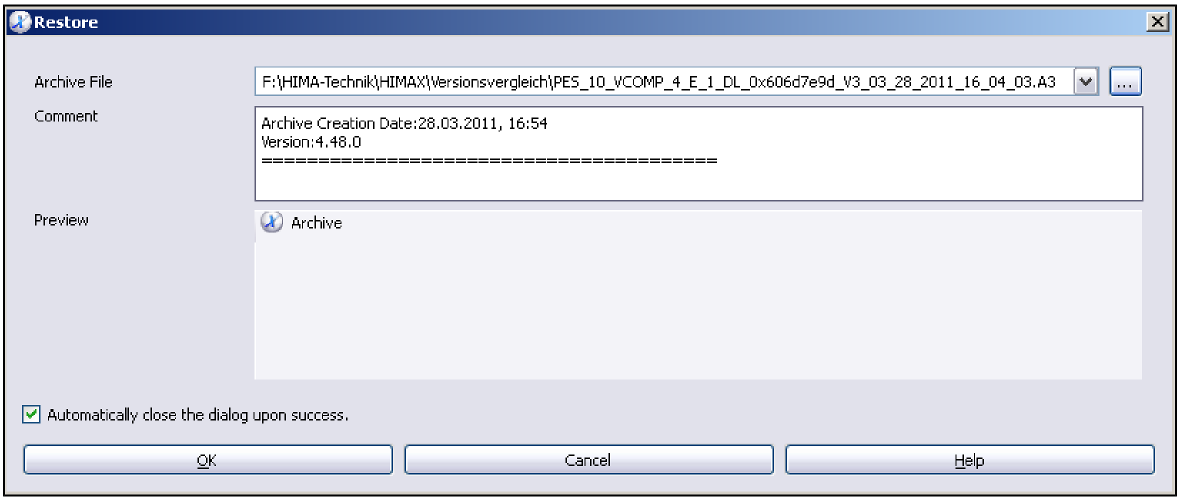
Look in: Versionsvergleich

Bilder_D
Bilder_E
PES_10_Test_VVGL_4_1_CG_0x1305acd5_V4_28_03_2011_12_02_13.A3
PES_10_VCOMP_4_E_1_DL_0x606d7e9d_V3_03_28_2011_16_04_03.A3
PES_10_VVGL_4_D_1_DL_0xf9403ba0_V3_28_03_2011_15_53_58.A3

File name: PES_10_VCOMP_4_E_1_DL_0x606d7e9d_V3

Files of type: A3 Files (*.A3)

Observe the details specified in the *Comment* and *Preview* fields. They can help identify the archive.



Archive File: F:\HIMA-Technik\HIMAX\Versionsvergleich\PES_10_VCOMP_4_E_1_DL_0x606d7e9d_V3_03_28_2011_16_04_03.A3

Comment:

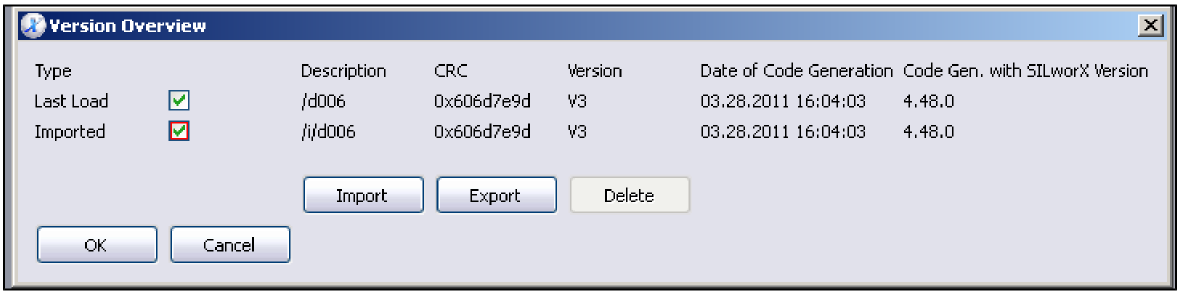
```
Archive Creation Date:28.03.2011, 16:54
Version:4.48.0
========================================
```

Preview: Archive

☑ Automatically close the dialog upon success.

Click **OK** to restore the archive.
The successfully restored archive is displayed and the *Version Overview* dialog box appears.

Select the versions that should be compared with one another. To do so, check the boxes to the right of *Last Load* and *Imported.*



Type		Description	CRC	Version	Date of Code Generation	Code Gen. with SILworX Version
Last Load	☑	/d006	0x606d7e9d	V3	03.28.2011 16:04:03	4.48.0
Imported	☑	/i/d006	0x606d7e9d	V3	03.28.2011 16:04:03	4.48.0

Click **OK** to start the version comparison. The result is presented in the version comparator in tabular form.

4 Information Displayed in the Version Comparison

4.1. CRC Comparison

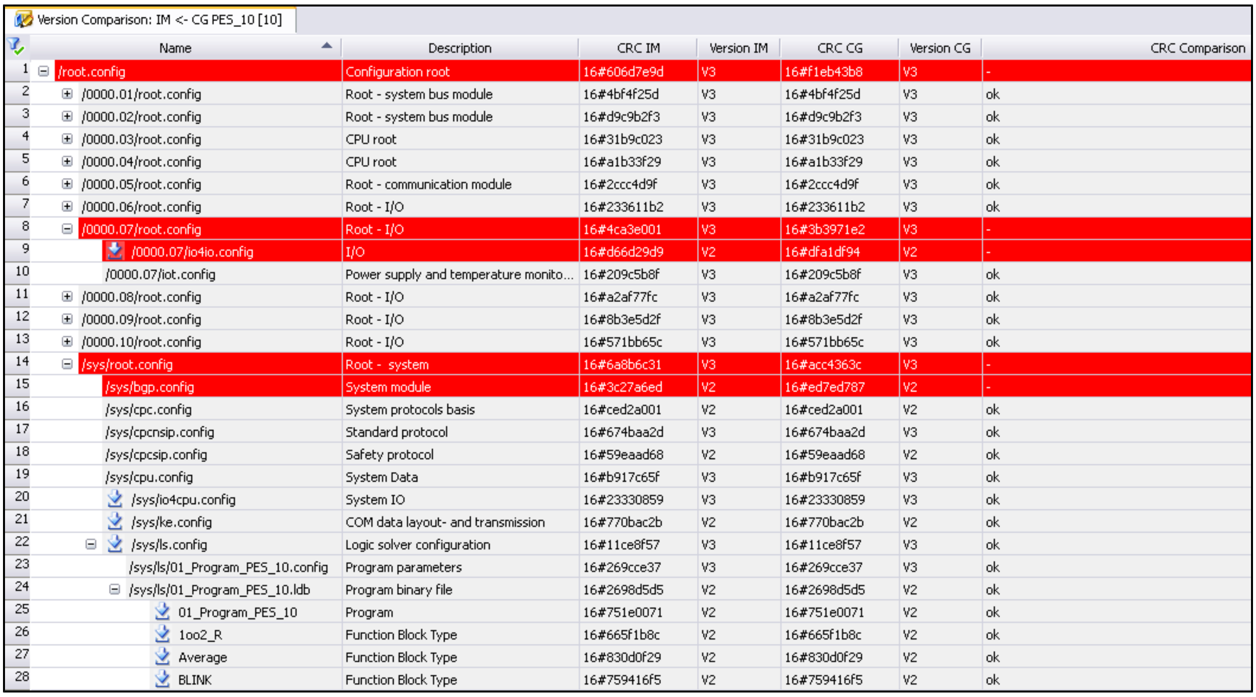
The version comparison is based on the checksums (CRCs) created by the code generator for the various function groups of the project. The function groups have a hierarchical structure and at list one configuration file.

The configuration files are contained in a list and highlighted in color if changes are performed.

Red	The function group described through this configuration file changed.
Yellow	The function group described through this configuration file is new or was deleted.

The first */root.config* line corresponds to the higher-level code version such as it is displayed in the logbook or in the Control Panel's system data. The higher-level code version includes the code versions of all function blocks. Click the (+) sign to the left of the line to display the subordinated objects.

A detailed check of the code comparison results is only necessary if the higher-level code version changed.



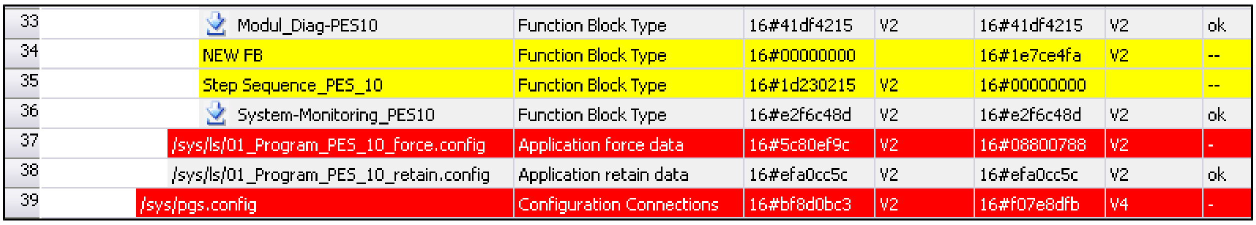
Version Comparison: IM <- CG PES_10 [10]

	Name	Description	CRC IM	Version IM	CRC CG	Version CG	CRC Comparison
1	⊟ /root.config	Configuration root	16#606d7e9d	V3	16#f1eb43b8	V3	-
2	⊞ /0000.01/root.config	Root - system bus module	16#4bf4f25d	V3	16#4bf4f25d	V3	ok
3	⊞ /0000.02/root.config	Root - system bus module	16#d9c9b2f3	V3	16#d9c9b2f3	V3	ok
4	⊞ /0000.03/root.config	CPU root	16#31b9c023	V3	16#31b9c023	V3	ok
5	⊞ /0000.04/root.config	CPU root	16#a1b33f29	V3	16#a1b33f29	V3	ok
6	⊞ /0000.05/root.config	Root - communication module	16#2ccc4d9f	V3	16#2ccc4d9f	V3	ok
7	⊞ /0000.06/root.config	Root - I/O	16#233611b2	V3	16#233611b2	V3	ok
8	⊟ /0000.07/root.config	Root - I/O	16#4ca3e001	V3	16#3b3971e2	V3	-
9	/0000.07/io4io.config	I/O	16#d66d29d9	V2	16#dfa1df94	V2	-
10	/0000.07/iot.config	Power supply and temperature monito...	16#209c5b8f	V3	16#209c5b8f	V3	ok
11	⊞ /0000.08/root.config	Root - I/O	16#a2af77fc	V3	16#a2af77fc	V3	ok
12	⊞ /0000.09/root.config	Root - I/O	16#8b3e5d2f	V3	16#8b3e5d2f	V3	ok
13	⊞ /0000.10/root.config	Root - I/O	16#571bb65c	V3	16#571bb65c	V3	ok
14	⊟ /sys/root.config	Root - system	16#6a8b6c31	V3	16#acc4363c	V3	-
15	/sys/bgp.config	System module	16#3c27a6ed	V2	16#ed7ed787	V2	-
16	/sys/cpc.config	System protocols basis	16#ced2a001	V2	16#ced2a001	V2	ok
17	/sys/cpcnsip.config	Standard protocol	16#674baa2d	V3	16#674baa2d	V3	ok
18	/sys/cpcsip.config	Safety protocol	16#59eaad68	V2	16#59eaad68	V2	ok
19	/sys/cpu.config	System Data	16#b917c65f	V3	16#b917c65f	V3	ok
20	/sys/io4cpu.config	System IO	16#23330859	V3	16#23330859	V3	ok
21	/sys/ke.config	COM data layout- and transmission	16#770bac2b	V2	16#770bac2b	V2	ok
22	⊟ /sys/ls.config	Logic solver configuration	16#11ce8f57	V3	16#11ce8f57	V3	ok
23	/sys/ls/01_Program_PES_10.config	Program parameters	16#269cce37	V3	16#269cce37	V3	ok
24	⊟ /sys/ls/01_Program_PES_10.ldb	Program binary file	16#2698d5d5	V2	16#2698d5d5	V2	ok
25	01_Program_PES_10	Program	16#751e0071	V2	16#751e0071	V2	ok
26	1oo2_R	Function Block Type	16#665f1b8c	V2	16#665f1b8c	V2	ok
27	Average	Function Block Type	16#830d0f29	V2	16#830d0f29	V2	ok
28	BLINK	Function Block Type	16#759416f5	V2	16#759416f5	V2	ok

NOTE:	In exceptional cases, changes to a function group may have no functional effect, see Renaming Variables. No further check is necessary in this case, since the code version did not change.

Indication of added, deleted and changed function groups

The following picture shows added and deleted function blocks (highlighted in yellow) as well as modified function blocks (highlighted in red).



33	Modul_Diag-PES10	Function Block Type	16#41df4215	V2	16#41df4215	V2	ok
34	NEW FB	Function Block Type	16#00000000		16#1e7ce4fa	V2	--
35	Step Sequence_PES_10	Function Block Type	16#1d230215	V2	16#00000000		--
36	System-Monitoring_PES10	Function Block Type	16#e2f6c48d	V2	16#e2f6c48d	V2	ok
37	/sys/ls/01_Program_PES_10_force.config	Application force data	16#5c80ef9c	V2	16#08800788	V2	-
38	/sys/ls/01_Program_PES_10_retain.config	Application retain data	16#efa0cc5c	V2	16#efa0cc5c	V2	ok
39	/sys/pgs.config	Configuration Connections	16#bf8d0bc3	V2	16#f07e8dfb	V4	-

4.2. Content and Relevance of the Most Important Files

4.2.1 Hardware, Modules

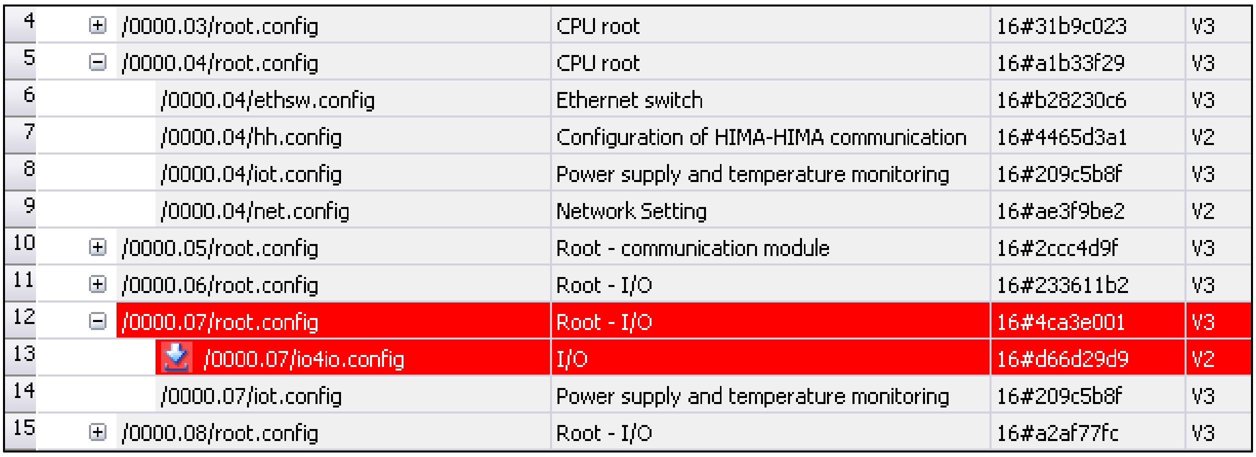
For each module, a file exists that groups all configuration files associated with it. Changing a module configuration (e.g., IP settings, scaling values, line monitoring, channel activation, etc.), modifies the file content.

Evaluation details are specified in Chapter io4io.config.

The information is displayed as follows:

Column	Description
Name	Module position in the Rack.Slot format, followed by the configuration file name.
Description	Short description of the configuration file.
CRC IM	Checksum of the imported configuration file.
Version IM	Minimum operating system version required for the module (imported), see Chapter Operating System Version Required for an Object.
CRC CG	Checksum of the configuration file created by the code generator.
Version CG	Like Version IM, but for the configuration file created by the code generator.

Example



4	⊞ /0000.03/root.config	CPU root	16#31b9c023	V3
5	⊟ /0000.04/root.config	CPU root	16#a1b33f29	V3
6	/0000.04/ethsw.config	Ethernet switch	16#b28230c6	V3
7	/0000.04/hh.config	Configuration of HIMA-HIMA communication	16#4465d3a1	V2
8	/0000.04/iot.config	Power supply and temperature monitoring	16#209c5b8f	V3
9	/0000.04/net.config	Network Setting	16#ae3f9be2	V2
10	⊞ /0000.05/root.config	Root - communication module	16#2ccc4d9f	V3
11	⊞ /0000.06/root.config	Root - I/O	16#233611b2	V3
12	⊟ /0000.07/root.config	Root - I/O	16#4ca3e001	V3
13	/0000.07/io4io.config	I/O	16#d66d29d9	V2
14	/0000.07/iot.config	Power supply and temperature monitoring	16#209c5b8f	V3
15	⊞ /0000.08/root.config	Root - I/O	16#a2af77fc	V3

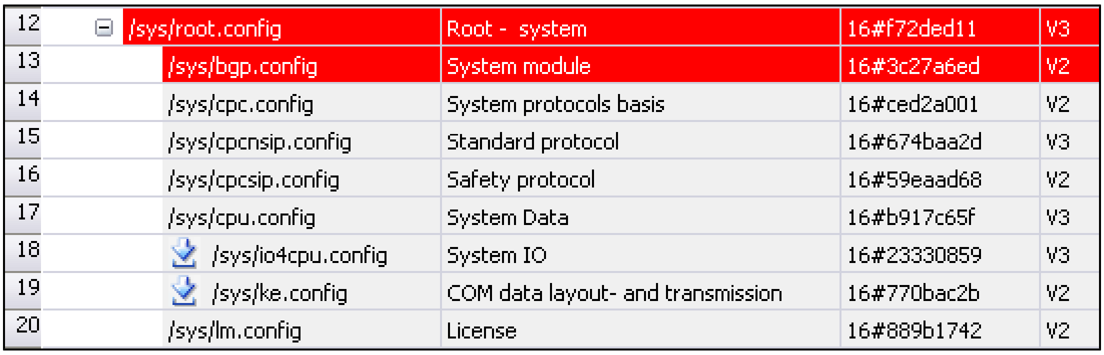
Line	Configuration file	Description
5	/0000.04/root.config	Main file of the CPU module in rack 0, slot 4. This configuration file is referenced to subordinated configuration files and always changes if one of the subordinated file is modified.
6	/0000.04/ethsw.config	Properties of the Ethernet switches on CPU modules
7	/0000.04/hh.config:	safeethernet communication properties for CPU modules.
8	/0000.04/iot.config	Power supply (single or redundant) and temperature monitoring of the CPU module.
9	/0000.04/net.config	Network settings of the CPU module.

Line	Configuration file	Description
10	/0000.05/root.config	Main file of the communication module in rack 0, slot 5. See also *Line 5*.
11	/0000.06/root.config	Main file of the I/O module in rack 0, slot 6. See also *Line 5*.
12	/0000.07/root.config	Main file of the I/O module in rack 0, slot 7. See also *Line 5*. Red: The version comparator detected a difference between the imported and the generated configuration files.
13	/0000.07/io4io.config	Configuration file for the I/O module Red: The version comparator detected a difference between the imported and the generated configuration files.
14	/0000.07/iot.config	Power supply (single or redundant) and temperature monitoring of the I/O module.

4.2.2 CPU Configuration and System Data

Central and higher-level data of the CPU module are grouped in the *sys/root.config* configuration file.

Example



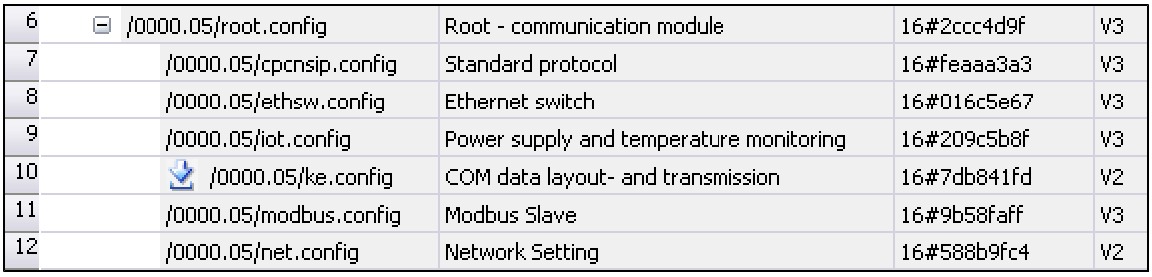
12	⊟ /sys/root.config	Root - system	16#f72ded11	V3
13	/sys/bgp.config	System module	16#3c27a6ed	V2
14	/sys/cpc.config	System protocols basis	16#ced2a001	V2
15	/sys/cpcnsip.config	Standard protocol	16#674baa2d	V3
16	/sys/cpcsip.config	Safety protocol	16#59eaad68	V2
17	/sys/cpu.config	System Data	16#b917c65f	V3
18	/sys/io4cpu.config	System IO	16#23330859	V3
19	/sys/ke.config	COM data layout- and transmission	16#770bac2b	V2
20	/sys/lm.config	License	16#889b1742	V2

Line	Configuration file	Description
18	/sys/root.config	Main file for the CPU module. This configuration file is referenced to sub-ordinated configuration files and always changes if one of the subordinated file is modified.
19	/sys/bgp.config	The module configuration describes all module data such as the assignment of modules to slots. This file almost always changes if the module data is modified (→ Hardware, Modules).
20	/sys/cpc.config	Number of protocols, communication time slice ASYNC, SYNC.
21	/sys/cpcnsip.config	General rules for the protocols used to transfer data from the COM module to the CPU module, e.g., behavior if the connection is lost.
22	/sys/cpcsip.config	safeethernet parameters, connection properties.
23	/sys/cpu.config	Resource settings such as allowed actions, safety time or watchdog time.
24	/sys/io4cpu.config	Redundancy details of the I/O modules, scaling of analog values and counter inputs.
25	ke.config	Configuration file for assigning (using) global variables to hardware, protocols, POUs, etc. (*ke* indicates the communication endpoint, i.e., global variable).
26	lm.config	Configuration file for license management.

4.2.3 COM Configuration and Protocols

The COM module data (e.g., protocols, interfaces, etc.) are saved in individual configuration files subordinated to *root.config*, the main configuration file for the COM module.

Example



6	⊟ /0000.05/root.config	Root - communication module	16#2ccc4d9f	V3
7	/0000.05/cpcnsip.config	Standard protocol	16#feaaa3a3	V3
8	/0000.05/ethsw.config	Ethernet switch	16#016c5e67	V3
9	/0000.05/iot.config	Power supply and temperature monitoring	16#209c5b8f	V3
10	/0000.05/ke.config	COM data layout- and transmission	16#7db841fd	V2
11	/0000.05/modbus.config	Modbus Slave	16#9b58faff	V3
12	/0000.05/net.config	Network Setting	16#588b9fc4	V2

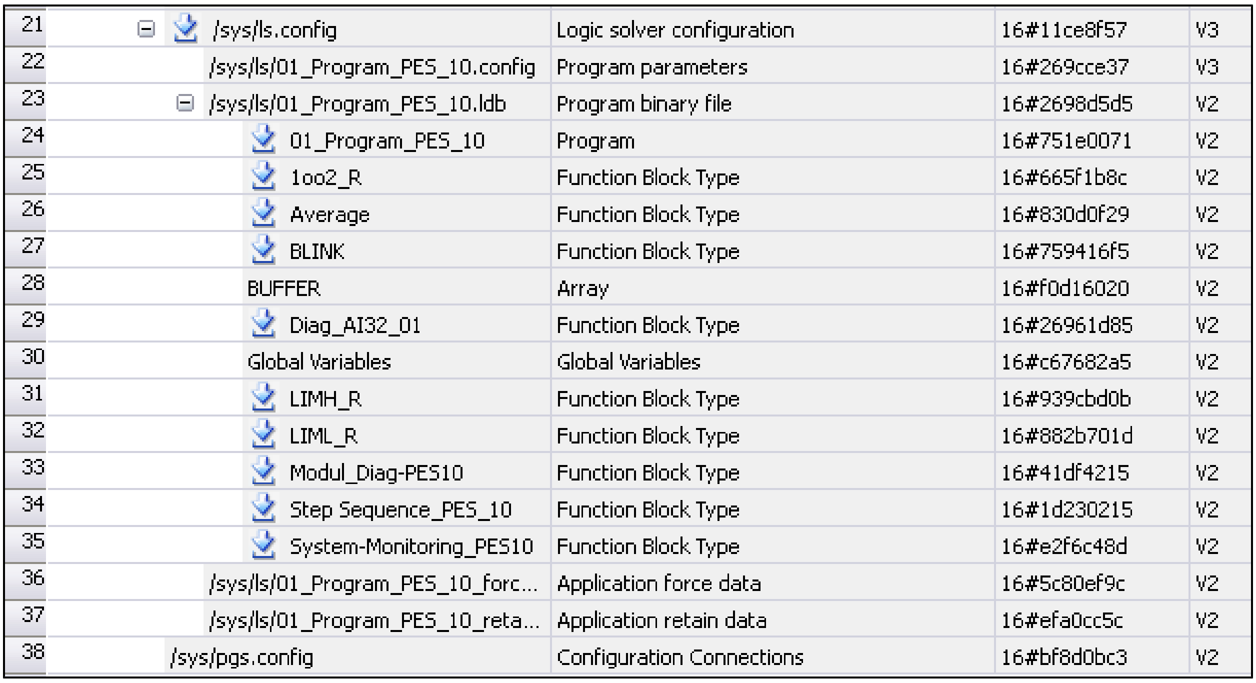
Line	Configuration file	Description
6	/0000.05/root.config	Main file for the COM module. This configuration file is referenced to subordinated configuration files and always changes if one of the subordinated file is modified.
7	/0000.05/cpcnsip.config	General rules for the protocols used to transfer data from the COM module to the CPU module, e.g., behavior if the connection is lost.
8	/0000.05/ethsw.config	Properties of the Ethernet switches on COM modules
9	/0000.05/iot.config	Power supply (single or redundant) and temperature monitoring of the COM module.
10	/0000.05/ke.config	Configuration file for reading and writing global variables in protocols (*ke* indicates the communication endpoint, i.e., global variable).
11	/0000.05/modbus.config:	Configuration file for the Modbus protocol properties
12	/0000.05/net.config	Configuration file for the network settings of this COM module.

4.2.4 Logic Data

The checksums of the displayed function blocks (POUs) are so-called source code CRCs. Changes to the executable code of a function block always result in a changed binary file (.ldb) for the program in which the function block is used.

Not all function block changes affect the executable code, e.g., renaming a local variable. The code comparison identifies a change in the source code, i.e., the line becomes red, but the binary file (loadable) does not change. In this case, no code-relevant changes are to be expected, i.e., no additional check is necessary.

Example



21	/sys/ls.config	Logic solver configuration	16#11ce8f57	V3
22	/sys/ls/01_Program_PES_10.config	Program parameters	16#269cce37	V3
23	/sys/ls/01_Program_PES_10.ldb	Program binary file	16#2698d5d5	V2
24	01_Program_PES_10	Program	16#751e0071	V2
25	1oo2_R	Function Block Type	16#665f1b8c	V2
26	Average	Function Block Type	16#830d0f29	V2
27	BLINK	Function Block Type	16#759416f5	V2
28	BUFFER	Array	16#f0d16020	V2
29	Diag_AI32_01	Function Block Type	16#26961d85	V2
30	Global Variables	Global Variables	16#c67682a5	V2
31	LIMH_R	Function Block Type	16#939cbd0b	V2
32	LIML_R	Function Block Type	16#882b701d	V2
33	Modul_Diag-PES10	Function Block Type	16#41df4215	V2
34	Step Sequence_PES_10	Function Block Type	16#1d230215	V2
35	System-Monitoring_PES10	Function Block Type	16#e2f6c48d	V2
36	/sys/ls/01_Program_PES_10_forc...	Application force data	16#5c80ef9c	V2
37	/sys/ls/01_Program_PES_10_reta...	Application retain data	16#efa0cc5c	V2
38	/sys/pgs.config	Configuration Connections	16#bf8d0bc3	V2

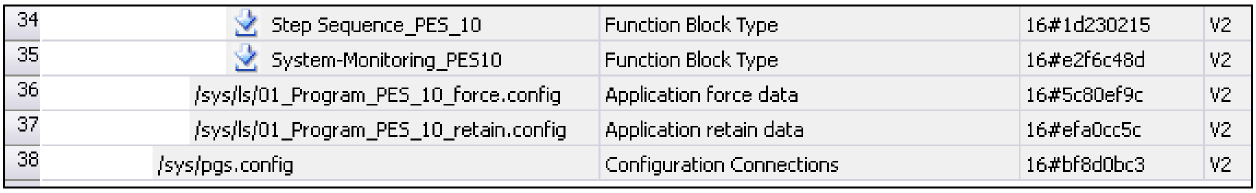
Line	Configuration file, object data	Description
27	/sys/ls.config	Main logic file (logic solver). This configuration file is referenced to subordinated configuration files and always changes if one of the subordinated file is modified, e.g., multitasking properties.
28	/sys/ls/Programm.config	Program properties, multitasking settings, allowed actions, etc.
29	/sys/ls/Programm.ldb	The binary file (loadable) is the executable code of the entire logic and always changes if changes are performed to the logic.
30	01_Programm01	CRC of the program as POU.
31	-	CRC of the function block as POU.
36	BUFFER	CRC of a user-defined data type.
38	Global Variables	Properties for how global variables are used in function blocks, e.g., data type, sequence (sorting order), etc. Should a change performed to one of this property affect the executable

Line	Configuration file, object data	Description
		code, the binary file (.ldb) also changes. In this case, further information is displayed in the detail view of the concerned function block. If applicable, changes to the *ke.config*file are also displayed. Further details can also be found in the *ke.config* file.
45	/sys/ls/force.config	Additional information for supporting forcing in the logic. It also possibly changes if the use of a global variable is modified in the logic. For further details, refer to the detail view of *ke.config.*
46	/sys/ls/retain.config	Retain information about the global variables used in the logic. It also possibly changes if the use of a global variable is modified in the logic. For further details, refer to the detail view of *ke.config.*

4.2.5 PGS Data (Config. Connections), User Management

The module protocol data, the RIO connections and the user management are saved in the *pgs.config* configuration file.

Example



34	Step Sequence_PES_10	Function Block Type	16#1d230215	V2
35	System-Monitoring_PES10	Function Block Type	16#e2f6c48d	V2
36	/sys/ls/01_Program_PES_10_force.config	Application force data	16#5c80ef9c	V2
37	/sys/ls/01_Program_PES_10_retain.config	Application retain data	16#efa0cc5c	V2
38	/sys/pgs.config	Configuration Connections	16#bf8d0bc3	V2

Line	Configuration file	Description
41	/sys/pgs.config	Configuration connection data, e.g., Max. Duration of Configuration Connections, PES User Management.

4.2.6 Operating System Version Required for an Object

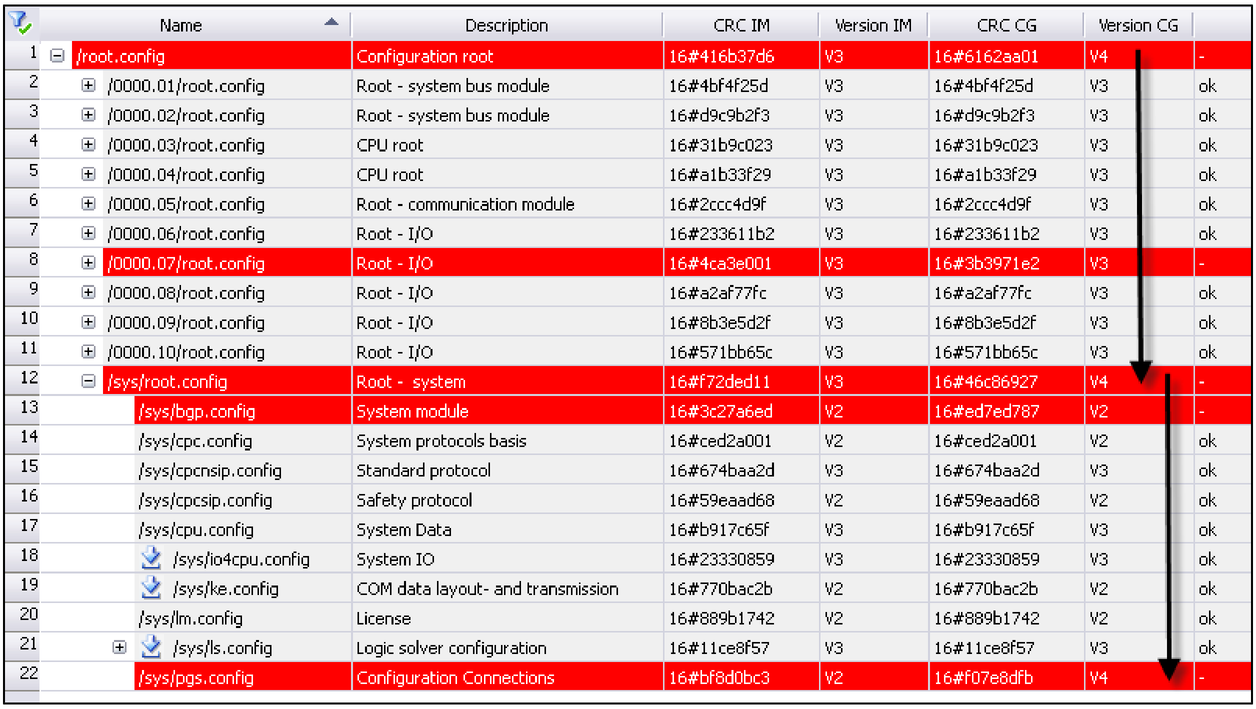
The checksum as well as the operating system version are displayed for each configuration file. The operating system version required for a module depends on the used functions. For instance, *Max. Duration of Configuration Connections* (PGS time) can only be modified with operating system V4 and beyond (see the example below).

The CRC and the required operating system version are already displayed in the logbook upon completion of the code generation.

In the */root.config* line, i.e., the main configuration file, the version comparison also displays the required operating system version.

A configuration can only be loaded into a controller if all modules in use are equipped with at least the operating system version determined by the code generator. Modules with unsuitable operating system versions reject the configuration as invalid.
For all files that cannot be allocated to an individual module according to the SRS format (System Rack Slot), the configuration is rejected by the CPU. In such a case, the CPU must be equipped with the required operating system version.

To display the subordinated objects, click the (+) sign to the right of the line number in the hierarchical list of the configuration files.



	Name	Description	CRC IM	Version IM	CRC CG	Version CG	
1	⊟ /root.config	Configuration root	16#416b37d6	V3	16#6162aa01	V4	-
2	⊞ /0000.01/root.config	Root - system bus module	16#4bf4f25d	V3	16#4bf4f25d	V3	ok
3	⊞ /0000.02/root.config	Root - system bus module	16#d9c9b2f3	V3	16#d9c9b2f3	V3	ok
4	⊞ /0000.03/root.config	CPU root	16#31b9c023	V3	16#31b9c023	V3	ok
5	⊞ /0000.04/root.config	CPU root	16#a1b33f29	V3	16#a1b33f29	V3	ok
6	⊞ /0000.05/root.config	Root - communication module	16#2ccc4d9f	V3	16#2ccc4d9f	V3	ok
7	⊞ /0000.06/root.config	Root - I/O	16#233611b2	V3	16#233611b2	V3	ok
8	⊞ /0000.07/root.config	Root - I/O	16#4ca3e001	V3	16#3b3971e2	V3	-
9	⊞ /0000.08/root.config	Root - I/O	16#a2af77fc	V3	16#a2af77fc	V3	ok
10	⊞ /0000.09/root.config	Root - I/O	16#8b3e5d2f	V3	16#8b3e5d2f	V3	ok
11	⊞ /0000.10/root.config	Root - I/O	16#571bb65c	V3	16#571bb65c	V3	ok
12	⊟ /sys/root.config	Root - system	16#f72ded11	V3	16#46c86927	V4	-
13	/sys/bgp.config	System module	16#3c27a6ed	V2	16#ed7ed787	V2	-
14	/sys/cpc.config	System protocols basis	16#ced2a001	V2	16#ced2a001	V2	ok
15	/sys/cpcnsip.config	Standard protocol	16#674baa2d	V3	16#674baa2d	V3	ok
16	/sys/cpcsip.config	Safety protocol	16#59eaad68	V2	16#59eaad68	V2	ok
17	/sys/cpu.config	System Data	16#b917c65f	V3	16#b917c65f	V3	ok
18	/sys/io4cpu.config	System IO	16#23330859	V3	16#23330859	V3	ok
19	/sys/ke.config	COM data layout- and transmission	16#770bac2b	V2	16#770bac2b	V2	ok
20	/sys/lm.config	License	16#889b1742	V2	16#889b1742	V2	ok
21	⊞ /sys/ls.config	Logic solver configuration	16#11ce8f57	V3	16#11ce8f57	V3	ok
22	/sys/pgs.config	Configuration Connections	16#bf8d0bc3	V2	16#f07e8dfb	V4	-

5 Detailed Evaluation

5.1. Hardware Changes

5.1.1 io4io.config

The *io4io.config* configuration file for an I/O module changes if changes are performed to the configuration data of the I/O module . This includes:

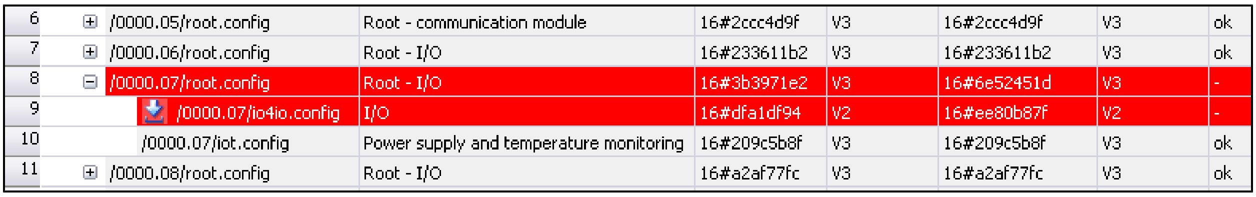
- Changes to the *Module* tab, e.g., noise blanking.
- Changes to the *I/O Submodule*, e.g., a supply's activation.
- Changes to fixed values, such as OC Limit or scaling values.

Note that assigning a module system variable to a global variable does not result in changed configuration data.

Because the *Channel Used* parameter is implicitly configured, changing, adding and deleting a global variable causes changes in the configuration files of the following modules:

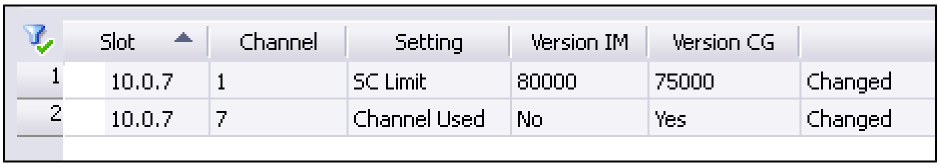
- X-DI 32 02
- X-DI 32 05
- X-AI 32 01
- X-AI 32 02
- X-CI 24 01

Example



6	/0000.05/root.config	Root - communication module	16#2ccc4d9f	V3	16#2ccc4d9f	V3	ok
7	/0000.06/root.config	Root - I/O	16#233611b2	V3	16#233611b2	V3	ok
8	/0000.07/root.config	Root - I/O	16#3b3971e2	V3	16#6e52451d	V3	-
9	/0000.07/io4io.config	I/O	16#dfa1df94	V2	16#ee80b87f	V2	-
10	/0000.07/iot.config	Power supply and temperature monitoring	16#209c5b8f	V3	16#209c5b8f	V3	ok
11	/0000.08/root.config	Root - I/O	16#a2af77fc	V3	16#a2af77fc	V3	ok

Double-click the *io4io.config* configuration file line to open the detail view.



	Slot	Channel	Setting	Version IM	Version CG	
1	10.0.7	1	SC Limit	80000	75000	Changed
2	10.0.7	7	Channel Used	No	Yes	Changed

The detailed list specifies the following information:

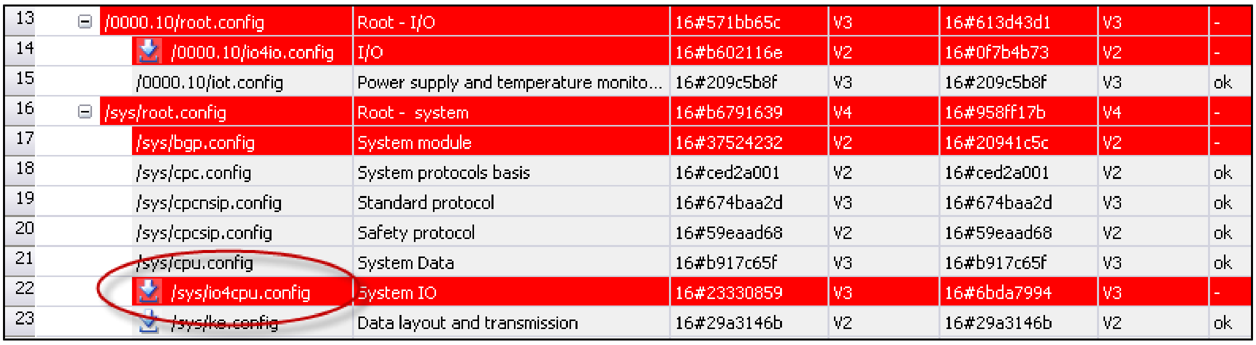
Column	Description
Slot	I/O module slot in the System.Rack.Slot format.
Channel	I/O module channel concerned.
Setting	Relevance of the parameter or function.
Version IM	Value in the imported configuration file version.
Version CG	Value in the configuration file version created by the code generator.
Version DL	Value in the loaded configuration file version.

5.1.2 io4cpu.config

Some changes affect the io4io.config configuration file as well as the *io4cpu.config* configuration file. For instance, changes to the scaling values of input modules.

Because the CPU module supports the I/O modules in the scaling calculation, changes to the scaling values affect both the *io4io.config* and *io4cpu.config* configuration files (see the following picture).

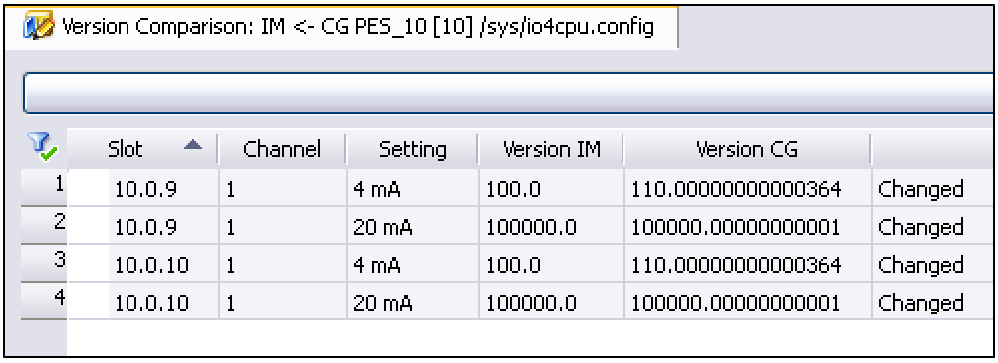
Example



13	⊟ /0000.10/root.config	Root - I/O	16#571bb65c	V3	16#613d43d1	V3	-
14	/0000.10/io4io.config	I/O	16#b602116e	V2	16#0f7b4b73	V2	-
15	/0000.10/iot.config	Power supply and temperature monito...	16#209c5b8f	V3	16#209c5b8f	V3	ok
16	⊟ /sys/root.config	Root - system	16#b6791639	V4	16#958ff17b	V4	-
17	/sys/bgp.config	System module	16#37524232	V2	16#20941c5c	V2	-
18	/sys/cpc.config	System protocols basis	16#ced2a001	V2	16#ced2a001	V2	ok
19	/sys/cpcnsip.config	Standard protocol	16#674baa2d	V3	16#674baa2d	V3	ok
20	/sys/cpcsip.config	Safety protocol	16#59eaad68	V2	16#59eaad68	V2	ok
21	/sys/cpu.config	System Data	16#b917c65f	V3	16#b917c65f	V3	ok
22	/sys/io4cpu.config	System IO	16#23330859	V3	16#6bda7994	V3	-
23	/sys/ke.config	Data layout and transmission	16#29a3146b	V2	16#29a3146b	V2	ok

Double-click the *io4cpu.config* configuration file line to open the detail view.

In the following picture, the modules in slot 9 and slot 10 are redundantly connected such that the change affects both modules.



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

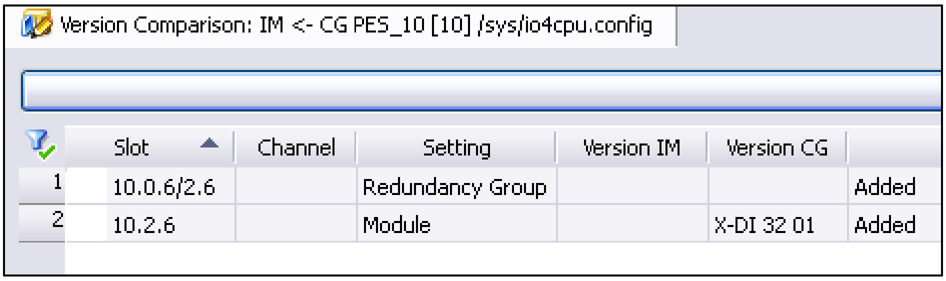
	Slot	Channel	Setting	Version IM	Version CG	
1	10.0.9	1	4 mA	100.0	110.00000000000364	Changed
2	10.0.9	1	20 mA	100000.0	100000.00000000001	Changed
3	10.0.10	1	4 mA	100.0	110.00000000000364	Changed
4	10.0.10	1	20 mA	100000.0	100000.00000000001	Changed

NOTE: As a result of the internal structure with mantissa and exponent, REAL numbers might be represented with decimal places (see *Version CG* column).

The decimal places can be ignored!

Even if only one parameter has changed, usually both the *4 mA* and the *20 mA* base points are affected by the underlying mathematical function.

The following picture shows an X-DI 32 01 module in a redundancy group. Since the redundancy evaluation is performed in the CPU module, this action also affects the *io4cpu.config* file.



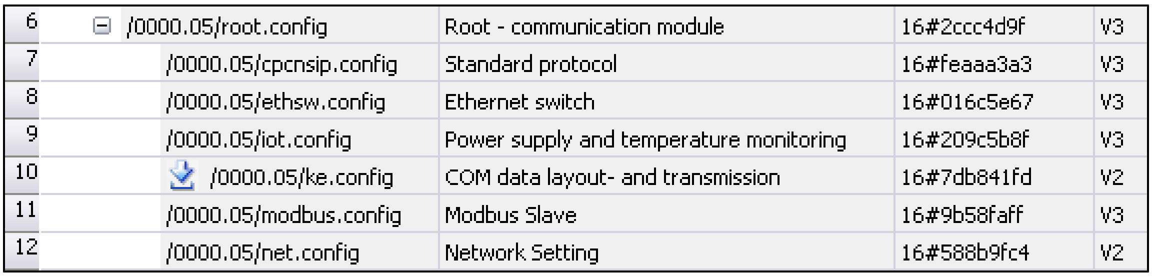
Version Comparison: IM <- CG PES_10 [10] /sys/io4cpu.config

	Slot	Channel	Setting	Version IM	Version CG	
1	10.0.6/2.6		Redundancy Group			Added
2	10.2.6		Module		X-DI 32 01	Added

5.1.3 cpnsip, ethsw, iot

cpnsip.config

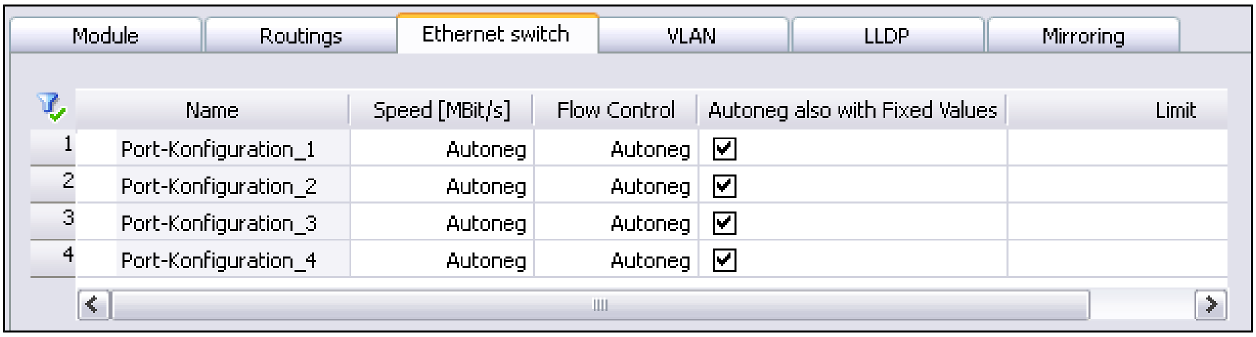
All settings for transferring non-safe protocol data between CPU und COM module are saved in the *cpnsip.config*configuration file.



6	⊟ /0000.05/root.config	Root - communication module	16#2ccc4d9f	V3
7	/0000.05/cpcnsip.config	Standard protocol	16#feaaa3a3	V3
8	/0000.05/ethsw.config	Ethernet switch	16#016c5e67	V3
9	/0000.05/iot.config	Power supply and temperature monitoring	16#209c5b8f	V3
10	/0000.05/ke.config	COM data layout- and transmission	16#7db841fd	V2
11	/0000.05/modbus.config	Modbus Slave	16#9b58faff	V3
12	/0000.05/net.config	Network Setting	16#588b9fc4	V2

ethsw.config

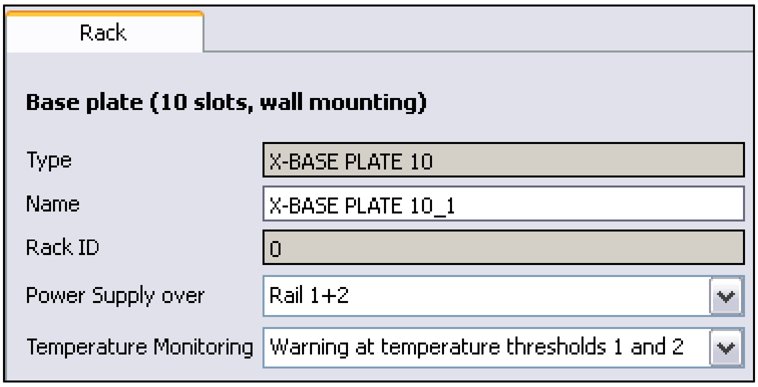
The settings associated with the Ethernet switches and configured in the **Ethernet Switch**, **VLAN**, **LLDP** and **Mirroring** tabs are saved in the *ethsw.config* configuration file.



	Name	Speed [MBit/s]	Flow Control	Autoneg also with Fixed Values	Limit
1	Port-Konfiguration_1	Autoneg	Autoneg	☑	
2	Port-Konfiguration_2	Autoneg	Autoneg	☑	
3	Port-Konfiguration_3	Autoneg	Autoneg	☑	
4	Port-Konfiguration_4	Autoneg	Autoneg	☑	

iot.config

The settings associated with the redundant power supply and temperature monitoring are saved in the *iot.confi*gconfiguration file. Since these settings apply to the entire rack, the changes also affect the *io.config* configuration files of all modules contained in that rack.



Base plate (10 slots, wall mounting)

Type	X-BASE PLATE 10
Name	X-BASE PLATE 10_1
Rack ID	0
Power Supply over	Rail 1+2
Temperature Monitoring	Warning at temperature thresholds 1 and 2

All changes to the communication settings of system bus, communication or processor modules (module's detail view in the Hardware Editor) result in a changed module configuration.

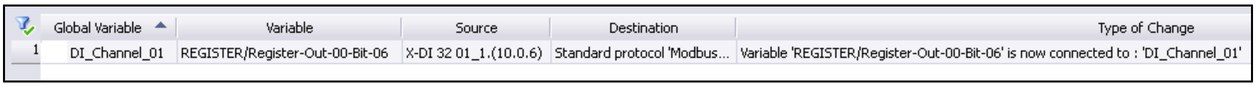
5.1.4 ke, modbus, net

ke.config

The *ke.config* configuration file changes if the assignment of global variables (new source, new destination) is modified for one of the protocols used on the module. A new variable was added to the Modbus protocol.

Double-click the *ke.config* configuration file line to open the detail view.

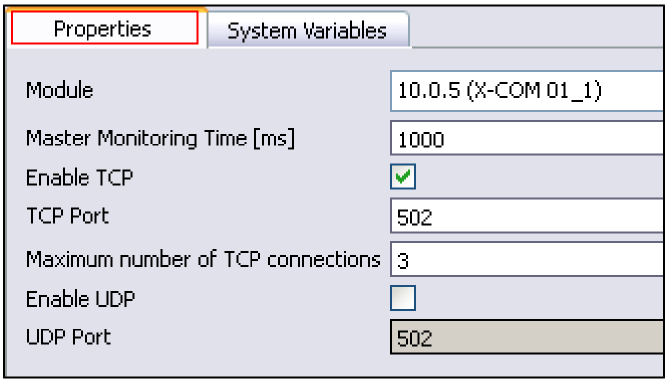
The information is displayed in plain text format. The following picture shows the *DI_Channel_01* global variable, which has the source in the *X-DI 32 01_1* hardware and is connected to a communication variable in *Standard protocol 'Modbus Slave_1'*.



	Global Variable	Variable	Source	Destination	Type of Change
1	DI_Channel_01	REGISTER/Register-Out-00-Bit-06	X-DI 32 01_1.(10.0.6)	Standard protocol 'Modbus...	Variable 'REGISTER/Register-Out-00-Bit-06' is now connected to : 'DI_Channel_01'

modbus.config

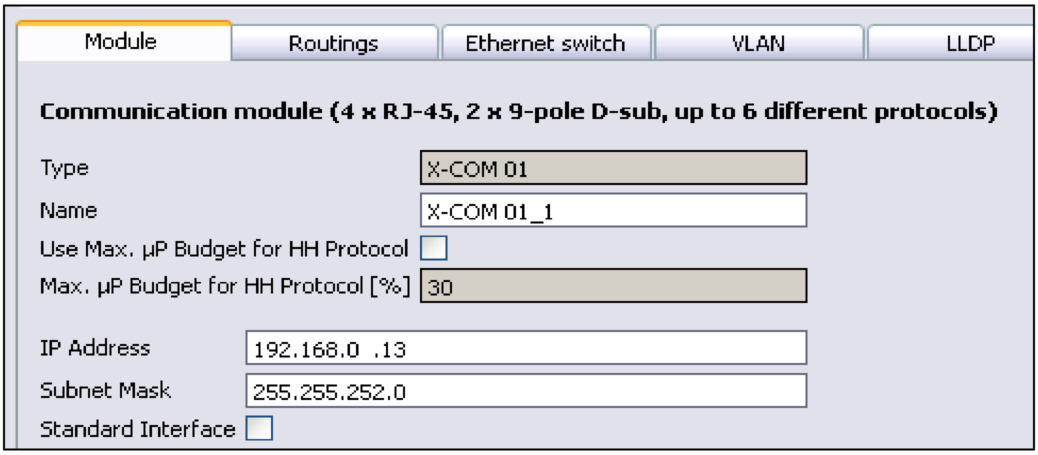
The settings for the Modbus protocols are saved in the *modbus.config* configuration file. The file changes if the protocol properties are modified, e.g., the monitoring time in case of the Modbus master.



Module	10.0.5 (X-COM 01_1)
Master Monitoring Time [ms]	1000
Enable TCP	☑
TCP Port	502
Maximum number of TCP connections	3
Enable UDP	☐
UDP Port	502

net.config

The general settings for Ethernet communication from the **Module** and **Routings** tabs, e.g., the IP address, are saved in the *net.config* configuration file.



Communication module (4 x RJ-45, 2 x 9-pole D-sub, up to 6 different protocols)

Type	X-COM 01
Name	X-COM 01_1
Use Max. µP Budget for HH Protocol	☐
Max. µP Budget for HH Protocol [%]	30
IP Address	192.168.0 .13
Subnet Mask	255.255.252.0
Standard Interface	☐

5.2. Logic Changes

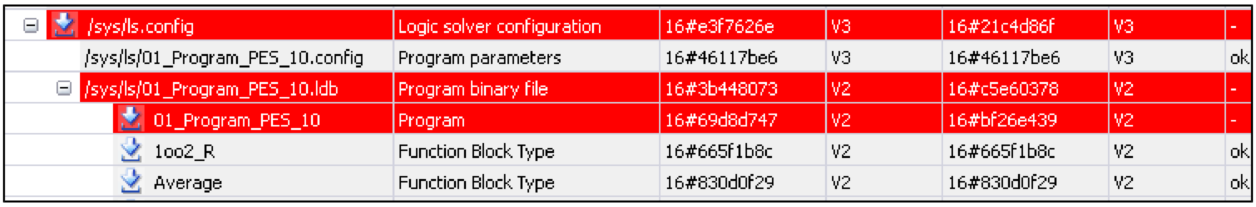
5.2.1 Logic Solver

If changes are performed to the logic, at least one configuration file subordinated to the */sys/ls.config* logic solver configuration file changes. Additionally, changes also affect the *Program.ldb* program binary file.

NOTE:

The source code is displayed for all POUs. The source code is converted to executable code during the code generation. The program binary file (loadable) will only change if the executable code has changed (functional change).

Not all changes to the source code result in functional changes. Renaming a VAR input or VAR output results in a changed checksum (CRC) for the function block and the corresponding line in the version comparison is highlighted in red, but the program binary file does not change. This means that no functional change was performed and no further check is necessary.

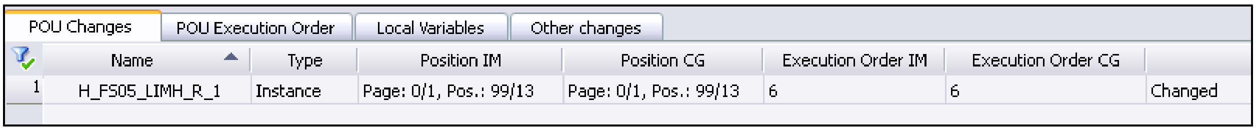


⊟ /sys/ls.config	Logic solver configuration	16#e3f7626e	V3	16#21c4d86f	V3	-
/sys/ls/01_Program_PES_10.config	Program parameters	16#46117be6	V3	16#46117be6	V3	ok
⊟ /sys/ls/01_Program_PES_10.ldb	Program binary file	16#3b448073	V2	16#c5e60378	V2	-
01_Program_PES_10	Program	16#69d8d747	V2	16#bf26e439	V2	-
1oo2_R	Function Block Type	16#665f1b8c	V2	16#665f1b8c	V2	ok
Average	Function Block Type	16#830d0f29	V2	16#830d0f29	V2	ok

Double-click the line associated with the changed POU to open the detail view. The following examples explains some evaluation details.

5.2.2 Changing Value Fields at the Input of Function Blocks

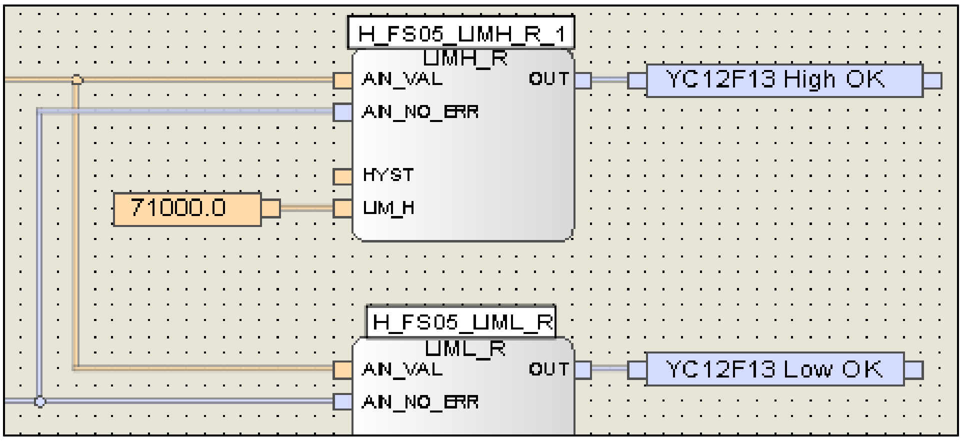
In the picture below, *Changed* noted in the **POU Changes** tab indicates that changes were performed to the input information of the *LIMH_R_1* function block instance.



	Name	Type	Position IM	Position CG	Execution Order IM	Execution Order CG	
1	H_FS05_LIMH_R_1	Instance	Page: 0/1, Pos.: 99/13	Page: 0/1, Pos.: 99/13	6	6	Changed

Double-click the *LIMH_R_1* line to open the FBD Editor and to centre the logic to the modified POU.

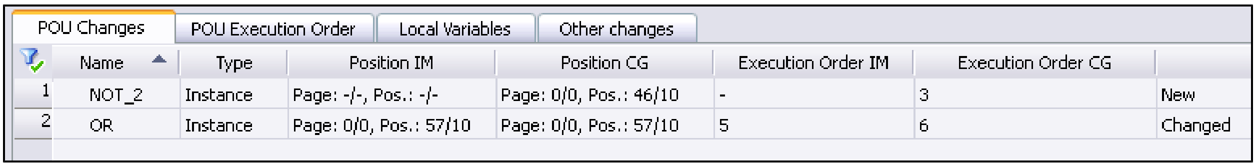
To identify the change, use the direct comparison of the current logic with the previous version. In the example below, the value field changed to `70010.0`.



5.2.3 Adding NOT to an Existing Connection Line

The empty columns *PositionIM* and *Execution Order IM* in the *POU Changes* tab indicate that the *NOT_2* object does not exist in the imported project version. The new version created by the code generator contains details on *Position CG* and *Execution Order CG*. Additionally, the POU is marked as *New*.

The input information for the *OR* object changed.

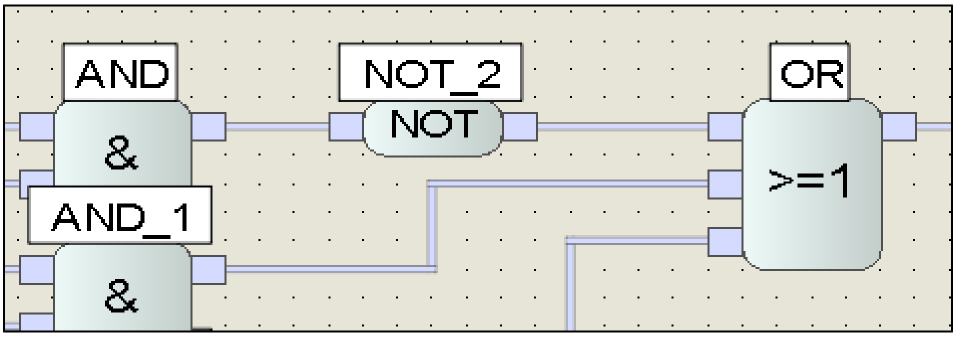


POU Changes | POU Execution Order | Local Variables | Other changes

	Name	Type	Position IM	Position CG	Execution Order IM	Execution Order CG	
1	NOT_2	Instance	Page: -/-, Pos.: -/-	Page: 0/0, Pos.: 46/10	-	3	New
2	OR	Instance	Page: 0/0, Pos.: 57/10	Page: 0/0, Pos.: 57/10	5	6	Changed

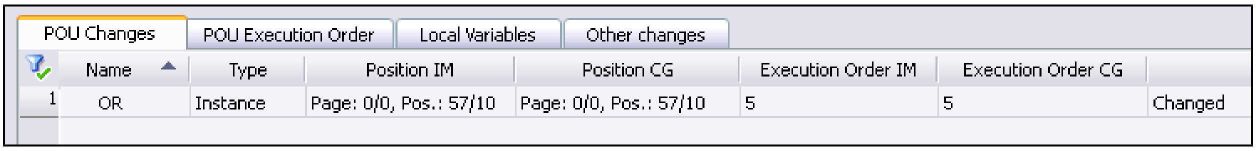
Double-click the *NOT_2* line to open the FBD Editor and to centre the logic to the modified POU.

Additionally, the *NOT_2* POU instance was added to the logic.



5.2.4 Inverting a Function Output

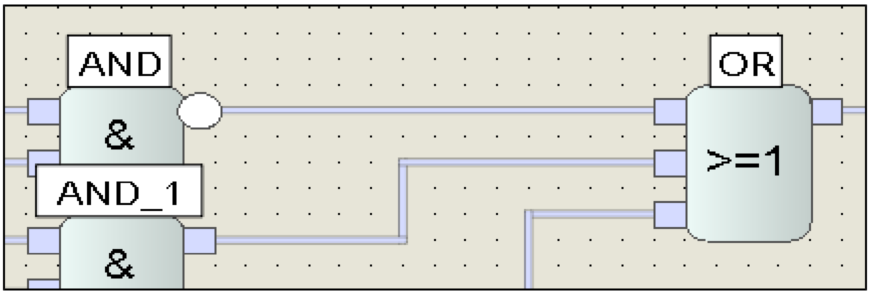
Changed noted in the **POU Changes** tab indicates that logic changes were made to one of the inputs of the *OR* object.



POU Changes | POU Execution Order | Local Variables | Other changes

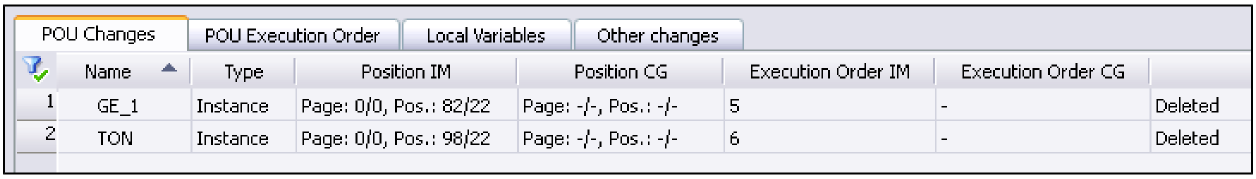
	Name	Type	Position IM	Position CG	Execution Order IM	Execution Order CG	
1	OR	Instance	Page: 0/0, Pos.: 57/10	Page: 0/0, Pos.: 57/10	5	5	Changed

Double-click the *OR* line to open the FBD Editor and to centre the logic to the modified POU. In the logic, the signal associated with the first input is inverted.



5.2.5 Deleting the POU from the Logic

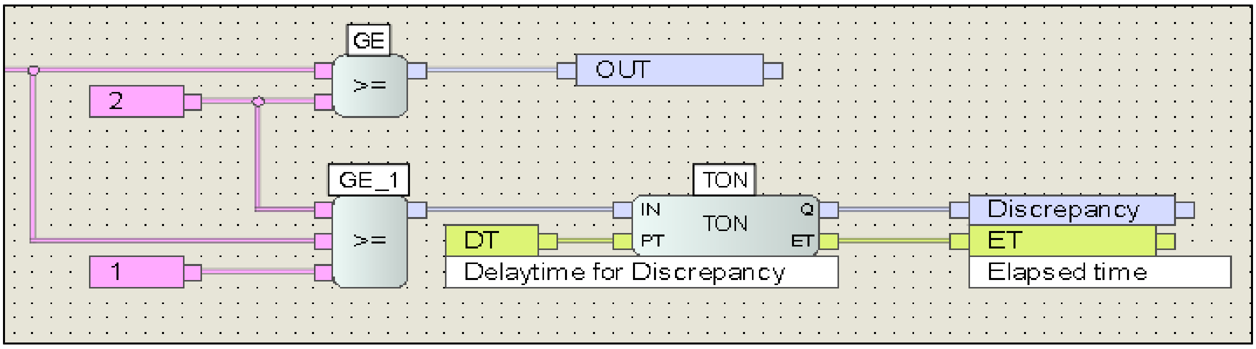
The empty columns **Position CG** and *Execution Order CG* indicate that the *GE_1* and *TON_15* POU instances do not exist in the new project version created by the code generator. The imported version contains details on *Position IM* and *Execution Order IM*. Additionally, the POU instances are marked as *Deleted*.



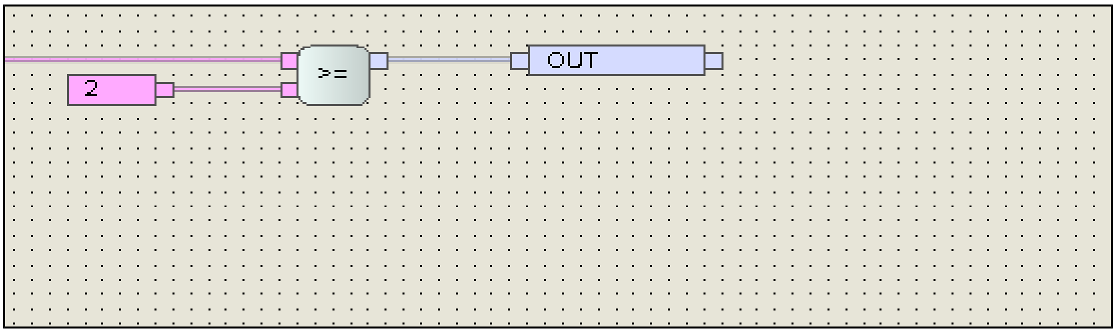
	Name	Type	Position IM	Position CG	Execution Order IM	Execution Order CG	
1	GE_1	Instance	Page: 0/0, Pos.: 82/22	Page: -/-, Pos.: -/-	5	-	Deleted
2	TON	Instance	Page: 0/0, Pos.: 98/22	Page: -/-, Pos.: -/-	6	-	Deleted

Double-click one of the lines to open the FBD Editor. The logic, however, cannot be centered to the deleted POU instances. This step must be performed manually.

Part of the logic of the imported version:

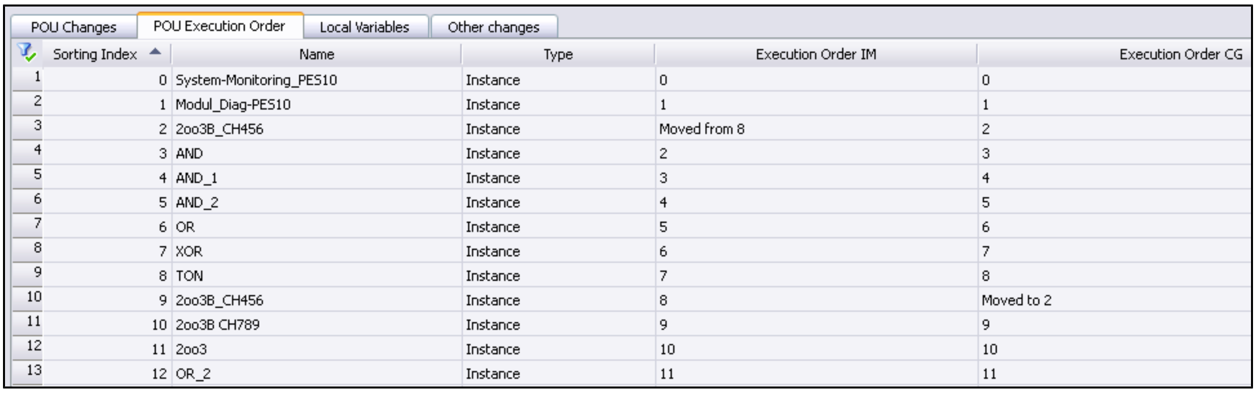


Part of the modified logic (code generator)



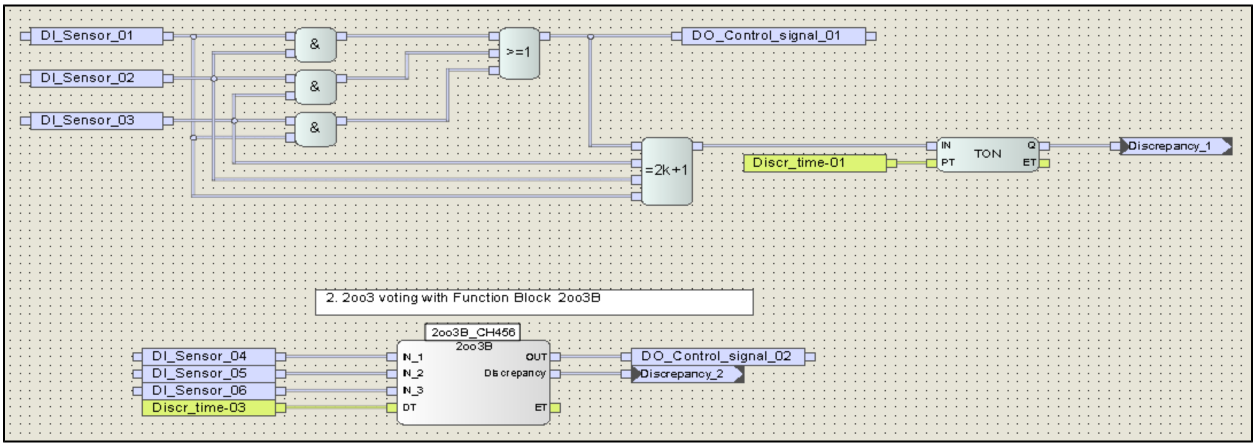
5.2.6 Changing the Network Sequence

In the **POU Execution Order** tab, the details contained in the *Execution Order IM* and *Execution OrderCG* columns indicate that the *2oo3B* POU instance was moved. The position of all the remaining instances did not change.

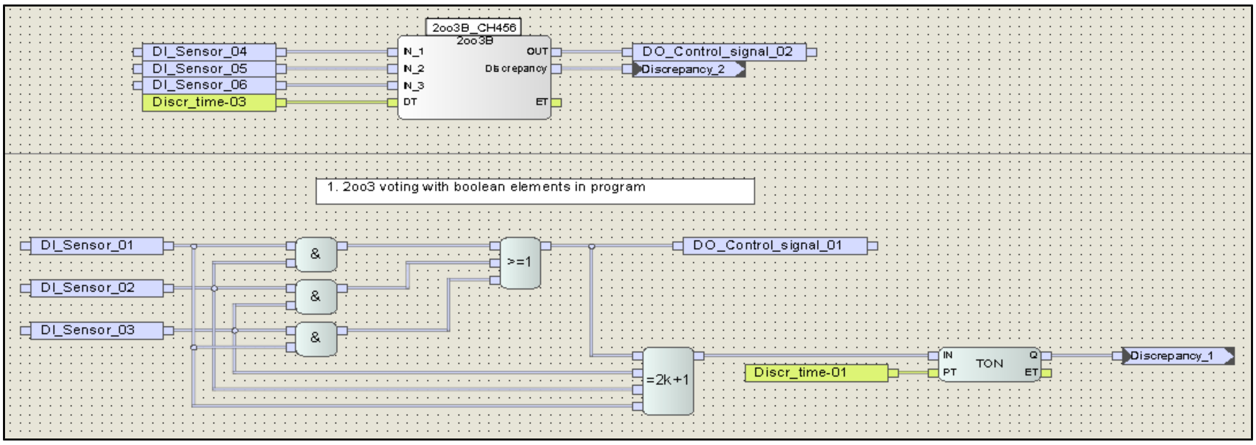


	Sorting Index	Name	Type	Execution Order IM	Execution Order CG
1	0	System-Monitoring_PES10	Instance	0	0
2	1	Modul_Diag-PES10	Instance	1	1
3	2	2oo3B_CH456	Instance	Moved from 8	2
4	3	AND	Instance	2	3
5	4	AND_1	Instance	3	4
6	5	AND_2	Instance	4	5
7	6	OR	Instance	5	6
8	7	XOR	Instance	6	7
9	8	TON	Instance	7	8
10	9	2oo3B_CH456	Instance	8	Moved to 2
11	10	2oo3B CH789	Instance	9	9
12	11	2oo3	Instance	10	10
13	12	OR_2	Instance	11	11

Part of the logic of the imported version:



Part of the modified logic (code generator)

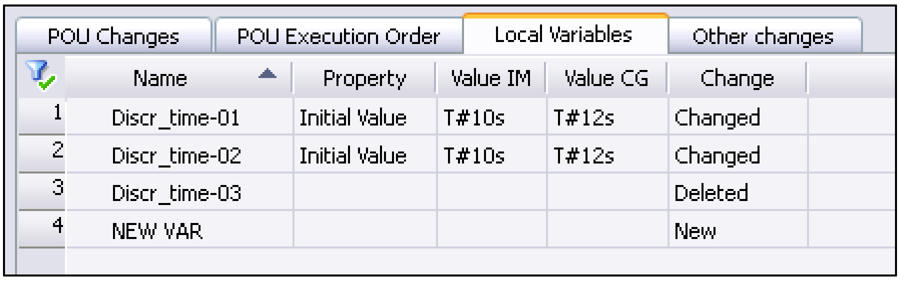


5.2.7 Changing Local Variables

In the **Local Variables** tab, the information specified in the *Change* column indicates the following changes:

- The NEW VAR variable was added.
- The initial values of the *Discr_Time-01* and *Discr_Time-02* variables changed (see the *Value IM* and *Value CG* columns).
- The *Var_1* variable was deleted.

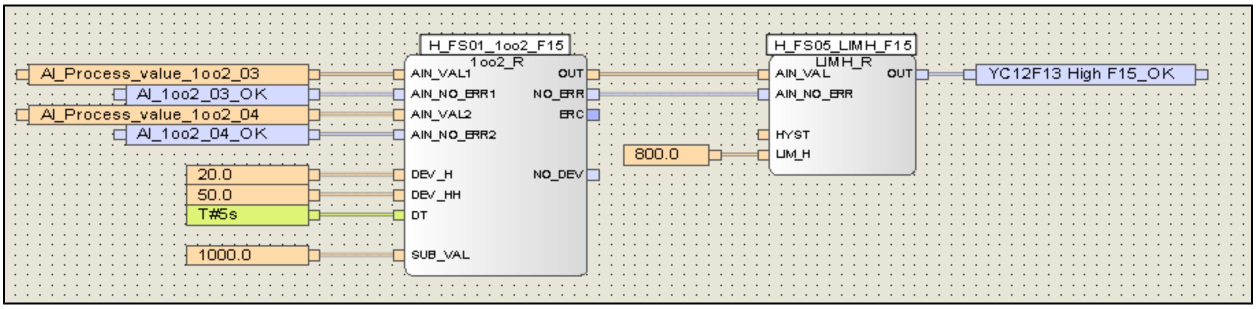
Double-click a line to open the FBD Editor and to centre the logic to the changed variables (it does not apply for deleted variables).



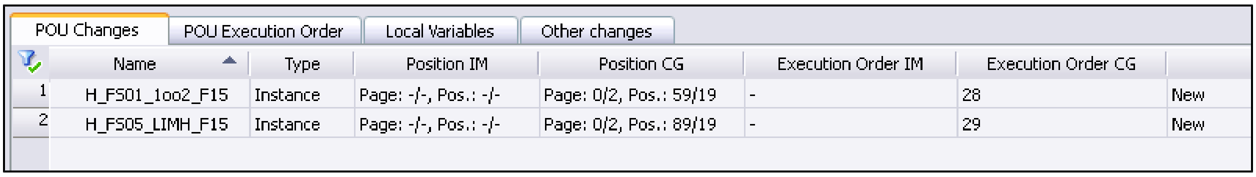
	Name	Property	Value IM	Value CG	Change
1	Discr_time-01	Initial Value	T#10s	T#12s	Changed
2	Discr_time-02	Initial Value	T#10s	T#12s	Changed
3	Discr_time-03				Deleted
4	NEW VAR				New

5.2.8 Creating New Networks

An existing logic (IM) is extended with the following objects and recompiled (CG):

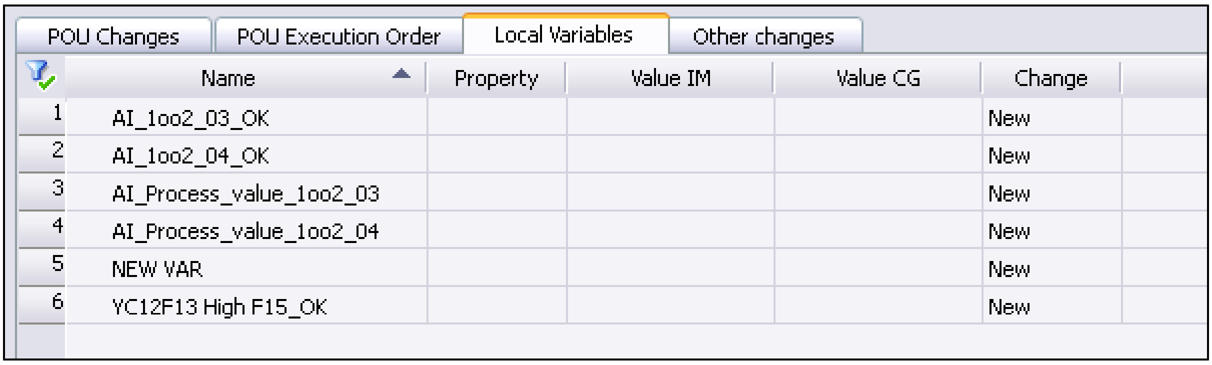


The empty columns *Position IM* and *Execution Order IM* indicate that the *1002_F15* and *LIMH_15* POU instances do not exist in the imported project version. The new version created by the code generator contains details on *Position CG* and *Execution Order CG*. Additionally, the POU instances are marked as *New*.



	Name	Type	Position IM	Position CG	Execution Order IM	Execution Order CG	
1	H_FS01_1oo2_F15	Instance	Page: -/-, Pos.: -/-	Page: 0/2, Pos.: 59/19	-	28	New
2	H_FS05_LIMH_F15	Instance	Page: -/-, Pos.: -/-	Page: 0/2, Pos.: 89/19	-	29	New

In the **Local Variables** tab, the information specified in the *Change* column indicates that the three local variables are new.



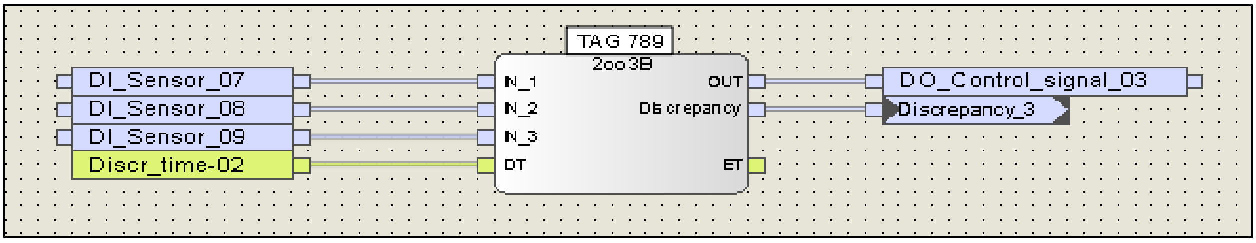
	Name	Property	Value IM	Value CG	Change
1	AI_1oo2_03_OK				New
2	AI_1oo2_04_OK				New
3	AI_Process_value_1oo2_03				New
4	AI_Process_value_1oo2_04				New
5	NEW VAR				New
6	YC12F13 High F15_OK				New

Double-click one of the lines to open the FBD Editor and to centre the logic to the selected POU.

5.2.9 Renaming Instances

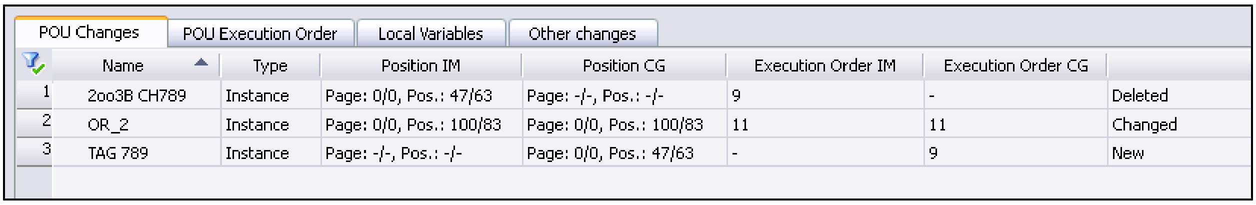
Renaming an instance is handled as if an instance is deleted and a new instance is added. This means that the old instance name no longer exists and a new instance name is created.

The *2oo3B_1* POU instance was renamed to *TAG 789*.



The empty columns *Position IM* and *Position CG* in the **POU Changes** tab and the additional information *Deleted/New* indicate:

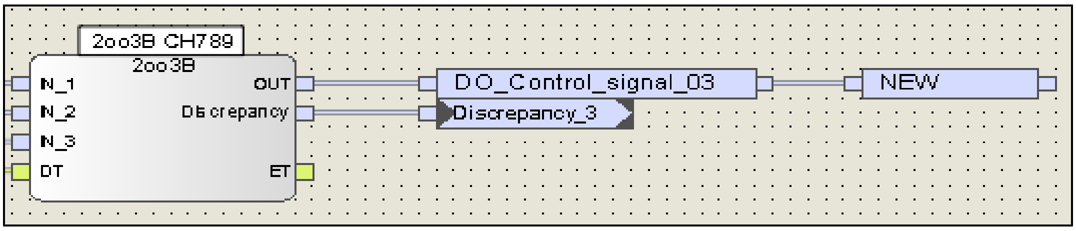
- The *2oo3B_1* POU instance was deleted.
- A new *TAG 789* POU instance was created.
- The new instance is located at the same position as the deleted instance and the execution order is identical.
- The *OR_2* instance is marked as *Changed* since the *Discrepancy_3* connector is connected to the new *TAG 789* POU instance.



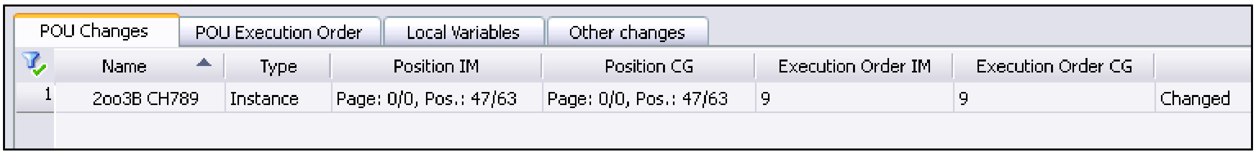
	Name	Type	Position IM	Position CG	Execution Order IM	Execution Order CG	
1	2oo3B CH789	Instance	Page: 0/0, Pos.: 47/63	Page: -/-, Pos.: -/-	9	-	Deleted
2	OR_2	Instance	Page: 0/0, Pos.: 100/83	Page: 0/0, Pos.: 100/83	11	11	Changed
3	TAG 789	Instance	Page: -/-, Pos.: -/-	Page: 0/0, Pos.: 47/63	-	9	New

5.2.10 Assigning new Global Variables

The *NEW* global variable is added to the existing logic (IM) and connected to it as previously described.

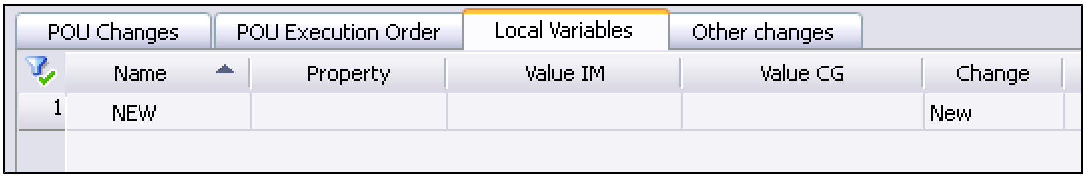


In the **POU Changes** tab, the *2oo3B_1* POU instance is marked as *Changed*. Since variables have no instance name, the instance describing the variable is displayed.



	Name	Type	Position IM	Position CG	Execution Order IM	Execution Order CG	
1	2oo3B CH789	Instance	Page: 0/0, Pos.: 47/63	Page: 0/0, Pos.: 47/63	9	9	Changed

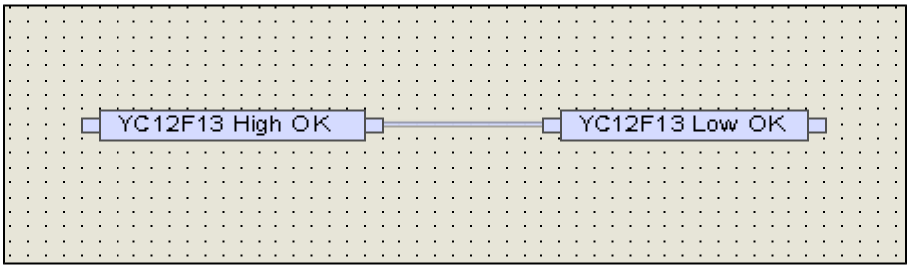
In the **Local Variables** tab, the *NEW* variable is marked as *New*.



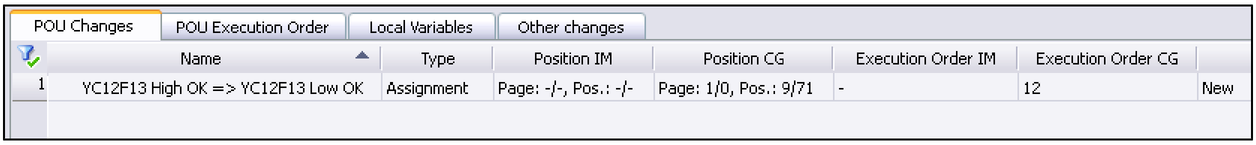
	Name	Property	Value IM	Value CG	Change
1	NEW				New

5.2.11 Adding new Variable Assignments

A *YC12F13 Low OK* variable is added to the existing logic and is assigned the value of the *YC12F13 High OK* variable.

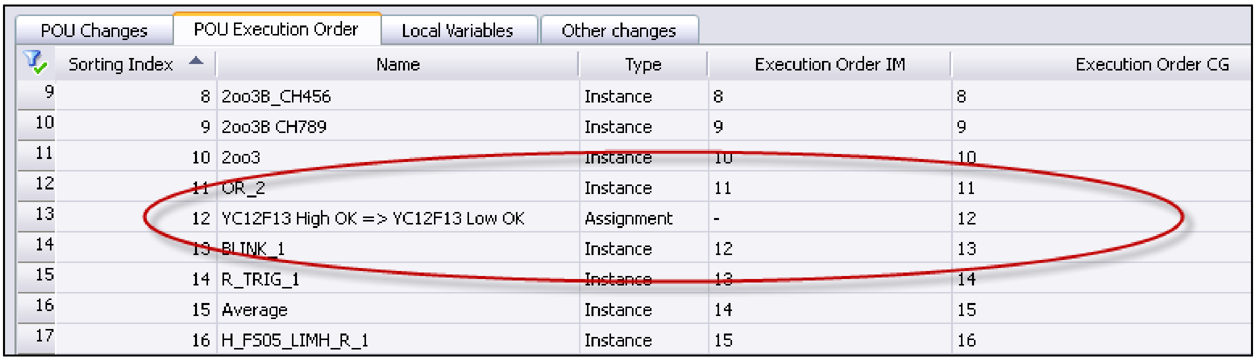


In the **POE Changes** tab, the type of change is classified as *Assignment*. The direction of the assignment is indicated in the *Name* column. The empty columns *Position IM* and *Execution Order IM* indicate that the assignment does not exist in the imported project version. The new version created by the code generator contains details on *Position CG* and *Execution Order CG*. Additionally, the assignment is marked as *New*.



	Name	Type	Position IM	Position CG	Execution Order IM	Execution Order CG	
1	YC12F13 High OK => YC12F13 Low OK	Assignment	Page: -/-, Pos.: -/-	Page: 1/0, Pos.: 9/71	-	12	New

Changes in the execution order caused by an assignment are indicated in the **POU Execution Order** tab.

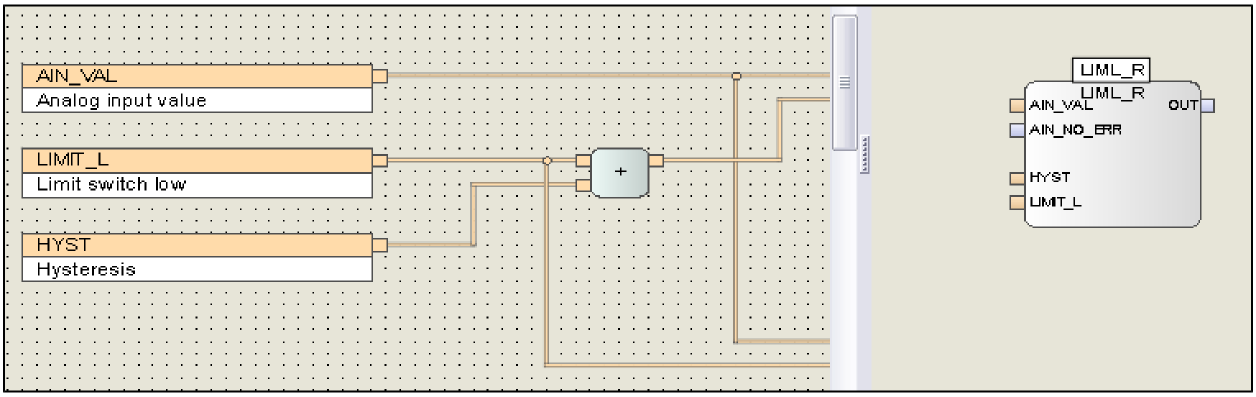


	Sorting Index	Name	Type	Execution Order IM	Execution Order CG
9	8	2oo3B_CH456	Instance	8	8
10	9	2oo3B CH789	Instance	9	9
11	10	2oo3	Instance	10	10
12	11	OR_2	Instance	11	11
13	12	YC12F13 High OK => YC12F13 Low OK	Assignment	-	12
14	13	BLINK_1	Instance	12	13
15	14	R_TRIG_1	Instance	13	14
16	15	Average	Instance	14	15
17	16	H_FS05_LIMH_R_1	Instance	15	16

5.2.12 Renaming Variables

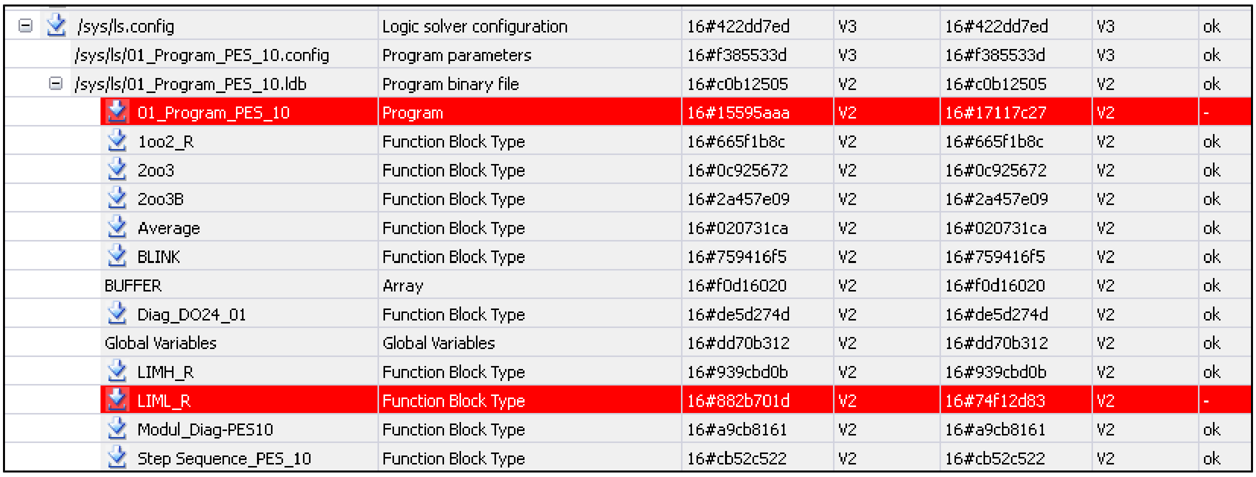
Renaming a variable of VAR_INPUT type is handled as if the variable is deleted and a new variable is added. This means that the previous variable no longer exists and a new variable is created.

The *LIM_L* input variable was renamed to *LIMIT_L*.



Renaming input variables does not result in a functional change of the logic. The new generated binary file */sys/ls/01_Program01.ldb* is identical to the imported version. The checksums of the two versions are identical.

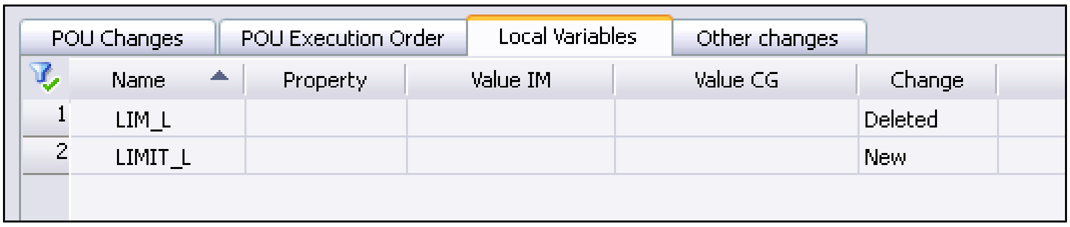
Renaming the input variable, however, results in a changed checksum for the LIML_R function block type. This change is displayed in the version comparison.



/sys/ls.config	Logic solver configuration	16#422dd7ed	V3	16#422dd7ed	V3	ok
/sys/ls/01_Program_PES_10.config	Program parameters	16#f385533d	V3	16#f385533d	V3	ok
/sys/ls/01_Program_PES_10.ldb	Program binary file	16#c0b12505	V2	16#c0b12505	V2	ok
01_Program_PES_10	Program	16#15595aaa	V2	16#17117c27	V2	-
1oo2_R	Function Block Type	16#665f1b8c	V2	16#665f1b8c	V2	ok
2oo3	Function Block Type	16#0c925672	V2	16#0c925672	V2	ok
2oo3B	Function Block Type	16#2a457e09	V2	16#2a457e09	V2	ok
Average	Function Block Type	16#020731ca	V2	16#020731ca	V2	ok
BLINK	Function Block Type	16#759416f5	V2	16#759416f5	V2	ok
BUFFER	Array	16#f0d16020	V2	16#f0d16020	V2	ok
Diag_DO24_01	Function Block Type	16#de5d274d	V2	16#de5d274d	V2	ok
Global Variables	Global Variables	16#dd70b312	V2	16#dd70b312	V2	ok
LIMH_R	Function Block Type	16#939cbd0b	V2	16#939cbd0b	V2	ok
LIML_R	Function Block Type	16#882b701d	V2	16#74f12d83	V2	-
Modul_Diag-PES10	Function Block Type	16#a9cb8161	V2	16#a9cb8161	V2	ok
Step Sequence_PES_10	Function Block Type	16#cb52c522	V2	16#cb52c522	V2	ok

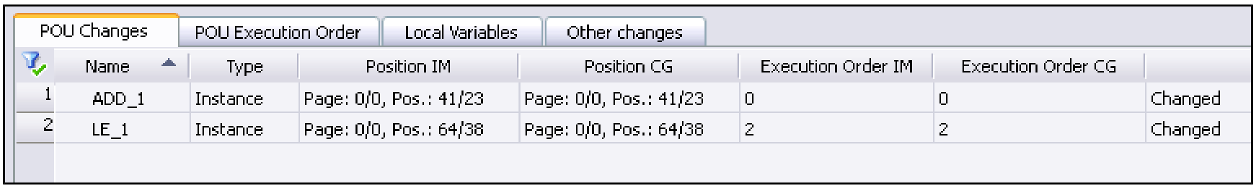
Double-click the *LIML_R* line to display further details on the POU.

LIMIT_L, the new name for the *LIM_L* variable, is displayed in the **Local Variables** tab.



	Name	Property	Value IM	Value CG	Change
1	LIM_L				Deleted
2	LIMIT_L				New

The *ADD_1* and *LE_1* instances used in the *LIML_R* function block type are also marked as *Changed* in the *POU Changes* since both instances are connected to the *LIMIT_L* input variable.

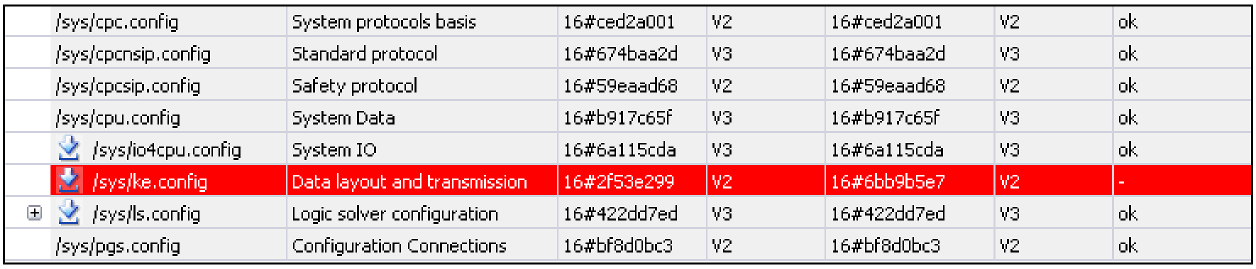


	Name	Type	Position IM	Position CG	Execution Order IM	Execution Order CG	
1	ADD_1	Instance	Page: 0/0, Pos.: 41/23	Page: 0/0, Pos.: 41/23	0	0	Changed
2	LE_1	Instance	Page: 0/0, Pos.: 64/38	Page: 0/0, Pos.: 64/38	2	2	Changed

5.3. Modifying the Assignment of Global Variables

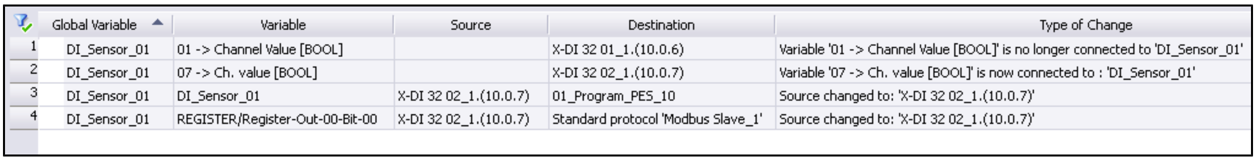
5.3.1 Assigning Global Variables to Another Hardware Input

Modifying the assignment of global variables (new source, new destination, modified initial values) always affects */sys/ke.config*, the configuration file for reading and writing global variables in protocols.



	/sys/cpc.config	System protocols basis	16#ced2a001	V2	16#ced2a001	V2	ok
	/sys/cpcnsip.config	Standard protocol	16#674baa2d	V3	16#674baa2d	V3	ok
	/sys/cpcsip.config	Safety protocol	16#59eaad68	V2	16#59eaad68	V2	ok
	/sys/cpu.config	System Data	16#b917c65f	V3	16#b917c65f	V3	ok
	/sys/io4cpu.config	System IO	16#6a115cda	V3	16#6a115cda	V3	ok
	/sys/ke.config	Data layout and transmission	16#2f53e299	V2	16#6bb9b5e7	V2	-
⊞	/sys/ls.config	Logic solver configuration	16#422dd7ed	V3	16#422dd7ed	V3	ok
	/sys/pgs.config	Configuration Connections	16#bf8d0bc3	V2	16#bf8d0bc3	V2	ok

Double-click the *sys/ke.config* line to display further details on the configuration file.



	Global Variable	Variable	Source	Destination	Type of Change
1	DI_Sensor_01	01 -> Channel Value [BOOL]		X-DI 32 01_1.(10.0.6)	Variable '01 -> Channel Value [BOOL]' is no longer connected to 'DI_Sensor_01'
2	DI_Sensor_01	07 -> Ch. value [BOOL]		X-DI 32 02_1.(10.0.7)	Variable '07 -> Ch. value [BOOL]' is now connected to : 'DI_Sensor_01'
3	DI_Sensor_01	DI_Sensor_01	X-DI 32 02_1.(10.0.7)	01_Program_PES_10	Source changed to: 'X-DI 32 02_1.(10.0.7)'
4	DI_Sensor_01	REGISTER/Register-Out-00-Bit-00	X-DI 32 02_1.(10.0.7)	Standard protocol 'Modbus Slave_1'	Source changed to: 'X-DI 32 02_1.(10.0.7)'

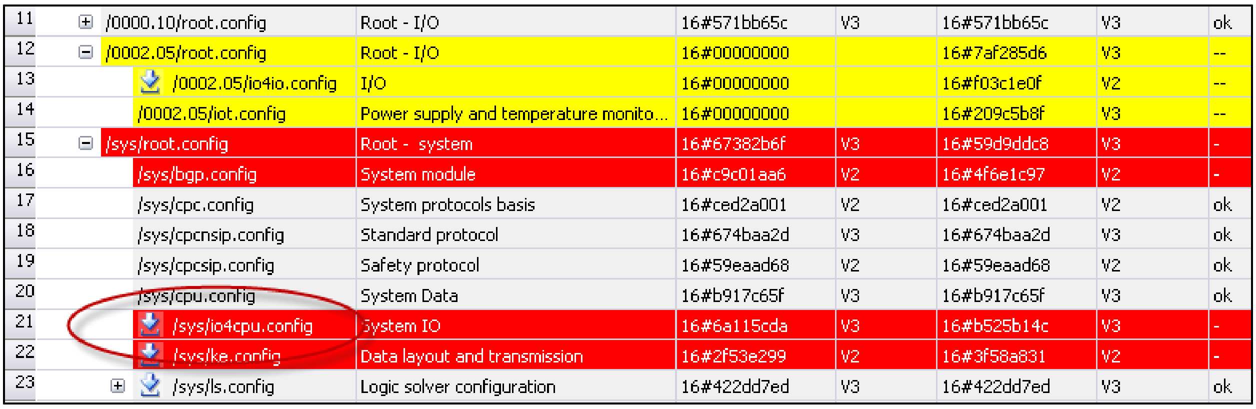
Line	Description
1	Channel 7 of module *X-DI 32 02_1* in slot 7 was connected to the *DI_Sensor_01* global variable.
2	The *DI_Sensor_01* global variable used in the *01_Program_PES_10* program has the new source *X-DI 32 02_1*.
3	Channel 1 of the *X-DI 32 01_1* module in slot 6 was disconnected from the *DI_Sensor_01* global variable.
4	The source of the *DI_Sensor_01* global variable transferred via the *Modbus-Slave_1* standard protocol changed.

The version comparison takes all the effects of the change into account. The users must decide which changes they want to check. Even if the logic did not change, but another physical source was used, the program can behave differently.

5.3.2 Adding new Modules

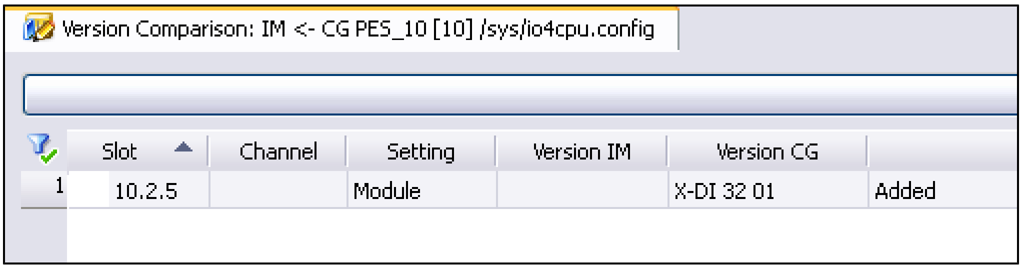
If a module is added to a system, the configuration file specific to the new module is highlighted in yellow in the version comparison.

Line 12 of the following picture show that a new module was added to rack 2 in slot 5. This also affects the */sys/bgp.config* module management file. The module configuration is saved in the */sys/io4cpu.config* and */sys/ke.config* files.
The module must be equipped with operating system version 3 since the rack was configured for temperature monitoring (line 14).



11	⊞ /0000.10/root.config	Root - I/O	16#571bb65c	V3	16#571bb65c	V3	ok
12	⊟ /0002.05/root.config	Root - I/O	16#00000000		16#7af285d6	V3	--
13	/0002.05/io4io.config	I/O	16#00000000		16#f03c1e0f	V2	--
14	/0002.05/iot.config	Power supply and temperature monito...	16#00000000		16#209c5b8f	V3	--
15	⊟ /sys/root.config	Root - system	16#67382b6f	V3	16#59d9ddc8	V3	-
16	/sys/bgp.config	System module	16#c9c01aa6	V2	16#4f6e1c97	V2	-
17	/sys/cpc.config	System protocols basis	16#ced2a001	V2	16#ced2a001	V2	ok
18	/sys/cpcnsip.config	Standard protocol	16#674baa2d	V3	16#674baa2d	V3	ok
19	/sys/cpcsip.config	Safety protocol	16#59eaad68	V2	16#59eaad68	V2	ok
20	/sys/cpu.config	System Data	16#b917c65f	V3	16#b917c65f	V3	ok
21	/sys/io4cpu.config	System IO	16#6a115cda	V3	16#b525b14c	V3	-
22	/sys/ke.config	Data layout and transmission	16#2f53e299	V2	16#3f58a831	V2	-
23	⊞ /sys/ls.config	Logic solver configuration	16#422dd7ed	V3	16#422dd7ed	V3	ok

Double-click the */sys/io4cpu.config* line to display further details on the added module.



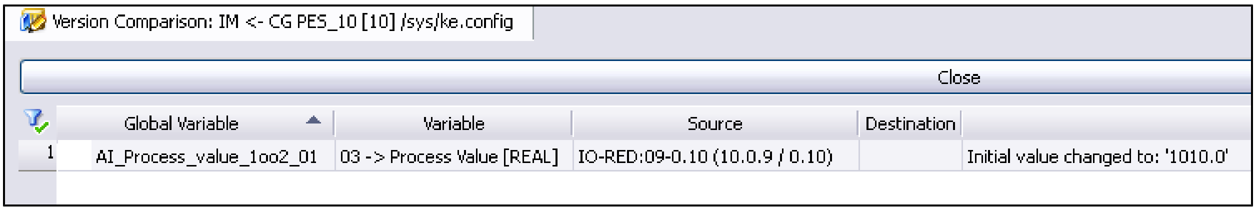
Version Comparison: IM <- CG PES_10 [10] /sys/io4cpu.config

	Slot	Channel	Setting	Version IM	Version CG	
1	10.2.5		Module		X-DI 32 01	Added

5.3.3 New Initial Value for a Global Variable

The initial value of a global variable can be set or changed in the Global Variable Editor. This information is saved in the */sys/ke.config* configuration file. The change of one or multiple initial values results in a changed */sys/ke.config* file.

Double-click the */sys/ke.config* line to display further details on the global variable. The following picture shows that the initial value of the *AI_Process_value_1oo2_03* global variable changed to `1010.0`.



	Global Variable	Variable	Source	Destination	
1	AI_Process_value_1oo2_01	03 -> Process Value [REAL]	IO-RED:09-0.10 (10.0.9 / 0.10)		Initial value changed to: '1010.0'

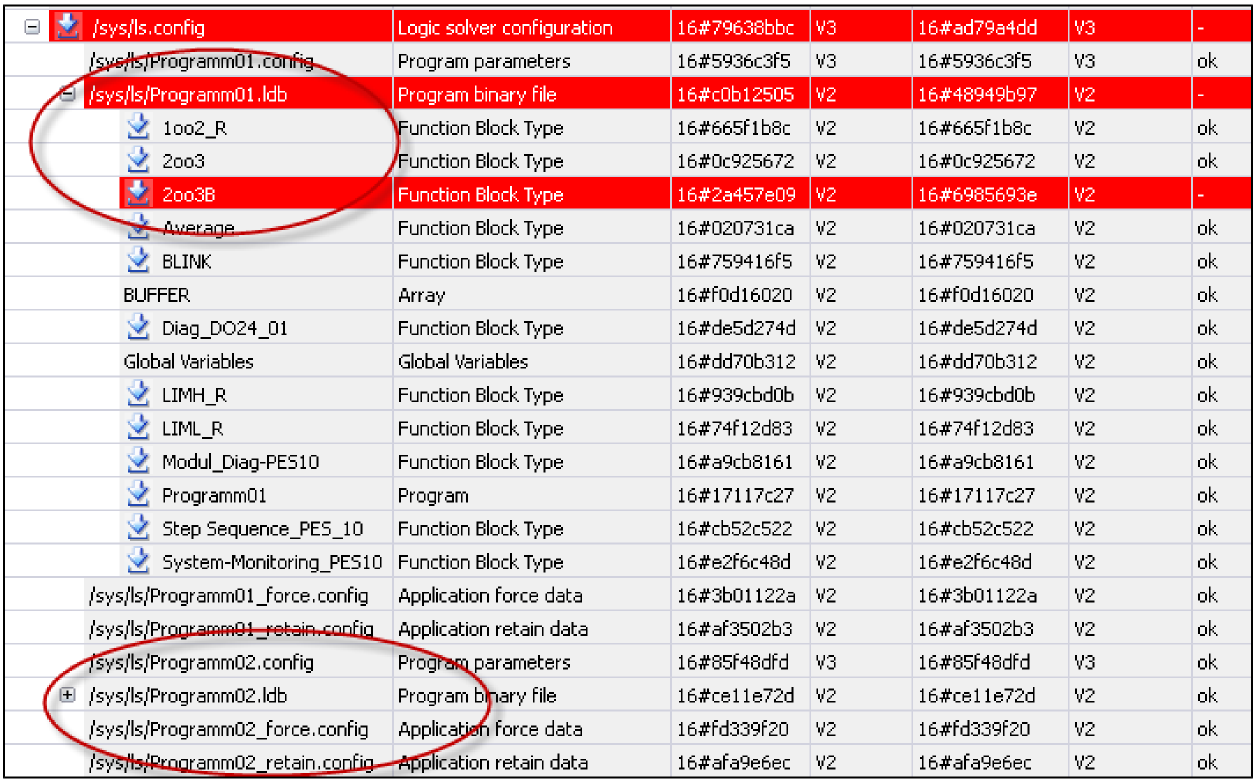
5.4. Multitasking

SILworX offers the opportunity to divide the system tasks in multiple programs and therefore to distinguish between safety-relevant (e.g., ESD functions) and non-safety-relevant logic parts (e.g., data preparation for a control system).

The version comparison allows one to identify changes performed to the logic. If safety-relevant logic parts changed, a safety-relevant test (e.g., performed by the TÜV) could become necessary.

In the following pictures, the version comparison detected a change in the binary file of *sys/ls/Program_ 01.ldb*. A change was performed to the *2oo3B* function block type. This POU must be tested.

The binary file of */sys/ls/Program02,* on the other hand, did not change.



⊟ /sys/ls.config	Logic solver configuration	16#79638bbc	V3	16#ad79a4dd	V3	-
/sys/ls/Programm01.config	Program parameters	16#5936c3f5	V3	16#5936c3f5	V3	ok
⊟ /sys/ls/Programm01.ldb	Program binary file	16#c0b12505	V2	16#48949b97	V2	-
1oo2_R	Function Block Type	16#665f1b8c	V2	16#665f1b8c	V2	ok
2oo3	Function Block Type	16#0c925672	V2	16#0c925672	V2	ok
2oo3B	Function Block Type	16#2a457e09	V2	16#6985693e	V2	-
Average	Function Block Type	16#020731ca	V2	16#020731ca	V2	ok
BLINK	Function Block Type	16#759416f5	V2	16#759416f5	V2	ok
BUFFER	Array	16#f0d16020	V2	16#f0d16020	V2	ok
Diag_DO24_01	Function Block Type	16#de5d274d	V2	16#de5d274d	V2	ok
Global Variables	Global Variables	16#dd70b312	V2	16#dd70b312	V2	ok
LIMH_R	Function Block Type	16#939cbd0b	V2	16#939cbd0b	V2	ok
LIML_R	Function Block Type	16#74f12d83	V2	16#74f12d83	V2	ok
Modul_Diag-PES10	Function Block Type	16#a9cb8161	V2	16#a9cb8161	V2	ok
Programm01	Program	16#17117c27	V2	16#17117c27	V2	ok
Step Sequence_PES_10	Function Block Type	16#cb52c522	V2	16#cb52c522	V2	ok
System-Monitoring_PES10	Function Block Type	16#e2f6c48d	V2	16#e2f6c48d	V2	ok
/sys/ls/Programm01_force.config	Application force data	16#3b01122a	V2	16#3b01122a	V2	ok
/sys/ls/Programm01_retain.config	Application retain data	16#af3502b3	V2	16#af3502b3	V2	ok
/sys/ls/Programm02.config	Program parameters	16#85f48dfd	V3	16#85f48dfd	V3	ok
⊞ /sys/ls/Programm02.ldb	Program binary file	16#ce11e72d	V2	16#ce11e72d	V2	ok
/sys/ls/Programm02_force.config	Application force data	16#fd339f20	V2	16#fd339f20	V2	ok
/sys/ls/Programm02_retain.config	Application retain data	16#afa9e6ec	V2	16#afa9e6ec	V2	ok

Changes to the global variable assignment (new source, new destination) require particular attention since they not necessarily result in a changed syntax and do not therefore always modify the program binary file. However, these changes may have safety relevance! Changes to global variables always affect the *ke.config* configuration file.

If global variables of multiple programs are read and changes are performed to the *ke.config* configuration file, the programs must be checked individually.

The cross-reference list displayed in the Global Variable Editor can be used, e.g., to determine if a variable that has had its hardware assignment changed was written to by a safety-relevant or a non-safety-relevant program. If the variable is used by a safety-relevant program, the change is also safety-relevant!

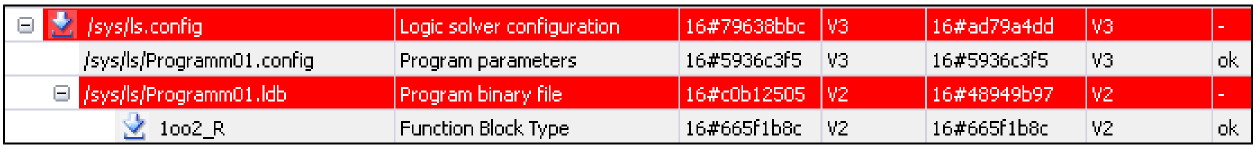
The safety relevance of changes to other central configuration files (e.g., module configuration files, system settings, etc.) must be checked on an individual basis. Because the changes to these central configuration files have no direct reference to the programs, the distinction made between *safe* and *non-safe* programsis not relevant in this context.

5.5. Memory Overview for Code and Data

A HIMax system has a total of 10 MB memory available for program code and data. The memory is partitioned if multiple programs are executed on a single controller.

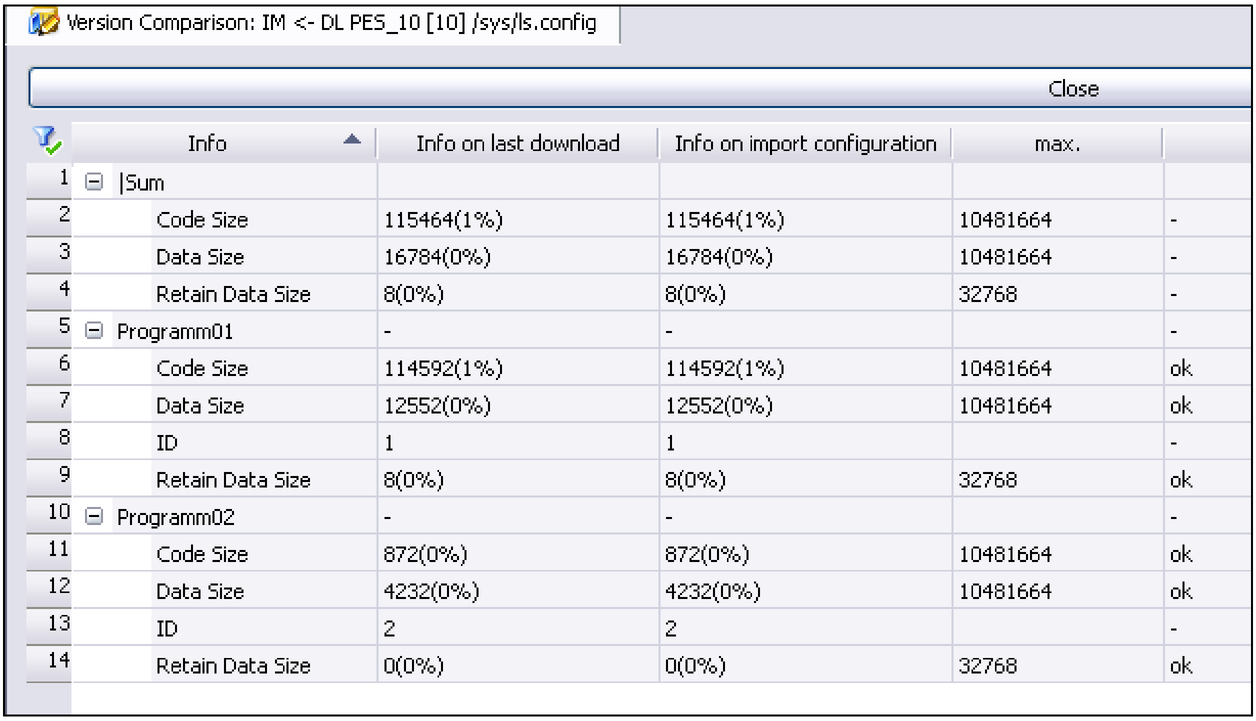
The memory overview can be accessed from within the version comparison.

Double-click the */sys/ls.config* line to open the memory overview.



⊟ /sys/ls.config	Logic solver configuration	16#79638bbc	V3	16#ad79a4dd	V3	-
/sys/ls/Programm01.config	Program parameters	16#5936c3f5	V3	16#5936c3f5	V3	ok
⊟ /sys/ls/Programm01.ldb	Program binary file	16#c0b12505	V2	16#48949b97	V2	-
1oo2_R	Function Block Type	16#665f1b8c	V2	16#665f1b8c	V2	ok

The overview shows the memory usage altogether and individually for each program. The percentages refer to the total memory of 10 MB.



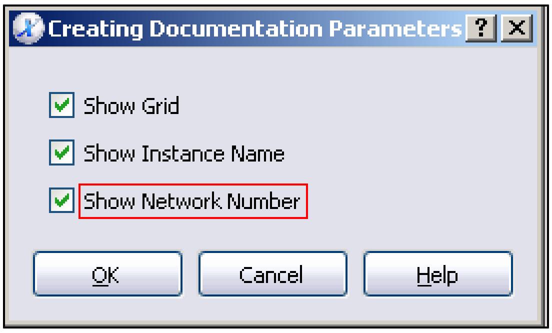
Version Comparison: IM <- DL PES_10 [10] /sys/ls.config

	Info	Info on last download	Info on import configuration	max.	
1	⊟ |Sum				
2	Code Size	115464(1%)	115464(1%)	10481664	-
3	Data Size	16784(0%)	16784(0%)	10481664	-
4	Retain Data Size	8(0%)	8(0%)	32768	-
5	⊟ Programm01	-	-		-
6	Code Size	114592(1%)	114592(1%)	10481664	ok
7	Data Size	12552(0%)	12552(0%)	10481664	ok
8	ID	1	1		-
9	Retain Data Size	8(0%)	8(0%)	32768	ok
10	⊟ Programm02	-	-		-
11	Code Size	872(0%)	872(0%)	10481664	ok
12	Data Size	4232(0%)	4232(0%)	10481664	ok
13	ID	2	2		-
14	Retain Data Size	0(0%)	0(0%)	32768	ok

6 Printing out the Version Comparison

The version comparison evaluation is printed out in the documentation editor. Click the **Documentation** button from the Action Bar to open the documentation editor.

Before the documentation editor appears, one can define if grid (the background grid), the instance name and the network number should be displayed on the screen. The printed document will contain all the details displayed on the screen.



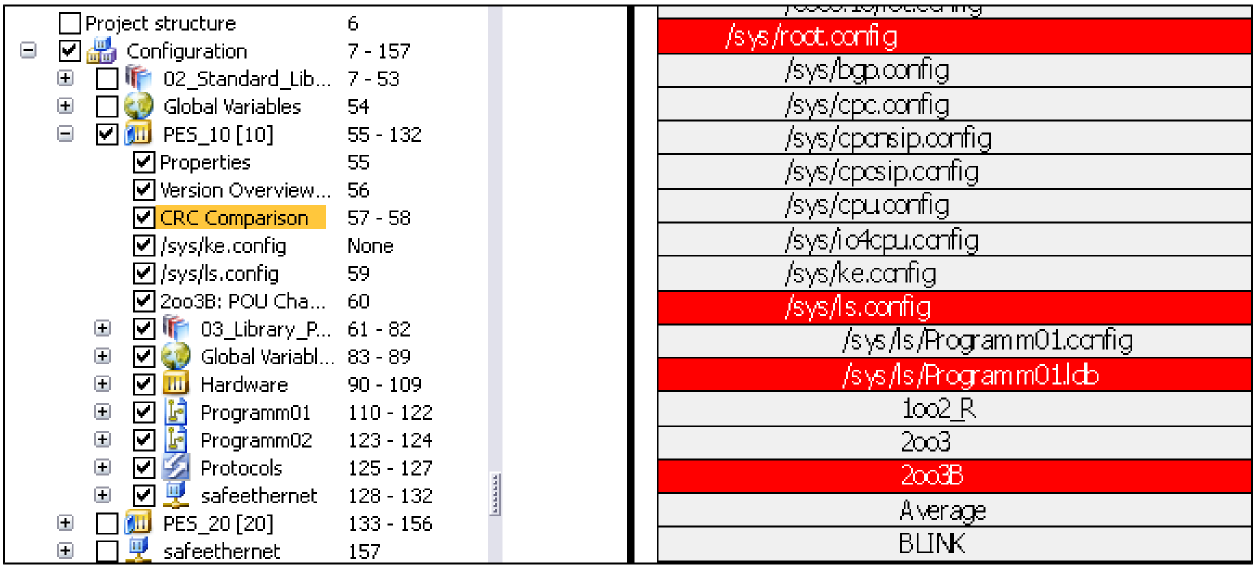
Creating Documentation Parameters

☑ Show Grid

☑ Show Instance Name

☑ Show Network Number

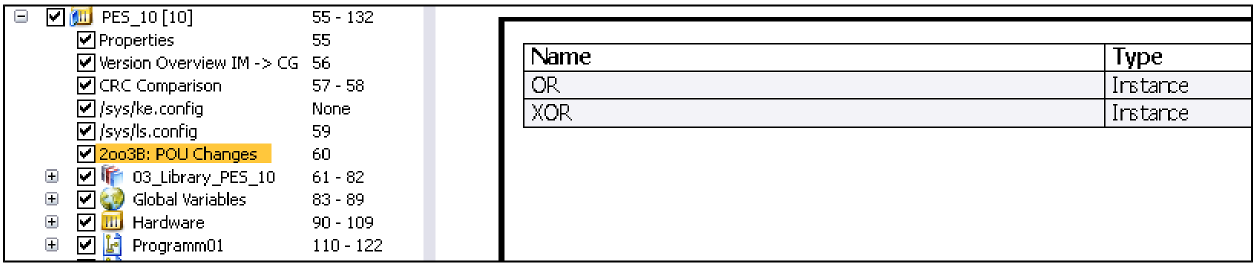
The required pages are marked in the project element list, e.g., *CRC Comparison* and */sys/ls.confg* of the version comparison performed last. All version comparison results are displayed with the specification of coordinates (grid), concerned instance and execution order.



□ Project structure	6
⊟ ☑ Configuration	7 - 157
⊞ □ 02_Standard_Lib...	7 - 53
⊞ □ Global Variables	54
⊟ ☑ PES_10 [10]	55 - 132
☑ Properties	55
☑ Version Overview...	56
☑ CRC Comparison	57 - 58
☑ /sys/ke.config	None
☑ /sys/ls.config	59
☑ 2oo3B: POU Cha...	60
⊞ ☑ 03_Library_P...	61 - 82
⊞ ☑ Global Variabl...	83 - 89
⊞ ☑ Hardware	90 - 109
⊞ ☑ Programm01	110 - 122
⊞ ☑ Programm02	123 - 124
⊞ ☑ Protocols	125 - 127
⊞ ☑ safeethernet	128 - 132
⊞ □ PES_20 [20]	133 - 156
⊞ □ safeethernet	157

/sys/root.config
/sys/bgp.config
/sys/cpc.config
/sys/cpcnsip.config
/sys/cpcsip.config
/sys/cpu.config
/sys/io4cpu.config
/sys/ke.config
/sys/ls.config
/sys/ls/Programm01.config
/sys/ls/Programm01.ldb
1oo2_R
2oo3
2oo3B
Average
BLINK

Additional information can be displayed for *2oo3B,*, the function block type that was identified as changed by the version comparator.



⊟ ☑ PES_10 [10]	55 - 132
☑ Properties	55
☑ Version Overview IM -> CG	56
☑ CRC Comparison	57 - 58
☑ /sys/ke.config	None
☑ /sys/ls.config	59
☑ 2oo3B: POU Changes	60
⊞ ☑ 03_Library_PES_10	61 - 82
⊞ ☑ Global Variables	83 - 89
⊞ ☑ Hardware	90 - 109
⊞ ☑ Programm01	110 - 122

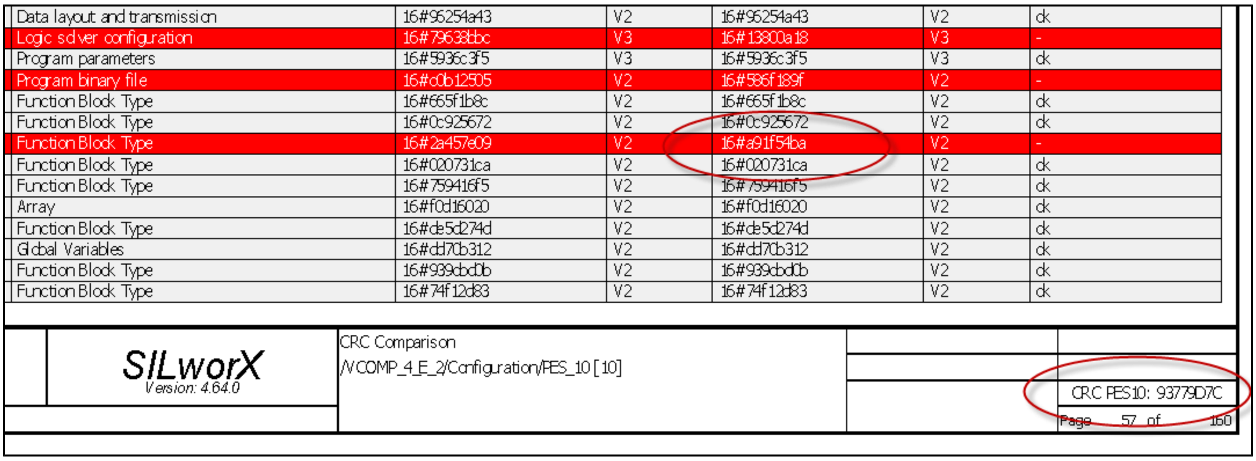
Name	Type
OR	Instance
XOR	Instance

To provide a documented proof of the changes, HIMA recommends to print out the current versions of all objects mentioned in the version comparison, such as function blocks or hardware, and to note significant revision and change details on the cover sheet.

Additionally, ensure that the print-out is provided with the current date.

Example

The cover sheet details include the sum CRC, which was specified manually.



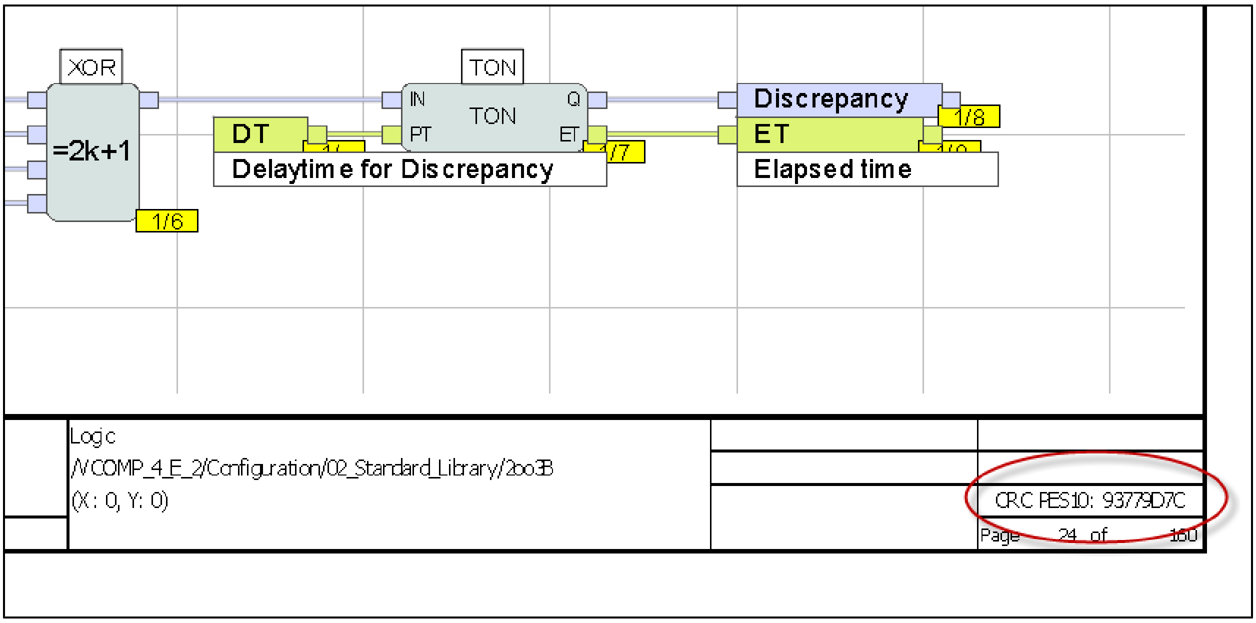
Data layout and transmission	16#96254a43	V2	16#96254a43	V2	ok
Logic solver configuration	16#79638bbc	V3	16#13800a18	V3	-
Program parameters	16#5936c3f5	V3	16#5936c3f5	V3	ok
Program binary file	16#c0b12505	V2	16#586f189f	V2	-
Function Block Type	16#665f1b8c	V2	16#665f1b8c	V2	ok
Function Block Type	16#0c925672	V2	16#0c925672	V2	ok
Function Block Type	16#2a457e09	V2	16#a91f54ba	V2	-
Function Block Type	16#020731ca	V2	16#020731ca	V2	ok
Function Block Type	16#759416f5	V2	16#759416f5	V2	ok
Array	16#f0d16020	V2	16#f0d16020	V2	ok
Function Block Type	16#de5d274d	V2	16#de5d274d	V2	ok
Global Variables	16#dd70b312	V2	16#dd70b312	V2	ok
Function Block Type	16#939dbd0b	V2	16#939dbd0b	V2	ok
Function Block Type	16#74f12d83	V2	16#74f12d83	V2	ok

SILworX
Version: 4.64.0

CRC Comparison
/VCOMP_4_E_2/Configuration/FES_10 [10]

CRC FES10: 93779D7C
Page 57 of 160

This CRC is then also indicated on the print-out of the modified function block.



Logic
/VCOMP_4_E_2/Configuration/02_Standard_Library/2oo3B
(X: 0, Y: 0)

CRC FES10: 93779D7C
Page 24 of 160

HI 801 286

HIMA Paul Hildebrandt GmbH + Co KG
P.O. Box 1261 I 68777 Brühl I Germany
Phone +49 6202 709-0 I Fax +49 6202 709-107
www.hima.com

SAFETY
NONSTOP

For a detailed list of all our subsidiaries and representatives,
please visit our website: www.hima.com/contact