Content

1   Release Notes ELOP II V5.6

This document contains supplementary notes on removed errors and improvements of ELOP II V5.6 that are not yet described in the online help.

To maintain the safety and availability of a controller programmed with ELOP II, the restrictions described in this document must be observed!

The following information is displayed in the *About ELOP II* dialog box:

Text displayed in ELOP II	Complete version designation
ELOP II V5.6 B1501.9810	ELOP II V5.6.Build 1501.9810IV1

1.1   Compatibility

ELOP II V5.6 can be used for all the following HIQuad operating system versions:

- BS41q/51q V7.0-8

ELOP II V5.6 can be used in PCs with the following operating system:

- Microsoft® Windows 7 Professional/Ultimate, 64-bit
- Microsoft® Windows 8.1

The minimum requirements for the computer used to run ELOP II V5.6 are specified on the current HIMA DVD.

In particular with very large projects, old PCs may require long processing times and thus be inappropriate for this task. Therefore, state-of-the-art computers should be used whenever possible. Enhanced hardware features such as computing power and memory space result in improved performance.

2 Improvements Compared to V5.1 (build 730 IV5)

1 The view in the Variable Declaration Editor is automatically synchronized.
A synchronization process occurs when one of the following elements is selected in the drawing area of the Function Block Editor:

- A value field containing a variable.
- An action block.
- A function block instance.

If the synchronization behavior is not desired, it can be disabled using the ⟳ (**Synchronize View**) button.

2 New ⊞⊷⊪⊪⊹⊒⊟⊠⊞⊟⊸ buttons on the Symbol Bar for aligning, resizing and spacing actions.

These buttons can be used to align, resize and space the selected function block elements.

These layout changes were already available in V5.1 and could be accessed through the context menu functions of the selected function block.

3 The extra **Ctrl** key is no longer required to drag function chart snippets onto the drawing area.

Starting with V5.1, the **Save FUP-Snippet as** function can be used to create user-defined function chart snippets. A function chart snippet is a function chart portion that can be reused. It may consist of any function chart elements.

- The extra **Ctrl** key is no longer required to drag function chart snippets created with a version higher than V5.6.
- The extra **Ctrl** key is still required to drag function chart snippets created with V5.1.

4 New ⬛ button (**Set Block)** on the Symbol Bar

To set function blocks in the drawing area of the Function Block Editor more easily and quickly, proceed as follows:

a Click the **Set Block** button on the Symbol Bar.
Result: The *Set Block* dialog box appears.
b Enter the function block name and press the Enter key.
Result: The function block preview is displayed in the drawing area.
c Click the mouse button to set the function block in the drawing area.
d Press Ctrl+Shift+N to return to the *Set Block* dialog box.
e Repeat steps b through d, to set as many function block as required.

5 New ⬛ button (**Delete Block and Reconnect Lines**) on the Symbol Bar

This button is used to delete selected function blocks from the drawing area and simultaneously connect the attached input and output lines.

This action could already be performed in V5.1 using the **Delete/Reconnect Lines** function from the context menu of the function block.

6 New ⬛ button (**Check All**) for the Documentation Editor.

The new button is available on the Symbol Bar of the Documentation Editor.

It helps detect problems in the print-out and locate them.

This action was already available in V5.1 using the **Check All** function from the context menu of the Documentation Editor.

7 New ↰ (**Undo**) and ↱ (**Redo**) buttons on the Symbol Bar.

The last changes performed to the function block (POU) are undone or redone using the **Undo** and **Redo** buttons, respectively. The **Undo** and **Redo** functionalities are only available in POUs, but not in program or type instances!

For instance, if the wrong object has been selected in the drawing area or the wrong variable has been renamed in the Variable Declaration Editor, **Undo** can be used to undo the last change.

The **Undo** action can also be triggered using Ctrl+Z.

The **Redo** action can also be triggered using Ctrl+Y.

---

i In versions prior to V5.6, Ctrl+Y was used to extend an existing sequence. In V5.6 and higher versions, Ctrl+Y triggers the **Redo** action.
To continue or extend an existing sequence, use the new shortcut Ctrl+Shift+A.

Important restrictions for **Undo** and **Redo** actions:

- The **Undo** actions are limited to 20 changes.
- How long the **Undo** or **Redo** action will take, depends on the size of the function block (POU).
- Some actions performed in the function blocks (POUs) cannot be undone:
  - Saved function blocks (POUs). Once the function blocks (POUs) have been saved, the saved changes can no longer be undone.
  - Layout changes in the Variable Declaration Editor or in the page list.
  - View-relevant changes (a moved pane view), e.g., performed with the scrollbar or with commands such as **Goto.**
  - When changes in the logic (e.g., drawing a line, adding a comment, etc.) are undone, the view related to the zoom factor is automatically adapted accordingly.

8 New shortcut Ctrl+Shift+A for continuing or extending a sequence.

In versions prior to V5.6, Ctrl+Y was used to extend an existing sequence. In V5.6 and higher versions, Ctrl+Y triggers the **Redo** action.

To continue or extend an existing sequence, use the new shortcut Ctrl+Shift+A. Observe that Ctrl+Shift+A only triggers the function if the cursor points to the SFC element.

9 Continue the connector using the shortcut or the **Create Continuation** context menu function.

Starting with V5.6, connectors can be continued using the **Create Continuation** function from the context menu of the connector.

The action can also be triggered using Ctrl+K. To this end, click the connector break and press Ctrl+K to create a connector continuation.

---

i If no connector has been selected, the Ctrl+K shortcut creates a new connector break instead of a connector continuation.

---

10 Redesigned Data Type Editor.

The Data Type Editor was redesigned as follows:

- Bigger declaration entry field.
- Revised elements for graphical representation; all colors are now simultaneously displayed within a preview.
- If more space is required for the Declaration field, the dialog box can be enlarged.

Similarly, the dialog box of the Variable Declaration Editor can be enlarged if more space is required for the comment field.

11 Analog values are displayed in the online changing dialog box as hexadecimal and signed integer values.
So far, the dialog box for showing the changes to force values and constants only

displayed the analog values in hexadecimal format. With V5.6 and higher, the analog values are displayed as hexadecimal as well as signed integer values.

---

i Analog values are displayed as signed integer values irrespective of the used data type. For other data types, these values must be converted.

---

12 The variable selection list is sorted alphabetically
The selection list of variables in the Element Selection dialog box for OLT fields is sorted alphabetically. So far, the variables were organized based on their creation date/time.

3 Problems Resolved in V5.6

This chapter describes problems in versions prior to V5.6 that have been resolved in V5.6.

1 ELOP II terminated when trying to import an Excel file
The problem only occurred when trying to import an Excel file. The problem did not occur if the file was created in a common text editor. [HE22046]

2 Warning: "LTC026: The statement list is not sorted"
For each POU used in the project, the *Check Object Data* function created the warning: "LTC026: the statement list is not sorted". [HE23299]

3 Administrator rights are not necessary after Windows time change
As a result of the Windows automatic time change (daylight savings time), ELOP II could no longer be started if the Windows user only had the privileges associated with the *Standard User* group. [HE23449]

4 ELOP II terminated after setting breakpoints
The problem occurred during the offline simulation when a breakpoint in an instance was set to a POU which was within the memory range of > 96 Kbyte. [HE23630]

5 Online use of global cross-references no longer cause the revision to change
The use of global cross-references resulted in a code-relevant change of the POU. If the global cross-reference was used to navigate during the online test and then saved, the online test was aborted when it was started again; an error message appeared informing that the loaded code does not match all the POUs. [HE26046]

6 Reload generation is now possible even after changed BUSCOM assignment

The reload code generation for a project was aborted with error under the following conditions:

- The loaded project configuration contained a variable with an I/O assignment and BUSCOM attribute.
- This variable still had an I/O assignment in the project configuration that was being processed, but no BUSCOM attribute.

These conditions were detected as an inconsistency of the BUSCOM assignment. A subsequent download code generation was aborted with error as well, but it was possible after restarting ELOP II. [HE26153]

7 Different blinking frequencies of the *Acknowledge* button

During the online test, the *Acknowledge* button was blinking faster than in the Control Panel.

[HE23626]

8 ELOP II: Faulty description of *Displaying/Forcing Values*

The description of the actual value was used in the online help for describing the force value, and vice versa.

[HE24926]

9 ELOP II terminated during the tag name assignment

Using the *Tag Assignment* dialog box in the *Edit Cabinet Layout* editor could cause ELOP II to terminate.

[HE26000]

10 The HTML documentation for project elements created by the user was overwritten

The *HTML Documentation* context menu item was removed for all the project elements in the structure tree.

[HE26565]

4 Restrictions of V5.6

1 The event number cannot be assigned.

The event number can be assigned twice if the event attribute and a hardware system variable are set in the *HW Assignment* menu item for a variable. After the variable is deleted, the hardware system variable retains the event number. This event number cannot be assigned to another variable as long as it is used for the hardware system variable.

Workaround:
- In the user program for the variable, open the declaration dialog box.
- Activate *Assign Tag Name* and select the hardware system variable previously connected.
- Click *Update.*
- Deactivate *Assign Tag Name.*
- Click *Update.*
- Save the user program.

[HE21204]

2 When generating non-reloadable code, ELOP II displays POU instances as changed. If POUs are added and non-reloadable code is then generated, ELOP II can display the POU instances (HEADER variables) as changed even if they were not modified. Workaround: Generate reloadable code. [HE12676]

3 Large force images cannot be loaded into the controller. Operating system version BS41q/51q V7.0-8 (07.14) refuses to load large force images with more than 60 modified forced variables.
Workaround: Perform major changes to the force image in smaller steps with a maximum of 60 changes each and load them.

It is also possible to save force images after a maximum of 60 changes and then to load them into the controller. If this is done, ensure that the images are loaded in the same order as they have been saved! [HE19490]

4 Online change of system parameters may have no effects.
Safety parameters may be changed online and loaded into the controller. The controller applies or ignores the changed settings depending on the **Parameter Online Change** safety parameter. However, since these settings are stored in the controller's working memory due to the loading process, they are also identified and displayed as safety parameter changes, although their application is prevented by the safety parameter.
In particular, this occurs if the **Parameter Online Change** is set to FALSE and then transferred online to the PES. The displayed details no longer correspond to the settings actually used in the controller.

Workaround: Manually document the settings in use when **Parameter Online Change** is set to FALSE, so that they can be reset. [HE19818]

5 Code generation aborted after language switching.
If one or more system function blocks from the ELOP LIB are used in the user program and code is generated, code may no longer be generated after switching the language in ELOP II. After aborting the code generation, the following message in German or English is displayed:

```
Fehler beim Erhalten von Typ-Informationen für Typ-ID
Error on obtaining type information for type ID
```

Workaround: After switching the language in ELOP II, open and save the user program. In doing so, the system function blocks from the ELOP LIB used in the user program are updated. The code generation can then be performed.
[HE25508]

6 ELOP II DEMO version only accessible with write protection
The ELOP II DEMO version does not work as described in the official description "DemoRestrictions_local.html", but can only be accessed read-only.
Unlike previous versions, the demo version of ELOP II V5.1 (and higher versions) no longer allows creation of small user programs with up to 10 POUs. [HE26632]

5 Upgrading from a Previous Version to ELOP II V5.6

Observe the following points:

- ELOP II V3.0 and higher versions can be installed simultaneously and operated by the same Windows user. Separate installation paths must be specified to install different ELOP II versions.
  A new registration is required whenever switching between ELOP II versions. To do so, administrator rights are needed.

- Projects that were created with V5.1 and higher versions need not be converted.

- Projects that were created with a version up to V5.1 need be converted. Observe the conversion instructions specified in the *New Features ELOP II V5.1* manual (HI 800 185 E).