



1 New SILworX Version

- Version 1.20, for HIMax Controllers only

This version was released with immediate effect.

2 Notes

When using SILworX, version 1.20, some restrictions must be observed. These restrictions have no influence on safety and on the availability of the code generated for a HIMax controller.

2.1 Safety-Related Faults during Reload

- **A project created with SILworX , version 1.12 (or below), must not be loaded into the controller by performing a reload, if the sequential function chart (SFC) was used in the user program!**
Workaround: create a new project using version 1.20 and load it into the controller by performing a download. [HE12597]
- **When multidimensional arrays with index variables are accessed, the array elements are improperly addressed.** In particular, the array elements that are accessed when index variables are used differ from those accessed when literal indexes with identical values are used.
Workaround: Use "Array of Array of" instead of a multidimensional array. [HE12800]

2.2 Restrictions when Executing a Program

- The function blocks behave differently depending on whether EN/ENO are displayed or not.
The function blocks SHL, SHR, ROL, ROR, DIV, DIV_TIME and MOD behave differently depending on whether EN/ENO are displayed or not. If ENO is set to FALSE while one of these function blocks is being processed, the values determined in the function block are not assigned to the value fields. This is only the case if EN and ENO are displayed.
Please take this different behaviors into account while programming. [HE12168, Restriction]
- Strong variations of the cycle time during REAL/LREAL calculations
If REAL/LREAL variables are used, the cycle time can strongly vary. This effect is particularly noticeable if trigonometric functions are performed.
Workaround: Try to approximate the cycle time by using real values during tests. [HE12115]

2.3 Restriction with Respect to FSC Networks

- Incorrect processing sequence with transitions vertically arranged one below the other
If transitions are vertically arranged one below the other, the processing sequence is not determined by the graphical order. The transitions are rather processed following their creation order. [HE12581]

2.4 Restrictions with Respect to the Version Comparator

- Function blocks have the same CRC, in spite of modified logic
After changing function blocks in a program, it could happen that a new CRC is calculated for the program, but not for the changed function blocks.
No change of the function blocks is thus reported by the version comparator!
For this reason, it is important to manually note if function blocks have been changed and to thoroughly test the user program! [HE12284]
Eliminated in version 1.20: For most of the POU's, the associated CRC is probably changed after the translation. This is not a problem if the CRC of the binary remains unchanged.

2.5 Restrictions with Respect to Reload

- Variables without a source for their value are differently treated during a reload
Depending on the type, variables without a source for their value are treated as follows during a reload:
 - Variables of type VAR_Const are initialized
 - Variables of types VAR_Global are initialized
 - Local variables, that are not of type VAR_Const, are not initialized

Workaround: A variable without a source of its value can be initialized by assigning it to a constant having a suitable value or by converting it into a constant. [HE12058]

- Reloadable code cannot be created if the user program did not change
This also applies if changes, that do not affect the program, were made in comment fields. [HE12096, Restriction]
- The reload switch for the user program seems to be not operative
The switch "Reload" of the user task controls the reload of the user task's parameters. Setting it to "Reload not allowed" prevents the user program's parameters, but not the user program, from being reloaded.
If you want to prevent reloading the user program, set Reload Enable of the user program to "non-enabled" (no checkmark). [HE12222]

2.6 Restrictions with Respect to the FBD Editor

- Copying online test fields terminates SILworX
If drag&drop is used in the online mode to copy variables from the table to the logic, independent online test fields are created.
Copying one of these independent online test fields in the FBD Editor terminates SILworX.

Release of the Planning Tool SILworX

Workaround: Do not copy independent online test fields in the editing mode. [HE12099]

- Updating a function block instance terminates SILworX
Updating a function block instance terminates SILworX, if an input or output was previously added at the top of the function block's interface. This has shifted the remaining inputs or outputs downwards by one position.
Workaround: If a function block interface was modified such that inputs or outputs were shifted downwards, release all connections from/to the function block instance prior to performing the update. Re-add the connections after completing the update. [HE12215]
- Copied connection lines are not connected
If in the FBD Editor a connection line that is connected on both sides to objects is copied or cut, and pasted between objects having the same distance, the connection line actually is not connected to the new objects. The connection line is graphically represented as connected, but it is not updated when the objects are moved; the verification and the code generator report corresponding errors.
This also applies if one of the source objects was included in copying.
Workaround: Always create new connection lines. [HE12124]
- Input connectors are wrongly copied and pasted
If an input connector is cut and pasted, also the output connector associated with it is deleted. The pasted input connector has a new name. This deletes all connections created by the connector.
Workaround: Do not use the copy&paste function with connectors. [HE11563]
- Moving crosspoints may terminate SILworX
Moving crosspoints of connection lines may terminate SILworX.
For this reason, we recommend saving the changes previously made prior to moving crosspoints to prevent that the changes are lost. [HE11759]
- FBD Editor blocked
When working with the FBD Editor, it could happen that the FBD Editor no longer reacts to mouse clicks.
Workaround: Double-click a function block or close a function block and re-open it. [HE11378]

2.7 Restrictions with Respect to the Online Test

- In value fields containing literals, the logic state is not displayed
If the value is entered as a text in a value field of a Boolean variable (TRUE), the logic state of the out-bound connection line is displayed not red in the online test, but blue. [HE11854]
- No online test after changes performed in the FBD Editor
If, in the FBD Editor, changes were made compared to the loaded user program, online test becomes impossible. This also applies after changes of non-code-relevant elements such as comment fields.
Workaround: Edit/Browse the variables in the tabular Force Editor or generate a new code (with/without reload), and load the controller newly. [HE12056]
- In the online test, the background color for forcing is not correct

If forcing is only prepared and the force main switch is still OFF, the background color of the online test field is yellow instead of gray. The color should only be yellow if the force value is being actually processed. [HE10884]

- Online test fields ignore negation on function block outputs
Online test fields connected to negated function block outputs display the non-negated value. [HE11371]
- Connection lines are not always refreshed during the online test
During the online test, connection lines of type BOOL are not refreshed if their source is not located in the visible area. [HE11710]

2.8 Restrictions with Respect to Redundant Modules

- Default for the combination of redundant input modules can trigger a safety-related reaction
If two input modules are redundantly combined, a Boolean operation exists with which the value that is processed by the user program is determined from the redundant input signals. The default setting for digital input modules is "AND". If one of the two input modules is removed, it could temporary create a malfunction message for some or for all channels. The Boolean operation "AND" ensures that this malfunction message is sent to the user program event if the redundant input module still provides undisturbed values. This can trigger the safety-related reaction.
Workaround: For digital inputs, enter the Boolean operation "OR" to increase availability, for analog inputs enter "MAX".
The Boolean operation "AND" will no longer be supported in future. [HE12354]

2.9 Other Restrictions

- If an array data type is copied and pasted, SILworX terminates
Workaround: Do not copy array data types. [HE12236]
- VISTA is not released for SILworX
Only install SILworX on a PC with Windows XP. [HE11694]
- Reconnecting diagnostic view causes a module login, instead of a system login
If the diagnostic view is opened during a system login and the connection is closed, the module login is offered when attempting to re-establish the connection.
Workaround: Establish the connections to the system either using the Hardware Online View or the Control Panel.
If the module login was opened, close all module views: the diagnostic and the module view. [HE11926]
- Is it impossible to save changes in an Editor?
After specific changes made within an Editor, the message "Impossible to save changes" appears while attempting to save. After confirming the message, however, the changes are saved.
If the Editor is then closed and re-opened, the message "The required data is being processed" appears.
Examples of changes in which this problem occurs are the cyclic renaming of variables (A => B, B => C, C => A).
Workaround: Avoid exchanging names.

If required, restart SILworX. [HE11613, Restriction]

3 Importing Projects from V1.12

Remark: The code generated with version 1.12, can still be loaded. Changes made with version 1.20, however, must be tested.

Projects created with version 1.12 can be converted to version 1.20. To do so, proceed as follows.

1. Translate the resource with version 1.12 and note the CRC of *root.config*!
 2. Install version 1.20.
 3. Open the project with version 1.20. SILworX converts the project to version 1.20.
 4. Translate the resource with version 1.20. Compare the calculated CRC of *root.config* with the CRC previously noted.
- If the two CRCs are identical, the conversion was successful.
 - If the two CRCs are not identical, the project must be tested to ensure that the function is still achieved.

3.1 Restrictions with FCS Networks in V 1.12

- Incorrect processing sequence in networks with FSC actions
FSC actions change the processing sequence in discontinuous, partial networks. [HE12301]
Eliminated in version 1.20: This can cause the user program to change its behavior . If the binary CRC is changed with respect to version 1.12, the program must be tested and, if required, modified!
- Inverted outputs of FSC actions are not taken into account
If the Q output of a FSC action is inverted, the inversion is not executed. [HE12313]
Eliminated in version 1.20: This can cause the user program to change its behavior . If the binary CRC is changed with respect to version 1.12, the program must be tested and, if required, modified!
- Improperly manual assigning of priorities causes incorrect behavior
If the transition priorities are manually assigned within FSC networks, they are not verified. The transition priorities in the generated code are stochastic. Transition might possibly not be executed.
Workaround: Use the automatic function for assigning priorities (graphical priority) or thoroughly verify the priorities assigned manually, check in particular their completeness. [HE12322]
Eliminated in version 1.20: This can cause the user program to change its behavior . If the binary CRC is changed with respect to version 1.12, the program must be tested and, if required, modified!
- Error message concerning FSC connections, even if the connection exists
After operating errors, SILworX may wrongly consider FSC connections as unconnected and refuse to generate the code. [HE12251]
Eliminated in version 1.20: This can cause the user program to change its behavior . If the binary CRC is changed with respect to version 1.12, the program must be tested and, if required, modified!

- Multiply used action block is reset.
If an action block with the same action qualifier is called within several steps, only the instance that was processed last due to its graphical position is effective. [HE12465]
Eliminated in version 1.20: This can cause the user program to change its behavior . If the binary CRC is changed with respect to version 1.12, the program must be tested and, if required, modified!

3.2 Restrictions with Function Blocks in V 1.12

- Incorrect processing sequence with function blocks with unconnected inputs
If a program logic contains function blocks without inputs or with unconnected inputs, they are processed in the network before the function blocks with connected inputs. This particularly affects the functionality of function blocks that use global variables. [HE12175]
Eliminated in version 1.20: This can cause the user program to change its behavior . If the binary CRC is changed with respect to version 1.12, the program must be tested and, if required, modified!
- Incorrect processing sequence with function blocks with unconnected outputs
If a program logic contains function blocks without outputs or with unconnected outputs, they are processed in the network after the function blocks with connected outputs. This particularly affects the functionality of function blocks that use global variables. [HE12176]
Eliminated in version 1.20: This can cause the user program to change its behavior . If the binary CRC is changed with respect to version 1.12, the program must be tested and, if required, modified!

3.3 Restrictions with Respect to the FBD Editor in V.1.12

- If parts of the program logic that contain the connection lines without corresponding input objects are copied, faulty logic parts are created that cause SILworX to terminate.
Workaround: If such faulty logic parts have been created in V.1.12, first delete the function blocks with unconnected outputs and then the connection lines. [HE12047]
Eliminated in version 1.20: This can cause the user program to change its behavior . If the binary CRC is changed with respect to version 1.12, the program must be tested and, if required, modified!