

PLC Designer



Migration of V2.x projects

Software Manual

EN



Contents

1	About this documentation	3
1.1	Conventions used	6
1.2	References used	7
2	Safety instructions	8
3	The "Controller-based Automation" system	9
4	General information	11
4.1	Preparing PLC projects in the »PLC Designer« V2.x	11
4.2	»PLC Designer« - preparing internal visualisation	12
4.3	Control configuration	13
4.4	Network variables	13
4.5	Access to the CANopen master or CANopen slave	13
4.6	L-force Motion	13
4.6.1	AxisGroup not available anymore	13
4.6.2	Structure of the L-force motion devices	14
4.6.3	Function blocks	14
4.7	System events	14
4.8	System libraries from the SysLibs23 range	15
5	Programming languages	16
5.1	Structured text - ST	16
5.2	AWL / IL	16
5.3	FUP / FBD	16
5.4	KOP / LD	16
5.4.1	Function blocks	16
5.4.2	Blocks with EN/ENO	16
5.5	AS / SFC	17
5.5.1	Step variables	17
5.5.2	Identifier	17
5.6	CFC	17
5.6.1	Big blocks	17
5.6.2	Comments	17
5.6.3	Macros	17
6	Update of a project from V2.x to V3.x	18
6.1	Components of the V2.x example project	18
6.2	Import of the example project into the »PLC Designer« V3.x	19
7	Update of a project from V3.0 to V3.x	25
7.1	Updating the target system	25
7.2	Updating the project environment	25
	Your opinion is important to us!	28

2 About this documentation

This documentation contains information on how to import older »PLC Designer« projects into the »PLC Designer«.

This manual is assigned to the "Controller-based Automation" manual collection which consists of the following documentations:

Documentation	Subject
System manuals	• "Controller-based Automation" • Visualisation
Communication manuals	• "Controller-based Automation" EtherCAT® • "Controller-based Automation" CANopen® • "Controller-based Automation" PROFIBUS®
Software manuals (SW)	• »PLC Designer« • »Engineer« • »VisiWinNET® Smart« • »Backup & Restore«



Information on how to use the L-force Controller beyond control technology can be found in the system manuals dealing with the respective application case.

More technical documentation for Lenze components

More information on Lenze components that can be used in connection with "PC-based automation" can be found in the following documentation:

Mounting & wiring
 MAs Controller
 MA i700 Servo inverter
 MAs Servo Drives 9400
 MAs Inverter Drives 8400
 MA I/O-System 1000 (EPM-Sxxx)
 MAs Servosystem ECS
 MAs communication card (MC-xxx)
 MAs communication modules

Parameterisation, configuration, commissioning
 SW Controller
 SW Servo-Inverter i700
 SW Servo Drive 9400 HighLine/PLC
 9400 HighLine commissioning guideline
 SW Inverter Drive 8400 Baseline/StateLine/HighLine/TopLine
 SHB I/O-System 1000 (EPM-Sxxx)
 BAs for the ECS servo system
 KHBs for the communication modules

Programming
 SW 9400 function library

Reuse
 SW Application Sample i700
 SW Application Samples
 SW ApplicationTemplate

Legend:

-  Printed documentation
-  Online help/PDF-File

Abbreviations used:

- BA** Operating instructions
- KHB** Communication manual
- MA** Mounting instructions
- SW** Software manual
- SHB** System manual

Target group

This documentation is intended for all persons who want to parameterise, configure, and diagnose a Controller-based control system using a Controller in conjunction with the »PLC Designer« engineering software.

About this documentation

Information on the validity

The information in this documentation is valid for the following Lenze software:

Software	From software version
»PLC Designer«	3.x

Document history

Version			Description
1.0	10/2010	TD11	First edition - control technology 3.x
1.1	05/2011	TD11	Updated for control technology 3.1 • Migration of projects 3.0 -> 3.x added
1.2	02/2013	TD11	Updated for control technology 3.5 • Advice for updating libraries added



Tip!

Current documentation and software updates for Lenze products can be found on the Internet in the "Download" area at:

<http://www.Lenze.com>

About this documentation

Conventions used

2.1 Conventions used

This documentation uses the following conventions to distinguish between different types of information:

Type of information	Writing	Examples/notes
Spelling of numbers		
Decimal separators	Point	The decimal point is always used. For example: 1234.56
Text		
Version information	Text colour blue	Information that is only valid for or from a certain software version of the controller is marked accordingly in this manual. Example: This function extension is available from software version V3.0!
Program name	» «	»PLC Designer«...
Window	<i>italics</i>	The <i>Message</i> window ... / The <i>Options</i> dialog box...
Variable identifier		Set <i>bEnable</i> to TRUE to...
Control element	bold	The OK button... / The Copy command... / The Properties tab... / The Name input field...
Sequence of menu commands		If the execution of a function requires several commands, the individual commands are separated by an arrow: Select File→Open to...
Shortcut	< bold >	Press < F1 > to open the online help.
		If a command requires a combination of keys, a "+" is placed between the key symbols: Use < Shift >+< ESC > to...
Program code	Courier	IF var1 < var2 THEN a = a + 1 END IF
Keyword	Courier bold	
Hyperlink	<u>underlined</u>	Optically highlighted reference to another topic. In this documentation it is activated by mouse-click.
Symbols		
Page reference	 6	Optically highlighted reference to another page. In this documentation it is activated by mouse-click.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

About this documentation

References used

2.2 References used

This documentation uses the following signal words and symbols to indicate dangers and important information:

Safety instructions

Structure of the safety instructions:



Pictograph and signal word!

(characterise the type and severity of the danger)

Note

(describes the danger and gives information about how to prevent dangerous situations)

Pictograph	Signal word	Meaning
	Danger!	Danger of personal injury through dangerous electrical voltage Indicates an impending danger that may lead to death or severe personal injury if the corresponding measures are not taken.
	Danger!	Danger of personal injury through a general source of danger Indicates an impending danger that may lead to death or severe personal injury if the corresponding measures are not taken.
	Stop!	Danger of damage to material assets Indicates a potential danger that may lead to damage to material assets if the corresponding measures are not taken.

Application notes

Pictograph	Signal word	Meaning
	Note!	Important note to ensure trouble-free operation
	Tip!	Useful tip for easy handling
		Reference to another documentation

3 Safety instructions

Please observe the following safety instructions when you want to commission a controller or system using the L-force Controller.



Read the documentation supplied with the corresponding field device thoroughly before starting to commission the devices with the L-force Controller!

The device documentation contains safety instructions which must be observed!



Danger!

According to our present level of knowledge it is not possible to ensure the absolute freedom from errors of a software.

If necessary, systems with built-in controllers must be provided with additional monitoring and protective equipment according to relevant safety regulations (e.g. law on technical equipment, regulations for the prevention of accidents), so that an impermissible operating status does not endanger persons or facilities.

During commissioning persons must keep a safe distance from the motor or the machine parts driven by the motor. Otherwise there would be a risk of injury by the moving machine parts.



Stop!

If you change parameters in the »Engineer« or the »WebConfig« while a device is connected online, the changes will be directly accepted by the device!

A wrong parameter setting can cause unpredictable motor movements. By unintentional direction of rotation, too high speed or jerky operation, the driven machine parts may be damaged!

4 The "Controller-based Automation" system

Central control technology keeps finding its way into automation technology. Due to their scaling options and various possibilities of combining visualisation and control in one device, Industrial PCs offer great advantages for many applications.

Lenze Controllers are available in the following software equipments:

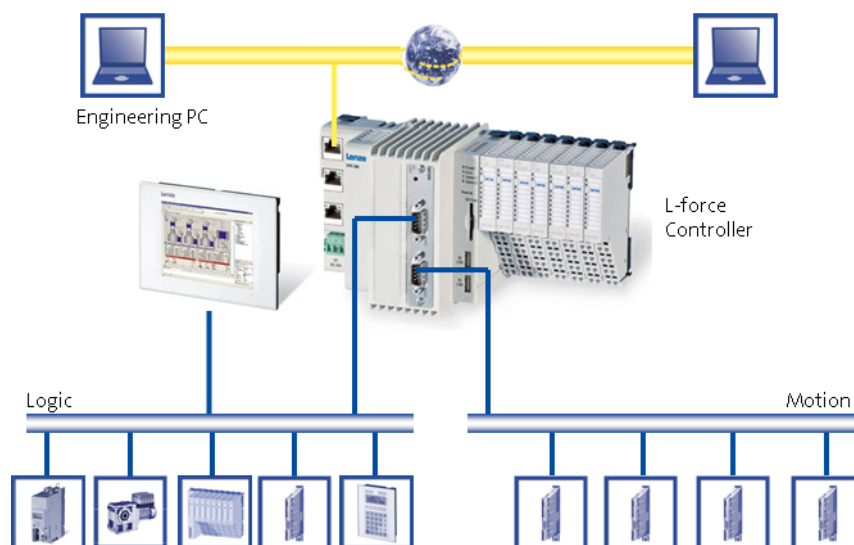
- Controllers as component, on request with operating system, without further software
- Controller as visualisation system (depending on the mode, an additional panel may be available)
- Controller as control system

The "Controller-based Automation" system enables central control of Logic & Motion systems.



Note!

- When the system is installed, we recommend to connect the Logic & Motion devices to separate phases!
- When the EtherCAT bus system is used, no separate phases for Logic & Motion are required.



More information on how to parameterise and configure the single bus systems can be found in the following communication manuals:

- Control technology CANopen®
- Control technology EtherCAT®
- Control technology PROFIBUS®



Read the Mounting Instructions accompanying the controller first before you start working!

The Mounting Instructions contain safety instructions which must be observed!

The "Controller-based Automation" system

Lenze provides well-matched system components:

- Controller as control and visualisation system
 - A Controller the central component of the PC-based Automation system that uses the Runtime Software to control the Logic & Motion functionalities.
 - The Controller uses the fieldbus to communicate with the field devices.
- Engineering tools for the Engineering PC
 - The Engineering PC uses Ethernet to communicate with the L-force Controller.
 - Use the various Engineering tools to configure and parameterise the system.
- Fieldbuses
- Field devices

General information

Preparing PLC projects in the »PLC Designer« V2.x

5 General information

This chapter describes the alterations necessary for importing the projects created in the »PLC Designer« V2.x into the version V3.0.

5.1 Preparing PLC projects in the »PLC Designer« V2.x

Preconditions/requirements

In order to be able to import a PLC project from the »PLC Designer« V2.x to V3.x, the following preconditions are required:

Precondition	Information/example	Remedy
It must be possible to compile the project in the »PLC Designer« version 2.3 <u>without any errors</u> .	When an error-free »PLC Designer« V2.x project is compiled, many warning signals are displayed in the »PLC Designer« V3. These are implicit conversions where information can get lost. (for instance a sign reversal)	The warning signals must be evaluated individually.
	The initialisation of variables with constants is checked more carefully in the »PLC Designer« V3.x: • Example: byte_var : BYTE := word_constant; if V2.x <u>no</u> error is in the »PLC Designer« V2.x - an error message is generated during the import to V3.	Correcting in the V2 project: Observe correct type data.
	In the »PLC Designer« V3.x, cases are checked against the switch variable: • CASE USINT OF INT is not checked in the »PLC Designer« V2.x - during the import into V3 an error message is generated.	Correcting in the V2 project: Observe correct type data in a case instruction.

Libraries used in the project...

- must not have any references to external files
- must be able to be compiled. ▶ [Convert libraries](#) (📖 21)

5.2 »PLC Designer« - preparing internal visualisation

- Visualisation: All variables in the visualisation must be declared. In the »PLC Designer« V2, a warning is displayed, in the »PLC Designer« V3, an error is displayed.
 - **Remedy:** Check visualisation: Remove non-declared variables.

"Single" wildcards

- The following wildcards can be imported error-free in the »PLC Designer« V3:

`$Var$.StructName.Varname`

`ArrName[$Index$]`

"Double" wildcards

- The following wildcards cannot be imported in the »PLC Designer« V3:

`$Instance$. $VarName$. . .`

`$Achse$$Spur$. . .`

Problematic wildcards

1. The wildcard occurs in a fixed text:

`Text: $axle$-AchselocalVar : STRING;`

`Text: %s-Achse TextVariable: localVar`

- **Remedy:**

`localVar : STRING;`

`Text: %s-axis`

`TextVariable: localVar`

2. The wildcard describes only a part of a variable name:

`achse$achse$spur$spur$.fActPosition`

- **Remedy:** Only one wildcard is defined for the `achse$achse$spur$spur$` wildcard:

`achse_spur : MyFunctionBlock;`

- Then the corresponding instance of the function block is passed directly:

`achse_spur := PLC_PRG.achse1Spur2;`

3. The wildcard is replaced by an expression:

`$Expression$ ?? PLC_PRG.var1 + PLC_PRG.var2`

- **Remedy:** The expression must be transmitted to an auxiliary variable. This auxiliary variable is then transmitted as an instance.

4. The wildcard describes a program:

`$Program$.bToggle ??PLC_PRG.bToggle`

- This type of wildcard replacement cannot be imported into the »PLC Designer« V3.x. (This wildcard replacement is only used in exceptional cases.)

5. The wildcard is replaced by different types:

`$Var$`

→ Replacement 1 : `PLC_PRG.n` (INT)

→ Replacement 2 : `PLC_PRG.st` (STRING)

- **Remedy:** Two different wildcards must be defined in the interface.

6. The visualisation is in a library. The wildcards will be replaced later from any project.

- **Remedy 1:** The TYPE_NONE data types must be replaced manually.
- **Remedy 2:** The library must be implemented into a project and the wildcard must be replaced correctly. If this project is then imported, the data type is determined correctly in the library as well.

Grouping

- Groups and (not yet) available elements cannot be imported into the »PLC Designer« V3.

5.3 Control configuration

The control configuration of a »PLC Designer« V2 project cannot be imported into the »PLC Designer« V3.

As a consequence...

- the device configuration must be created **anew**.
- the variables in the control configuration must be declared anew.

5.4 Network variables

- In the »PLC Designer« V3.x, GVL objects are created. The variable declarations can be imported.
- The network properties are not imported.

5.5 Access to the CANopen master or CANopen slave

The access to internal variables of the CANopen master/CANopen slave (e.g. `nStatus`) via the **global** data structures existing in the »PLC Designer« V2 project is not possible in the »PLC Designer« V3.

- The access takes place in the »PLC Designer« V3 via function calls of the CiA405 library (new in V3).

5.6 L-force Motion

5.6.1 AxisGroup not available anymore

The axis groups used in »PLC Designer« V2 are not available in V3.

- Hence, there is no need for the SMC_ResetAxisGroup function block that caused a reset of all motion axes available in this axis group.
- In the »PLC Designer« V3, this functionality is replaced by the SMC3_ReinitAxis block (new in V3).
 - This FB must be called individually for each axis.
 - A higher-level block for resetting several axes is not available in the »PLC Designer« V3.

5.6.2 Structure of the L-force motion devices

The single structures of the L-force motion devices in the »PLC Designer« V3 differ regarding ECS and 9400 CiA 402. This affects e.g. the control word and the status word.

5.6.3 Function blocks

Some function blocks are not needed in the »PLC Designer« V3 or replaced by other blocks (e.g. SMC_ResetAxisGroup, L_SMC_WriteParameter). In some cases, the numbers of inputs and outputs of the function blocks have changed (e.g. MC_Home).

The changes must be carried out manually after the conversion.

5.7 System events

The system events of the »PLC Designer« V2 project are visible in the task configuration after the import into the »PLC Designer« V3 but cannot be used.

- **Remedy:** (Before the import into »PLC Designer« V3), remove the system events in the »PLC Designer« V2 project. Use the functions of the CmpEventManager . lib library.

5.8 System libraries from the SysLibs23 range

Libraries that have been transmitted (due to reasons of compatibility) from the »PLC Designer« V2 to version 3, may respond differently than under V2.

- **Example:** *SysFileOpen* with Mode = a (append) does not create a file under 3.x (unless available). The »PLC Designer« V2.3 creates this file.
- **Remedy:** The use of the functions must be adapted manually (after the conversion) on an individual basis.



Note!

Import of variables of a »PLC Designer« V2.3 project into the »PLC Designer« V3

- The declarations of the **Retain variables** remain stored (and have an unchanged same effect).
- The declarations of persistent variables must be revised and re-created. Create an own global variable in the »PLC Designer« V3!
- An "AT Declaration" must not be used in combination with VAR RETAIN or VAR PERSISTENT!
- The size of the provided memory area in the »PLC Designer« V3 depends on the target system. If required, adapt the number and value of the retain variables!

6 Programming languages

This chapter describes the restrictions of the single programming languages when a »PLC Designer«-V2 project is imported into the »PLC Designer« V3.

6.1 Structured text - ST

No restriction known

6.2 AWL / IL

No restriction known

6.3 FUP / FBD

No restriction known

6.4 KOP / LD

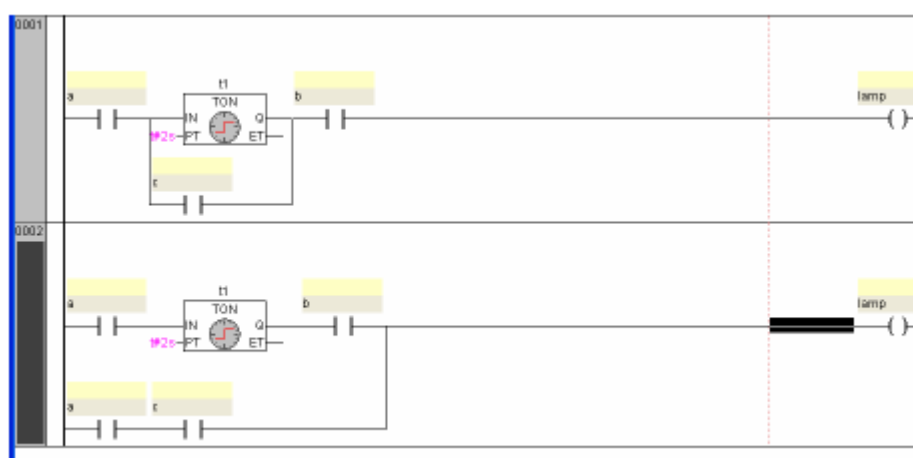
6.4.1 Function blocks

Function blocks may **only** be used in parallel branches that are directly connected to the power rail.

- Example: Network 1 cannot be imported, network 2 (same function as network 1) can be imported.

6.4.2 Blocks with EN/ENO

Blocks with EN/ENO must not be provided with any further blocks at the inputs. A variable is permitted.



6.5 AS / SFC

6.5.1 Step variables

Step variables explicitly declared by the user...

- must be declared locally in the SFC.
- must not be declared as VAR_INPUT, VAR_OUTPUT or VAR_INOUT since the calls cannot be adapted automatically.

Steps do not use any boolean variables for managing the internal states in the »PLC Designer« V3.x but structures of *SFCStepType* type.

6.5.2 Identifier

The following must not begin with an underscore...

- IEC actions in the tree
- boolean variables that are called in an IEC association list.
- the names of programmed transitions.

In the »PLC Designer« V3.x, the implicit variables (created for actions) get an underscore as prefix. An invalid identifier with a double underscore would be generated.

6.6 CFC

6.6.1 Big blocks

The graphical presentation of big blocks is insufficient after the import into »PLC Designer« (e.g., graphical fields may overlap).

6.6.2 Comments

When a multiline comment is imported, only the first line is displayed.

6.6.3 Macros

Macros cannot be imported into the »PLC Designer« V3.

Update of a project from V2.x to V3.x

Components of the V2.x example project

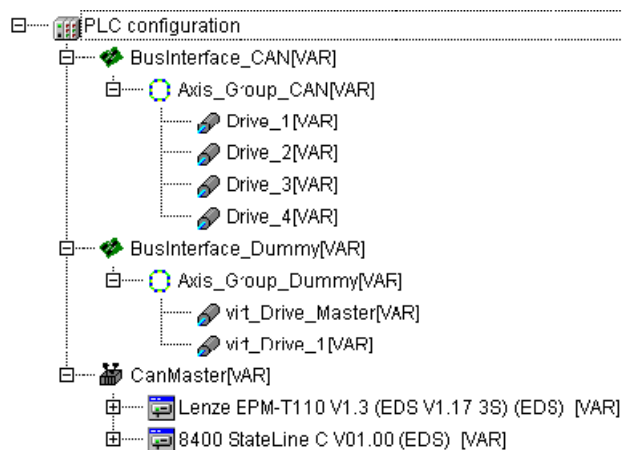
7 Update of a project from V2.x to V3.x

This chapter describes how to change a PLC program created in the »PLC Designer« V2.x in order to be able to import it into the »PLC Designer« V3.x.

7.1 Components of the V2.x example project

Control configuration

The control configuration of the example project is composed as follows:

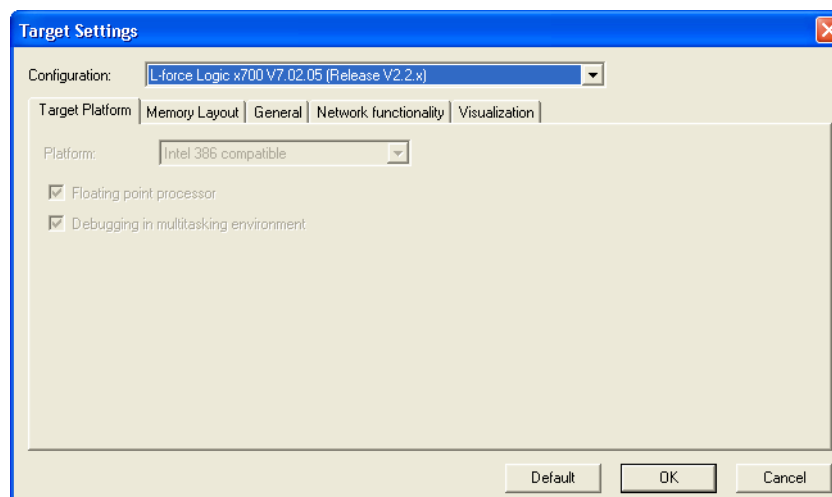


[7-1] Example: Control configuration in the »PLC Designer« V2.x

Contains Motion and Logic devices triggered via CAN bus by the Controller.

Target system

Set target system: **L-force Motion x700 V7**



Update of a project from V2.x to V3.x

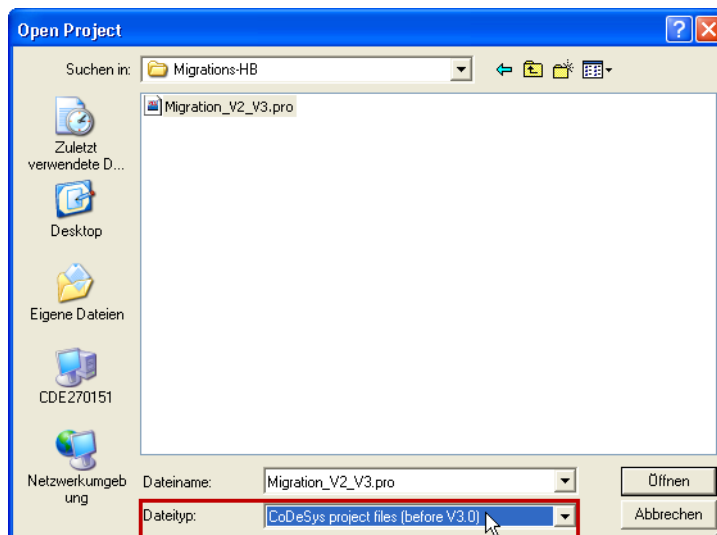
Import of the example project into the »PLC Designer« V3.x

7.2 Import of the example project into the »PLC Designer« V3.x

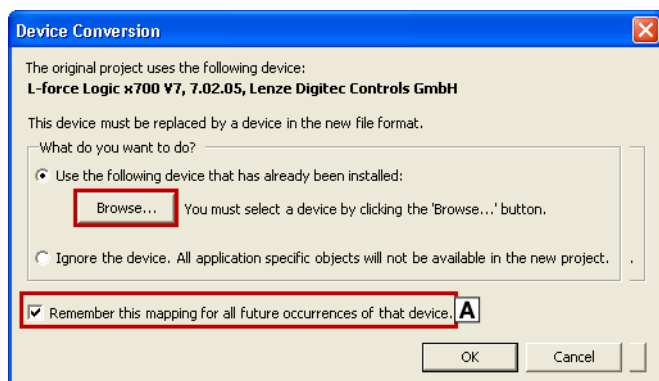


How to import a V2 project into the »PLC Designer« V3.x:

1. Start »PLC Designer« V3.x.
2. Open the »PLC Designer« V2 project (**File→Open project**):
 - Select file extension *.pro (CoDeSys projects (before V3.0)).



3. Convert devices
 - Replace the target system set in the V2 project by a corresponding V3 target system.

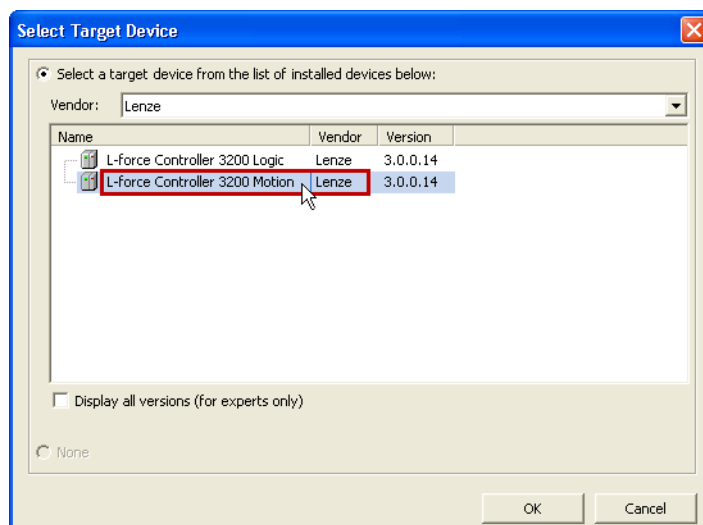


Note: You have the option to save the assignment for future projects: Activate the control field **A** to replace the target system automatically by this V2 target system during the next project import.

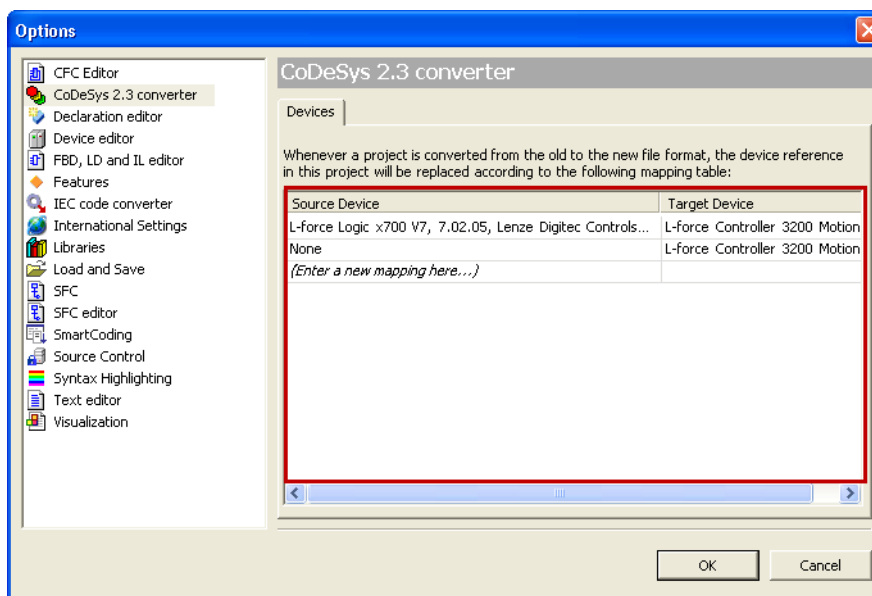
Update of a project from V2.x to V3.x

Import of the example project into the »PLC Designer« V3.x

- Replace the **L-force Motion x700 V7** target system by the **L-force Controller 3200 C Motion** target system:



The target system replacement already saved on the system can be checked and edited via **Tools→Options→CoDeSys 2.3 converter** :



Update of a project from V2.x to V3.x

Import of the example project into the »PLC Designer« V3.x

4. Convert libraries

The libraries used in the »PLC Designer« V2 project have to be replaced manually by the suitable libraries in the »PLC Designer« V3.

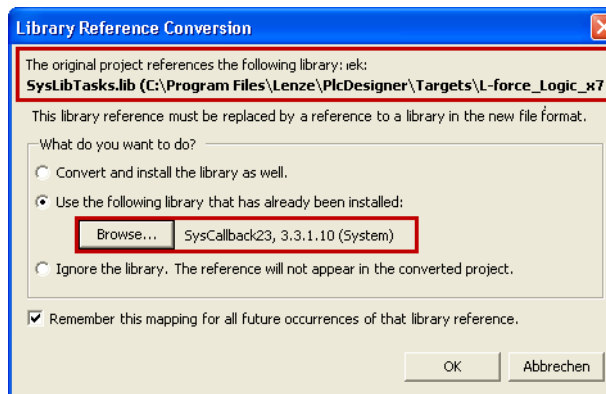
The following libraries have to be converted, for example (V2 → V3):

- Standard (V2) → Standard (V3)
- IecSfc (V2) → IecSfc (V3)
- All system libraries: SysLibxxx (V2) → SysLibxxx23 (V3)
Example: Replace SyslibPlcCtrl (V2) by SysPlcCtrl23 (V3)

Exceptions:

- A conversion of the 3S_CAN libraries is not required. → Click "Ignore library".
After inserting a CAN bus into the project, the suitable libraries are inserted automatically.
- A conversion of the Analyzation library is not possible in the »PLC Designer« V3. Thus, these function cannot be used in the »PLC Designer« V3.
- All Motion libraries (examples: SM_DriveBasic, SM_PLCOpen, DummyDrive, SM_CNC, SM_Trafo, SM_Error) have to be ignored. → Click "Ignore library".
When a Motion target system is inserted into the project, the required Motion libraries are automatically available due to SM3_Basic and SM3_CNC.

Example: Convert the reference to the **SysLibCallback.lib** library to V3:



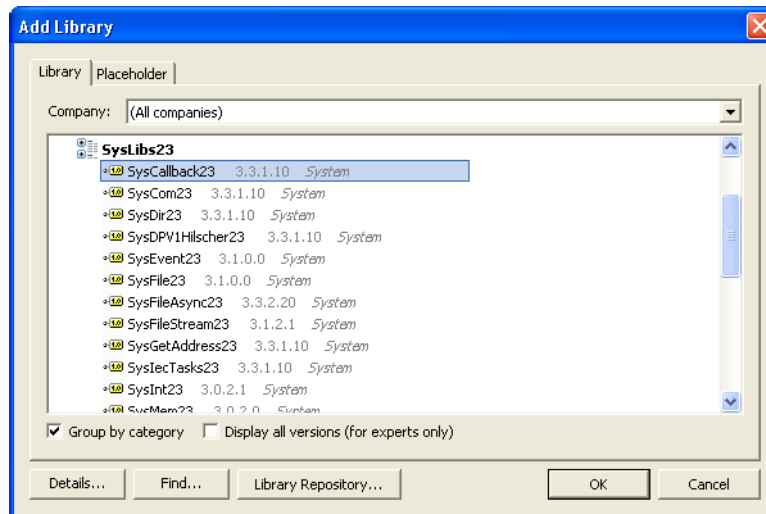
- Version 2-compatible system libraries are located on the V3 system in the \SysLibs23 directory.

Update of a project from V2.x to V3.x

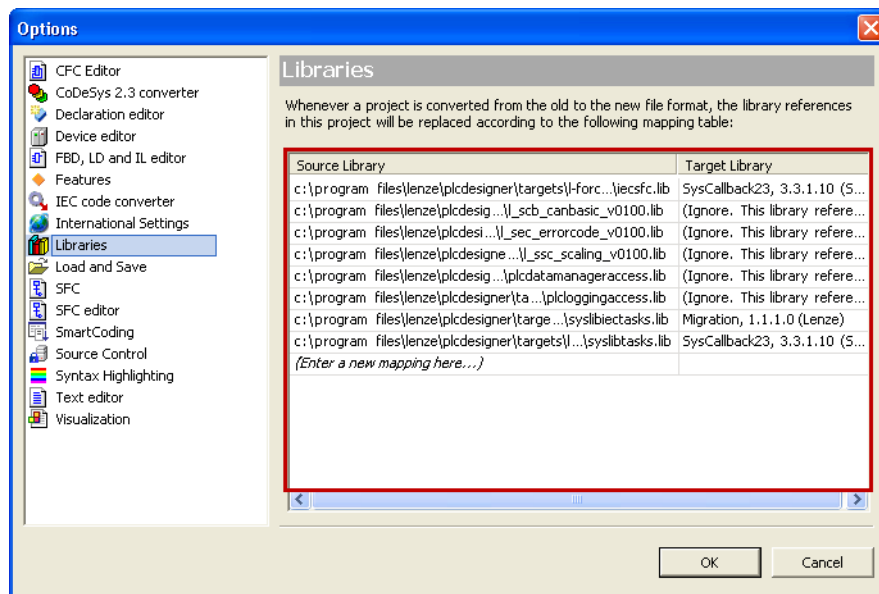
Import of the example project into the »PLC Designer« V3.x

Select the matching library from this folder.

- In this case, replace: **SysLibCallback.lib** (V2) by **SysCallback23.lib** (V3).



The library replacements already saved on the system can be checked and edited via **Tools→Options→Libraries**:



The installation of the »PLC Designer« V3 contains a reference list with predefined standard replacements. The libraries contained in this list are replaced automatically. When a device is added (Controller or fieldbus) to the control configuration, the required libraries are loaded automatically.

Update of a project from V2.x to V3.x

Import of the example project into the »PLC Designer« V3.x

5. Recreate control configuration.



Note!

Libraries

Device-internal and fieldbus-internal libraries are automatically available after the import of the »PLC Designer« V2 project. No manual update of the libraries required!

Control configuration

When a V2 project is imported into V3, the control configuration gets lost. Thus, it has to be recreated in the »PLC Designer« V3.

Logic and Motion devices

- When the system is installed, we recommend to connect the Logic and Motion devices to separate phases!
- Separate bus phases for Logic and Motion devices are not mandatory in the »PLC Designer« V3 (in contrast to the »PLC Designer« V2.3) .

This means the following for this example:

- Insert dummy axes as virtual axes below the available node *L-force Motion General Drive Pool*.
- Insert CAN bus, adapt baud rate
- Insert CANopen manager
- Insert devices
- Adapt device settings (e.g. names, node number, basic configuration, mapping, PDO features).



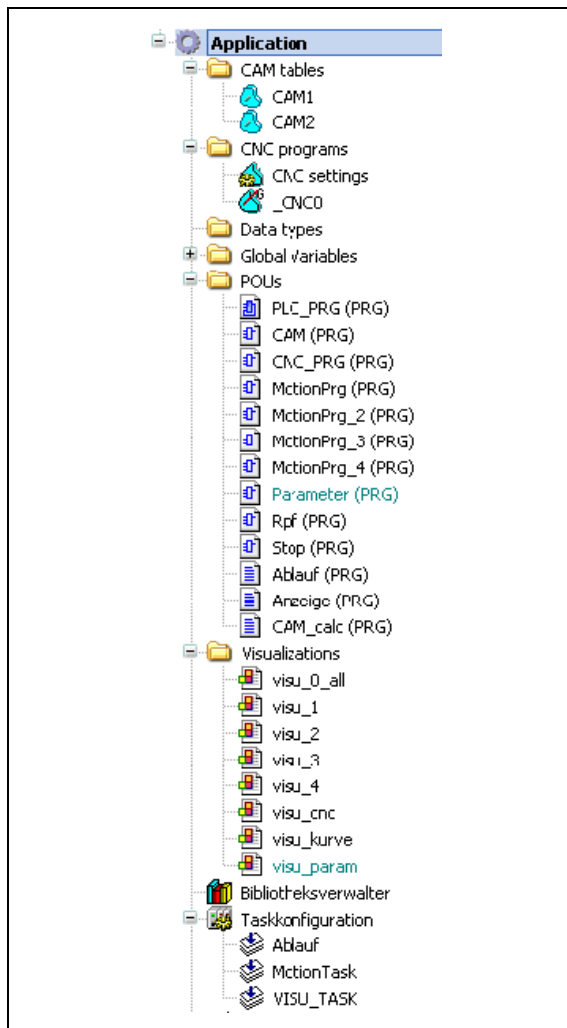
More information on the bus-specific configuration can be found in the following documentation:

- Controller-based Automation EtherCAT communication manual
- Controller-based Automation CANopen communication manual
- Controller-based Automation PROFIBUS communication manual

Update of a project from V2.x to V3.x

Import of the example project into the »PLC Designer« V3.x

The following CAN device configuration arises in the example project:



The following components have been automatically imported and updated in this sample project:

- Cams
- CNC programs
- Program blocks
- »PLC Designer« - internal visualisation
- Task configuration

After the device configuration has been created, the example project has been completely converted from V2 to V3 and can be executed.

Update of a project from V3.0 to V3.x

Updating the target system

8 Update of a project from V3.0 to V3.x

This chapter describes what you have to observe if a PLC project created in the »PLC Designer« V3.0 is to be imported into the »PLC Designer« (with a higher version) V3.x.

8.1 Updating the target system

Requirements

- A firmware of version 3.1 (or higher) must be installed on the L-force Controller.
- To update an existing controller from software version 3.0 to version 3.x, please contact the Lenze support.

8.2 Updating the project environment

- For a full migration of a project created in the »PLC Designer« V3.0, the corresponding device descriptions are required.



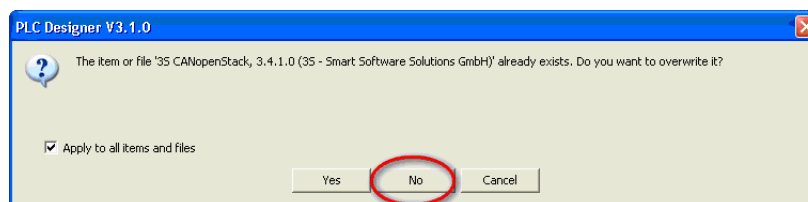
Note!

- Save the project in question prior to migration to the »PLC Designer« V3.0 as **Project Archive** to ensure that the device descriptions which apply to the »PLC Designer« V3.0 project are available during migration.
 - The **Project Archive** includes the PLC program and the device descriptions.
- During the installation of the »PLC Designer«, the setup program will only add new device descriptions. The device descriptions of the previous version which are still available in the system remain unchanged and can be used as before.



How to proceed:

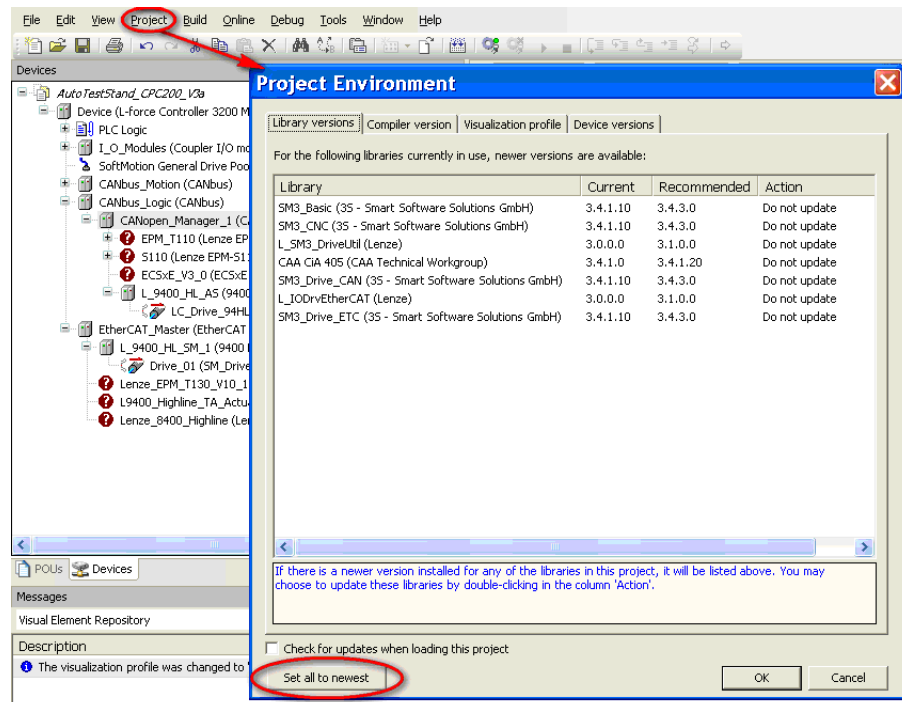
1. Open the desired project in the »PLC Designer« V3.x (**File→Open project**).
 - Do not overwrite any available objects or files:



Update of a project from V3.0 to V3.x

Updating the project environment

2. Upgrade the project environment to the current version (**Project→Project Environment**):



- Update the following devices added to the device tree to the current version:

L-force Controller 3200

CANopen Manager (if available)

EtherCAT Master (if available)

PROFIBUS Master (if available)

All I/O modules connected to the backplane bus

- It is not required to update further devices.

3. Save project (**File→Save project**).
4. Clear project (**Build→Clean all**)
5. Generate project (**Build→Generate code**)
6. Transfer project to the V3.x target system (**Online→Download**)
7. Exit »PLC Designer« (**File→Exit**)
8. Navigate to the directory where the Windows-specific **Documents and Settings** are saved (the drive and name depend on the computer, for example: *C:\Documents and Settings\All Users\Application Data*)
 - Change to the ...**Lenze\PLCDesigner\Devices** subdirectory.
 - Delete the **devicecache.xml** file.
9. Restart »PLC Designer«.

Index

A

Application notes (depiction) [7](#)

AxisGroup [13](#)

C

Comments [17](#)

Conventions used [6](#)

E

E-mail to Lenze [28](#)

F

Feedback to Lenze [28](#)

Function blocks [14](#)

L

L-force Motion [13](#)

M

Macros [17](#)

P

Programming language

AS/SFC [17](#)

AWL/ILO [16](#)

FUP/IL [16](#)

KOP/LD [16](#)

ST [16](#)

R

References used [7](#)

S

Safety [8](#)

Safety instructions (depiction) [7](#)

System events [14](#)

T

Target group [4](#)



Your opinion is important to us!

These instructions have been created to the best of our knowledge and belief to provide you with the best possible support for handling our product.

If you think we have fallen short of this aim and have suggestions for improvements, please e-mail us at:

feedback-docu@lenze.de

Thank you for your support.

Your Lenze documentation team



© 02/2013



Lenze Automation GmbH
Hans-Lenze-Str. 1
D-31855 Aerzen
Germany



+49 (0)5154 – 82 -0



+49 (0)5154 – 82 - 2800



Lenze@Lenze.de



www.Lenze.com

Service Lenze Service GmbH
Breslauer Straße 3
D-32699 Extertal
Germany



00 80 00 / 24 4 68 77 (24 h helpline)



+49 (0)51 54 / 82-11 12



Service@Lenze.de

SHPPLCDMIG ■ 13427067 ■ EN 1.2 ■ TD11

10 9 8 7 6 5 4 3 2 1