HOW-TO CONVERT FROM VER 22 TIL VER 23

What needs to happen in order to continue working with old projects in the new version.

Contents

Background

In some areas, version 23 is fundamentally different from version 22, and that means that you must *install* this version instead of *update* to the version, like you usually do.

The most important changes from version 22 to version 23

From ver23 you can download components to your own database on a component portal. To be able to use data from the portal in its own database, ver23 sets requirements for the structure of this. We have therefore created a whole new database structure that supports this. It also means that if you want to continue working with your own 'old' components, you must convert data. How this is done is reviewed in the following first section.

We have renamed all standard symbols and the new symbol names are used on the portal. However, it also means that in order to work with existing components, you must be able to get hold of old symbols. The next section reviews how to get hold of (selected) user settings from ver22.

Finally, we have expanded our plc features in ver23. New symbols have been added, new symbol data fields, which support some long-awaited functions. The new symbol data fields are created in the program so that they function as 'real' data fields on the plc symbols, and this requires a new installation. You will get a brief overview of these features at the end of this document.

You need to keep your old installation

These three things mean that you cannot 'just' update your current installation but must make a somewhat more comprehensive conversion if you want to work further with existing projects.

Therefore, we also recommend that you keep your old ver22 installation, and only transfer the projects you want to work on. And you do it as the need arises.

New database structure

The database contains many new fields, and all these fields MUST be there, ie you cannot delete uninteresting fields as before, but you can hide selected fields. Be aware that all ‘our’ fields are meant for something specific; there are hints in all fields with info on what they are intended for.

You can create your own fields as before. For example, there are several fields that keep track of the connection to the Portal and other fields with information about old item numbers.

There are double fields on symbols - pcs and usr, so you can have both our version of symbol selection etc. and your own on the same component. And importantly, all pin files are gone, as there is now room for data in the database itself.

Open the Component database, select and open a component and see it.

Import components from ver22

Components are imported from the old database. To be able to convert, you need to know both the name of the database and its location. The latter because we extract all pin files and at the same time retrieve any data in the Panelbuilder database, so that in future you only have one database, and because we use your old setup to make the most correct conversion possible.

Clean up manufacturers

Manufacturers will in future be included in the ID of components, and therefore they must be spelled correctly! When you import components from ver22, you also go through all manufacturer names.

Import of own menu table

Our default menu is similar to the one we use in ver22. If you have made your own menu table, you can also import this. Please note that components on the portal do not know your own menu table and therefore cannot put the components in the correct folder if you use it.

Database settings

To get the full benefit of the program, it is a good idea to keep track of how the two programs Automation and the Database talk to each other. This can be seen under Database settings. Try to open both and see the difference.

The important difference between ver22 and ver23 Component’s article no

In ver22, the link between the project components and the database is not unambiguous: The component's item number is retrieved (as a starting point) in the database's EANnr field. However, you can also freely choose another field to contain another item number, most often a ‘Navision number’, and finally you can choose an alternative field, which is automatically used in the event that the EAN number is empty. These options mean that it is not possible to uniquely identify a component across different projects, but it works fine in the individual project.

In ver23 it is different, as each component is linked directly to a componentID in the database. In the project, you can then choose to display another number, which can be selected from the database's standard fields and possibly your own fields, eg a ‘Navision number’.

In the individual project, you choose which item number to use and a prioritization order, if necessary.

When we convert a file from ver22 to ver23, we automatically set the project to retrieve item no. from the database's HistoricalArticleNo field, in order to display the same value as before. However, keep in mind that if the components of the file do not already exist in the database, a correct link to the database cannot be created.

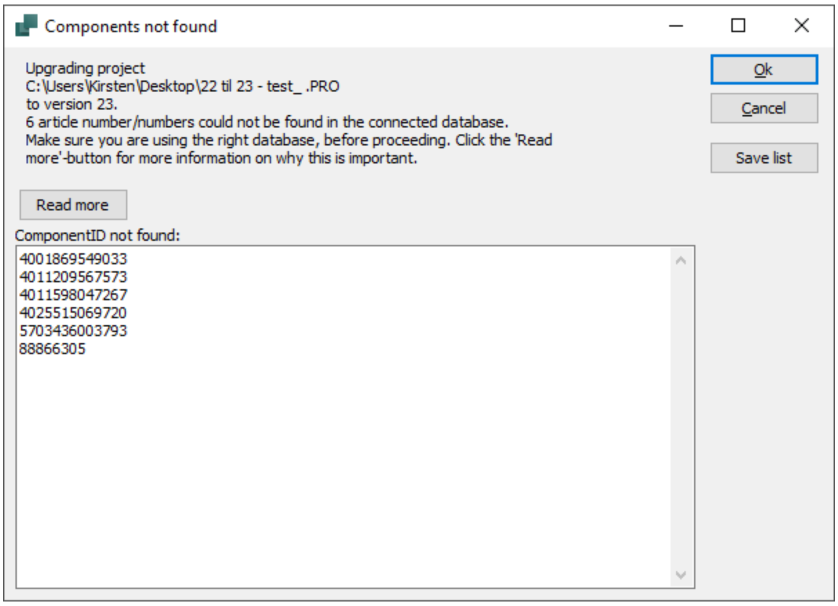
How-to: Conversion of a project from ver22 to ver23

When you first load a ver22 project, you must consider all questions. Hence this short guide, with a few tips:

- Know the setup of your current database - it helps in understanding later
- The simplest way to get through all the questions is by finding (or making) a small project with quite a few components in it, and using this project to get through the routine

Open the project in ver23

- The program/ database cannot find all the components in the associated database, so you save a list of these components
- Now press Cancel and close the project.
- Please click on the 'Read more' button



Components not found

Upgrading project
C:\Users\Kirsten\Desktop\22 til 23 - test_ .PRO
to version 23.
6 article number/numbers could not be found in the connected database.
Make sure you are using the right database, before proceeding. Click the 'Read more'-button for more information on why this is important.

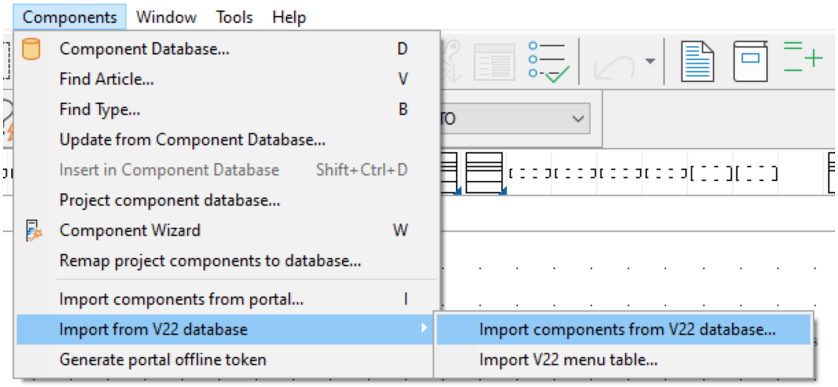
ComponentID not found:

```
4001869549033
4011209567573
4011598047267
4025515069720
5703436003793
88866305
```

Import components from the old database

In ver23, in the Components menu, select Import from V22 database.

The first time you do this, you will be guided through a whole lot of questions, and it is these questions that we answer here.

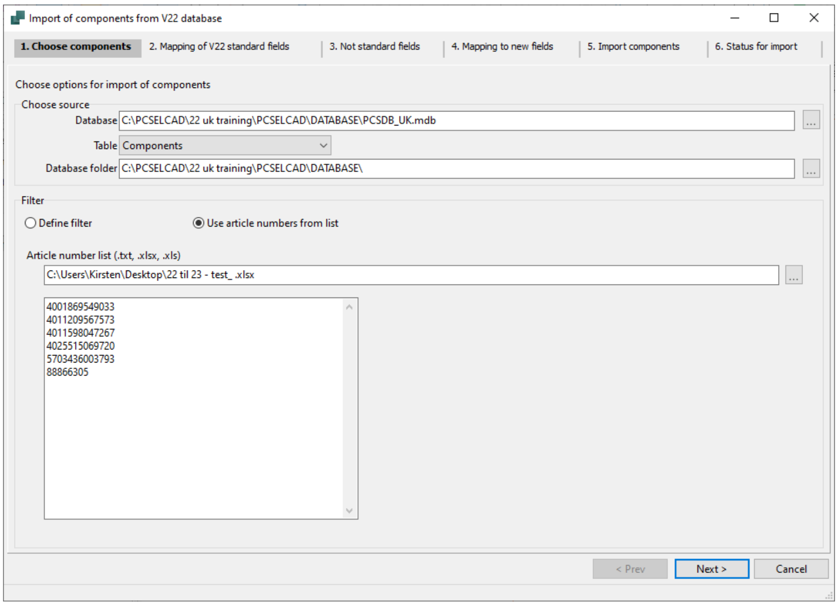


Components
Window
Tools
Help
Component Database... D
Find Article... V
Find Type... B
Update from Component Database...
Insert in Component Database Shift+Ctrl+D
Project component database...
Component Wizard W
Remap project components to database...
Import components from portal... I
Import from V22 database
Generate portal offline token
Import components from V22 database...
Import V22 menu table...

In short, select your old database, which can be an Access database (* .mdb or * accdb) or an SQL database (* .udl).

You select the database, the component table and finally the folder that contains pin and plc files, idb file and the Panelbuilder database. Most often, all the files are in the same folder.

The import tool reads your old idb file, which contains all your old settings. And these then need to be converted, along with the components.



Import of components from V22 database

1. Choose components | 2. Mapping of V22 standard fields | 3. Not standard fields | 4. Mapping to new fields | 5. Import components | 6. Status for import

Choose options for import of components

Choose source

Database: C:\PCSELCAD\22 uk training\PCSELCAD\DATABASE\PCSDB_UK.mdb

Table: Components

Database folder: C:\PCSELCAD\22 uk training\PCSELCAD\DATABASE\

Filter

○ Define filter ● Use article numbers from list

Article number list (.txt, .xlsx, .xls): C:\Users\Kirsten\Desktop\22 til 23 - test_ .xlsx

4001869549033
4011209567573
4011598047267
4025515069720
5703436003793
88866305

Some good advice

When you select only a few components, either using the list or another filter, you get through the various choices faster, and if you find errors, you have not wasted a lot of time. If your ver22 database is in 'good shape', you can easily import all your old components, but if this is not the case, we strongly advise that you do the import in smaller bits.

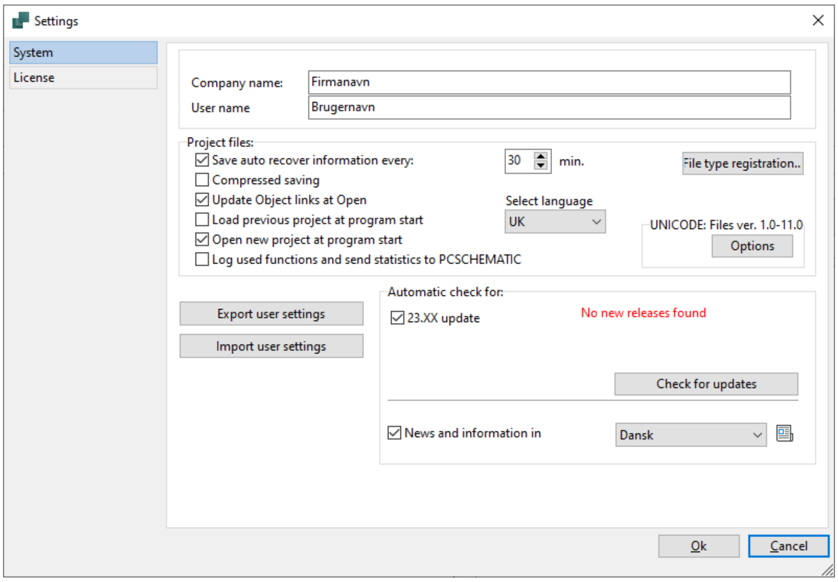
Old settings

When you update, as you have done the last many times, you get to continue your user settings for better or worse.

It also means that many of the changes in the program that new users see when they open the program remain 'secret' to old users.

Old settings are many things, and some of these settings you would like to include in the new version.

On the Settings tab | System in ver22 you can Export user settings, and in ver23 you can both import and export.



Settings

System
License

Company name: Firmanavn
User name Brugernavn

Project files:
☑ Save auto recover information every: 30 min.
☐ Compressed saving
☑ Update Object links at Open
☐ Load previous project at program start
☑ Open new project at program start
☐ Log used functions and send statistics to PCSCHEMATIC

Select language UK

UNICODE: Files ver. 1.0-11.0

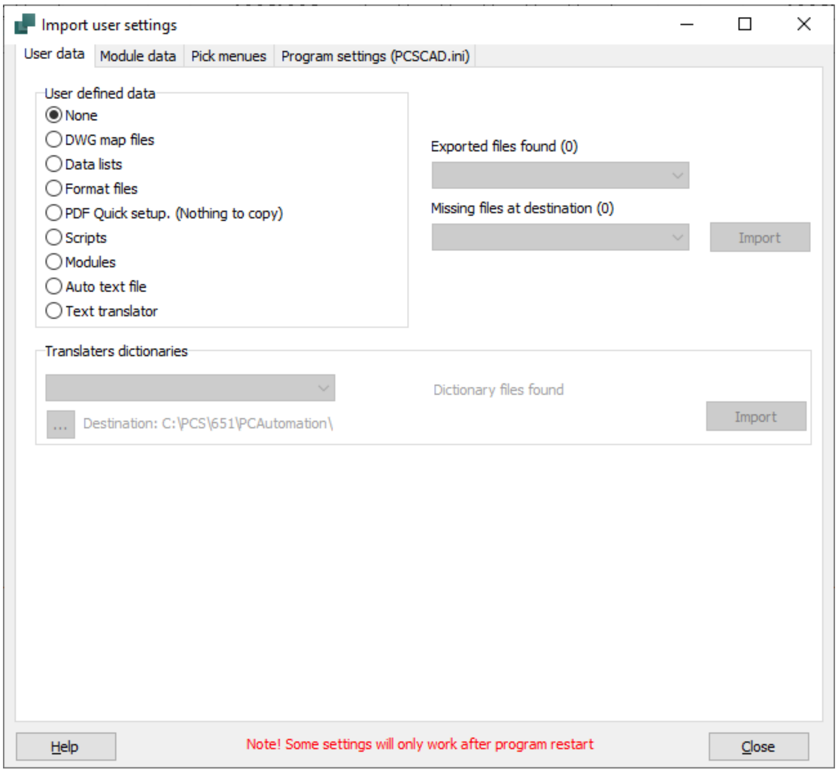
Automatic check for:
☑ 23.XX update
No new releases found

☑ News and information in Dansk

You will need to use this feature in the future if you want to copy settings to colleagues, and therefore do NOT copy the ini file - it does not get everything anyway.

When importing user settings, you enter this dialog, where you are guided through the various settings in the program.

You can tap the Help button and get a small document explaining the various settings.



Import user settings

User defined data
● None
○ DWG map files
○ Data lists
○ Format files
○ PDF Quick setup. (Nothing to copy)
○ Scripts
○ Modules
○ Auto text file
○ Text translator

Exported files found (0)

Missing files at destination (0)

Translaters dictionaries

Dictionary files found

Destination: C:\PCS\651\PCAutomation\

Note! Some settings will only work after program restart

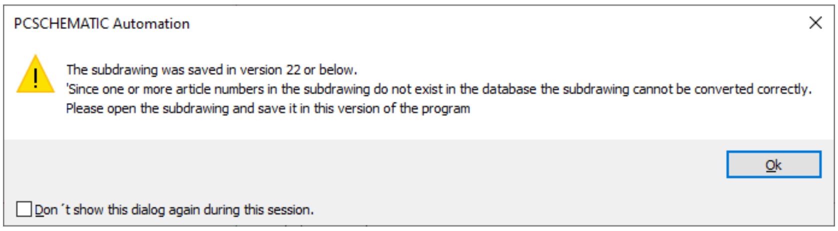
Subdrawings and templates

When we convert a file, we reconnect to the database, which also means that you can not use it in 'old' installations.

To continue working with both version 22 and version 23, make a copy of the templates and sub-drawings you will use in the future.

Templates and sub-drawings are also files that we change in connection with the database, and therefore you need to convert each of these separately.

As you get your components into the new database, the warning will appear less frequently.



The subdrawing was saved in version 22 or below.
'Since one or more article numbers in the subdrawing do not exist in the database the subdrawing cannot be converted correctly.
Please open the subdrawing and save it in this version of the program

□ Don ´t show this dialog again during this session.

We have created new PLC symbols and several PLC functions in ver23

And maybe it could be an idea to consider going over to these.

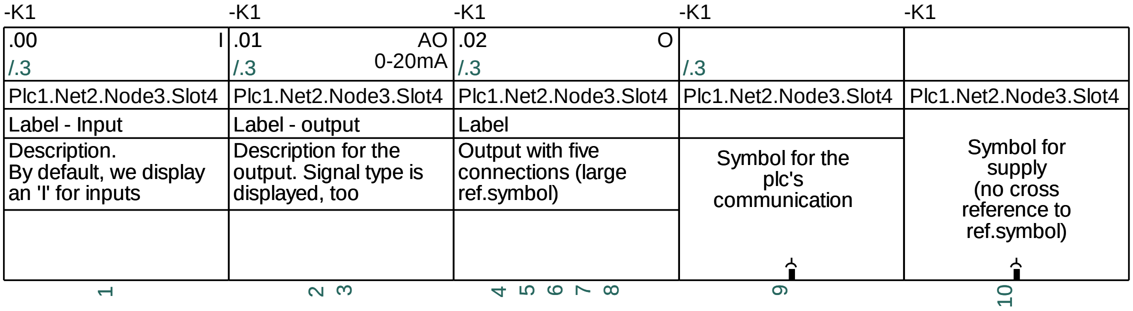
All plc components on the portal use these symbols.

Among the advantages is that all components use the same symbol when it comes to the same function. In addition, we have expanded the Component Guide's functionality, so that you can create PLCs in a simple way, and you can edit (some) existing PLCs.

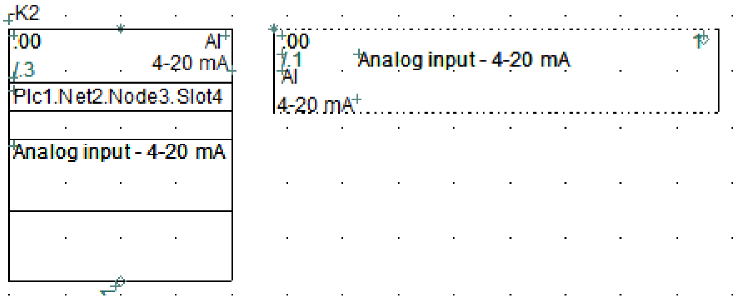
Finally, the symbols contain other functions that have been in demand for a long time, such as PlcID (plc.net.node.slot) and signal type, both of which are displayed on the symbols.

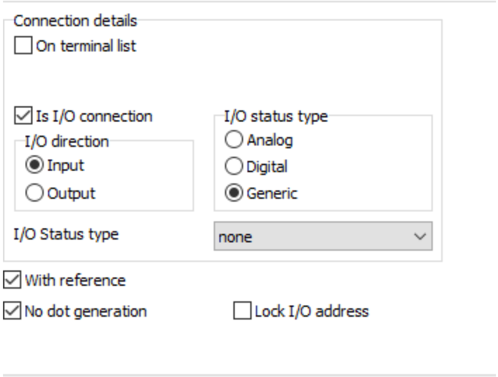
New plc symbols

The IO symbols themselves and the other symbols in the 'field' have been given a uniform appearance:


-K1
.00
I
/.3
Plc1.Net2.Node3.Slot4
Label - Input
Description.
By default, we display an 'I' for inputs
1
-K1
.01
AO
0-20mA
/.3
Plc1.Net2.Node3.Slot4
Label - output
Description for the output. Signal type is displayed, too
2
3
-K1
.02
O
/.3
Plc1.Net2.Node3.Slot4
Label
Output with five connections (large ref.symbol)
4
5
6
7
8
-K1
/.3
Plc1.Net2.Node3.Slot4
Symbol for the plc's communication
9
-K1
Plc1.Net2.Node3.Slot4
Symbol for supply (no cross reference to ref.symbol)
10

- The width is 1 current path (40 mm)
- All connections point downwards
- New symbol data field – PlcID (composed of Plc.Net.Node.Slot) – on all symbols
- Space for label and description
- Space for description on connections
- All texts in Arial, 2.5 mm




Connection details

☐ On terminal list

☑ Is I/O connection

I/O direction

● Input

○ Output

I/O status type

○ Analog

○ Digital

● Generic

I/O Status type: none

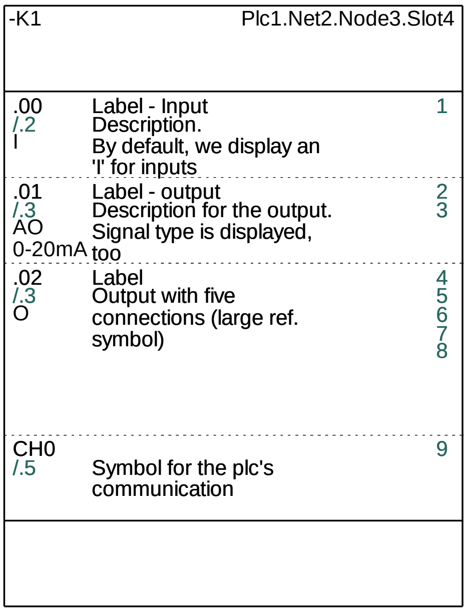
☑ With reference

☑ No dot generation

☐ Lock I/O address

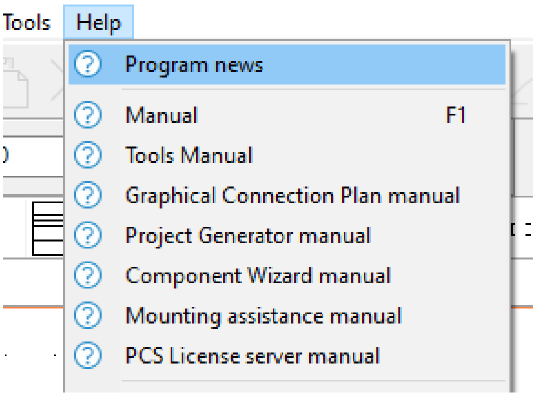
The reference symbols have also changed appearance:

- Width is 2 current paths (80 mm), meaning you can have 4 columns on a page
- 15/30 mm high per address
- Common top, common bottom which is automatically grouped together with the symbols per address
- All texts in Arial, 2.5 mm



You can read more ...

You can read much more in the Program News document for ver23. Find it in the Help menu.


Tools
Help
Program news
Manual F1
Tools Manual
Graphical Connection Plan manual
Project Generator manual
Component Wizard manual
Mounting assistance manual
PCS License server manual