Release Notes V4.5

08 04 24 cs

Table of Contents

Upgrading to Proview V4.5.0....5
New functions....5
Mysql development database....5
ClassVolume stored in database....5
ConversionOn in ChanDi, ChanIi and ChanAi handled by Profibus....5
Modbus TCP I/O....5
Additional text for alarm messages....5
Tooltip in plc trace....5
Documentation....6
NMpsCell lock....6
Local translationtables....6
General Frequency Converter components for Profibus PPO 5 and PPO 3....6
Draw layout of plc functionobjects in Ge....7
Ge dynamic type DigCommand....7
Configuration file for BerkeleyDb....7
Runtime backup format....7
New Classes....7
BaseComponent:BaseFcPPO3....7
BaseComponent:BaseFcPPO3Fo....7
BaseComponent:BaseFcPPO3Sim....8
BaseComponent:BaseFcPPO3PbModule....8
BaseComponent:BaseFcPPO3MotorAggr....8
BaseComponent:BaseFcPPO3MotorAggrFo....8
BaseComponent:BaseFcPPO3MotorAggrSim....8
BaseComponent:BaseFcPPO3MotorAggr....8
BaseComponent:BaseFcPPO3FanAggr....8
BaseComponent:BaseFcPPO3PumpAggr....8
BaseComponent:BaseFcPPO5....8
BaseComponent:BaseFcPPO5Fo....8
BaseComponent:BaseFcPPO5Sim....8
BaseComponent:BaseFcPPO5PbModule....8
BaseComponent:BaseFcPPO5MotorAggr....8
BaseComponent:BaseFcPPO5MotorAggrFo....9
BaseComponent:BaseFcPPO5MotorAggrSim....9
BaseComponent:BaseFcPPO5MotorAggr....9
BaseComponent:BaseFcPPO5FanAggr....9
BaseComponent:BaseFcPPO5PumpAggr....9
OtherIo:Modbus_TCP_Slave....9
OtherIo:Modbus_Module....9
ABB:ABB_Converter_ACS800_PPO3....9
ABB:ABB_Converter_ACS800_PPO5....9
OtherManufacturer:Danfoss_Converter_FC300_PPO3....9
OtherManufacturer:Danfoss_Converter_FC300_PPO5....9
Additions in V4.5.0-5....9
Xtt FileView....9
New Xtt commands....11

open fileview....11
write object....11
read object....11
New Objects....11
NMps:DataFWrite....11
NMps:DataFRead....12
Upgrade procedure 12
Make a copy of the project....12
upgrade.sh....12
dumpdb....12
classvolumes....12
renamedb....13
dirvolume....13
loaddb....13
compile....13
createload....13
createboot....13

Upgrading to Proview V4.5.0

This document describes new functions i Proview V4.5.0, and how to upgrade a project from V4.4.0 to V4.5.0.

New functions

Mysql development database

As a complement to BerkeleyDb the development database now also can be stored as a mysql database.

The database choice is configured in the RootVolumeConfig, SubVolumeConfig, SharedVolumeConfig or ClassVolumeConfig object in the directory volume. Here you also specify the mysql server.

ClassVolume stored in database

A classvolume is normally stored in a text file (.wb_load). Now it is possible to store the volume in a database instead. The storage type (Wbl, BerkeleyDb or Mysql) is configure in the ClassVolumeConfig object in the directory volume.

ConversionOn in ChanDi, Chanli and ChanAi handled by Profibus

Reading of Profibus Di, Ii and Ai can be disabled by setting ConversionOn in the channel object to 0.

The attribute ConversionOn in ChanDi, ChanIi and ChanAi was previously not used by the profibus driver. Now the received value is not converted and moved to the signal if ConversionOn is zero.

NOTE !!!

The default value for ConversionOn was previously 0 for some channel objects. Use for example the spreadsheet editor to check ConversionOn for all ChanDi, ChanIi and ChanAi.

Modbus TCP I/O

Support for Modbus TCP I/O is added in V4.5. Configuration objects are found in the OtherIO volume: Modbus_TCP_Slave and Modbus_Module.

Additional text for alarm messages

The attribute MoreText is added to DSup and ASup objects. MoreText is viewed as tooltip for an alarm or event in the alarm and eventlists. The text has a maximum size of 255 characters, and can have 10 rows.

Tooltip in plc trace

Tooltip are used in plc trace to view the description of referenced signals in GetXx, StoXx and similar objects.

For function object code containing $PlcMain and $PlcFo symbols, also the signal name and channel name are displayed.

Documentation

Guide to I/O Systems, is a new guide describing the I/O systems that can be used with Proview, and how to connect other I/O systems to Proview.

A chapter, Class Editor, added to Designer's Guide. The chapter describes how to build classes and function objects in the class editor.

NMpsCell lock

Lock added to handling of NMps cells in applications, using the nmpsappl_Mirror interface. It is no longer necessary to run the plc thread handling the cells, and the application on the same priority. Note that an application on low priority can delay a plcthread on high priority.

Local translationtables

Project local translationtables to translate alarm messages and texts in Ge graphs can be placed on $pwrp_exe. rt_xtt will translate an english text that is found in the en_us table, to the corresponding text in the current language.

Syntax

The first letter on a row can be E or B. If E, only an exact match of the string will be translated. For B the beginning of the string should be equal, and only the beginning

The three numbers separated by points, are the key to identify corresponding texts of different languages. In the first number, 0-99999 is reserved.

Example

English table $pwrp_exe/lng_en_us.dat

```
# Eurotherm TC3001
E 100000.0.1 "ProcessValue"
E 100000.0.2 "LineVoltage"
E 100000.0.3 "LoadVoltage"
B 100000.0.21 "Engineering unit violation, "
E 100000.0.31 "Electrical cabinet"
```

Swedish table $pwrp_exe/lng_sv_se.dat

```
# Eurotherm TC3001
E 100000.0.1 "ÄrVärde"
E 100000.0.2 "LinjeSpänning"
E 100000.0.3 "LaddSpänning"
B 100000.0.21 "Engineering unit violation, "
E 100000.0.31 "Elskåp"
```

General Frequency Converter components for Profibus PPO 5 and PPO 3

There are two new frequencyconverter objects, BaseFcPPO5 and BaseFcPPO3, that can handle all frequency converters that are able to communicate on Profibus with PPO message type 5 and 3. The

classes can be subclassed to handle a converter of a specific type. Subclasses for Danfoss FC300 and ACS800 are included : Danfoss_Converter_FC300_PPO5, Danfoss_Converter_FC300_PPO3, ABB_Converter_ACS800_PPO5 and ABB_Converter_ACS800_PPO3.

Aggregates with the new converter objects for motors, fans and pumps are also added: BaseFcPPO5MotorAggr, BaseFcPPO5FanAggr, BaseFcPPO5PumpAggr, BaseFcPPO3MotorAggr, BaseFcPPO3FanAggr and BaseFcPPO3PumpAggr. The frequencyconverter object in the aggregate can be casted to get the properties of a specficic subclass.

Draw layout of plc functionobjects in Ge

Now it is possible to draw the layout of a plc function object in Ge. The drawing can only have the colors black, gray and red. It should be saved with the same name as the function object class, but with lower case. A .flwn file is created on $pwrp_exe when File/Export/PlcFo is activated in the menu.

In the class description of the function object, Graphmethod in GraphPlcNode should be set to 12.

Ge dynamic type DigCommand

A new dynamic type in Ge executes an xtt-command when the value of a digital signal changes from 0 to 1. This can be used for example to open a graph when some event occurs in the process.

Configuration file for BerkeleyDb

Large development databases (>~250000 objects) needs larger qoutas than the default values. These can be set in a configuration file, $pwrp_db/'volumename'.db.cnf, where lk_max_locks and lk_max_objects con be configured. The default values of lk_max_locks and lk_max_objects are 50000 and 20000.

Example

```
#
# Configuration file for BerkeleyDb, $pwrp_db/myvolume.db.cnf
#
lk_max_locks 100000
lk_max_objects 50000
```

Runtime backup format

The format of the runtime backup is changed. Previously attributes and objects where stored with the attributereference as key, which caused problems when classes where changed. Now each attribute are stored separatley with object identity and attribute name.

New Classes

BaseComponent:BaseFcPPO3

Frequency converter with Profibus PPO3 protocol.

BaseComponent:BaseFcPPO3Fo

Function object to BaseFcPPO3.

BaseComponent:BaseFcPPO3Sim

Function object to simulate BaseFcPPO3.

BaseComponent:BaseFcPPO3PbModule

Profibus module for BaseFcPPO3.

BaseComponent:BaseFcPPO3MotorAggr

Motor aggregate with BaseFcPPO3, circuit breaker, contactor safetyswitch etc.

BaseComponent:BaseFcPPO3MotorAggrFo

Function object to BaseFcPPO3MotorAggr.

BaseComponent:BaseFcPPO3MotorAggrSim

Function object to simulate BaseFcPPO3MotorAggr.

BaseComponent:BaseFcPPO3MotorAggr

Motor aggregate with BaseFcPPO3, circuit breaker, contactor safetyswitch etc.

BaseComponent:BaseFcPPO3FanAggr

Fan aggregate base on BaseFcPPO3MotorAggr.

BaseComponent:BaseFcPPO3PumpAggr

Pump aggregate based on BaseFcPPO3MotorAggr.

BaseComponent:BaseFcPPO5

Frequency converter with Profibus PPO5 protocol.

BaseComponent:BaseFcPPO5Fo

Function object to BaseFcPPO5.

BaseComponent:BaseFcPPO5Sim

Function object to simulate BaseFcPPO5.

BaseComponent:BaseFcPPO5PbModule

Profibus module for BaseFcPPO5.

BaseComponent:BaseFcPPO5MotorAggr

Motor aggregate with BaseFcPPO5, circuit breaker, contactor safetyswitch etc.

BaseComponent:BaseFcPPO5MotorAggrFo

Function object to BaseFcPPO5MotorAggr.

BaseComponent:BaseFcPPO5MotorAggrSim

Function object to simulate BaseFcPPO5MotorAggr.

BaseComponent:BaseFcPPO5MotorAggr

Motor aggregate with BaseFcPPO5, circuit breaker, contactor safetyswitch etc.

BaseComponent:BaseFcPPO5FanAggr

Fan aggregate base on BaseFcPPO5MotorAggr.

BaseComponent:BaseFcPPO5PumpAggr

Pump aggregate based on BaseFcPPO5MotorAggr.

OtherIo:Modbus_TCP_Slave

I/O object to configure a Modbus TCP slave.

OtherIo:Modbus_Module

I/O object to configure a Modbus TCP module.

ABB:ABB_Converter_ACS800_PPO3

Frequency converter ABB ACS800 with Profibus PPO3 protocol.

ABB:ABB_Converter_ACS800_PPO5

Frequency converter ABB ACS800 with Profibus PPO5 protocol.

OtherManufacturer:Danfoss_Converter_FC300_PPO3

Frequency converter Danfoss FC300 with Profibus PPO3 protocol.

OtherManufacturer:Danfoss_Converter_FC300_PPO5

Frequency converter Danfoss FC300 with Profibus PPO5 protocol.

Additions in V4.5.0-5

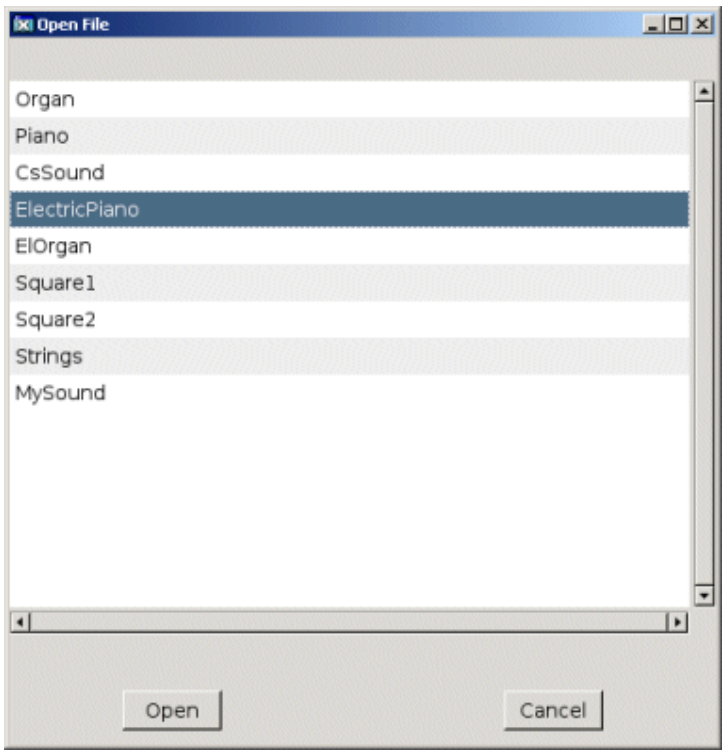
Xtt FileView

A fileview is opened from from Xtt with the command

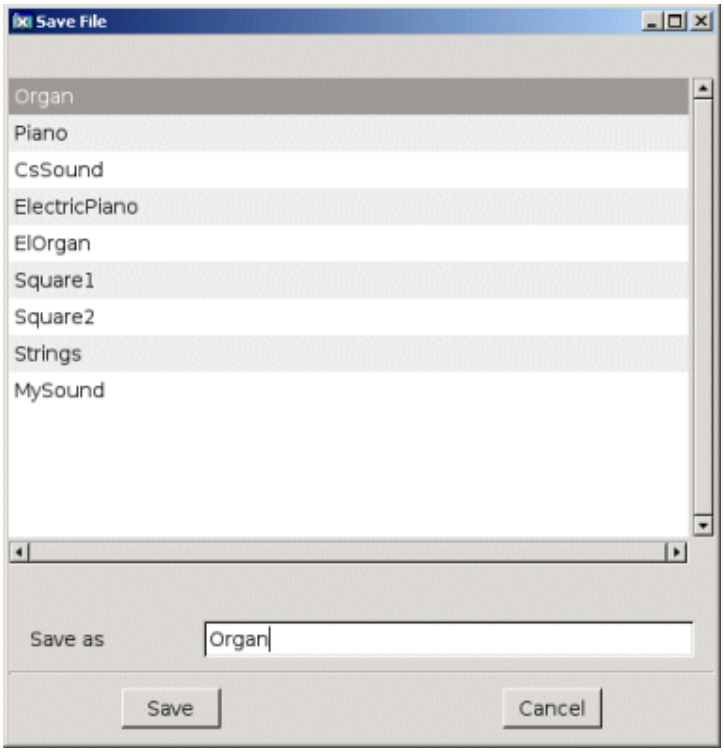
xtt> open fileview /file= /target= /trigger= /type=

- /file is a directory and a file pattern with wildcard that describes the file that is to be viewed int the fileview.
- /target is a string attribute in which the name of the selected file is stored.
- /trigger is a boolean attribute that it set to indicate that a save or opening should be performed.
- /type can be 'save' or 'open', where in the save mode an 'Save as' input entry is added.

The acual opening or save action can be programmed in the plc by DataFRead or DataFWrite objects, see New objects below.



Fileview in Open mode



Fileview in Save mode

New Xtt commands

open fileview

See Xtt FileView above.

write object

Write a data object to file. Writes all attribute with attribute name and value to text file.

xtt> write object /object= /file=

read object

Read a data object from file. Read a file written with the 'write object' function.

xtt> read object /object= /file=

New Objects

NMps:DataFWrite

Plc function object to write a data object to file.

NMps:DataFRead

Plc function object to read a data object from file.

Upgrade procedure

The upgrading has to be done from any version in the interval V4.2.0 – V4.4.4. If the project has a lower version, the upgrade has to be performed stepwise following the schema

V2.1 -> V2.7b -> V3.3 -> V3.4b -> V4.0.0 -> V4.1.3 ->V4.2.0->V4.5.0

The upgrade procedure is to change the version of the project in the projectlist, and then execute the script upgrade.sh.

NOTE !!

Do not activate Update Classes.

If the previous version should be kept, first make a copy of the project.

Make a copy of the project

Do `sdf` to the project and start the administrator

```
> pwra
```

Now the Projectlist is opened. Enter edit mode, login as administrator if you lack access. Find the current project and select Copy Project from the popup menu of the ProjectReg object. Open the copy and assign a suitable project name and path. Change the version to V4.5.0. Save and close the administrator.

upgrade.sh

Do `sdf` to the project.

upgrade.sh is a script that is divided into a number of passes. After each pass you you have to answere

whether to continue with the next pass or not.
Start the script with

```
> upgrade.sh
```

and go through all the passes.

dumpdb

Creates a dump file for each volume in the project. The name of the dumpfile is

```
$pwrp_db/'volumename'.wb_dmp
```

NB ! The dump has to be done with V4.4 dump program

classvolumes

Create loadfiles and structfiles for the class volumes.

renamedb

Store the old databases under the name `$pwrp_db/'volumename'.db.1.`

dirvolume

Create a directory database and load the dumpfile for the project volume into the database.

loaddb

Create databases and load the dumpfiles into them.

compile

Compile all the plc programs.

createload

Create loadfiles for the root volumes.

createboot

Create bootfiles for all nodes in the project.

If the project contains any application programs, these has to be built manually.

Delete files from the upgrading procedure:

`$pwrp_db/*.wb_dmp.*`
`$pwrp_db/*.db.1` (old databases, directories which content also should be removed)