



Release Notes V5.2

2014 08 29

Copyright SSAB Oxelösund AB 2014

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

Table of Contents

Upgrading to Proview V5.2.0.....	4
New functions.....	4
Inc3p MaxWidup.....	4
PID controller.....	4
Bumblers Manual/Auto switching for P and PD controllers	4
Limited windup for parts of output.....	4
MinWidup and MaxWidup.....	4
Event selection on type.....	4
Core plc program.....	6
Plc trace, not updated objects are dimmed.....	6
Build configuration.....	6
BuildDirectory.....	6
BuildCopy.....	6
BuildMake.....	7
BuildExec.....	7
Export.....	7
Import.....	7
Build and distribution of configuration files.....	7
Plc help texts.....	8
New time plc objects.....	11
New data reference plc objects.....	11
New string plc objects.....	12
New Classes.....	12
BuildConfig.....	12
Modified Classes.....	12
Upgrade procedure	12

Upgrading to Proview V5.2.0

This document describes new functions in Proview V5.2.0, and how to upgrade a project from V5.1.0 to V5.2.0.

New functions

Inc3p MaxWindup

Inc3p has a new attribute, MaxWindup, to avoid infinite accumulation at unsuitable parameters. If $\text{MaxWindup} > 0$ the time to action (Acc) is limited to $\pm \text{MaxWindup}$.

PID controller

Bumpless Manual/Auto switching for P and PD controllers

The new attribute PDAbsFlag makes bumpless switching from manual to auto possible also for P and PD controllers. Setting PDAbsFlag to zero will give the new function where and offset is calculated in manual mode that is added to the output when the controller is switched to auto.

Setting PDAbsFlag to 1 will give the old function.

Limited windup for parts of output

The new attribute WindupMask specifies which parts of the output has limited windup. WindupMask can be set to I, BI, PBI or BPID. BPID will give the old function.

With I or BI, windup for P disturbances are eliminated. This was previously achieved by setting MaxOut to 110%.

MinWindup and MaxWindup

The new attributes MinWindup and MaxWindup can be used for example with servo valve control with leakage compensation and limited I part.

Event selection on type

Previously the event selection in operator places has been done only on position in the object tree. Now the EventType has been extended with six new types: MaintenanceAlarm, SystemAlarm, UserAlarm1, UserAlarm2, UserAlarm3 and UserAlarm4, and the event selection can also be made on these types. With the new attribute EventSelectType the types of event that should be displayed is specified. The EventSelectType is present in the OpPlace, AlarmCategory and WebHandler objects.

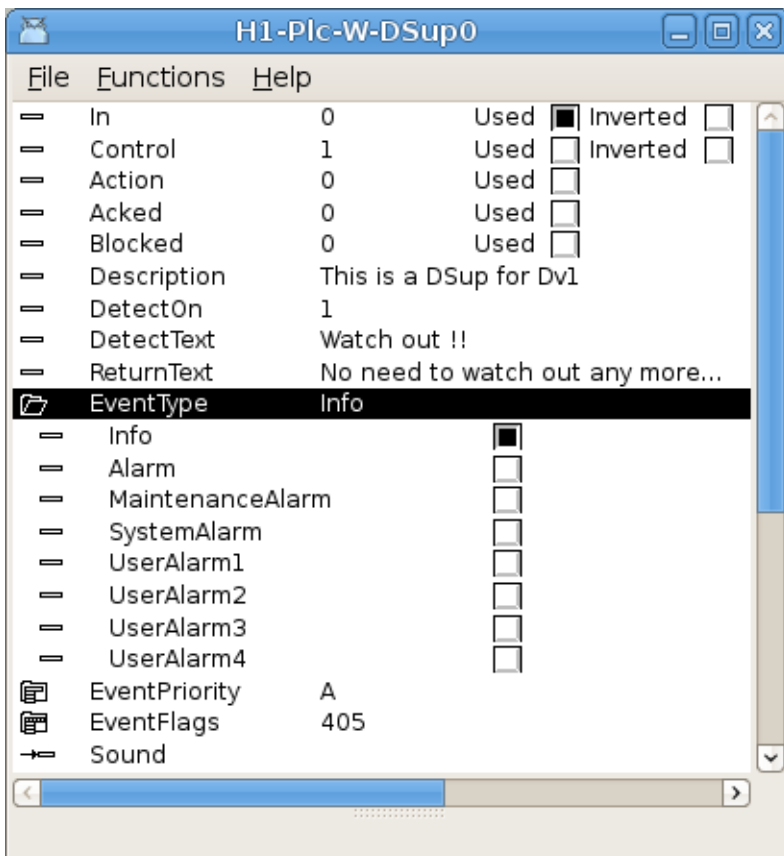


Fig Available event types in a DSup object

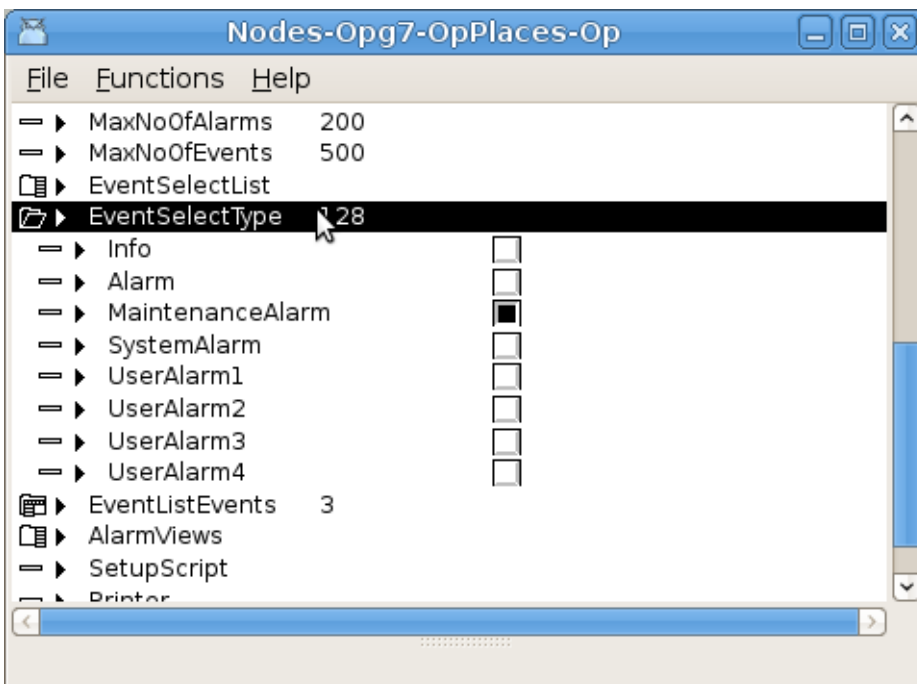


Fig Displayed event types in an OpPlace object

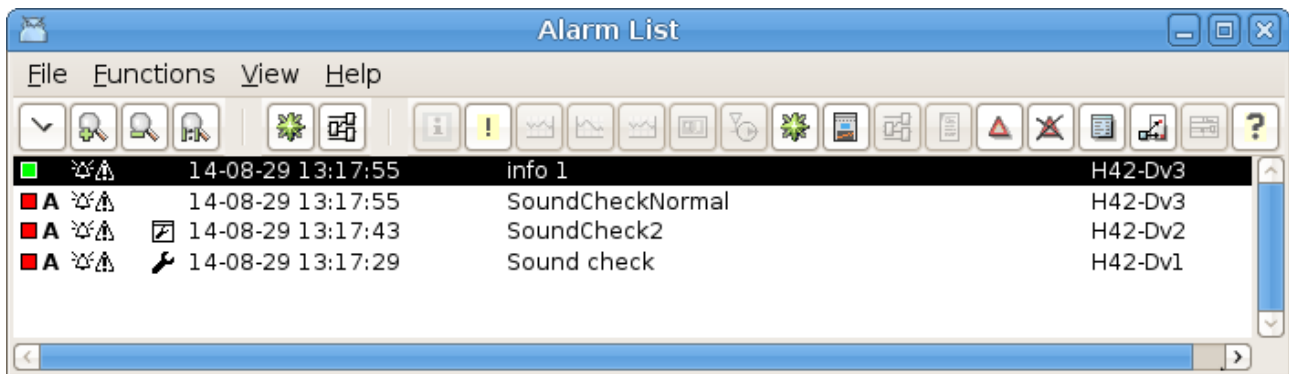


Fig System and maintenance types are marked with a wrench

Core plc program

Operator stations often doesn't need a plc program, but to get all functionality it has been adviced to create a dummy plc. This is no longer necessary as the core plc program wil be started if there is no PlcPgm's configured. The core plc program should be configured with a PlcProcess and a PlcThread object.

One advantage with the core plc program is that operator stations can easier be configured for other platforms than the development station. The dummy plc program doesn't need to be built one the development station any more.

Plc trace, not updated objects are dimmed

Plc trace are now checking the SubscriptionOldness property and function objects with old subscriptions are dimmed.

Build configuration

The build configuration makes it possible to handle

- building of applications, with execution of makefiles and copying of include files.
- copying of graph files.
- copying of configuration files from \$pwrp_cnf.
- importing files from other projects.
- exporting files to other projects.

The build is configured in the directory volume under a BuildConfig object, See Fig 1.

Other objects in the build configuration are

BuildDirectory

Configures how a directory, eg \$pwrp_appl or \$pwrp_pop is built. The actions executed when building the directory is specified with BuildCopy, BuildMake and BuildExecute object.

BuildCopy

Copies a file, or a number of files specified with wildcard from the source directory to the build tree.

BuildMake

Executes a make file.

BuildExec

Executes a shell command.

Export

Configures files that should be exported to other projects. Normally files are exported to a common directory from where they are imported by other projects. Specific files to export is configured with ApplExport objects.

Import

Configures files that is imported from other projects. Specific files are imported with ApplImport objects.

The build, export and import can be performed automatically when building a node, by configuring BuildBeforeNode or BuildAfterNode in the Option attribute of the BuildDirectory, Import and Export objects.

To perform an selective build, export or import, the Build Directories, Import and Export window is opened from the Functions menu in the configurator.

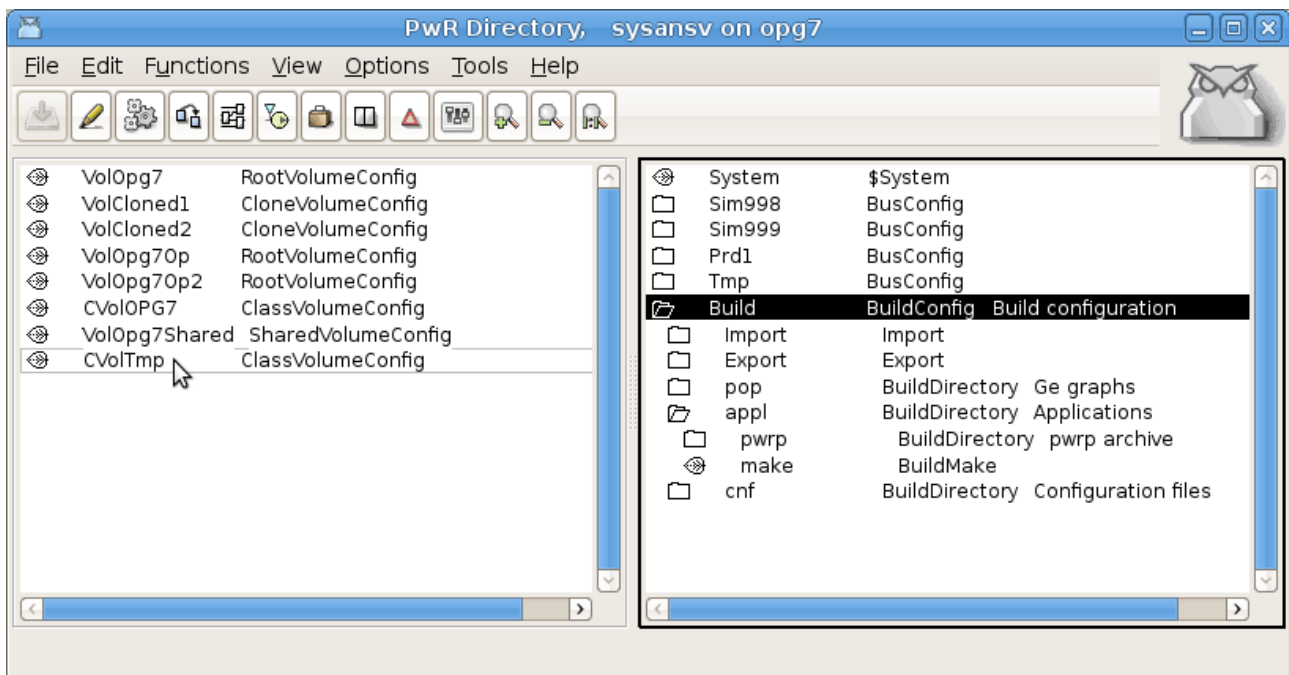


Fig 1 The build configuration

Build and distribution of configuration files

The build and distribution of configuration files is more strict regarding the separation in source and build tree. Now all configuration files reside in \$pwrp_cnf, and in \$pwrp_cnf there can be one subdirectory per node. Under each node there can be one subdirectory for each linux user on the node. No files should be distributed directly from \$pwrp_cnf, they should first be copied to \$pwrp_load in the build tree. When the node is built, a corresponding node and user structure is created under \$pwrp_load, and files are copied from \$pwrp_cnf to \$pwrp_load. The files handled

by the build methods are

\$pwrp_cnf/xtt_help.dat	→ \$pwrp_load/
\$pwrp_cnf/'node'/xtt_help.dat	→ \$pwrp_load/'node'/
\$pwrp_cnf/'node'/'user'/xtt_help.dat	→ \$pwrp_load/'node'/'user'/
\$pwrp_cnf/'node'/'node'_xtthelp.dat	→ \$pwrp_load/'node'/
\$pwrp_cnf/xtt_setup.rtt_com	→ \$pwrp_load/
\$pwrp_cnf/'node'/xtt_setup.rtt_com	→ \$pwrp_load/'node'/
\$pwrp_cnf/'node'/'user'/xtt_setup.rtt_com	→ \$pwrp_load/'node'/'user'/
\$pwrp_cnf/Rt_xtt	→ \$pwrp_load/
\$pwrp_cnf/'node'/Rt_xtt	→ \$pwrp_load/'node'/
\$pwrp_cnf/'node'/'user'/Rt_xtt	→ \$pwrp_load/'node'/'user'/
\$pwrp_cnf/pwrp_stop.sh	→ \$pwrp_load/
\$pwrp_cnf/'node'/pwrp_stop.sh	→ \$pwrp_load/'node'/
\$pwrp_cnf/pwrp_alias.dat	→ \$pwrp_load/

The copying is made when the node is built, but can also be made separately with the configurator command 'build cnf /node='. All copying can also be disabled from the configurator options dialog, in case one prefers to this with make files or scripts.

Plc help texts

The plc objects HelpText and HelpTextL are added to view and generate help texts from plc documents. The help texts inserted in these objects are viewed in the plc document, and can also be viewed in the help browser.

The text can contain tags to format the text and to insert images and links. Images will not be displayed in the plc document, they will only be shown in the help browser. Neither will links work in the plc document. The supported tags are

<h1>	Large header
<h2>	Header
<image>	Display an image.
<link>	Link to another help text topic

The maximum text size is for the HelpText object 1023 characters, and for HelpTextL 8191 characters.

The help texts will be concatenated to one help text file for the volume, when the volume is built. The file will contain one topic for each plc window, that contains the text from all HelpText objects in the window. The texts are displayed in the order the HelpText objects are found in the database.

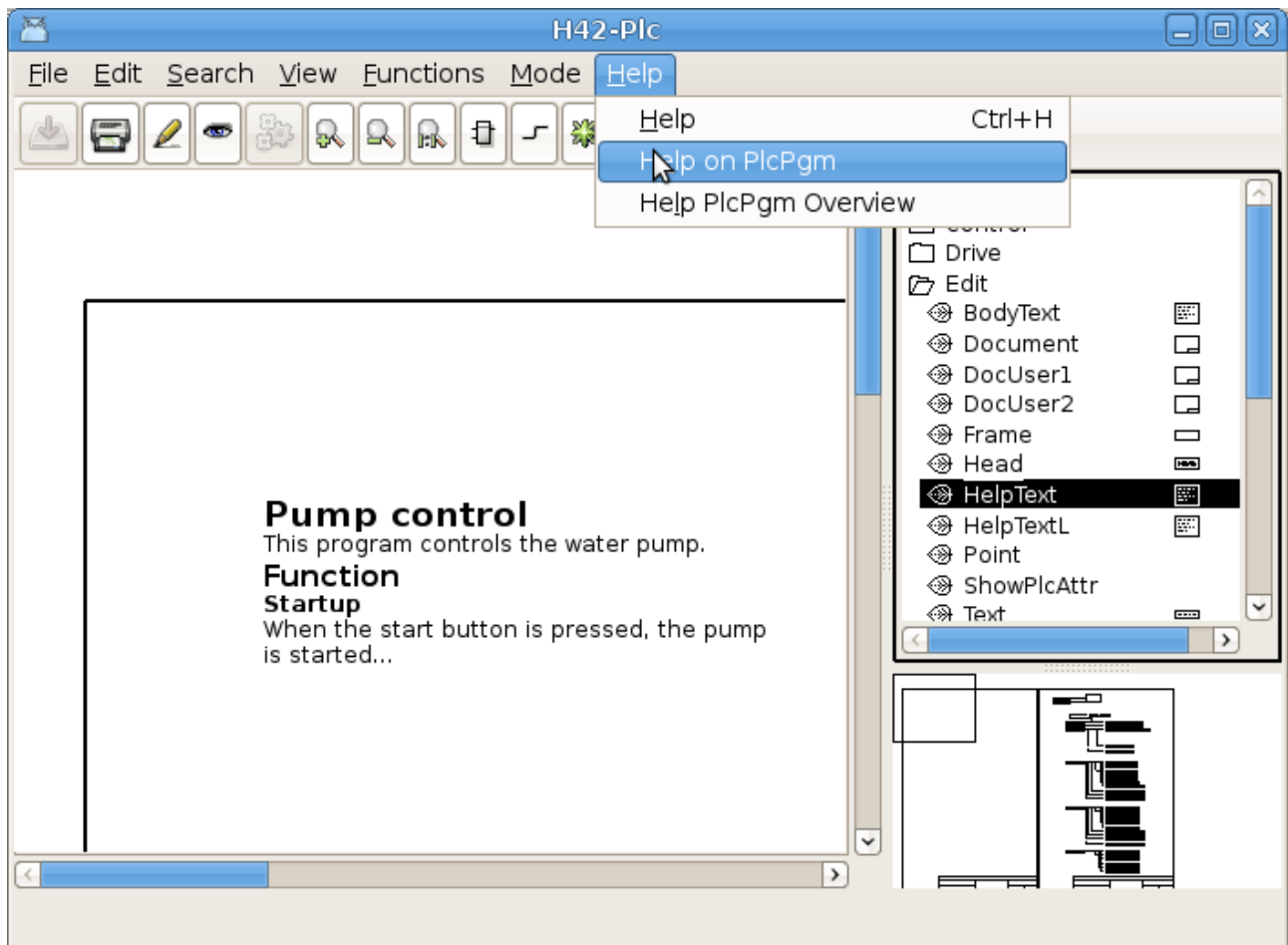


Fig Help text example

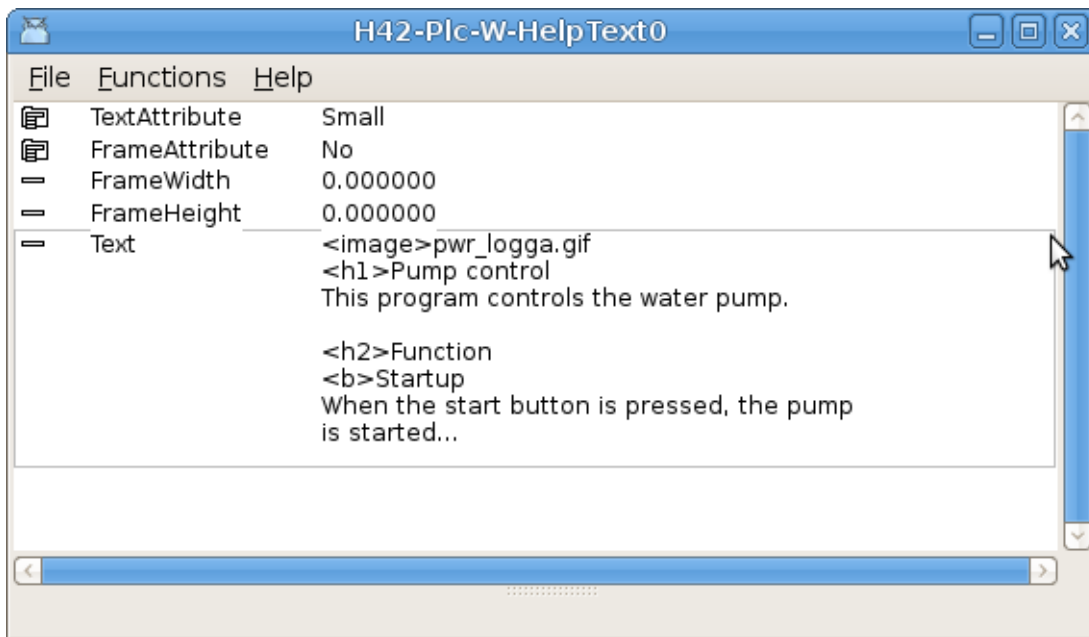


Fig Edited text

The help texts are viewed in the help browser, by activating help in the popup menu for the PlcPgm object, or from 'Help on PlcPgm' in the plc editor menu. Also in runtime the text can be displayed in the same way. The text in the help browser first shows a list of all subwindows with links to the help texts of the subwindows.

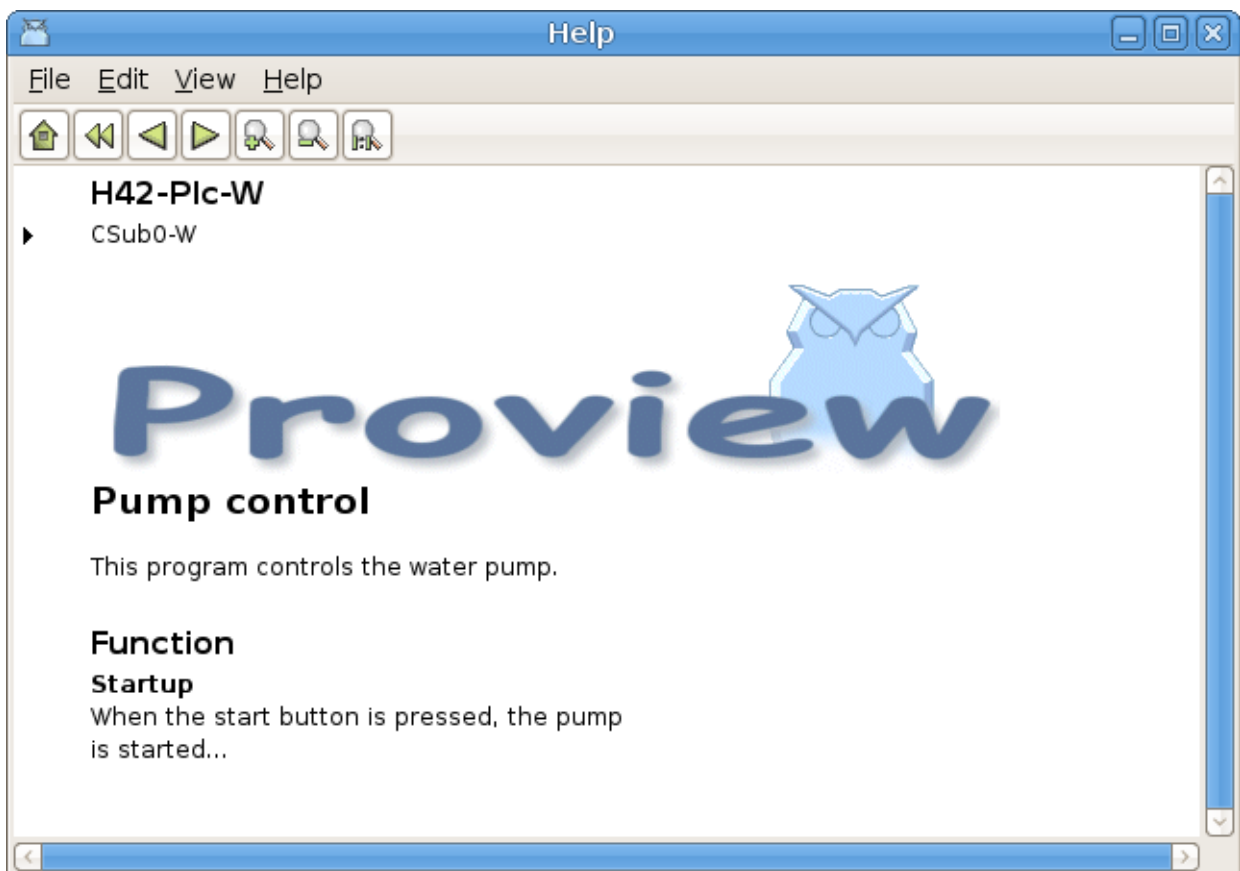


Fig Text in help browser

From 'Help PlcPgm Overview' in the plc editor menu, a list of all the PlcPgm's in the volume is displayed with description and links to the help texts of each PlcPgm.

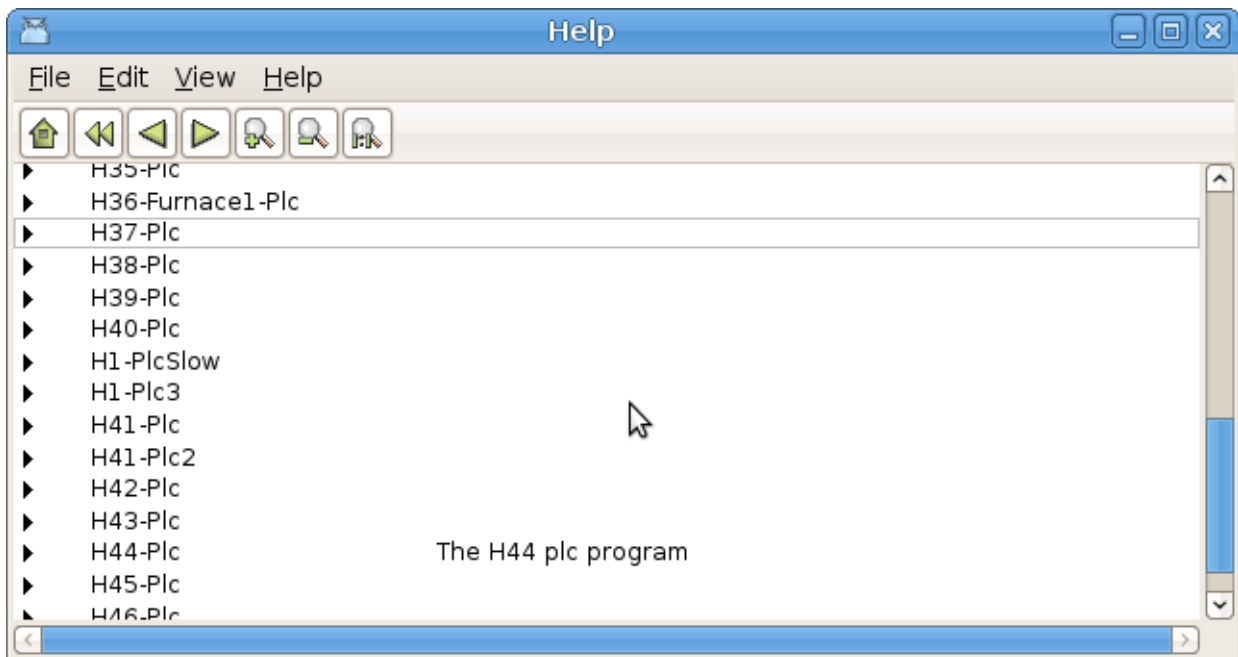


Fig Help PlcPgm overview

New time plc objects

A number of new time plc objects are added.

AtSel	Select one of two absolute times.
AtMux	Absolute time multiplexer.
AtDemux	Absolute time demultiplexer.
AtMin	Absolute time minimum function.
AtMax	Absolute time maximum function.
AtLimit	Absolute time limiter.
DtSel	Select one of two delta times.
DtMux	Delta time multiplexer.
DtDemux	Delta time demultiplexer.
DtMin	Delta time minimum function.
DtMax	Delta time maximum function.
DtLimit	Delta time delimiter.

New data reference plc objects

A number of new data reference plc objects are added.

DataSel	Select one of two data references.
DataMux	Data reference multiplexer.
DataEqual	Check of two data references are equal.
DataNotEqual	Check if two data references are not equal.

New string plc objects

A number of new string plc objects are added.

StrSel	Select one of two strings.
StrMux	String multiplexer.
StrEqual	Check if two strings are equal.
StrNotEqual	Check if two strings are not equal.
StrAdd	Add a number of strings.
StrTrim	Remove leading and trailing spaces in a string.
StrParse	Parse a string.

New Classes

BuildConfig

Modified Classes

Upgrade procedure

The upgrading has to be done from any version in the interval V4.7.0. 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->V4.6.0->V4.7.0->V4.8.6->(V5.0.0)->V5.1.0->V5.2

Enter the administrator and change the version of the project to V5.2.0. Save and close the administrator.

I you have any class volumes, enter the class editor and build the volume.

Enter the configurator for each root volume and activate 'Function/Update Classes' and build.

Note ! The update procedure upgrade.sh doesn't have to be executed.