Release Notes V4.3.0

07 01 21 cs

Table of Contents

Upgrading to Proview V4.3.0

This document describes new functions i Proview V4.3.0, and how to upgrade a project from V4.2.0 to V4.3.0.

New functions

Graphical user interface

The graphical user interface is change to GTK (The GIMP Toolkit), on which the Gnome desktop is built. The interface is totally rewritten and now have a more modern look and layout. Many windows now has a toolbar with faster access to functions also present in the menu. You can also scroll with the mouse wheel in most windows.

- The Configurator now has a toolbar with buttons for save, enter edit mode, zoom etc. The horizontal pane is changed to a vertical pane with the plant hierarchy to the left and the node hierarchy to the right.
- The Plc Editor also has a toolbar for save, edit/view mode, zoom, view different palettes etc. The width of the palette can be changed with a pane, and the plant hierarchy palette also shows the object description.
- In the Ge Editor, the toolbar is increased with buttons for group/ungroup, pop/push, align, equal distance, move vertical/horizontal, show grid, zoom.
- Xtt navigator has a toolbar with the methods of the selected object.
- In Plc Trace the navigation window is inserted into the main window.
- In the Curve window (for Trends, FastCurves and Logging) you can scroll horizontal with the mouse wheel, and width the arrow left and right keys. The arrow up and down keys zooms the curve.

The previous interface, Motif, is still available in the source code, but not included in the released packages.

Database locked dialog

If the database is locked, a dialog window is viewed where the user can choose whether to exit, enter the loadfile, or remove the lock and enter the database.

Extern Volumes in runtime

Extern volumes are implemented in runtime. An extern volume is a kind of plug-in for external functions to Proview. Normally, objects in a volume reside in the runtime database, but in an external volume, they appear to be in the database, but actually can be data of any kind, for example from an extern database, or from another process control system (Siemens, ABB). All data that can be presented as an object with a name and identity, in an object tree, could be connected as an Extern Volume. One has to write an interface to the extern function, implementing the rt_provider interface. When this is done, the data can be viewed in the runtime navigator, or subscribed from Ge graphs. In V4.3, extern volumes are used to connect Proview V2.7 systems to V4.3 systems.

Gateway between V4.3 and V2.7b

Communication between the nodes are performed by a client process in the V4.3 system and a server process in the V2.7b system. For every mounted volume, one client, and one server process has to be started.

On the client side, mounted volumes are presented as volumes of type Extern Volumed, i.e. volumes that is not of Proview type, but fed with information by a provider process. The provider process also works as a client process, communicating with the server process in the V2.7b system via UDP. Objects and subscription request from the nethandler, are forwarded to the server by the provider. The communication protocol is similar to the nethandler protocol, with the difference that attributes is referenced by objid and object name, instead of attribute references, i.e. objid and offset. Then, you are not dependent on that the classes are identical, when subscribing or requesting separate attributes. Though, if a whole object body is subscribed, the classes has to be identical. Also the class identity has to be identical, i.e. classes that has been moved can not be interpreted (this affects some classes in Nmps, remote and ssab cards). Userclasses that should be interpreted has to be defined on the client side with the same classidentity as on the server side.

The client is implemented for Linux V4.3, the server side for Linux V4.3, OpenVMS V2.7b and VAXELN V2.7b.

Client configuration

Any userclasses on the serverside has to be loaded also on the client side, that is, the wb_load files has to be upgraded and moved to the V4.3 system. A ClassVolumeLoad object should be placed in the project volume, which will load classvolume at proview startup.

Create a mount object for the volumes that is to be mounted and supply objid for the object that is to be mounted.

Start the client process remote_pvd_pwrcli with the arguments:

1. Ip-address of the mounted node.
2. Name of the mounted node.
3. Volumeidentity of the mounted volume.
4. Name of the mounted volume.
5. Server identity, i.e. an identity between 1 and 255 that is unique for each remote_pvd_pwrcli process on the node (default 200).
6. UDP-port. Has to coincide to the UDP port on the server side (default 3051).
7. Operative system on the server side: 0 Linux, 2 OpenVMS, 3 VAXELN.

Example

```
remote_pvd_pwrcli 192.168.1.1 ptlab1 0.1.99.6 VolPTLAB1 201 3052 3
```

Sever configuration

Start the server process remote_pvd_pwrsrv with the arguments

1. Ip-address of the client node.
2. Name of the client node.
3. UDP-port. Has to coincide with the UDP-port of the client (default 3051)

Example VAXELN

```
program SSABB_ROOT:<VAX_ELN.EXE>REMOTE_PVD_PWRSRV.EXE_ELN -
/warm_debug -
/argument=(“”””””,””””””,””””””,”””192.168.1.2”””,””pwr43”””,”””30
52”””)
```

Installation

The dependency of Motif and Imlib is removed in the released packages, which strongly simplifies the installation procedure.

Profibus interface for Softing Profiboard card

Proview has since some versions ago had support for a Profibus DP PCI-card manufactured by the company Softing (order no. PB-IF-1MS). Formerly we had written our own driver for this device. As from Proview v4.3 we instead use the driver provided by Softing themselves. The driver for the profibus card can be downloaded from ftp://ftp.softing.com/pub/outgoing/profi/linux/profibusV1.4.1.tar.gz. Build and install it according to the instructions.

The Proview I/O-routines has been adapted to work with the new driver. To configure it in the Node-hierarchy in the workbench you insert a Pb_Profiboard object and configure the profibus slaves as children to this object (read more in the Designer's Guide).

Property EscapeStore for dynamic type ValueInput

Dynamic type ValueInput has a new property, EscapeStore. Normally the input value is stored when 'Return' is pressed. However, when EscapeStore is set, the value is stored when the input field looses input focus, or when 'Return' is pressed.

Build method for Application objects

A build method is added to the Application object. If a bash command is inserted into the BuildCmd attribute, this command will be executed when the object is built. You can, for example, execute a makefile that will build the application if any source code is modified.

Build method for PlcProcess object

There is also a build method for the PlcProcess object, similar to the build method of the Applicatin object. Modules that are link with the plc program can be build here.

Note that there are no dependency to build the node (and to link the plcprogram) if any module are rebuilt.

New syntax in Function Key definition file

For operator stations using GTK the syntax in the function key definition file, $HOME/Rt_xtt, is changed. The statements should be written on separate lines, without any leading '*hotkeys:'. A comment is written with '#' in the first position of the row.

Example

```
#
# Function key definition file
#
Ctrl<Key>F1: SetDig(V10-PK1-PUSH-Nolla.ActualValue)
Ctrl<Key>F2: ToggleDig(V10-PK1-PUSH-Etta.ActualValue)
```

Web homepage filename

The name of the project homepage is changed from index_opwin.html to index.html, or if a filename is specified in the WebHandler object, from 'filename'_opwin.html to 'filename'.html.

Web archives configuration

Some object graph java archives are quite large and time consuming to load when opening the homepage for a project. It is no possible to configure whether to load them or not. The WebHandler object has a LoadArchives attribute, which is a bitmask where loading of the archives pwr_basecomponent.jar (object graphs for objects i basecomponent volume) or pwr_abb.jar (object graphs for objects in the abb volume).

Changed Classes

PlcProcess

PlcProcess has a new attribute BuildCmd and a build method (see above).

WebHandler

WebHandler as a new attribute LoadArchives (see above).

New Classes

CompOnOffBurner, CompOnOffZone

Objects for On/Off burner contol.

Upgrade procedure

The upgrading has to be done from V4.2.0. 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 -> V4.3.0.

The upgrade procedure is to change the version of the project in the projectlist, and then activate Functions/Update Classes from the meny in the Configurator. If the V4.2 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. Fined 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 V4.3.0. Save and close the administrator.

Update Classes

Do `sdf` to the project.

Start the Configurator, enter edit mode, and activate Functions/Update Classes in the menu.

Save and Build.