Release Notes V4.2.0

Revision 0.1	**Revision History** **2006-04-01** **v0.1**	**cs**

Release Notes V4.2.0:

by Claes Sjöfors

Table of Contents

Chapter 1. Introduction

1.1. Upgrading to Proview 4.2.0

This document describes new functions i Proview V4.2.0, and how to upgrade a project from V4.1.3 to V4.2.0.

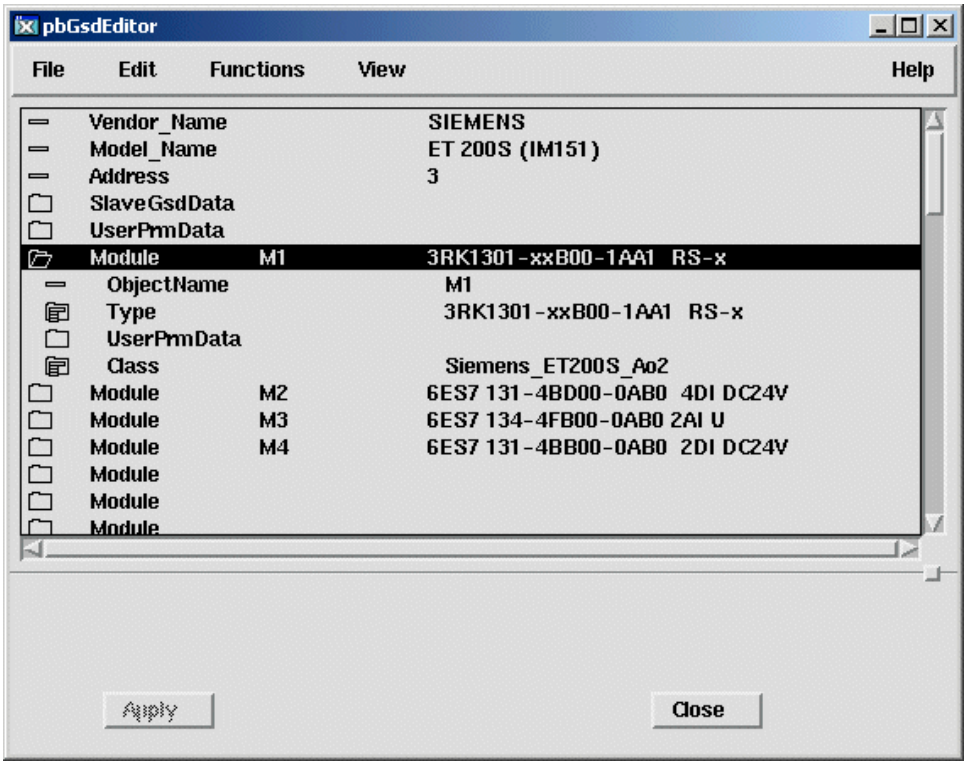
Chapter 2. New functions

2.1. Profibus configurator

The configuration of profibus is changed in V4.2.0, both the configuration procedure and the objects used for the configuration.

You start by creating a master object in the node hierarchy, for Softing profiboard the class **Pb_Profiboard** is used. Under this the slaves of the profibus circuit are configured with **Pb_DP_Slave** objects, or object that is a subclass of **Pb_DP_Slave**. If you use the **Pb_DP_Slave** object you put the name of a gsd-file, byteordering and possible floatrepresentation into the object. For some slaves, there are specific subclasses, for example **Siemens_ET200S_IM151**, **Siemens_ET200M_IM153** and **ABB_ACS_Pb_Slave**. In this case, the gsd-file is already specified in the object, and the file also comes with the proview release.

Next step is to open the profibus configuartor for each slave, by activating Configure Slave in the popup menu for the slave. The Profibus configurator reads the gsd-file and displayes data and configuration alternatives for the slave. Under the map `SlaveGsdData` information about the slave is displayed, and under the map `UserPrmData` configuration data for the slave is displayed.



The slave can keep a specific number of modues, and for each possible module there is a module entry in the configurator. By opening a module entry you can specify type, configuration data, objectname and object class for the module.

Type

Under Type all possible types are displayed for the actual slave. Select the desired type by clicking in the checkbox for the type.

Configuration data

Under UserPrmData the configuration alternatives of the selected module are displayed.You can specifiy data and choose between different alternatives to configure the module. See the datasheet for the module for more information about the alternatives.

Objekt name

When configurinn, the Profibus configuration creates a module object under the slave object. In ObjectName you specify a name of the module object. The name should be unic for the slave.

Modul class

Under ModuleClass the possible classes of the module object, that is created under the slave object, are displayed. The class you choose is dependent on the layout of the dataarea tranfered on the profibus circuit. There are a number of specific classes, e.g.**Siemens_ET200S_Ai2, Siemens_ET200SDi2, ABB_ACS_PPO4.** These contains a specified dataarea described by intern channel objects. If there is no matching module class, you choose **Pb_Module** and specify the layout of the datarea later, by creating channel objects under the module object.
Whan all the modules are configured, you click on apply, and the different module objects are created. Now the PrmUserData configuration of the slave and the modules is stored in the attribute PrmUserData in the slave object, together with some other data.

You also have to assign a Process and PlcThread for the the configuration objects, and configure channel object under Pb_Module object if necessary.

2.2. Build methods

Compiling PlcPgm, creation of loadfiles and bootfiles are now preformed by the Build function. The build function consists of build methods for node, volumes and objects.

2.2.1. Build methods for objects

PlgPgm

The build method for a PlcPgm checks if the plc-kode is modified since the last compilation. If it is changed, the program with all subwindows are compiled.

XttGraph

The build method for a XttGraph copies the .pwg file from $pwrp_pop to $pwrp_exe if the file on $pwrp_pop is more recent than the file on $pwrp_exe. If the graph is a java applet or java application, it is exported as java and compiled.

WebHandler

The build method for a WebHandler object creates a home sita for a node (calls Generate Web).

2.2.2. Build methods for volumes

Rootvolume

The build medhod of the rootvolume calls the build method of all PlcPgm, XttGraph and WebHandler objects in the volume. If the volume is modified since the last loadfile creation, new loadfiles are created. Also new crossreference files are created if this is specified in Options.

Classvolume

If the class volume is modified since the last creation of loadfiles for the volume, new loadfiles and structfiles are created for the volume.

2.2.3. Build methods for nodes

The build method of a node calls the build method of the opened volume, and thereafter creates a new bootfile for the node.

Note

Only the volumes that are opened is built. If the node contatins several volumes the other volumes must be built separatley, before the node is built.

2.3. PSS9000 Remote rack

A Proview system can now fetch data from a PSS9000 rack via ethernet. The rack is configured by a Ssab_RemoteRack object in the node hierarchy. Under the rack the cards are configured in the ordinary way.

2.4. ld_node_xxx.dat

ld_node file contains the nodes a node connects to via QCOM at proview startup. The files is generated from data in NodeConfig and FriendNodeConfig objects in the project volume.

Until now, the ls_node file has been common for all nodes of a project in the same QCOM bus. Now each node has a separate ld_nod file. This makes it possible to control individually which external nodes a node connects to.

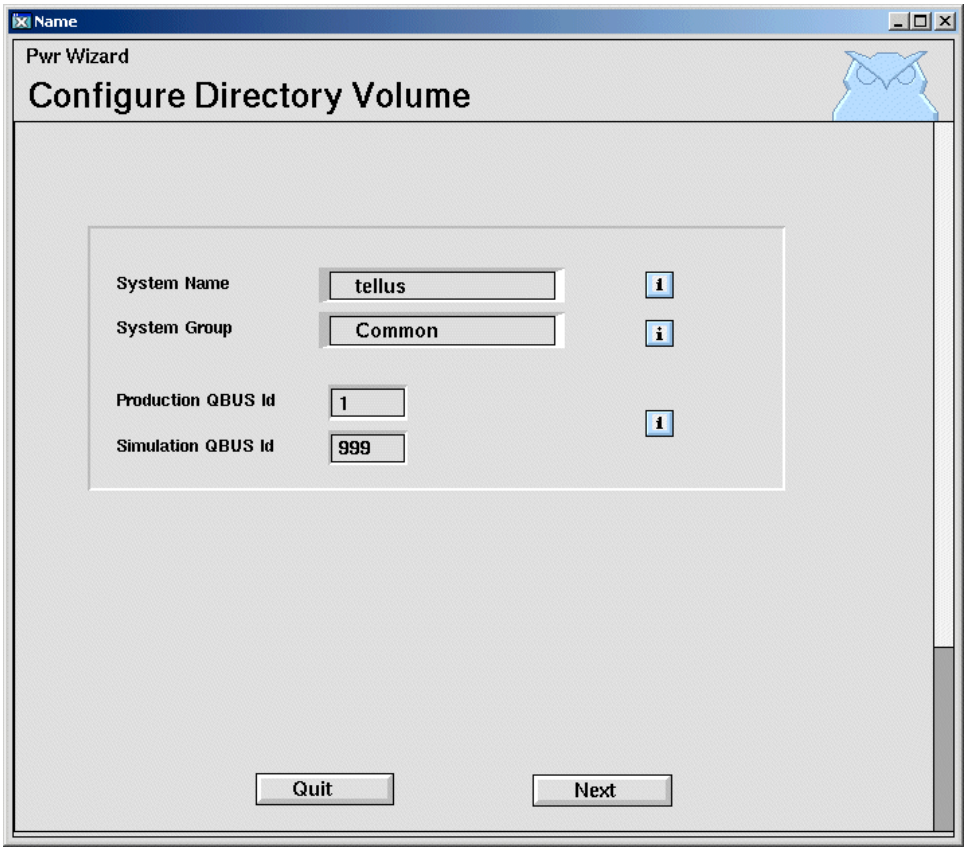
As before, this is configured by FriendNodeConfig objects in the project volume. These have been configured as siblings to the NodeConfig objects in a QCOM bus, and results in all local nodes connecting to this external node.Now a FriendNodeConfig object can also reside as a child to a NodeConfig object, implying that only this node connects to the external node.

2.5. Buffering of subscriptions removed

The buffering of subscriptions, which could lead to catch up fenomena at bad communicaion, is now removed.

2.6. Project configuration Wizard

The configuration of the project volume is now simplified by using a wizard that is automaticaly started when an empty project volumes is opened. The wizard feches the configured volumes of the project from the global volumelist, and creates volume and node configuration objects for these.



2.7. Update of classes

If a class in a classvolyme was modified, you previously had to dump the database to a textfile and the reload this, to update the instances of the modified class. Now there is a function that updates instances without dump and reload.

Every database stores loadfiles for classvolymes locally in the database directory. It is these local files, and not the global dbs-files in $pwr_load or $pwrp_load that is used when the workbench is opened. This makes you independent of changes in the global dbs-files. When the workbench is started, the versions of local and global dbs-files are compared, and if a new version in a global dbs-file is found, you get a warning message about this. The command 'check classes' displayes which classes are modified, and if there are any instances of the classes in the database. You should then activate Functions->Update Classes in the menu to update the instances and the local dbs-files.

For function object classes there are some restrictions. In some cases the connections to the function object has to be reconnected. If an input or output is removed, the input or output pin should not be visible in any instance, otherwise the connections should be redrawn. Also if an input or output is moved, the connections should be redrawn.

2.7.1. Objects for time handling

A number of new objects to handle times is added in V4.2.0. There are objects to store, add, subtract times etc.

Signals

The signal objects ATv (AboluteTimeValue) and DTv (DeltaTimeValue) store time values in the shape of an absolute time (of type pwr_tTime) or a delta time (of type pwr_tDeltaTime, i.e. a time intervall).

The objects are found under the signal map in the palette. IO-copying of the objects is not performed.

Plc objects

Addition and subtraction of times is perfomed in the plc program by the objects **AtAdd, DtAdd, AtSub, DtSub** and **AtDtSub**.

To fetch an ATv or DTv the objects **GetATv** and **GetDTv** is used. To fetch an attribute of type pwr_tTime or pwr_tDeltaTime in an object, the objects **GetATp** and **GetDTp**

To store a time value in an **ATv** or **DTv**, **StoATv** and **StoDTv**, or **CStoATv** and **CStoDTv** for conditional storeage. To store a time value in an attrbiute of type pwr_tTime or pwr_tDeltaTime, the objects **StoATp** and **StoDTp**, or **CStoATp** and **CStoDTp** for conditional storage.

To convert a deltatime to float DtToA is used, and vice vers AToDt.

All objects are found under the map Signals->Time in the palette of the plc editor.

2.7.2. Update Classes

Previously, if you made a change in a class, you had to reload the database, i.e. dump the database to a textfile, and then load the textfile into the database again. Now there is a funcion that converts the objects in a database to the now class description without a reload. When the workbench is started, any new version of a dbs-file for a classvolume is detected. If a new version i found, an error message is displayed in the message window. You can then either continue with the old class description, or update the objects to the new. The update is performed from Function->Update Classes in the menu. The objects that are influenced by the new class description can first be displayed by the command `wtt> check classes` that lists the modified classes and the number of instances found for each class.

Before executing a class update, be sure to have a backup of the database.

2.7.3. Ge

Object graph in Window and Folder

It is now possible to display an object graph in a window or folder object. The instance object of the object graph is inserted in the properties Window.Object and Folderx.Object.

Select color in Table

A property to modify the color of selected cells in tables is added to the Table object. Set the desired color in Table.SelectColor.

Bit type in Invisible

Access is often stored as bits in a bitmask, and it is now possible to influence the sensitivity and visibility for an object from a bit in a bitmask. The type for the attribute is written ##Bit#32[7] which means a 32-bit bitmask, bit number 7 (the first bit is bit 0).

2.7.4. Modified types

pwrb:DataRepEnum

The values Int32 and UInt24 is added.

2.7.5. Modified classes

Profibus:Pb_Module

The attribute ModuleName is added.

BaseComponent:CompLimit

The attribute DisableAlarm is added, which makes it possible to use the limitvalue supervision in a BaseSensor without alarm.

BaseComponent:CompModeDM, CompModeDMFo

Funktionality for local mode added.

BaseComponent:BaseMValveFo

Funktionality for local mode added.

RootVolume, SubVolume, SharedVolume

The attribute Modified is added, where the time of the latest save is stored.

2.7.6. New classes

CompPID, CompPID_Fo

The Pid controller divided into a main object and a function object. The controller can be a component of another object.

CompModePID, CompModePID_Fo

The mode object to the PID controller.

GetDatap

Plc object to fetch the reference to a data object, e.g. a data output in a DataArithm. Can also be used to attach data inputs in function objects with template plc code.

pwrb:ATv

Absolute Time Value, storage of an absolut time, pwr_tTime.

pwrb:DTv

Delta Time Value, storage of a delta time, pwr_tDeltaTime.

pwrb:AtAdd

Addition of an absolute time and a delta time.

pwrb:DtAdd

Addition of two deltatimes.

pwrb:AtSub

Subtract an absolute time from an absolute time.

pwrb:DtSub

Subtract a delta time from a delta time.

pwrb:AtDtSub

Subtract a delta time from an absolute time.

pwrb:AtEqual, pwrb:AtGreaterThan, pwrb:AtLessThan

Comparition two abolute times.

pwrb:DtEqual, pwrb:DtGreaterThan, pwrb:DtLessThan

Comparition of two delta times.

pwrb:CurrentTime

Fetches the system time.

pwrb:DtToA, pwrb:AToDt

Konverterar från detatid till flyttal och vice versa.

pwrb:GetATv

Fetches the value of an ATv.

pwrb:GetDTv

Fetches the value of a DTv.

pwrb:StoATv

Stores a value into an ATv.

pwrb:CStoATv

Conditional storage of a value into an ATv.

pwrb:StoDTv

Stores a value into a DTv.

pwrb:CStoDTv

Conditional storage of a value into a DTv.

pwrb:StoATp

Store a value into an absolute time attribute.

pwrb:CStoATp

Conditional storage of a value into an absolute time attribute.

pwrb:StoDTp

Store a value into a delta time attribute.

pwrb:CStoDTv

Conditional storage of a value into a delta time attribute.

ssabox:Ssab_RemoteRack

Configuration of a PSS9000 remote rack.

ABB_ACC800, ABB_ACC800Fo, ABB_ACC800Sim

Control of a motor aggregate using the crane macro in ASC800.

ABB_ACC_PPO5

Profibus module to ABB_ACC800.

ABB_ACS_Pb_Slave

Profibus slave to ABB_ACS800.

ABB_Sensor_Pb_PA, ABB_Sensor_Pb_PA_Fo

Baseclass for ABB Profibus PA sensor.

ABB_TempSensor_TF12, ABB_DiffPressure_265G, ABB_FlowSensor_FXE4000

Some ABB Profibus PA sensors.

Siemens_ET200S_IM151, Siemens_ET200M_IM153

Profibus slave objects for ET200S IM151 and ET200M IM153

Siemens_ET200M_Di32, Siemens_ET200M_Di16, Siemens_ET200S_Di8

Profibus module objects for ET200M digital input modules

Siemens_ET200M_Do32, Siemens_ET200M_Do16, Siemens_ET200S_Do8

Profibus module objects for ET200M digital outout modules.

Siemens_ET200M_Ai8, Siemens_ET200M_Ai4, Siemens_ET200S_Ai2

Profibus module objects for ET200M analoga input modules.

Siemens_ET200M_Ao8, Siemens_ET200M_Ao4, Siemens_ET200S_Ao2

Profibus module objects for ET200M analog output modules.

Siemens_ET200S_Di4, Siemens_ET200S_Di2

Profibus module objects for ET200S digital input modules.

Siemens_ET200S_Do4, Siemens_ET200S_Do2

Profibus module objects for ET200S digital output modules.

Siemens_ET200S_Ai2

Profibus module objects for ET200S analog input modules.

Siemens_ET200S_Ao2

Profibus module objects for ET200S analog output modules.

Chapter 3. Upgrade procedure

3.1. Procedure for upgradeing

The upgrading has to be done from V4.1.3. If the project as a lower version, the upgrade has to be performed stepwise following the scheme **V2.1 -> V2.7b -> V3.0 -> V3.3 -> V3.4b -> V4.0.0 -> V4.1.3 -> V4.2.0**

Note

Upgradeing a Debian project from V4.1.3 requires that a new debian release is installed. Dump the project with the reload.sh script before installing the new debian release, and while the project is still pointing at V4.1.3. Execute the only the dumpdb pass. Intall the new debian release, and when executing the upgrade.sh script, skip the dumpdb pass.

The upgrading is made in two steps:

- Make a copy of the project
- Execute upgrade.sh

3.1.1. 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 projectname and path. Change the vertion to V4.2.0. Save and close the administrator.

Do sdf to the project.

3.1.2. upgrade.sh

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`

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` (V4.1 databases, directories which content also is removed)