Release Notes V4.7

09 07 03 cs

Table of Contents

Upgrading to Proview V4.7.0

This document describes new functions i Proview V4.7.0, and how to upgrade a project from V4.6.1 to V4.7.0.

New functions

Project directories

The directory structure of a Proview project is changed.

A project is separated into two trees, a build tree under bld and source files under src.

Profinet

Ge action SetValue

New Ge action to set a specified value into an attribute.

Loadfile format

The loadfile format is changed (.dbs files), and not compatible with loadfile format i V4.6 and previous versions. Dbs-files of these versions can not be opened from the configurator (File/Open/Dbs file).

Linux 64

Proview 4.7 is ported to 64 bit linux and distributed in one 32 bit and one 64 bit version.

Mixed development environment

You can have a mixed development environment with both 32 bit and 64 bit linux machines and with a common database server. There are though some restrictions.

A BerkeleyDb database created on a 32 bit platform can not be opened on a 64 bit platform, and vice versa. A mysql database though, can be opened on both platforms.

A 32-bit system has to be built on a 32-bit platform, and a 64-bit system has to be built on a 64-bit platform. Classvolumes though, and the loadfile format, are platform independent.

Mixed runtime environment

There are no restictions in the runtime environment. Systems of different platforms can communicate without problems. If there is some application communication, keep in mind that the alignment of c data structures can differ between the platforms.

Directory volume database changed

The directory volume, previously stored in a BerkeleyDb database, is now stored as a wb_load file. That is $pwrp_db/directory.db is replace by $pwrp_db/directory.wb_load.

Definition of pointer attribute size

For pointer attributes that are not private, the size of the pointed entity has to be stated in the Size element of the $Attribute object defining the attribute.

Including external includefiles in the plc program

The environment variable PWR_EXT_INC can be used to define directories where include files included in the plc program are stored. If you, for example, has a common directory where you store includefiles that specify transactions, /usr/local/myinclude, the compiler will find these files by defining PWR_EXT_INC with the command

export PWR_EXE_INC="-I /usr/local/myinclude"

New Classes

PnControllerSoftingPNAK

I/O Agent object configuring the Sofing profnet stack.

PnDevice

I/O object configuring a Profinet device.

PnModule

I/O object configuring a Profinet module.

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)
```