Installation and maintenance of Process and Operator stations

2011 01 27

Installation

This section describes the the first installation of a Proview Process or Operator station. Note that there is no difference between the installation procedure of an process or operator station. The hardware and the software are identical.

The installation basically requires three steps: installation of the operating system, installation of the proview runtime package and the packages this is depending on, installation of the project package with the loadfiles and plcprogram of the current project. If you have a Proview loadstation available, all these steps are performed by the the load station. The first step by copying an image prepared with the needed packages, and also finding the node name and ip-address in the project configuration files and attaching these to the system. The second step by finding the appropriate version for the Proview runtime package and installing this. The third step by finding the latest project package and installing this.

Before starting the load station some preparation has to be done.

Check node configuration

DirectoryVolume

Open the directory volume (pwrs) and check the settings of the production bus and node.

Open the BusConfig object for the production bus and check that the bus is correct.

Open the NodeConfig for the process of operator station and check

- the node name.
- the ip-address in Address.
- the boot node, should be blank or the same as node name.
- the RemoteAccessType should be SSH.

Check the RootVolumeLoad object under the NodeConfig object, the name of the object should be the RootVolume of the node. Check also in the Distribute object, which states the files that are included in the project package.

RootVolume

Enter the rootvolume (pwrs 'volumname') and check some settings in the node hierarchy under the Node object:

- There should be a Security object with DefaultXttPriv set to 4 (System).
- The PlcThread objects, under the PlcProgram object, should have a Prio of about 20.
- Check also the MaxDelay in the CycleSup object under the PlcThread object. For alarm, we use to set this to the scantime or the thread, and for Halt 1 second. The alarm action should be Message for the Alarm object and EmergencyBreak for the Halt objects.
- Check the username in the OpPlace objects. This should be set to op1, op2, op3 or op4 for an operator station. Check also that the EventSelectList contains the hierarchies that the operator should receive alarms from.
- In the IOHandler object, IOReadWriteFlag should be set to 1 and IOSimulFlag to 0.
- In the MessageHandler object on operator stations, set EventLogSize to the number of alarms

you want to store in the historical event log (~5000).

Build the node

Build the node by activating Functions/BuildNode in the configurator for the rootvolume, and select the current process or operator station in the popup (if there are only one node configured the popup is not viewed). Check in the terminal window for error messages.

If external functions are called from the plc code, or if some additional modules are used, you might have to add archives or object modules in a .opt file on $pwrp_exe.

Installation with load station

Create a package for the node from the distributor. Open the distributor by activating Functions/Distribute in the configurator menu. Select the node, and activate Functions/Create Package in the menu. This will create a package with all the files needed on the runtime node. The package file is placed on $pwrp_load/pwrp_pkg_'nodename'_'version'.tgz.

Now everything is prepared to perform the installation on the load station.

Attach the disk and start up the load station. Enter the nodename of the process or operator station and wait until the installation is performed.

Then attach the disk to the process or operator station and boot the system.

Installation without load station

Install the operating system, and thereafter the Proview runtime package, following the instructions in the installation guide.

When the installation is done, enter the configurator and open the Distributor, select the node and activate Functions/Distribute in the menu. Select the node, and activate Functions/Distribute in the menu. This will create a package with all the files needed on the runtime node, copy the package to the runtime node and install the package. When copying the package, you will have to enter the password for pwrp in the terminal window. The package file is placed on $pwrp_load/pwrp_pkg_'nodename'_'version'.tgz.

Note! If you have selected RSH as RemoteAccessType in the NodeConfig object, the initial distribution will fail to unpack the package. The package has to be installed with pwr_pkg.sh. Log in to to runtime node as pwrp, check that the .tgz file is present and install it with

```
$ pwr_pkg.sh -i 'tgz-file'
```

Configure and check the runtime system

Configure busnumber

Before starting Proview, the bus number has to be configured in /etc/proview.cnf. To edit this file you have to be root. Login as root with 'su' and open the file in gedit. The busnumber stated in qcomBusId has to equal the busnumber in the BusConfig object in the directory volume.

Close the current terminal window and open a new one. Check that the busnumber is correct with

```
$ echo $PWR_BUS_ID
```

Check installation

Check also that the Proview runtime package is installed correctly

```
$ cat $pwr_exe/rt_version.dat
```

will show the version of the current package.

Check that the project package is installed, for example by looking at the plcprogram

```
$ ls -al $pwrp_exe/*plc*
```

If the project package is not installed for som reason (maybe the package was not created), a new package can be copied and unpacked from the distributor.

Start Proview

Reset any previous start attempt with

$ source pwr_stop.sh

Start proview with

$ pwr start

Check that the proview processes are started with

```
$ ps x
 3901 ?        Sl     0:00 /usr/pwrrt/exe/rt_ini
 3911 ?        S      0:00 rt_neth
 3912 ?        Sl     0:00 rt_qmon
 3913 ?        S      0:00 rt_neth_acp
 3914 ?        S      0:00 rt_io_comm
 3915 ?        S      0:00 rt_tmon
 3916 ?        S      0:00 rt_emon
 3917 ?        S      0:00 rt_alimserver
 3918 ?        Sl     0:00 rt_bck
 3919 ?        S      0:00 rt_linksup
 3921 ?        S      0:00 rt_fast
 3923 ?        S      0:00 rs_remote_logg
 3924 ?        S      0:01 rt_elog
 3925 ?        S      0:00 rt_sysmon
 3958 ?        Sl     0:00 plc_vhxo13_0507_00053
```

If the process are missing, reset and start rt_ini with the -i option, viewing the console messages in the terminal window.

```
$ source pwr_stop.sh
$ rt_ini -i

   Proview/R Version V4.6.0 for Linux on Unknown Hardware
   Copyright © 09-OCT-2008 12:00:00 by SSAB Oxelösund AB

   Proview/R is free software; you can redistribute it and/or
   modify it under the terms of the GNU General Public License.

   This program is distributed in the hope that it will be useful
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
   GNU General Public License for more details.

   PROVIEW/R Process Environment

I Reading Boot file
      /data0/pwrp/saturnus7/common/load/ld_boot_aristotle_0517.dat
I Created at 13-FEB-2009 16:16:53.85 for project: saturnus7
I This node vill run PLC file: plc_aristotle_0517_00014
```

```
I This node has node identity 0.1.1.4 (65796)
I Reading Node file
      /data0/pwrp/saturnus7/common/load/ld_node_aristotle_0517.dat
I Reading volume file         /data0/pwrp/saturnus7/common/load/volsaturnus7.dbs
I Created 13-FEB-2009 16:16:51.59
I Reading volume file   /data0/x4-6-0/rls/os_linux/hw_x86/exp/load/rt.dbs
I Created 09-JAN-2009 14:11:59.81
W Version missmatch for volume: pwrs, 09-OCT-2008 12:00:00.00 !=       09-
JAN-2009 14:11:44.30
...
```

Look for error messages ('E' or 'F') or warnings (W).

When the runtime has started correctly start rt_xtt and open 'System/System Status'/'current node'.

Check that all servers that are configured is running and has a green status indicator.

Click on the 'plc' button in the Process column and check that the LoopCnt of all the threads are counting. If not, the plc process is not alive.

Click on the 'rt_io' button in the Process column and check that the IOReadWriteFlag indicator is green. If not the IO has stalled.

Open 'System/System Messages' and look for any error messages (marked with red).

Maintenance

Distribution of project changes

When changes are made in the development environment, for example in a process graph or the plc program, these are distributed to the runtime stations by the Distributor. Open the distributor from Functions/Distribute in the configurator menu. Select the node and activate Functions/Distribute in the menu. Now a new package is created, containing all the files that is needed in the runtime environment for this node. The package is also copied to the node. Here you might have to enter the password for pwrp in the terminal window (this can be eliminated by creating a ssh key and copy this to the runtime node). Finally the package is unpacked on the runtime node, after removing the files from the previous package. Log in to to the runtime node and restart Proview.

Go back to a previous package

Sometimes the changes doesn't work as planned, and you want to return to a previous package. Then log in to the process or operator station as pwrp. ls -l *.tgz will show all copied packages and hopefully you will find the package you want to return to. The package is installed by the script pwr_pkg.sh. If you for example will install the package pwrp_pkg_vhxnu4_0009.tgz the command is

```
$ pwr_pkg.sh -i pwrp_pkg_vhxnu4_0009.tgz
```

You should be logged in as user pwrp when running pwr_pkg.sh.

Distribute without network

If you don't have any network contact between the development and process station, you can use som other media, for example a USB stick, to copy the package. Create a package in the distributor and copy it to the USB stick. Move the USB stick to the process station and copy the package to /home/pwrp. Install the package with pwr_pkg.sh

```
$ pwr_pkg.sh -i pwrp_pkg_vhxnu4_0010.tgz
```

Update of Proview runtime package

At rare occasions, bugs are found in a Proview release that has to be corrected. Also minor improvements and addition of new functionality can be made to a version. A new package is then built with the last number in the version number incremented. The three first figures has to correspond to the version of the development environment, for example if pwr46_4.6.0-7 is installed on the development station, pwrrt_4.6.0-x has to be installed on the runtime nodes, the last number though (x) can be any number.

To install a package you copy the package to the process or operator station, either by downloading it from a web brower and saving it to disk, or copying it with ftp. Login as root (su) and install the package with dpkg.

$ dpkg -i pwrrt_4.6.0-7.deb

Restart proview with 'pwr stop' and 'pwr start'.

Troubleshooting

Proview is not starting

Start Proview with the command 'rt_ini -i' to view the console messages in the terminal window (reset previous start attempts with 'source pwr_stop.sh')

```
$ source pwr_stop.sh
$ rt_ini -i
```

F Could not open file /data0/pwrp/saturnus7/common/load/ld_boot_..
The boot file could not be opened.

Check the busnumber

$ echo $PWR_BUS_ID

If it is not correct, change the busnumber in /etc/proview.cnf

Check that the bootfile $pwrp_load/ld_boot_'nodename'_'busid'.dat exists with the correct bus number. If the busnumber in the name is not correct, change it in the directory volume.

W Version mismatch for volume: ...
Usually a change is made in a classvolume, but the rootvolume for the node is not rebuilt. Update the classes and build the rootvolume for the node.

If the volume is a Proview base volume, check that the installed pwrrt version corresponds to the version in the development environment

```
$ cat $pwr_exe/rt_version.dat
```

If this doesn't help, the versions for the loadfiles can be viewed with wb_ldlist in the development environment. Check the versions of the .dbs files on $pwrp_load

```
$ wb_ldlist $pwrp_load/volsaturnus7.dbs
Volume     VolSaturnus7    13-FEB-2009 16:16:51.59 (1234538211,592936184) 65796
VolRef     VolSaturnus7    13-FEB-2009 16:16:51.59 (1234538211,592936184) 65796
VolRef     pwrs            09-OCT-2008 12:00:00.00 (1223546400,0) 1
VolRef     pwrb            09-OCT-2008 12:00:00.00 (1223546400,0) 2
VolRef     BaseComponent   09-OCT-2008 12:00:00.00 (1223546400,0) 10
VolRef     CVolSaturnus7   13-FEB-2009 16:09:38.61 (1234537778,610918602) 25372
VolRef     Profibus        09-OCT-2008 12:00:00.00 (1223546400,0) 64007
```

Here the volume rootvolume VolSaturnus7 references the class volume CVolSaturnus7. The time for CvolSaturnus7 has to equal the VolRef time. By listing the .dbs file for the classvolume we can compare the times.

```
$ wb_ldlist $pwrp_load/cvolsaturnus7.dbs
Volume     CVolSaturnus7   22-APR-2009 08:12:08.22 (1240380728,222698031) 25372
VolRef     CVolSaturnus7   22-APR-2009 08:12:08.22 (1240380728,222698031) 25372
VolRef     pwrs            09-JAN-2009 14:11:44.30 (1231506704,308475731) 1
VolRef     pwrb            09-JAN-2009 14:11:57.96 (1231506717,961138847) 2
VolRef     Profibus        21-APR-2009 13:43:05.27 (1240314185,278213917) 64007
```

The time 22-APR-2009 08:12:08.22 doesn't match the VolRef time in the rootvolume 3-FEB-2009 16:09:38.61 and this causes the mismatch. Updating the classes and building the rootvolume will fix the problem.

The plcprogram is not starting

Check that there is a plc executable on $pwrp_exe

```
$ ls -l $pwrp_exe/*plc*
```

No executable is found

Do you really have any PlcPgm objects in this node ? If there are no PlcPgm the executable is not created for the node.

Otherwise build the node and look for error messages when the plcprogram is linked. Any undefined references may be caused by plc windows that are not yet compiled, or PlcPgm objects which are not yet opened and where the plc window object is missing.

An executable exist

If the plc program terminates with a segmentation fault, it is usually caused by some code in CArithms, DataArithms or in external functions called by the plc program from arithm blocks. Erroneous pointers and array index out of bonds are common causes for this, but it can be quite hard to find where the faulty code is.

Start PlcPgm's one by one

One way is to stop the execution of all PlcPgm's at the startup of the plcprogram and start them one by one. This will at least tell you which PlcPgm and which window causes the problem.

The execution of the PlcPgm's are disabled by adding the row

```
'nodename'_setval  plcscan = Off
```

to the file $pwrp_load/pwrp_alias.dat (replace 'nodename' with the name of the node). Create the file if it doesn't exist.

The restart the plcprogram, this can be done from the prompt,

$ plc_aristotle_0517_00014

Start rt_xtt from another terminal window and start the PlcPgm's one by one by setting ScanOff in the PlcWindow object under the PlcPgm object to 0. Check when the plcprogram is terminating. The PlcPgm you were starting when the program terminated contains the erroneous code.

Use the debugger

By starting the plcprogram in the debugger, you will get the exact row causing the problem. First you have to compile suspected programs with debug, and also build the node with debug (build with debug is set in Options/Settings in the configurator). Copy the source files for the plc windows ($pwrp_tmp/*.gc) to the process station, and start the debugger from the directory where the sourcefiles are placed.

```
$ gdb plc_aristotle_0517_00014
```

Core file

Bye setting the system resource for maximum core file size, a core file will be created when a program terminates with for example segmentation fault.

ulimit -c unlimited

The core file can be opened by the debugger and the state when the crash occurred can be examined.

```
$ gdb -c core plc_aristotle_0517_00014
```

No connection to other node

Node in the same project

Node in other project