



Getting Started Guide

2009-05-20

Copyright SSAB Oxelösund AB 2009

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

About this guide.....	4
Download and installation.....	6
Create a project.....	7
Configure the Project.....	9
Configure the Process station.....	15
Create a simple Plc program.....	17
Build.....	20
Start runtime.....	21
Examine the runtime environment.....	22
Plant Graphics.....	23
Appendix.....	28
Glossary.....	28
Edit functions in the Configurator.....	30
Edit functions in the Plc Editor.....	31

About this guide

This guide will take you through the steps of configuring, developing, simulating and running a small Proview project, on a single-computer system. The guide does not intend to be comprehensive. For detailed documentation, please consult the Designer's Guide or the GE (graphical editor) Manual. These documents are available at the Proview [<http://www.proview.se>] site.

Download and installation

To configure a Proview system you need to install the Proview Development package, pwr46, which is available on the Download page at www.proview.se. Download the package for your Linux release and follow the instructions in the Intallation Guide to install for the package.

Create a project

- Login as user pwrp. During installation the user pwrp with password “pwrp” is added to the system.
- Open the ProjectList by doubleclicking on the Proview icon on the desktop. The [ProjectList](#) will open up in a new window.
- Enter Edit mode by activating Edit/Edit mode (Ctrl+E) in the menu. Once in Edit mode, an object palette will appear to the left in the window.
- Create a project hierarchy. The ProjectList at first only contains the “Bases” hierarchy. To [create a hierarchy](#) for the projects, select Hier in the palette, move the cursor to the “Bases” hierarchy and middle click. A new, empty Hier object is inserted below the existing one.

The hierarchy object is a container for arbitrary objects and it has two attributes: an ObjectName and a Description. To [expand the object](#) and expose the attributes, select the object and use the right arrow key. Now, to [edit an attribute](#), select the attribute and use the right arrow key to open the input field.

We choose the name “Test” for our hierarchy.

- We then [add a ProjectReg object](#) to the “Test” hierarchy and name it “GettingStarted”. We want to add the ProjectReg as a child and not as a sibling. To do this, middle click directly on the desired parent leaf, which is our “Test” hierarchy in this case. [Open the object](#) (select the object and use the arrow right key) and [set the name](#) to “GettingStarted”. Note that the project and path attributes also are influenced the name change.
- Save the changes (Ctrl+S) and leave Edit mode (Ctrl+E).

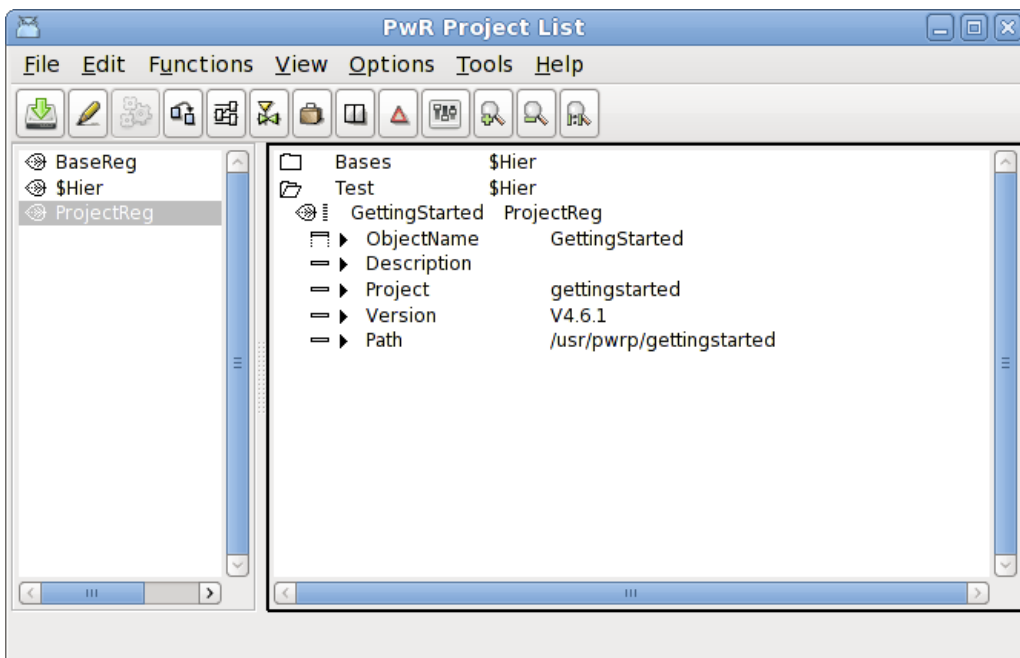


Fig The Project List

Tip

This configurator tool is used to configure most functions of the system, by creating

objects and set values to object attributes. See [*Edit functions in the Configurator*](#) on how to select/create/delete/move objects and set attribute values. Note that most actions can be activated in different ways: from the menu, from the PopupMenu (opened by rightclicking on an object), with a single mouseclick or from the keyboard.

Configure the Project

The next step is to attach the project and the [Directory Volume](#) by right clicking the ProjectReg object in the ProjectList, and choosing Open Project in the popup menu.

Tip

An alternative way is to open a terminal window and type

```
> sdf gettingstarted
```

The sdf command sets up environment variables for the project with default values. The Directory Navigator is then started with

```
> pwrns
```

Starting up the Directory Navigator for the first time in a project will activate a wizard that will configure the nodes and volumes of the project. To create a simple test project you basically just have to press the 'Next' button.

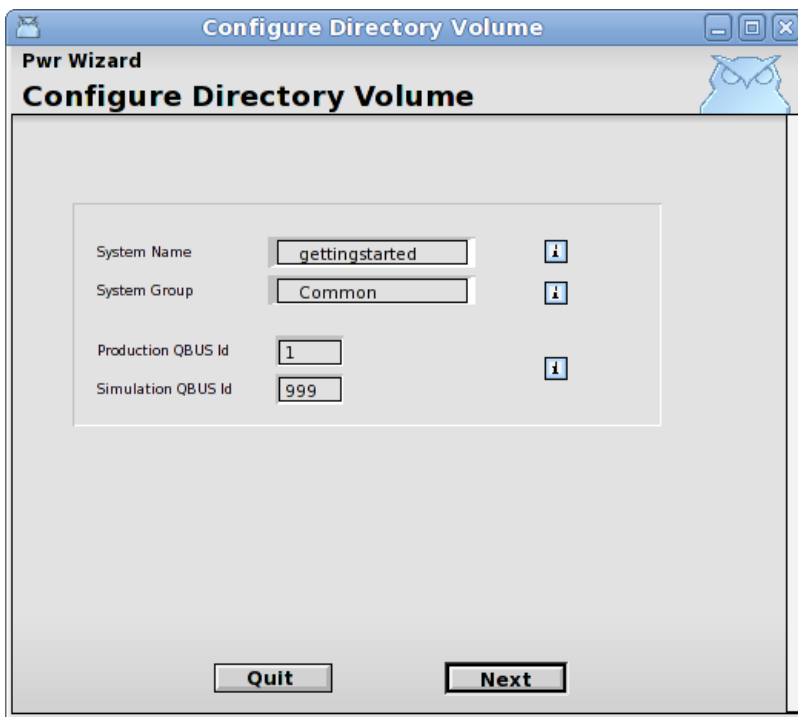


Fig Directory Volume Wizard 1

Press Next.

Comment

The systemgroup defines the users that will have access to the system. Common is a predefined group with the users pwrp and b55. By opening the UserDatabase you can define you can define another systemgroup with users and enter the name of this systemgroup here.

The Proview network is divided into busses, and if two nodes should be able to communicate with each other they have to be on the same bus. We will later start the runtime environment on the simulation bus id, which is stated to 999. The environment variable \$PWR_BUS_ID controls which bus the runtime is started at, and this is configured in the file /etc/proview.cnf.

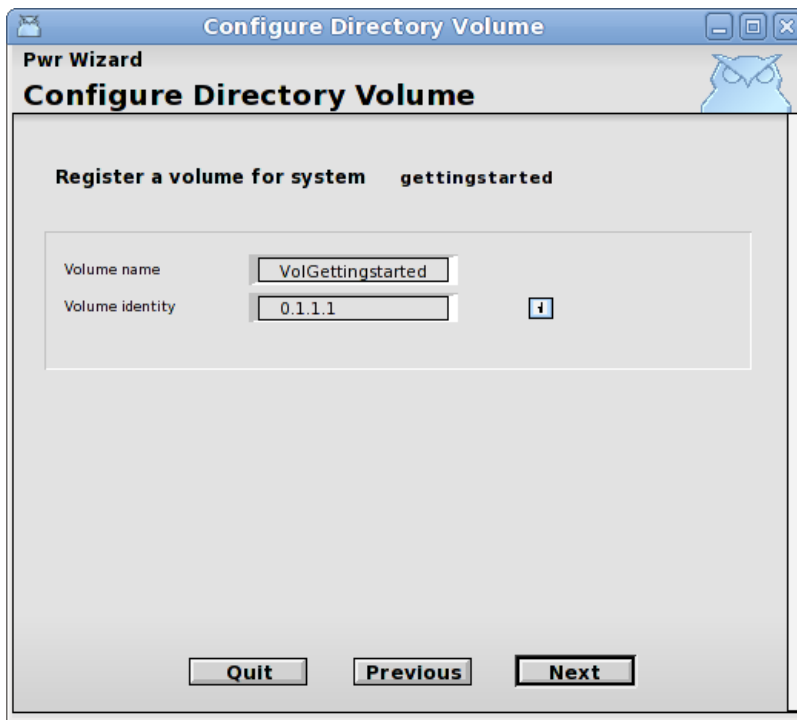


Fig Directory Volume Wizard 2

Press Next

Comment

This will register the volume VolGettingstarted in the GlobalVolumeList. In this volume we will later configure our process station.

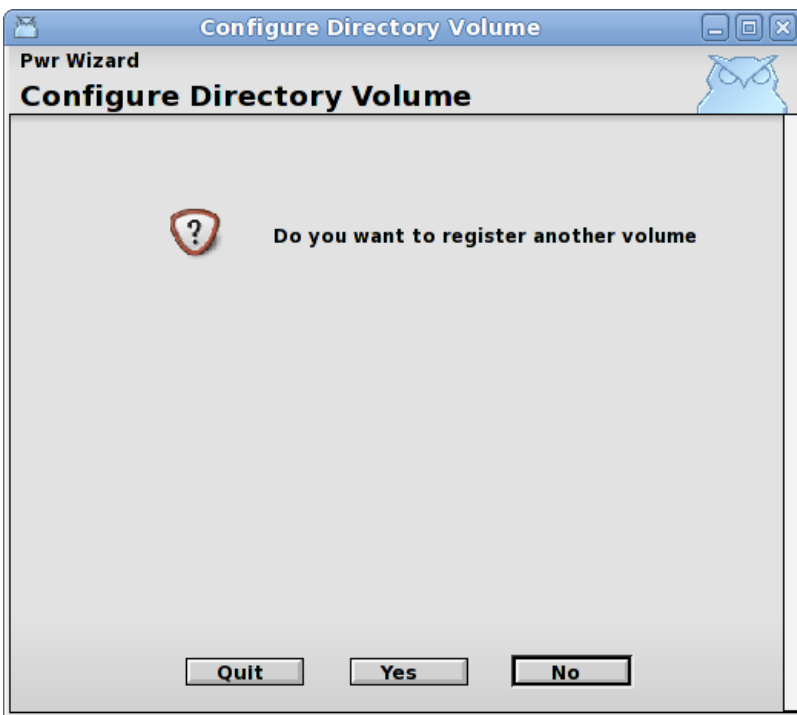


Fig Directory Volume Wizard 3

Press No

Comment

If we have more than one node (process, operator or storage station) in our project we should register one volume for each node. The standard name for a root volume is the nodename with the prefix 'Vol'. For our development and simulation nodes we don't register any volumes. They will load a volume for one of the process, operator or storage stations in the project.

We can also register a classvolume if we need to create specific classes in this project. The standard prefix for classvolumes is 'CVol'.

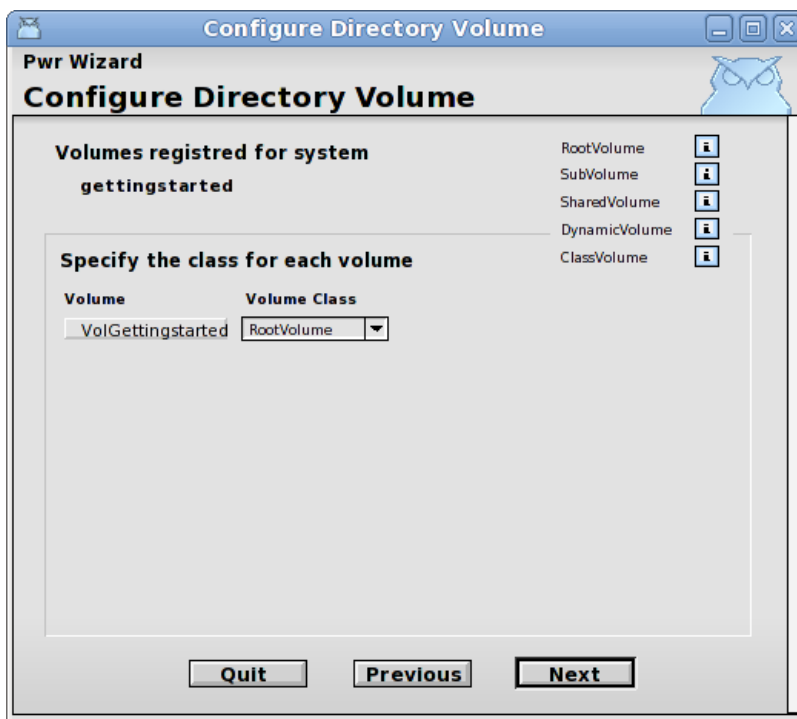


Fig Directory Volume Wizard 4

Press Next

Comment

In this page the type of the registered volumes are specified. In our case we want a root volume which is the default. A root volume is a volume that usually contains all the objects of a node, and is loaded into the node when the runtime environment is started.

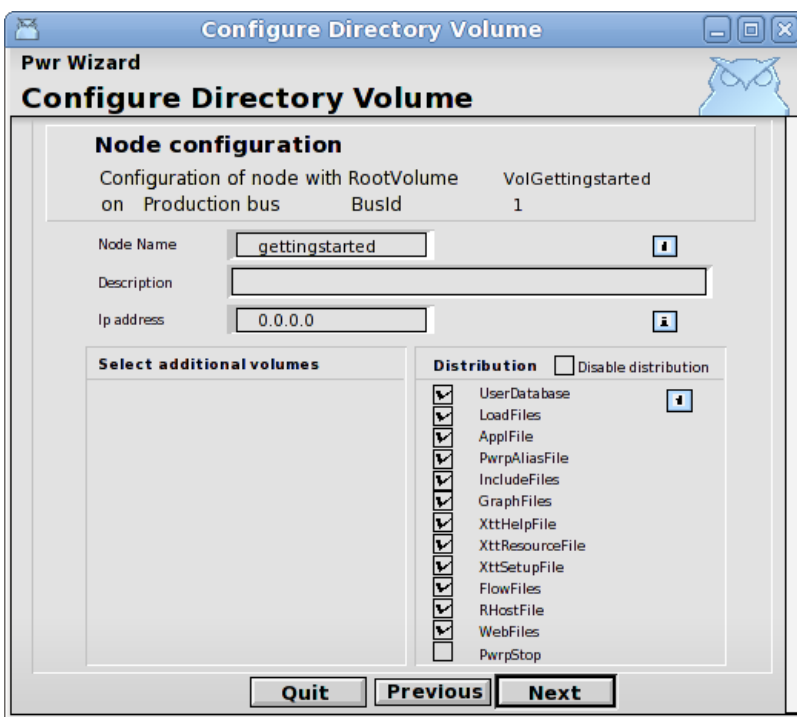
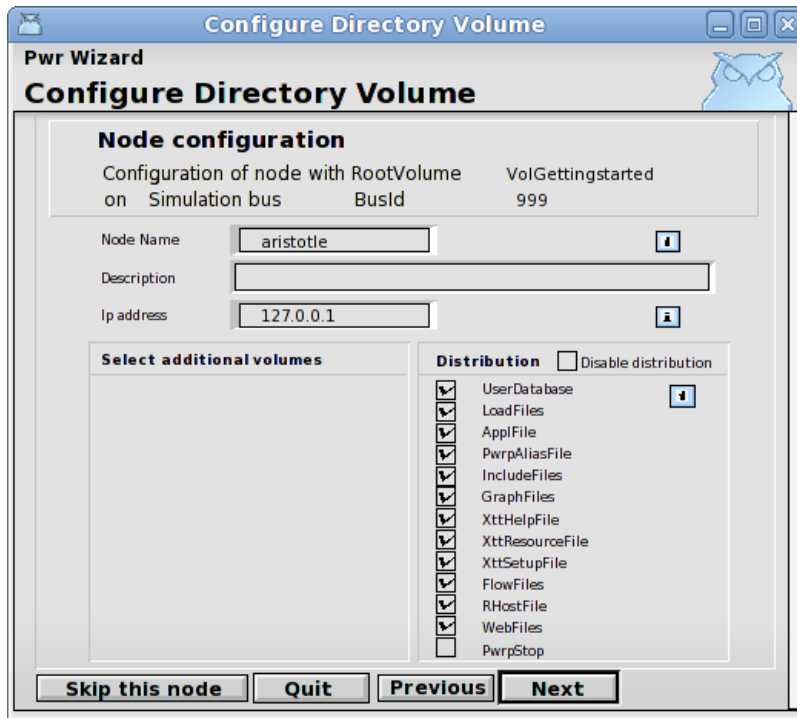


Fig Directory Volume Wizard 5

Press Next

Comment

This is the configuration of the process station. Normally we would enter a valid IP-address, but in our case, we are never going to start this node, instead we will start the simulation node which is configured on the next page.



Comment

This is the configuration of the current node, on which we later will start the runtime environment. Note that the default IP address is the loopback address 127.0.0.1, which will work as long as we don't have any communication with other nodes.

Fig Directory Volume Wizard 6

Press Next

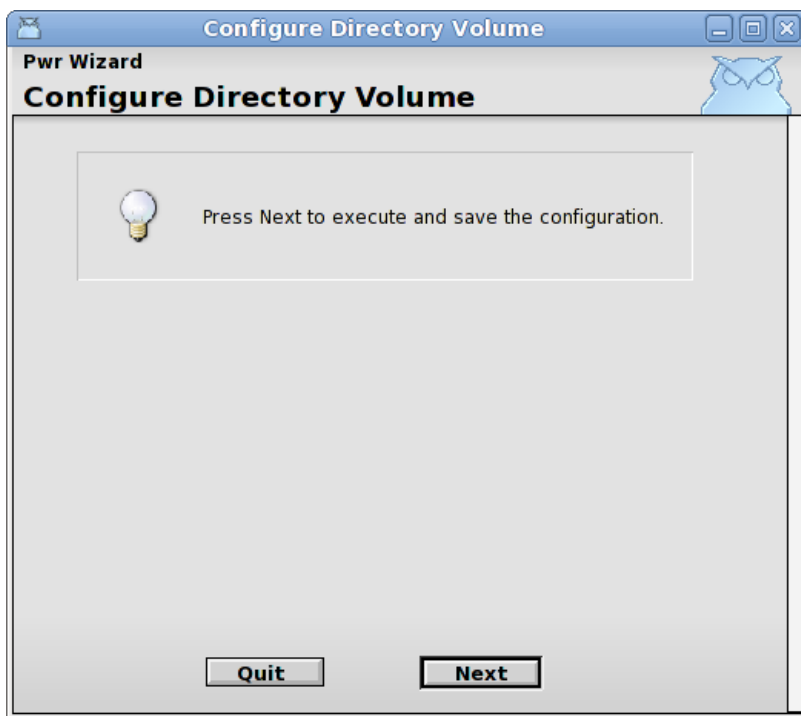


Fig Directory Volume Wizard 7

Press Next

Now the configuration object for volumes and nodes are created and stored in the directoryvolume. Let's take a brief look at the result.

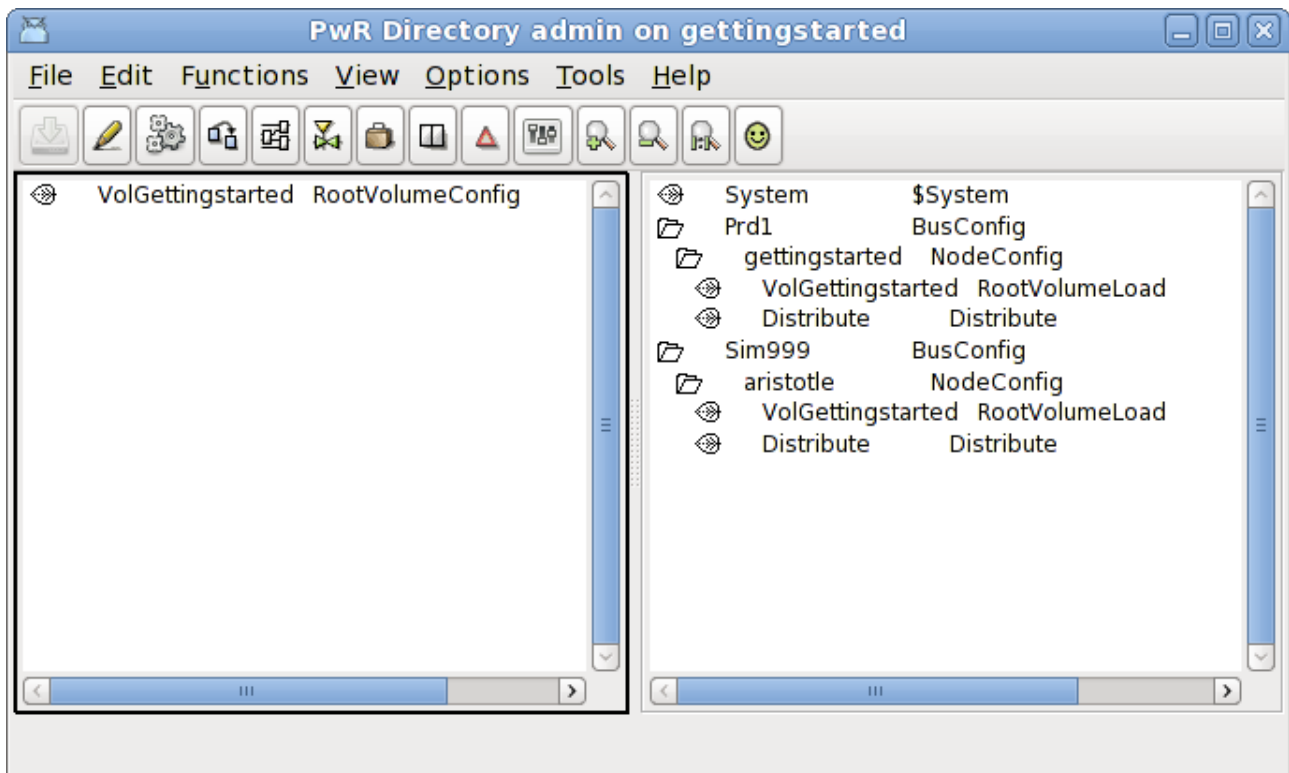


Fig Directory Volume

The view is separated into two windows, the left with the configured volume(s) and the right with the configured nodes. Under the production bus Prd1 the production node is configured and under the simulation bus Sim999 the current node is configured as a simulation node. Under each node a root volume is stated with a RootVolumLoad object, specifying which volume is loaded at Proview runtime startup. There is also a System object containing the systemname and systemgroup. If we later want to add volumes or nodes to the project, we enter edit mode, and create additional VolumeConfig and NodeConfig objects.

If we return to the wizard, there is still one remaining page.

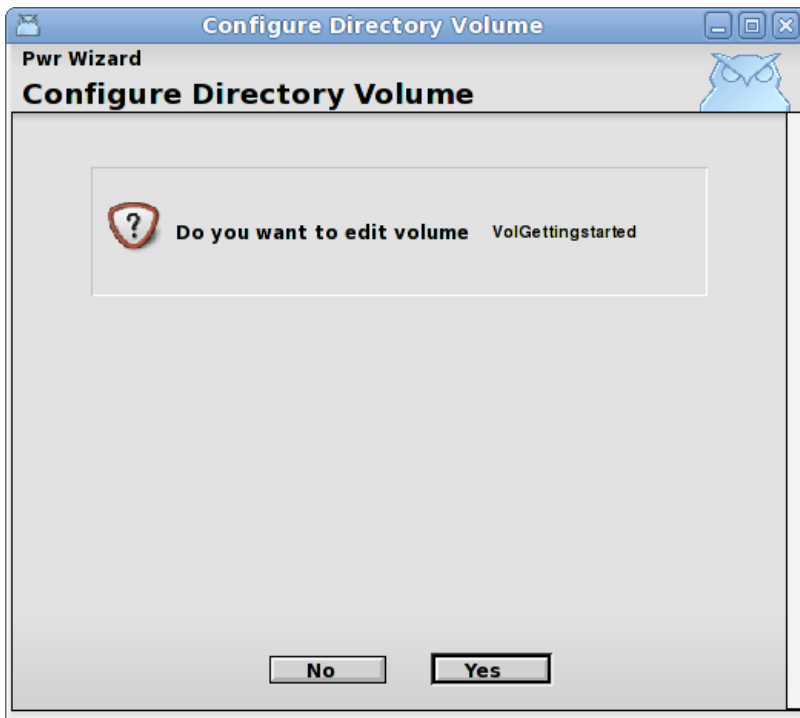


Fig Directory Volume Wizard 8

Press Yes

Comment

We will now open the root volume VolGettingstarted where the configuration of our process station is made.

Tip

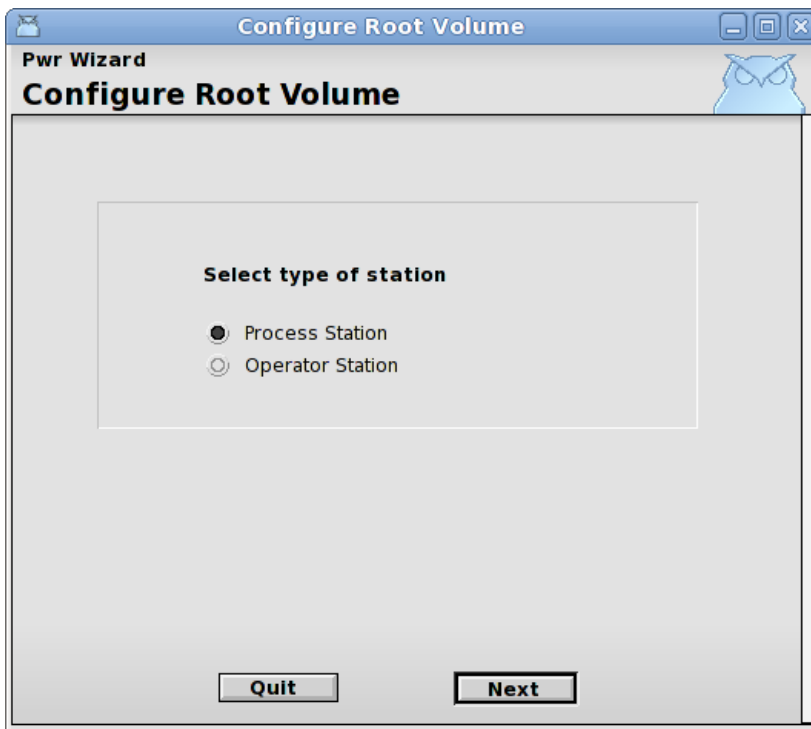
This volume can also be opened by rightclicking on the RootVolumeConfig object in the DirectoryVolume, and activating Open Project in the popupmenu.

It can also be opened from a terminal window with the command (do sdf to the project first)

```
> pwrs volgettingstarted
```

Configure the Process station

Opening an empty rootvolume will start the RootVolumeWizard



Comment

The process station contains also the functionality of the operator station (and storage station) so we will be able start the operator environment as well.

Fig Root Volume Wizard 1

We will configure a process station, press Next.

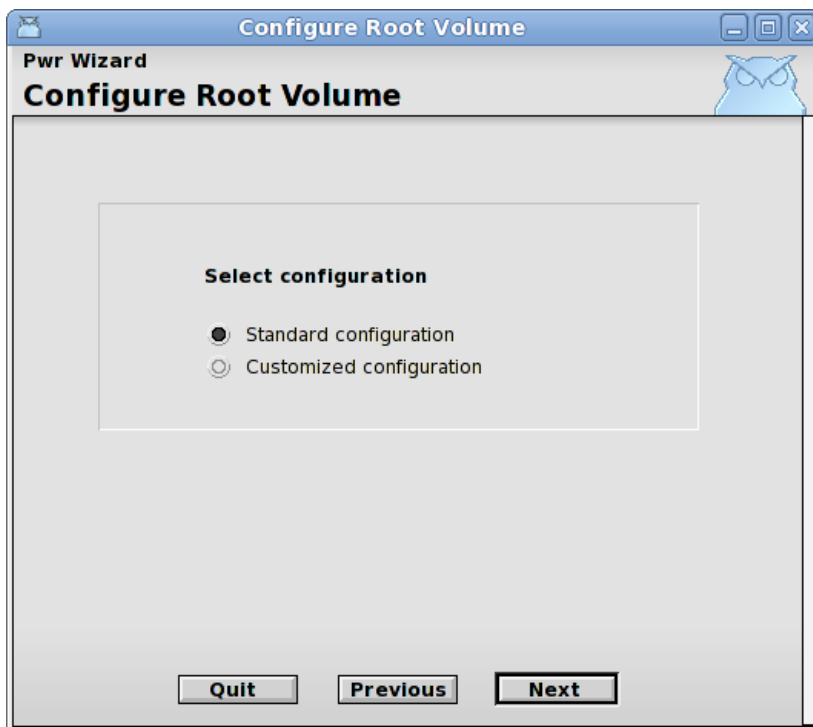


Fig Root Volume Wizard 2

Use the standard configuration and press Next.

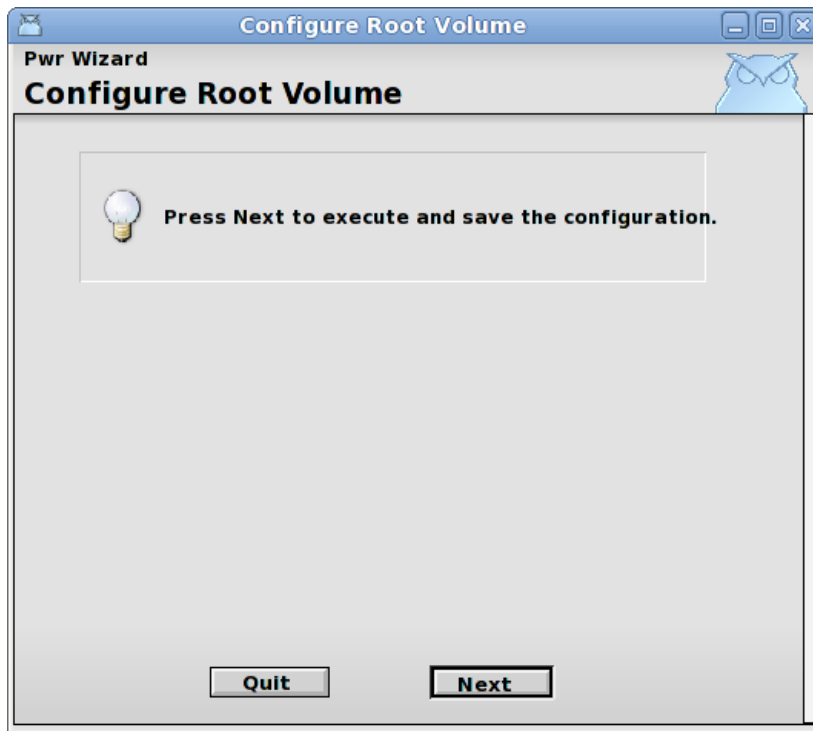


Fig Root Volume Wizard 3

Press Next to save the configuration.

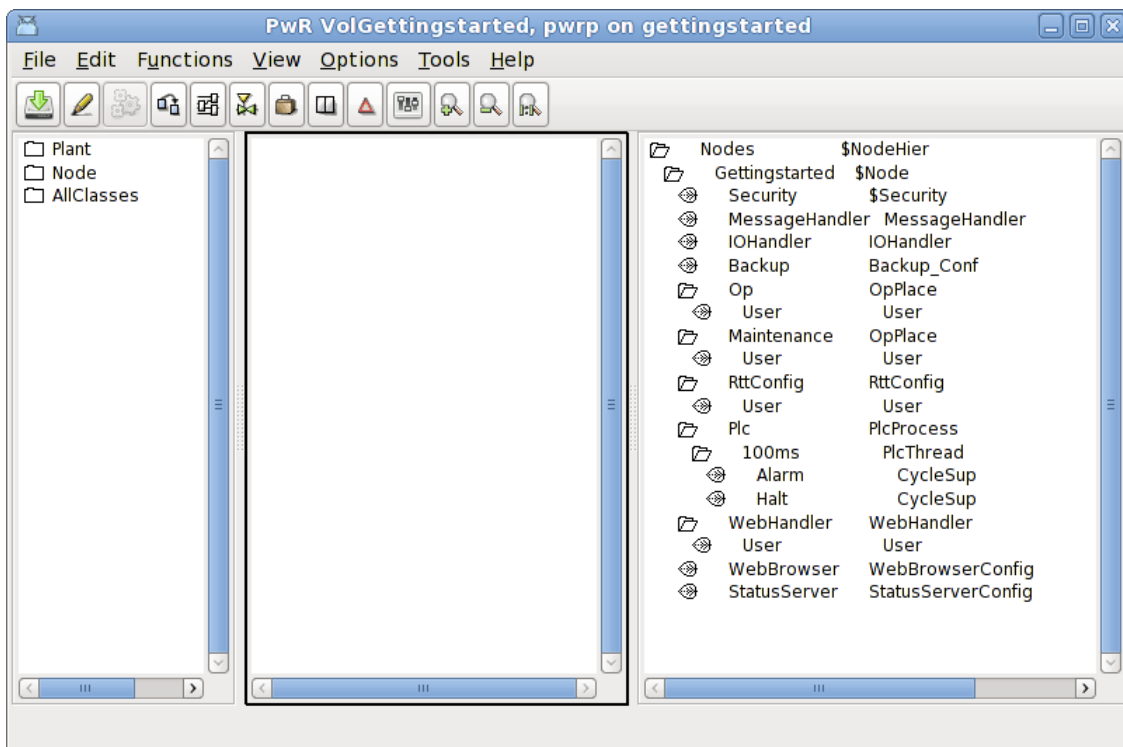


Fig VolGettingstarted initial configuration

Let's have a look at the result in the root volume VolGettingstarted window. The Configurator is separated into two windows, the Plant hierarchy to the left and the Node hierarchy to the right. Note that we already are in edit mode, and the palette is visible to the left.

The node hierarchy contains configuration objects for the hardware and for system processes, and is already configured to some extent, all the objects needed to run our simulation is there.

The plant hierarchy though, is empty. It will contain objects for the signals, sensors and components in the plant. We will also add code from the plc program in the plant hierarchy.

Create a simple Plc program

If you have followed the guide, the Configurator for the root volume, VolGettingstarted, is now open, and in edit mode.

Tip

To open the root volume again, login as pwrp, doubleclick on the Proview icon to open the ProjectList, find the GettingStarted project, rightclick on it and activate Open Project in the popup menu. This will open the directory volume. Here rightclick on the VolGettingstarted object in the left window, and activate Open Volume in the popup menu. Now the root volume will open. Enter edit mode with Ctrl+E.

- Create the hierarchy H1 in the Plant hierarchy: open the Plant map in the palette to the left, and select \$PlantHier. Middleclick in the Planthierarchy window (the left) and a \$PlantHier object is created. Select the object, and press the Arrow right key to open the object, and set the name to H1 (select ObjectName and use Arrow right key again to open the input field). Close the object with Arrow key left.
- Create four digital value objects (Dv): open the Signal map in the palette and select Dv. Middleclick on the leaf of the H1 object to create a child object.. Continue to create three additional Dv objects and name the Dv objects Dv1, Dv2, Dv3 and Dv4.

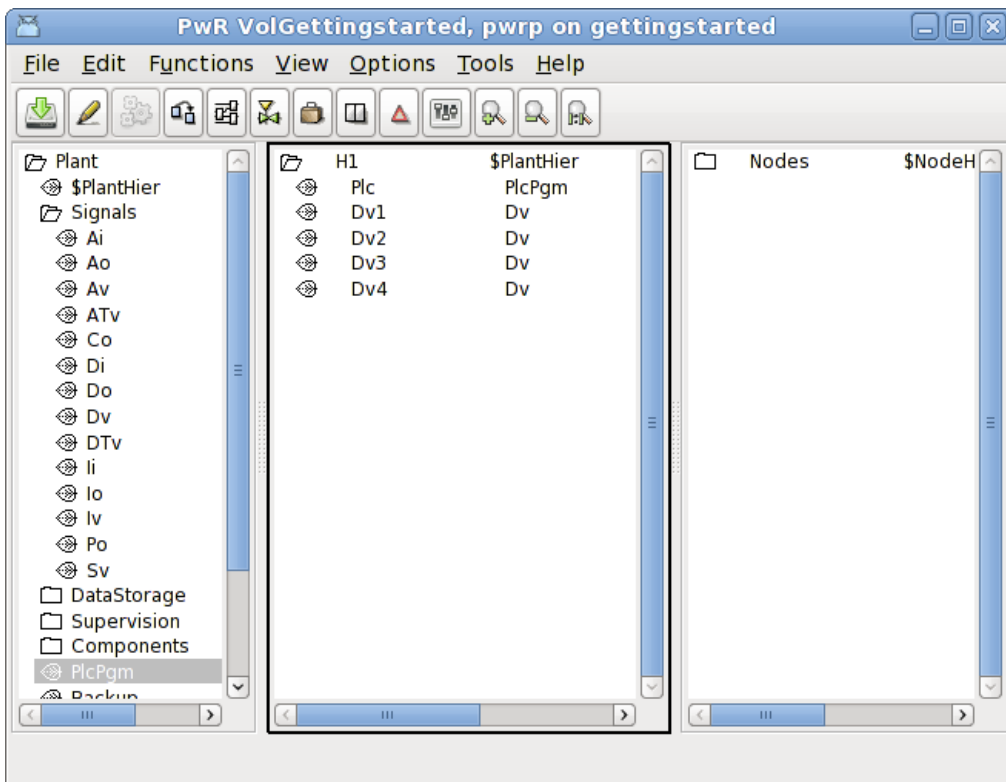
Tip Arrow up when the input field is open recalls the last inserted values.

- Create a PlcPgm object: select PlcPgm in the palette. Create the object as child to H1 and name it Plc. Set also the description to 'A simple program'. Note that the description is displayed on the configurator.

Tip Most objects have a description field. It's usually a good idea fill it in this.

- Save (Ctrl+S) and leave edit mode (Ctrl+E)

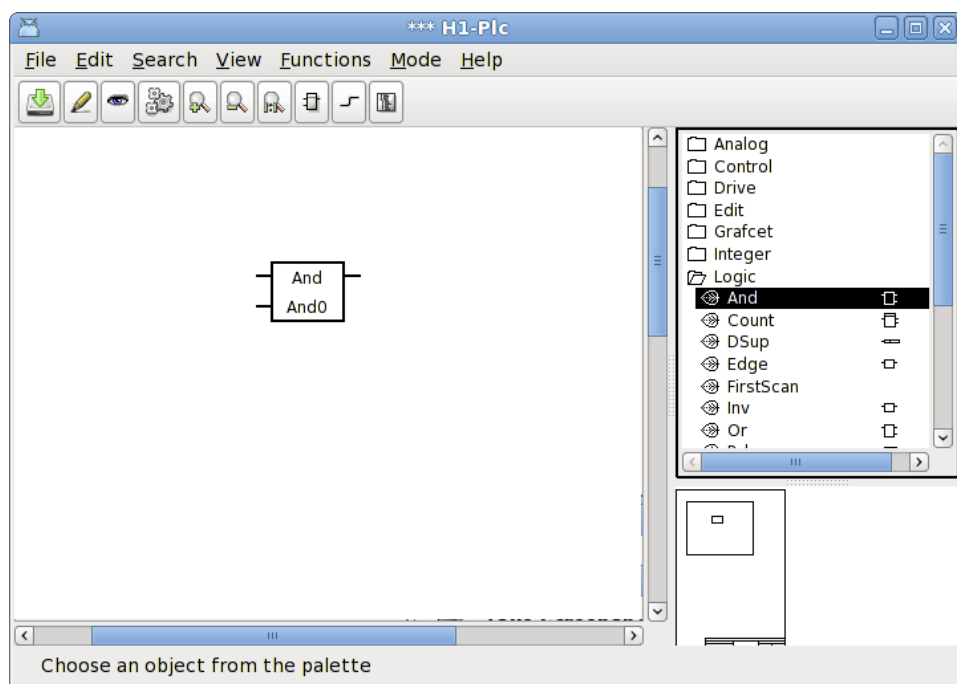
Fig Dv and PlcPgm configured



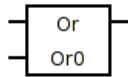
Now its time to open the plc editor. Rightclick on the Plc object and activate Open Program. Note that you have to leave edit mode to open the plc editor.

The Plc editor contains a working area with a palette and navigation window to the right. We will create an and gate, an or gate and a Wait (delay) function object, and we will use the Dv's we have created as inputs and output in the logical schema.

- Create an and gate by opening the Logic map in the palette and select And. Move the cursor to the a position in the working area where you want to place the and gate and click there. The mousebutton should not be pressed when moving the cursor.

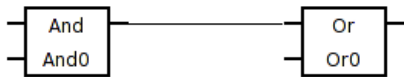


- Create an or gate by selecting Or and move the cursor to a position in the working area.

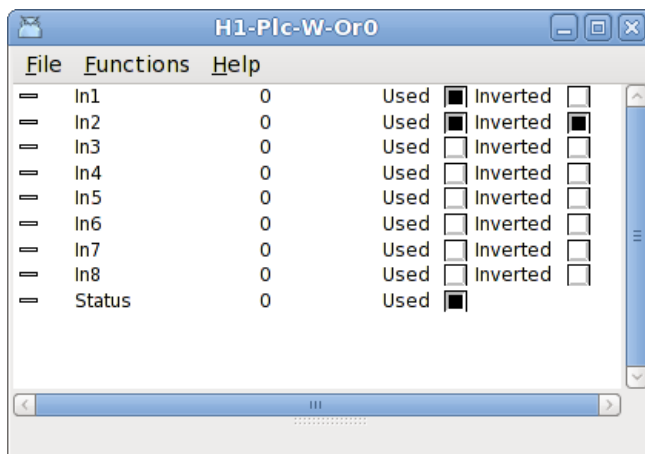
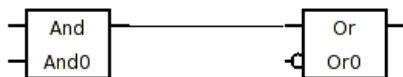


- Connect the output of the And to the first input of the Or by dragging with the middle button from the And output pin to the Or input pin.

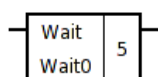
Tip It can be a little hard to hit a pin. Note that also the area inside the pin is sensitive.



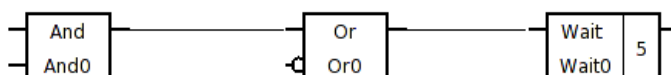
- We will now invert the second input of the Or gate. Rightclick on the Or gate and activate ObjectEditor in the popup menu. From the object editor we can add new inputs and invert them. Click on the Invert checkbox for In2 to invert the second input. Then close the object editor.



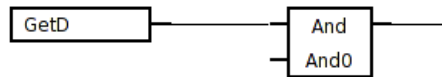
- Create a Wait object by selecting Wait in the palette and place the object in the working area. Open the ObjectEditor (from the popup menu) and enter the delay time in TimerTime to 5 (select TimerTime and press the Arrow right key to open the input field). Close the object editor. Note that the delay time is displayed in the Wait function object.



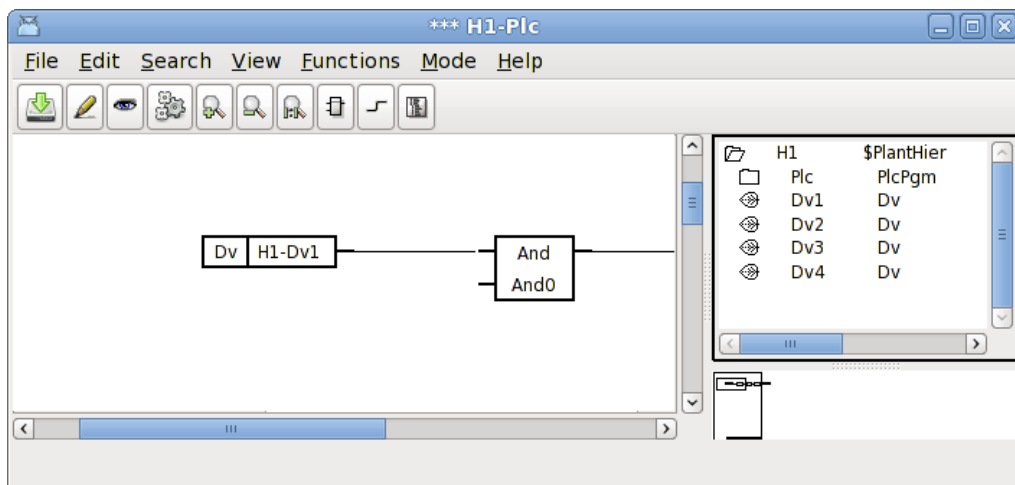
- Connect the output for the Or gate to the input of the Wait object.



- Next step is to connect the Dv's as input and output to the logical schema we have created. Create a GetDv object, which fetches the value of a Dv object, by dragging with the middle button from the input of the and object (as you created a connection before), and releasing the button in some empty space in the working area. This will create a GetD object that can be connected to the Dv.

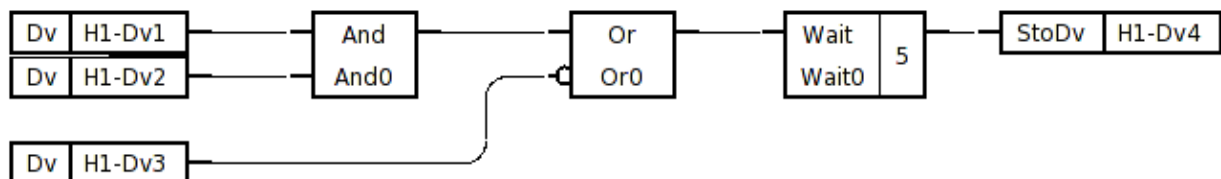


Activate View/Palette/Plant in the menu to replace the palette with the plant hierarchy. Select Dv1 and activate Connect in the popup menu for the GetD object.



- Continue to connect the other inputs to Dv2 and Dv3, and the output to Dv4.

Tip You can also use Ctrl+LeftDoubleClick to connect to the plant hierarchy.



Our program is now complete. Save and close the plc editor.

Build

Return to the VolGettingstarted configurator, and activate Functions/Build Node in the menu.

Select the current node, which is our simulation node (not the gettingstarted node which is our nonexistent process station).

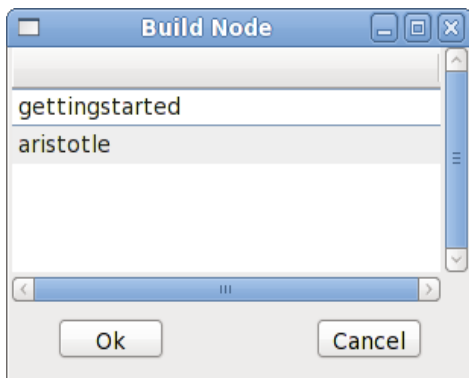


Fig Build node selection

Start runtime

It now time to start the runtime environment. Activate Tools/Runtime Monitor to open the runtime monitor. Click on the 'Start Runtime' button and wait for the 'Down' label to switch to 'Running'.

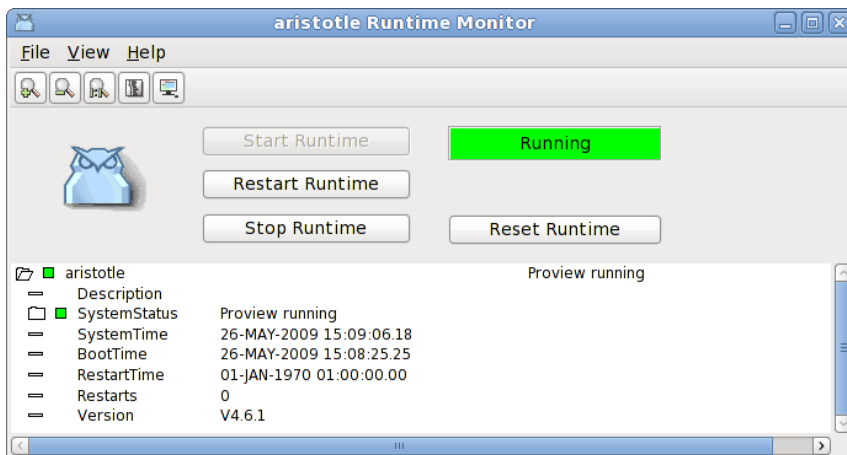


Fig Runtime Monitor

Tip

The runtime environment can also be started from a terminal window with the command (don't forget to do sdf to the project first)

```
> rt_ini &
```

and stopped with the command

```
> . pwr_stop.sh
```

If you have configured another Bus identity than the default (999) you have to set this to the environment variable PWR_BUS_ID

```
> export PWR_BUS_ID=998
```

This can be inserted in the login script \$pwrp_login/login.sh.

You can also change the default busid in /etc/proview.cnf.

Examine the runtime environment

The runtime environment is viewed by the runtime navigator (Xtt). It is started from File/Runtime Navigator in the runtime monitor menu.

Tip

You can also start it from a terminal window with

```
> rt_xtt
```

Under the Database map you will find the object tree with the H1 hierarchy and the PlcPgm we created.

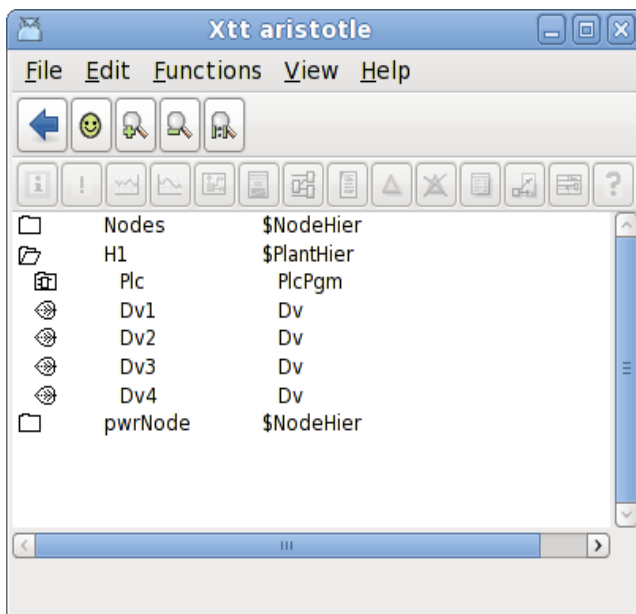


Fig Runtime Navigator Xtt

Rightclick on the Plc object and activate Open Plc in the popup menu to open the program. Now the logical schema is viewed in trace mode, with objects with high status marked with red.

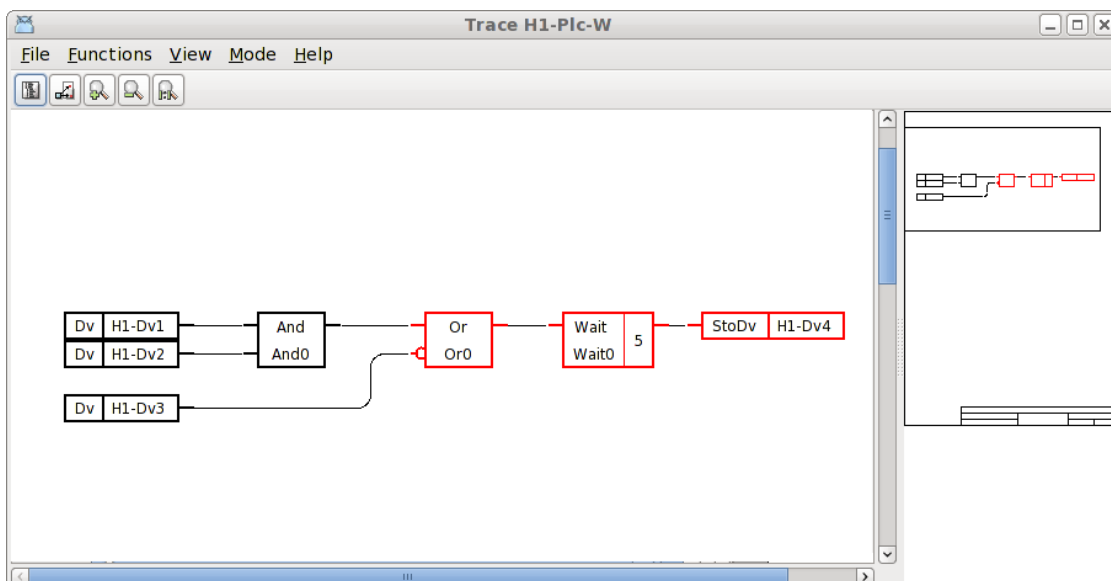


Fig Trace for the plc code

To change one of the input values, for example H1-Dv3, rightclick on the object and activate Object Graph in the popup menu.

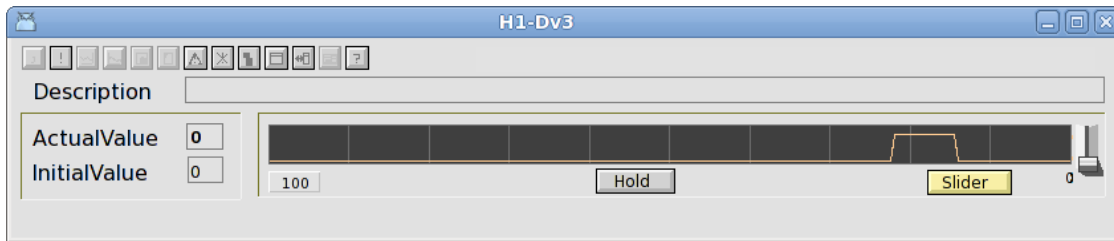


Fig Object graph for H1-Dv3

Click on the Slider button to enable the slider to the right and change the value to high with the slider. Notice how the highlight of the Or gate and following objects are removed. Change the value to low and check that the wait object gets high after 5 seconds. Check the program by setting and resetting the other Dv values as well.

Tip

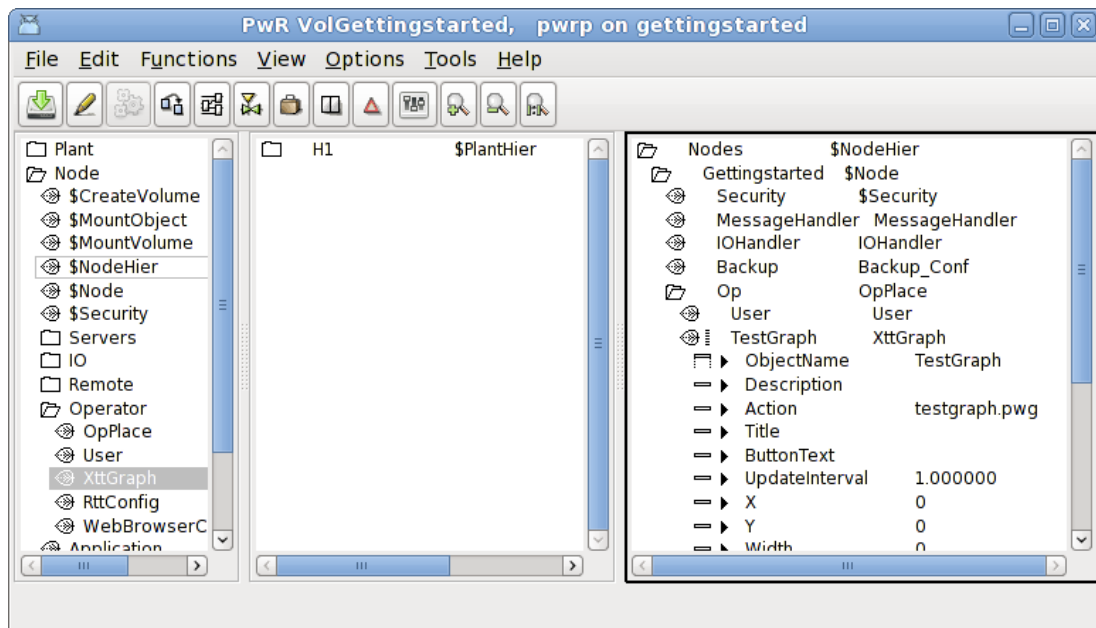
If you change the mode to Simulate (Mode/Simulate in the plc trace menu) you can toggle the Dv values by clicking with Ctrl+Shift+LeftDoubleClick on the function objects.

Plant Graphics

We will now draw a graph with some pushbuttons and indicators, and connect them to our Dv objects.

- Enter edit mode in the Configurator for the root volume VolGettingstarted (Ctrl+E) and create a XttGraph object in the node hierarchy under the OpPlace object Nodes-Gettingstarted-Op. You will find XttGraph in the palette under Node/Operator. Select XttGraph in the palette and middleclick on the User object below the Op object. Open the new XttGraph object (select the object and press the arrow right key) and set the name to TestGraph.

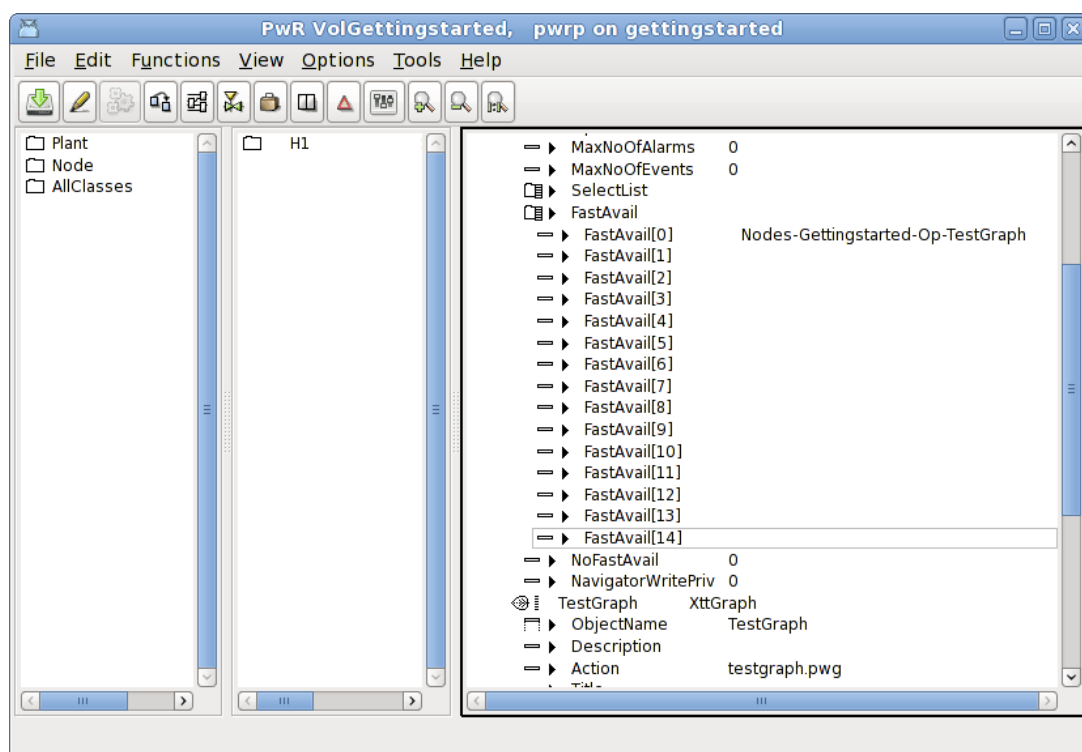
Fig XttGraph object



Tip

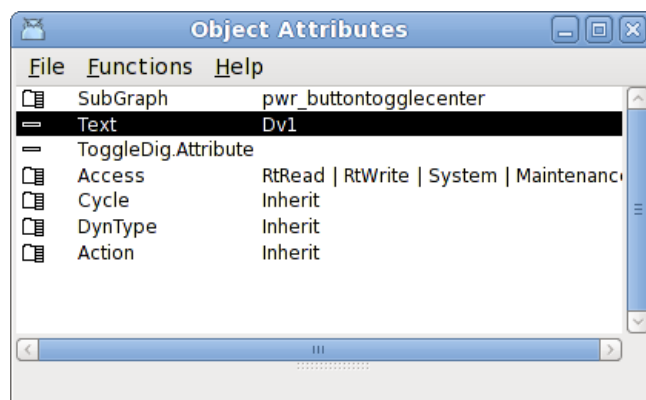
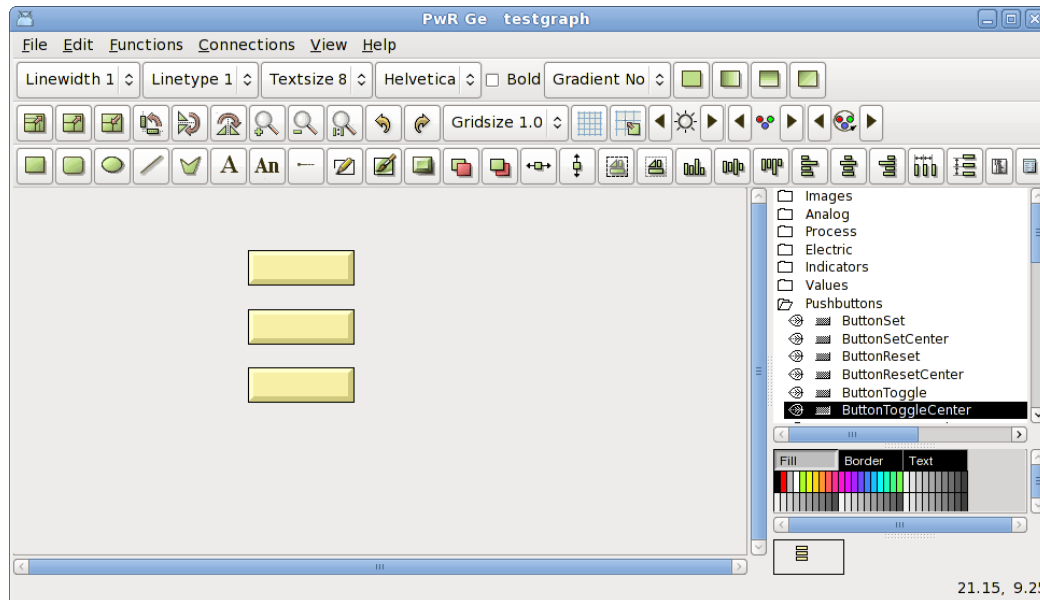
Note that the Action attributes is set to testgraph.pwg. This is the name of the graph file in the catalog \$pwrp_pop.

- To be able to open the graph from the operator window, open the User object, that is a sibling to the TestGraph object. In the User object there is in an array, FastAvail, with 15 elements. Every element corresponds to a pushbutton in the operator window. Open the array with the arrow right key. Insert the TestGraph object into the first element of the array by selecting TestGraph, rightclicking on FastAvail[0] and activating *Insert Selected Object* in the popup menu.



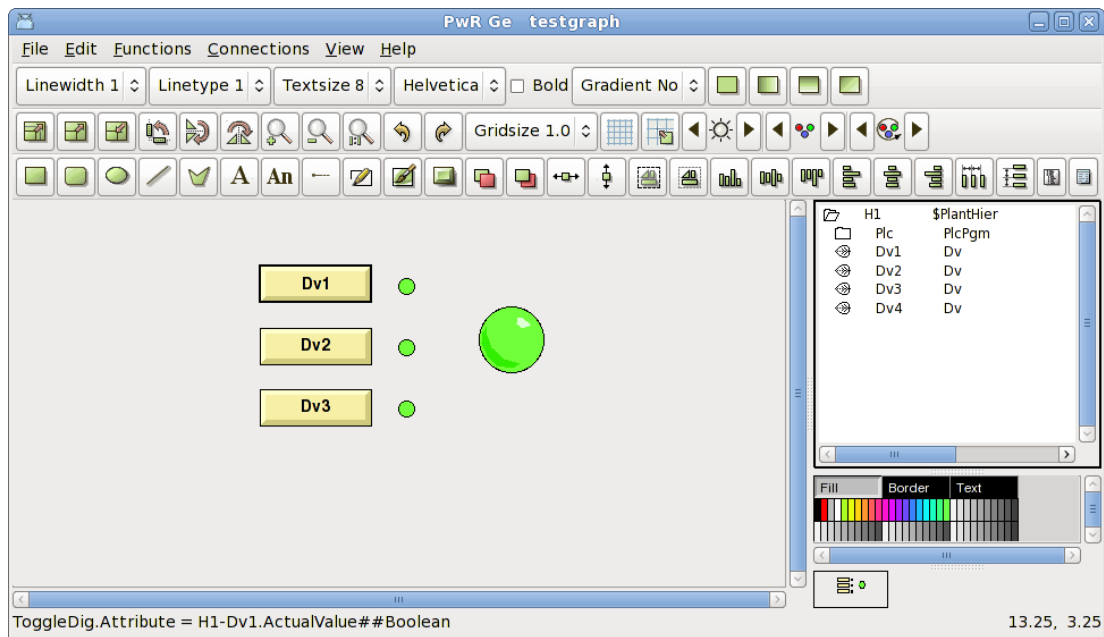
- Save (Ctrl+S) and leave edit mode (Ctrl+E).

- Rightclick on the TestGraph object and activate Open Ge in the popup menu. This will open the graphical editor. The graphical editor contains a working area and a palette and navigation window to the right.
- Create three pushbuttons to toggle the values of Dv1, Dv2 and Dv3. Open the Pushbutton map in the palette and select ButtonToggleCenter. Middleclick in the working area and create three pushbuttons. Doubleclick on a pushbutton to open the attribute editor for the push button. Set the Text attribute to Dv1, Dv2 and Dv3 for the buttons.



Note the ToggleDig.Attribute attribute, this is where we will insert the name of the Dv later.

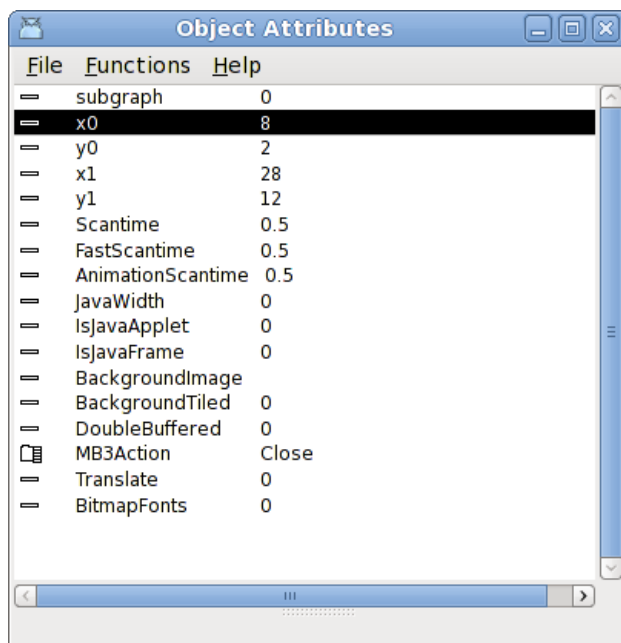
- Create also three indicators to show the values of Dv1, Dv2 and Dv3. Open the Indicator map in the palette and select IndRound. Create three indicators beside the pushbuttons by middleclicking in the working area.
- Create a large indicator for Dv4 by selecting IndRoundLarge in the palette and middleclicking in the working area.
- Now its time to connect the pushbuttons and indicators the the Dv objects. Activate View/View Plant i the menu to replace the palette with the plant hierarchy. Open the H1 map and select Dv1. Select the Dv1 pushbutton also and activate Function/Connect (Ctrl+Q) in the menu. Connect also the small indicator to Dv1, and the corresponding pushbuttons and indicators to Dv2 and Dv3. Finally connect the large indicator to Dv4.



Tip

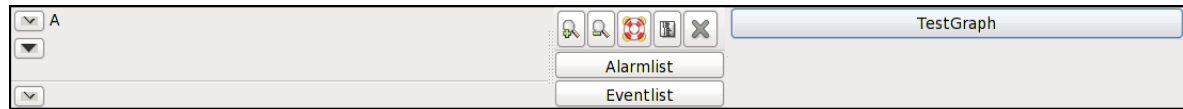
You can also connect by selecting the Dv in the planthierarchy and click with Ctrl+LeftDoubleClick on the pushbutton or indicator.

- We should also set the borders of the graph. Activate File/Graph Attributes in the menu to open the attributes for the graph. Measure what you want to be the upper left corner or the graph by placing the cursor there and read the coordinates in the lower right corner of the editor. Insert the coordinates in x0 and y0. Measure the lower right corner and insert these coordinates in x1 and y1.

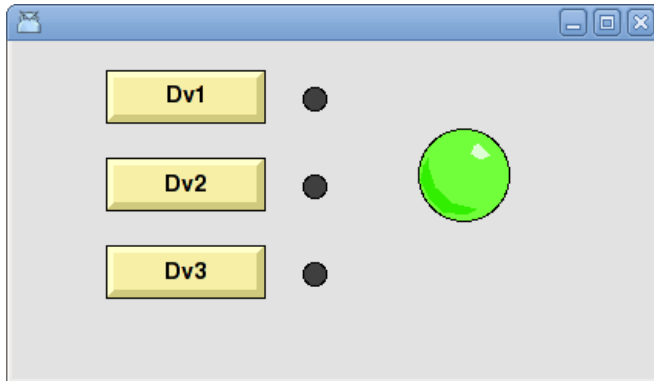


- Save the graph and close Ge.
- Build the node from the VolGettingstarted Configurator (Functions/Build Node in the menu).

- Open the [runtime monitor](#). Stop runtime and start it again.
- Activate File/Start Operator Environment in the runtime monitor menu.
- Select Nodes-Gettingstarted-Op operator place. The operator window is now opened with the TestGraph button to the right.



- Click on the TestGraph button to open the graph.



- Click on the pushbuttons and check that the indicators are responding.

Tip

Another way to open a graph is to start Xtt from a terminal window

```
> rt_xtt
```

Open the command line in xtt with Ctrl+B and enter the command

```
xtt> open graph testgraph
```

Tip

A XttGraph object is not necessary to open a graph. Open Ge from the Configurator menu, edit and save the graph. Then copy the graph file from the catalog \$pwrp_pop to \$pwrp_exe

```
> cp $pwrp_pop/testgraph.pwg $pwrp_exe
```

Start xtt and open the graph as described in the tip above.

Appendix

Glossary

ProjectList	A list of all projects. Projects are ordered in a tree structure and defined with ProjectReg objects. Projects can be opened from the ProjectList and this is also where new projects are created.
GlobalVolumeList	A list of all volumes in all projects. All volumes have to be registered in this list.
Volume	A container for objects. There are a number of different types of volumes. A rootvolume, for example, is loaded into a process or operator station at startup and contains all objects needed to configure the station.
DirectoryVolume	Every project contains one directory volume, that configures the nodes and volumes in the project.
Root Volume	The root volume is loaded into a process or operator station at Proview startup. It usually contains all the configuration objects for this node.
Object	An object contains a data area divided in attributes. The attributes store data configured in the Configurator, or data calculated in runtime. An object can also have methods, object graphs etc. Every object is specified by a class, where the attributes, methods, plc code etc. are defined. Objects are placed in an object tree, where objects can have parents, siblings and children.
Plcprogram	A control program built by logical and analog functionblocks.
Configurator	A tool to navigate in the object tree and to configure the.
UserDataBase	A database where Proview users are defined with username, password and privileges. Contains by default the systemgroup Common with users pwrp and b55. A project links to a systemgroup and gets the users for this group.
Project	A number of processstations, operatorstations and storagestations are gathered into a project. It contains a directory tree with databases and configuration files for the nodes and volumes that belongs to the project.
sdf	A shell command to set up the environment for a project (set default project) > sdf 'projectname'
Plant hierarchy	Configures the plant with objects for signals, sensors and components. Also PlcPgm's containing plc code is configured in the plant hierarchy.

Node hierarchy	Configures the hardware, IO-system, and system processes of the node.
----------------	---

Edit functions in the Configurator

Function		Action	Comment
Select Object	Mouse	LeftClick	
	Key	Arrow Up and Down	
Create Object	Menu	Edit/Create Object/After (Ctrl+D) Edit/Create Object/First Child (Shift+Ctrl+D)	Select an object class in the palette first, then select the destination object (parent or sibling), and then activate Create Object in the menu.
	Mouse	MiddleClick	Click on the icon (map or leaf) creates the object as child to the clicked object. Click on the text creates the object as sibling.
Delete Object	Menu	Edit/Delete Object (Delete key)	Deletes the selected objects.
	PopupMenu	Delete Object	Select the object first.
Move Object	PopupMenu	Move Selected Object	Select the object you want to move the object to first.
	Mouse	MiddleClick	Select the object you want to move, and middleclick on the destination object. Click on the icon to move as child. Click on the text to move as sibling.
Show Children	Mouse	LeftClick on map icon.	
	Key	Arrow Right.	Note that arrow right key also shows the attributes of the object if no children are present.
Open object	Key	Shift+Arrow Right. Arrow Right (if no children).	The arrow key displays the attribute in the current window.
	PopupMenu	Open Object	Opening from the menu opens a new window.
	Menu	Functions/Open Object (Ctrl+A)	
Change attribute value	Key	Arrow Right.	Enter the value into the input field and press enter.
	Menu	Functions/Change Value (Ctrl+Q)	

Table 1 Edit functions in Configurator

Edit functions in the Plc Editor

Function		Action	Comment
Select Object	Mouse	LeftClick	
	Key	Shift+Arrow Keys	
Create Object	Mouse	LeftClick	Select an object class in the palette with the mouse. A functionobject is now moved with the cursor. Place it with a LeftClick.
		MiddleClick	If a class already is selected, or is selected by the Arrow Up and Down keys, Middelclick in the work area to create an object.
	Menu	Edit/Create Object (Alt+D)	
Delete Object	Menu	Edit/Delete (Delete key)	Delets the selected objects.
	PopupMe nu	Delete	
	Mouse	MiddleDoubleClick	Hit on an object deltes the object. Hit in en empty space deletes all selected objects.
Move Object	Mouse	LeftDrag	If several objects are selected, and a selected objects is dragged, all selected objects will be moved.
	Key	Shift+Control+Arrow keys	
Create Connection	Mouse	MiddleDrag	Drag between two pins. If the connection is dragged from one pin and released in an empty space, a predefined object is created (normally a Get or Sto object).
	Key	Ctrl+D	Select a connection poiont with Ctrl+Arrow keys, lock this with Shift+Ctrl+D, and select a second connectino point with Ctrl+Arrow keys.
Open object	PopupMenu	ObjectEditor	Opening from the menu opens a new window.
	Menu	Functions/Open Object (Ctrl+A)	
Connect	Mouse	Ctrl+LeftDoubleClick	Select an object in the Plant hierarchy.
	Menu	Edit/Connect (Ctrl+Q)	

	PopupMenu	Connect	
Open Subwindow	Mouse	Shift+LeftDoubleClick	
	Menu	Functioniions/Open Subwindow	
	PopupMenu	SubWindow	

Table 2 Edit functions in the Plc Editor