Release Notes V5.3

2015 04 30

Table of Contents

Upgrading to Proview V5.3.0

This document describes new functions i Proview V5.3.0, and how to upgrade a project from V5.2.0 to V5.3.0.

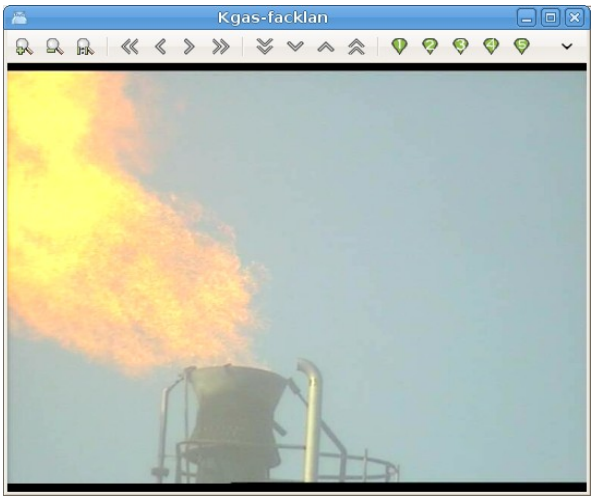
New functions

Softing Profinet stack in user space

The softing Profinet stack is now distributed as an archive. Previously in was distributed as a kernel module that had to be build with the current kernel by Softing. This vastly improves the usability of the Profinet stack for external users as the stack can be installed as an ordinary package.

Streaming video and camera control

A video stream from a network camera can be displayed with the XttCamera object.



The video can be displayed in a separate window or in a MultiView window. It can not be embedded in a Ge graph.

The protocol VAPIX for control of Axis cameras are supported. With buttons in the control panel the camera can be paned, tilted and zoomed. It is also possible to store 10 camera positions.

Camera method

A camera Xtt method for an object can be configured by placing a CameraPosition object under the object. Values for pan, tilt and zoom are specified in the object together with a link to the XttCamera object. When the method is activated the window for the specified camera is opened and the specified pan, tilt and zoom values are transferred to the camera.

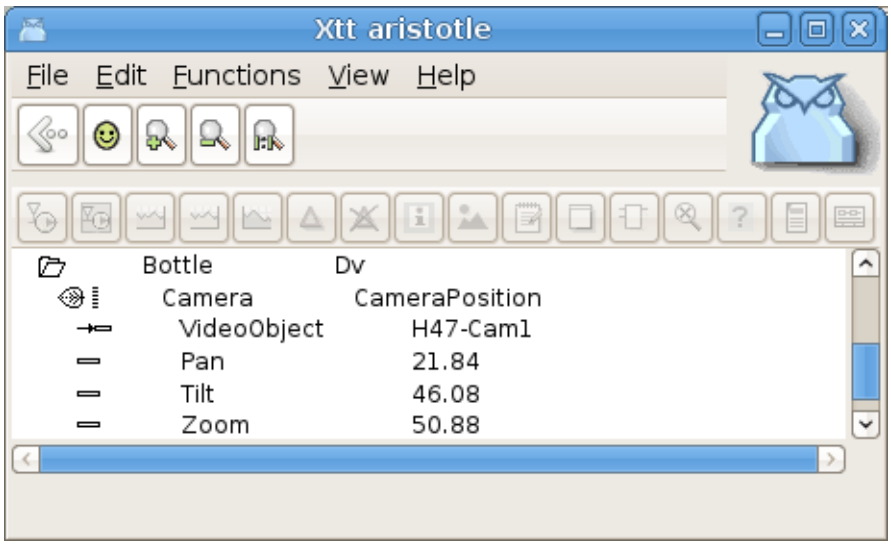
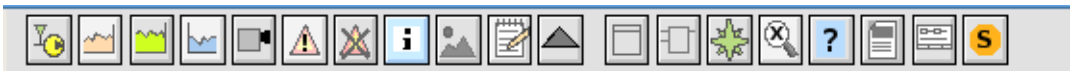


Fig Configuration of camera method

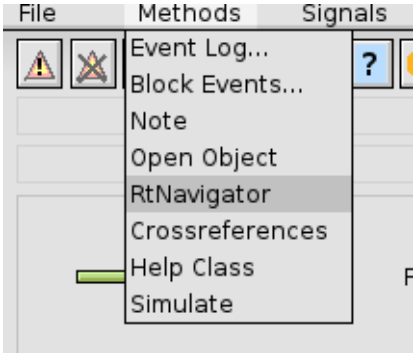
Method toolbar and menu for object graphs



There is a new design for the method buttons, and they are also gathered in a new toolbar component that will only view the configured methods. Configured but not authorized method buttons are dimmed.

There is also a new Ge action for the method pulldown menu, MethodPulldownMenu. The menu only displays the configured methods. There is also a HelpMenu property to only display the help methods, Help and HelpClass, which makes it possibe to use it as the help menu of the object graph.

Signal and components contain a XttMethodMask attribute with information about which methods buttons should be displayed. This can configured in the configurator, or it will be calculated the first time the object graph is opened.



Toolbar in eventlog window

There is a new toolbar in the Eventlog window to zoom in and out and to activate methods for the selected event.

Conditional confirm action for toggle dig

The confirm action in Ge has two new properties, OnSet and OnReset. These can be combined with

ToggleDig where OnSet is true, the confirm window will only be displayed when the toggled attribute is set, and if OnReset is true, the confirm window will only be displayed when the toggled attribute is reset.

Ge dynamics DigFourShift

The dynamic DigFourShift can used for subgraphs with several pages. Three boolean attributes are specified and the dynamics will shift between four pages. New subgraphs using the DigFourShift are TrafficLight2 and TrafficLight3.

Fig TrafficLight2 and TrafficLight3 with DigFourShift dynamics

Release attribute for Slider dynamics

The property ReleaseAttr is added to Slider dynamics. A boolean attribute can be specified in ReleaseAttr that will be set when the slider is released.

Plc object IOSimulFlag

This is a new function object to fetch the IOSimulFlag attribute in the IOHandler object. It can be used in function objects to test if simulation is active.

Components with embedded plc code

Previously components with plc code had to be separated in a main object containing signals and data, and a function object containing the plc code. Now it is possible to create components containing plc code without creating a function object. The plc code is embedded in the component and you just create the component object and the plc code will automatically be generated under the component. The component should contain an PlcEmbed attribute with the plc thread, reset object and execute order. As there is no function object, references to other object has to be done with another mechanism. The component can contain attributes of type AttrRef that points to attributes in other object where values should be fetched or stored in the plc code. In the code, GetRef and StoRef objects are used to fetch and store these values via an attribute reference. This kind of components are suitable for components with a limited interaction with other objects.

Components with embedded plc can be inserted into other component or aggregate classes. In the example below the Burner attribute is an array with 20 component object with embedded plc.

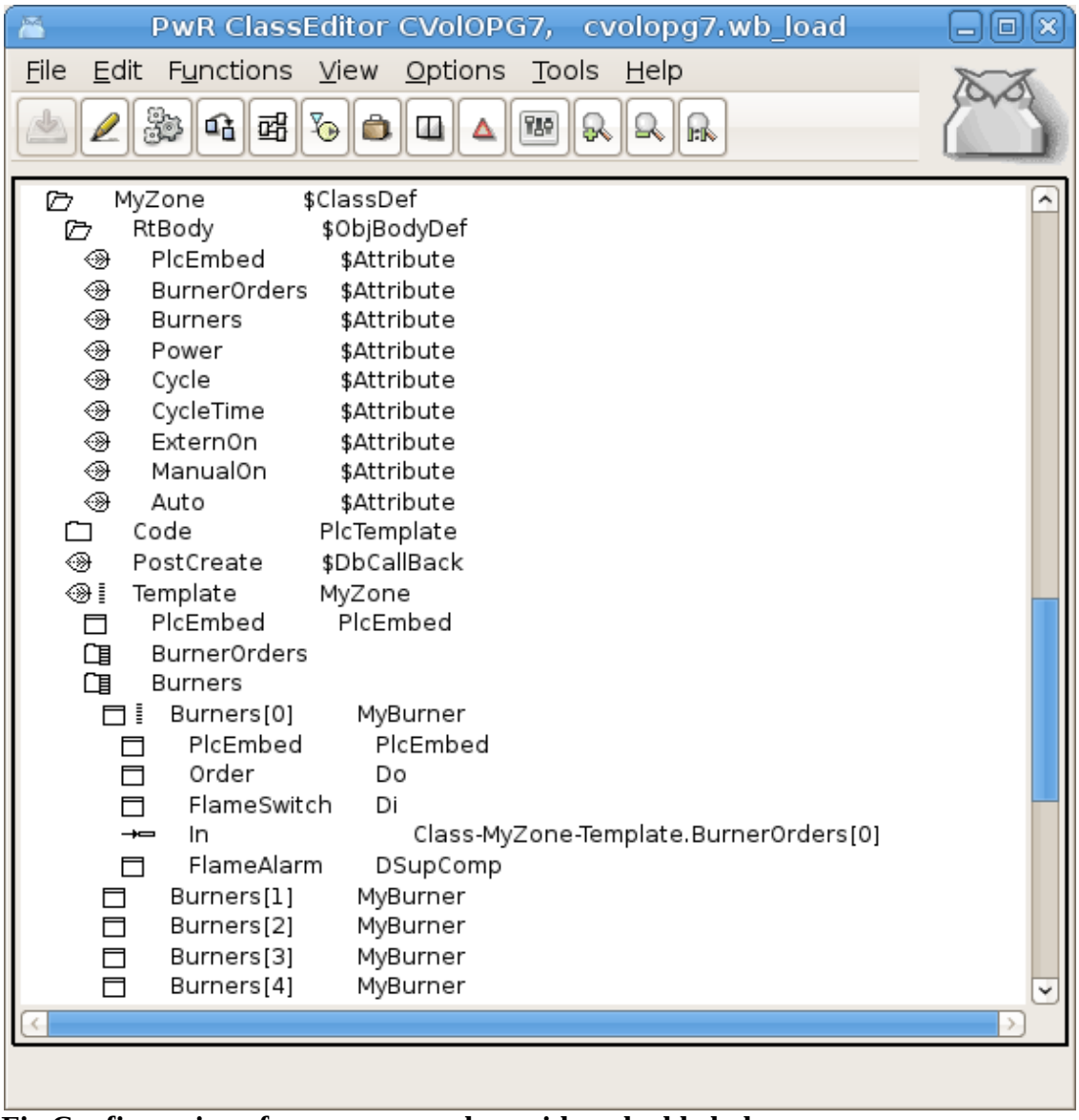


Fig Configuration of a component class with embedded plc

Editor for text and code attributes

A text editor can be opened from the plc editor to edit text and code attributes in BodyText, HelpText, Carithm and DataArithm objects. The editor is opened from EditText or EditCode in the plc editor popuop menu, thus making it easy to edit the text or code without opening the object editor first.

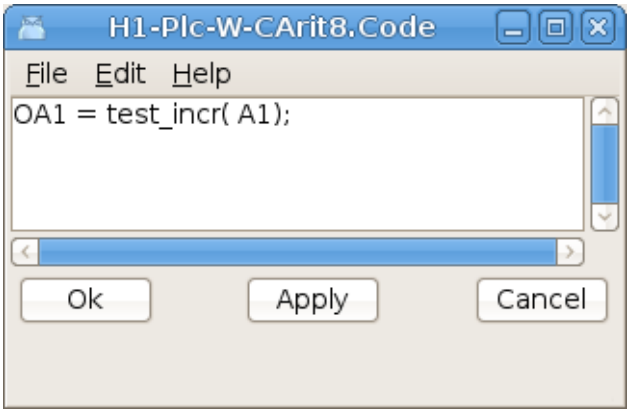


Fig Text and code editor

Plc editing with script

Some new script function has been added to be able create function objects and connections in a plc program with script. When editing a plc session is opened with OpenPlcPgm() and later closed with ClosePlcPgm(). In the plc session function objects are created with CreatePlcObject() and connections with CreatePlcConnection(). Attribute values can be set with SetPlcObectAttr() and function object can be connected with objects in the plant hierarchy with PlcConnect().

New functions

CreateObject()	Create an ordinary object in the planthier.
OpenPlcPgm()	Open a plc program and start an plc editing session.
CreatePlcObject()	Create a plc document or plc function object.
CreatePlcConnection()	Create a plc connection between two function objects.
SetPlcObjectAttr()	Set attribute values in a function object or plc document.
PlcConnect()	Activate the plc connect function.
ClosePlcPgm()	Close the plc editing session.

Script example

```
main()
  int first_child = 1;
  int after = 3;

  #Create some plant object s
  CreateObject("H2","$PlantHier","H1",after)
  CreateObject("Plc2","PlcPgm","IO-MOT1",first_child);
  CreateObject("ObjFan1","MCC2","H2-Plc",after)
  save

  # Open a plc session
  OpenPlcPgm("IO-MOT1-Plc2");

  # Create a document
  CreatePlcObject("Doc01","Document",2.0,2.0);
  SetPlcObjectAttr("Doc01.DocumentOrientation",1);
  SetPlcObjectAttr("Doc01.DocumentSize",3);

  CreatePlcObject("And1","And",1.0,1.0,"Doc01",7,1,5)
  CreatePlcObject("Fan01","MCC_Fo",1.3,1.0,"Doc01")
  CreatePlcObject("GetDi01","GetDi",0,4,1.0,"Doc01")
  PlcConnect("GetDi01","IO-MOT)-INP-R0_C02_DI_A1");
  CreatePlcConnection("GetDi01","Status","And1","In1");
  ClosePlcPgm();
```

```
endmain
```

Ge script update

The Ge script function has been updated and improved. Enumeration names can be used for different kind of enumerations, eg dynamic and action types.

Ge script example

```
create object/x1=1/y1=1/x2=10/y2=10/subgraph=movesub
set current annotation "Close"
set current gradient VerticalTube1
set current attr HostObject.Object H1-Dv1
set current attr dyntype1 DigLowColor|Move|DigColor
set current attr DigLowColor.Color White
set current attr DigLowColor.Color White
set current attr DigColor.Instances 2|3
set current attr DigColor2.Color GreenGreenYellow
set current attr action ToolTip|SetDig|ValueInput
set current attr ToolTip.Text "Some tooltip text..."
set current attr Access Operator1|System
```

Remove several objects with subwindows in plc editor

It's now possible to remove several objects with subwindows at the same time. Previously they had to be removed one by one.

Sev database hdf5, beta release

The database hdf5 is added as an alternative to mysql and sqlite as database in sev. For the moment there is no journal file in hdf5, and the database can be corrupt after for example a power failure. Journaling will be added to hdf5 this year. The advantage of hdf5 is that it requires much less space the sql databases, and data can be fetched much faster.

Attribute flag DevHideValue

Flag DevHideValue for an attribute will hide the value in the configurator. Combined with RtHide it can be used for password attributes.

Hierarchies in class volumes

It's now possible to insert one hierarchy level for $ClassDef objects in a class volume. The hierarchy object should be of class $ClassHier.

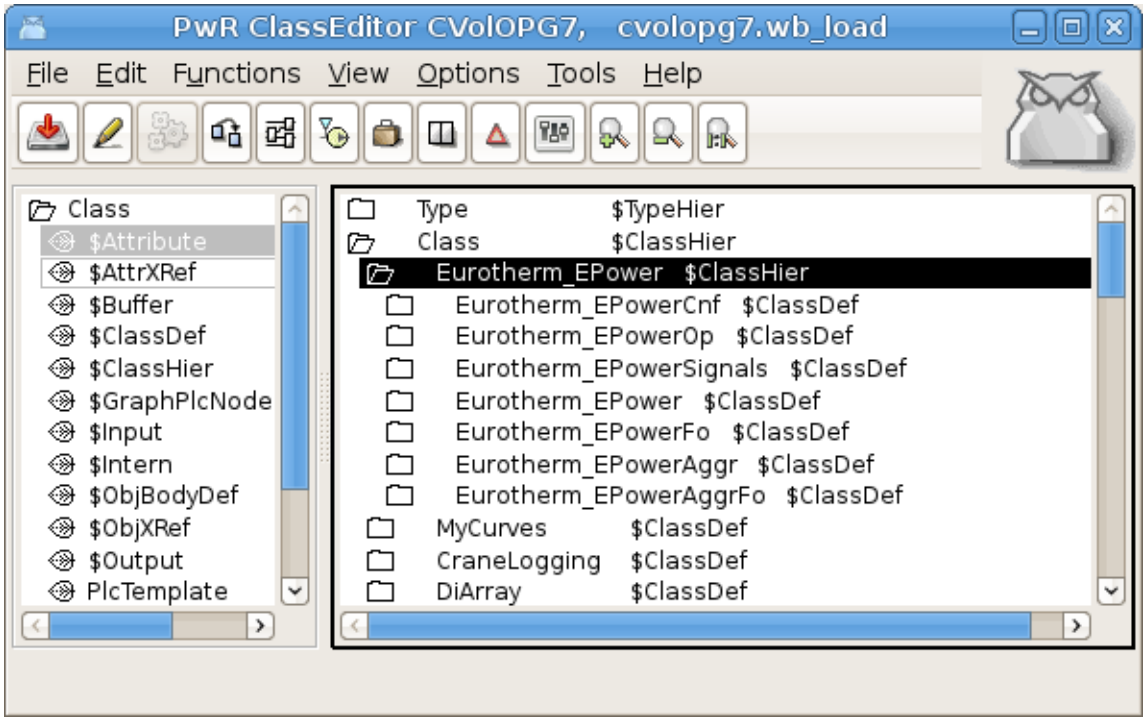


Fig Hierarchy in class volume

Multiview pane and fix layout

Pane and fix layout are added to the MultiView window.

Pane layout means that the border line between two windows can be moved to adjust the size of the two windows. Pane is only supported for MutliViews with two columns and one row.

With Fix layout the coordinates for each cell is specified and the cell positions will not be adjusted if the window size if changed.

Safety level indicator in the configurator

A safety level attribute is added to $Hierarchy, PlcPgm and XttGraph objects. The safety level is an enumeration with the values None, Low and High. Objects with low safety level is marked with a yellow warning triangle, and object with high safety level are marked with a red triangle.

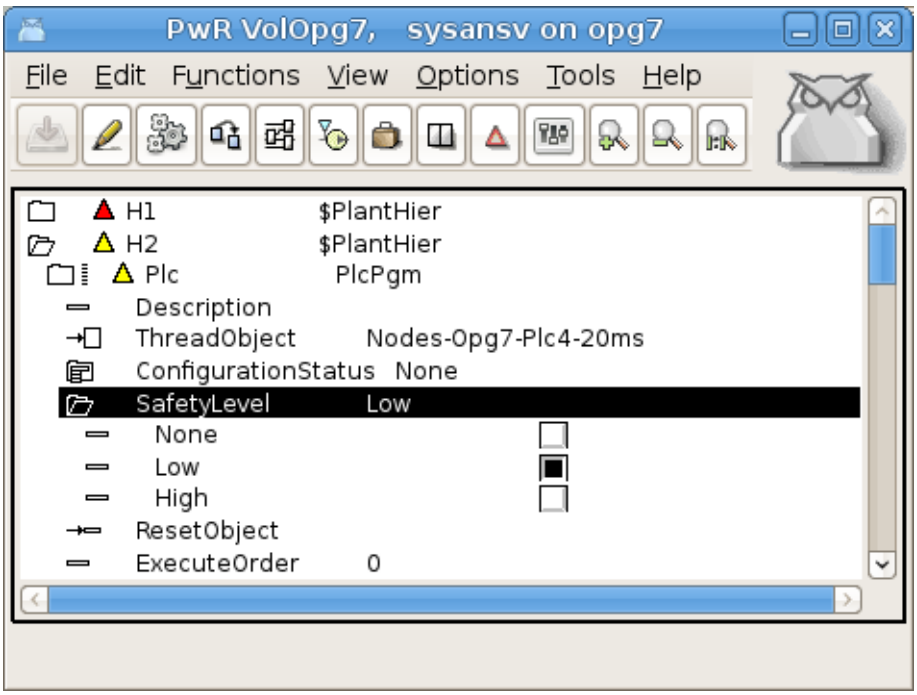


Fig Safety level indicator

Warning triangles in plc code

Warning triangles can also be inserted in the plc code with the YellowWarning and RedWarning objects. They should be used to mark code where special safety precautions have been made, and any modification should be preceded with risk analysis. These are found under the Edit folder in the plc editor palette.

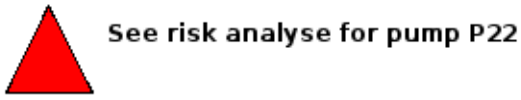


Fig Warning triangle in plc code

System events

A number of events and alarms generated by the system is added. These are configured by SystemSup objects in the MessageHandler object. These alarms can be modified or disabled by editing the SystemEvent array in the MessageHandler object.The following events will cause an alarm or event message

- Link down
- Node up
- Node restart
- Outunit restart
- Emergency break with reboot
- Emergency break with fixed output values
- Emergency break with fixed I/O
- Simulate load

- System status error
- System status warning
- IO soft error limit reached
- IO hard error limit reached
- Application alarm quota exceeded
- System alarm quota exceeded

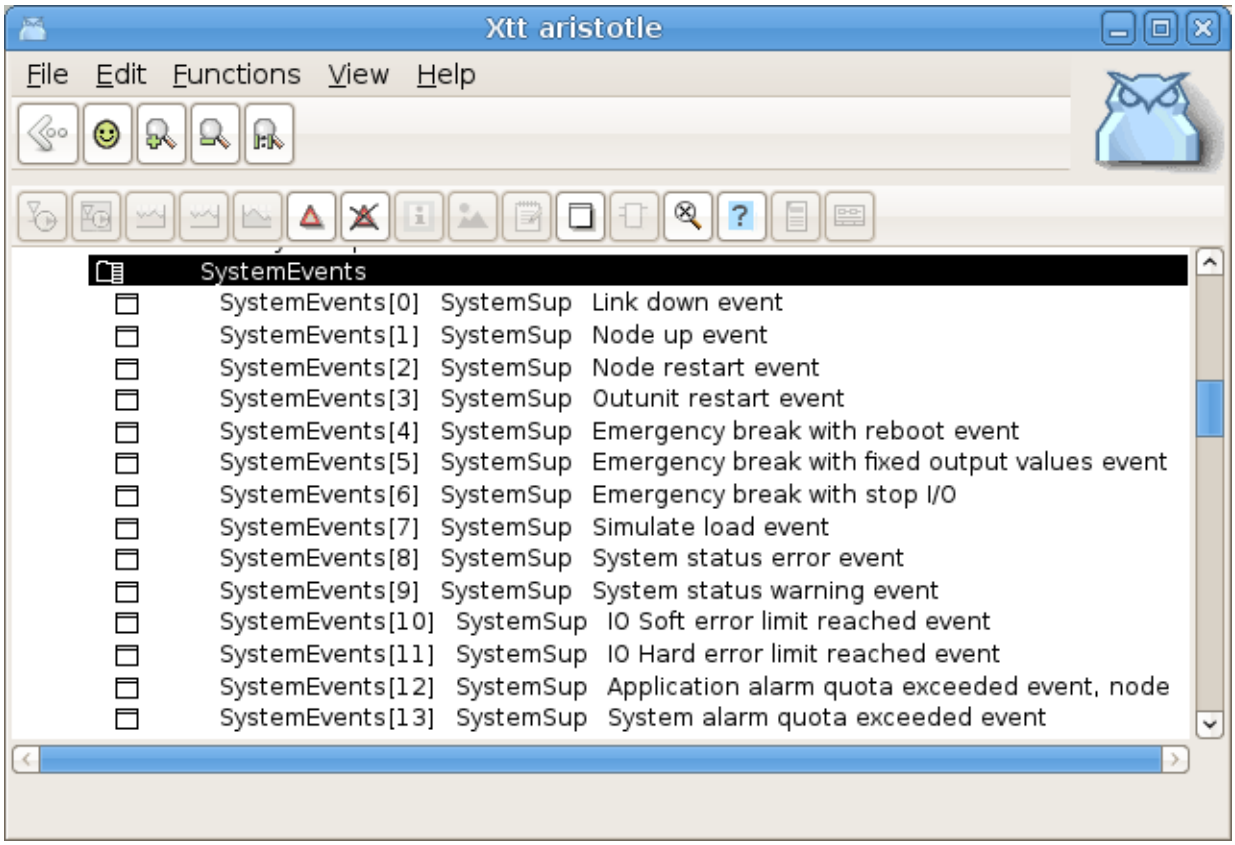


Fig System event array in the MessageHandler object

New Classes

XttMethodsMask

Contains bitmasks for Xtt methods. This class is inserted in all signal and component objects.

SystemSup

Supervision object for system events. Only present in the SystemEvent array in the MessageHandler object.

GetRefA, GetRefD, GetRefI, GetRefS

Get the value of an analog, digital, integer or string attribute by reference.

StoRefA, StoRefD, StoRefI, StoRefS

Store a value into an analog, digital, integer of string attribute by reference.

CStoRefA, CStoRefI, CStoRefS

Conditional storage of an analog, digital or string attribute by reference.

SetRefD, ResRefD

Set and reset of a digital attribute by reference.

IOSimulFlag

Plc object to fetch the value of IOSimulFlag in the IOHandler object.

Eurotherm_EPower, Eurotherm_EPowerAggr

A set of classes to control a Eurotherm Epower thyristor aggregate with Profinet communication.

PlcEmbed

Class for components with embedded plc code.

XttCamera

Configuration of a network video camera.

CameraPosition

Configuration object for the Xtt camera method. The camera method makes it possible to open the camera window and zoom in a specific part of the plant from the popup menu for an object.

YellowWarning, RedWarning

Yellow and red warning triangles in the plc editor.

Modified Classes

Upgrade procedure

The upgrading has to be done from any V5.2. 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->V4.6.0->V4.7.0->V4.8.6->(V5.0.0)->V5.1.0->V5.2.0->V5.3

Enter the administrator and change the version of the project to V5.2.0. Save and close the administrator.

Enter the directory volume and save.

I you have any class volumes, enter the class editor and build the volume.

Enter the configurator for each root volume and activate 'Function/Update Classes' and build.

Note ! The update procedure upgrade.sh doesn't have to be executed.

Modify OpPlace, PID and CompPID objects

- set the IsDefaultOp attribute in the OpDefault OpPlace object to 1.
- set PDAbsFlag = 1, WindupMask = BPID, MaxWindup = same vaue as MaxOut and MinWindup = same value as MinOut in PID and CompPID objects.