

# ProviewR

OPEN SOURCE PROCESS CONTROL



## Ge konstruktörshandbok

Copyright (C) 2005-2021 SSAB EMEA AB

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.

## 2 Inledning

Ge är en editor för att bygga bilder för underhåll och operatörskommunikation i ProviewR. Editorn är integrerad med ProviewR's utvecklingsmiljö och startas från konfiguratören. Bilderna byggs upp av bas-objekt i form av rektanglar, cirklar, linjer och texter, och av subgrafer, som är en slags sammansatta objekt. Ge innehåller en uppsättning subgrafer för några vanliga komponenter: ventiler, motorer, pumpar mm. Andra symboler kan konstruktören rita själv i Ge. Formen på objekten lagras som vektorer vilket, tillsammans med transformationsmatriser gör att de kan zoomas, skalas och roteras obegränsat. I runtime öppnas Ge-bilderna från operatörmiljön i Xtt, eller från webgränssnittet.

# 3 Terminologi

Här förklaras några termer som används i manualen.

## **Graf**

En bild från vilken operatören kan övervaka och kontrollera processen..

## **Subgraf**

En samling basobjekt kan sparas som en subgraf. En subgraf fungerar som en slags klass eller mall, och i en graf kan man skapa instanser av subgrafen, subgraf-objekt. Lagrade subgrafer hamnar i subgraf-plaetten.

## **Basobjekt**

Basobjekt är objekt av typen rektanglar, ellipser, linjer, polylinjer, texter, kopplingspunkter och noter.

## **Sammansatta objekt**

Mer komplexa objekt. Bar och trend objekt.

## **Annotation**

Text i en subgraf som kan vara unik för varje instans.

## **Komponent**

En instans av en subgraf eller en grupp.

## **Koppling**

En linje som sammanbinder två subgrafer. Linjen kan vara rak eller innehålla brytpunkter som delar upp den i vertikala och horisontala delar.

## **Kopplingspunkt**

Punkt i en subgraf till vilken man kan fästa en koppling.

## **Signal**

Signal används generellt som benämning på attribut i ett objekt i ProviewR's realtidsdatabas. Ofta är det fråga om värdet för en Di, Do, Dv, Ai, Ao eller Av men det kan vara ett attribut i vilket objekt som helst.

## **MB1**

Vänster musknapp.

## **MB2**

Musknappen i mitten (om denna saknas kan man ibland trycka ner vänster och höger musknapp samtidigt).

## **MB3**

Höger musknapp.

## **Rtdb**

ProviewRs realtids-databas.



# 4 Arbeta med editorn

## 4.1 Base objects

Vi ska börja med att titta på enkla objekt som linjer, rektanglar, cirklar och polyliner. Dessa objekt kallas för basobjekt. Basobjekten ligger till vänster i vertygpanelens nedersta rad.



**Figur** Knappar för att skapa basobjekt

### 4.1.1 Rektangel

Man ritar en rektangel genom att aktivera rektangelknappen i verktygspanelen, trycka ner höger musknapp där rektangelns ena hörn ska vara, dra med musknappen intryckt till motsatt hörn och släppa upp musknappen där. Med default-inställningarna skapas nu en rektangel med en svart ram.

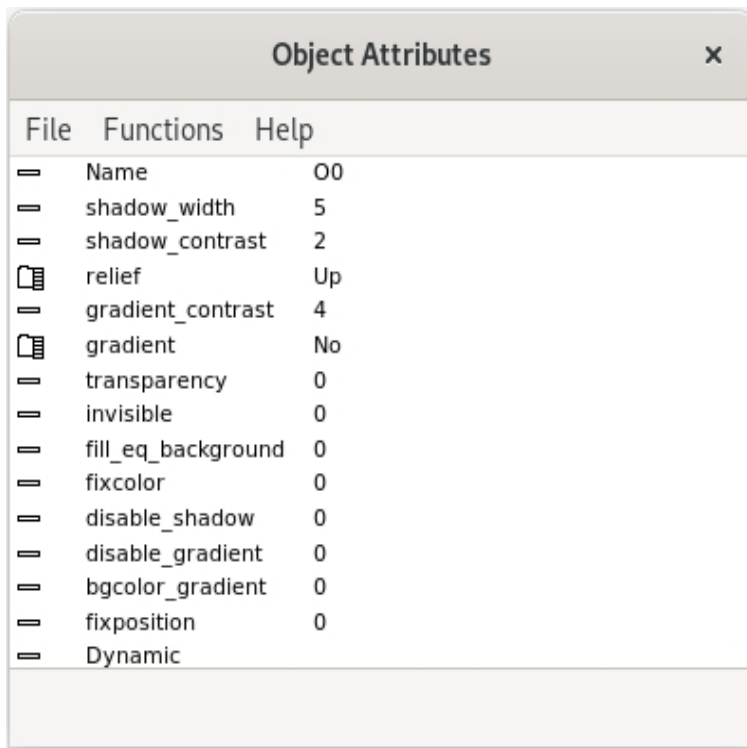


**Fig** Rektangel

Vi ska nu se hur man kan ändra utseendet på rektangeln. I verktygspanelen finns knapparna för kant, fyllnad och 3D. De fungerar så att om ett objekt är utvalt, ändas egenskapen på det utvalda objektet. Samtidigt utgör den aktuella inställningen default värde för nya objekt som skapas.

#### Objekts-editorn

Genom att dubbelklicka på rektangeln öppnas ett fönster där man kan sätta olika egenskaper på rektangeln. Ett attribut ändras genom att man väljer ut attributet, trycker på pil höger (eller Ctrl+Q) och matar in the nya värdet.



**Fig Objekteditorn**

### Fyllnadsfärg

Vi kan fylla rektangeln med färg genom att klicka på Fill knappen i verktygspanelen.



Färgen ändras genom att först välja ut rektangeln genom att klicka på den, och sedan klicka på önskad färg i färg-paletten.



**Fig Rektangel med fyllnadsfärg**

### Kant

Kanten på rektangeln kan tas bort med Kant knappen i verktygspanelen. Men det krävs att rektangeln är antingen fylld eller har en 3D ram (se nedan) för att kanten ska kunna tas bort. Är kanten borttagen kan man addera den med Kant knappen.



**Fig Rektangel utan kant**

### **Kanttjocklek**

Tjockleken på kanten kan ändras med Linewidth menyn i verktygspanelen. Välj ut rektangeln och ställ in önskad linjebredd. Linjebredden räknas i pixel och en bredd mellan 1 och 8 pixel kan väljas.



**Fig Rektangel med bredare kant**

### **Kantfärg**

Även färgen på kanten kan ändras. Välj ut rektangeln, välj 'Border' i färg-paletten och klicka på önskad färg.



**Fig Ändrad kantfärg**

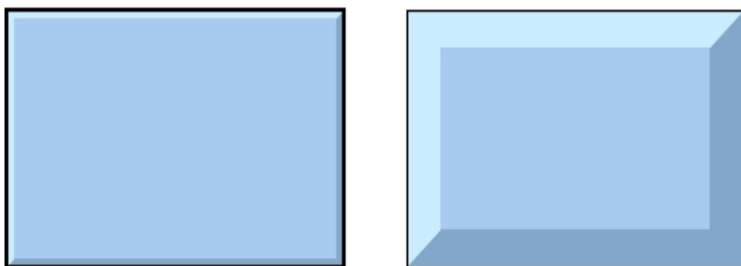
### **3D**

Med 3D knappen i verktygspanelen ritas kant på rektangeln som ger relief verkan.



### Fig Rektangel med 3D

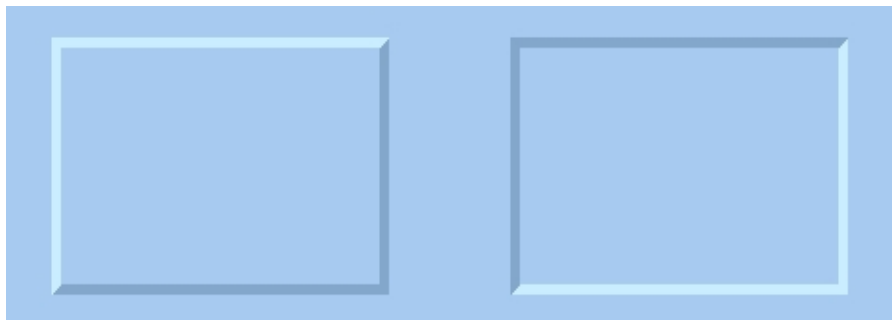
Bredden på relief kanten kan ändras med attributet `shadow_width`. Öppna objekts-editorn genom att dubbelklicka på rektangeln och sätt ett nytt värde på `shadow_width`. Enheten på `shadow_width` är % av bredden eller höjden (den minsta av dem).



### Fig Bredd 3 % till vänster och 15 % till höger

Man kan även ändra på relief kantens kontrast med `shadow_contrast`. Genom att öka värdet till 3 blir skuggan mörkare och den ljusa kanten ljusare. Tillåtna värden är 1-3.

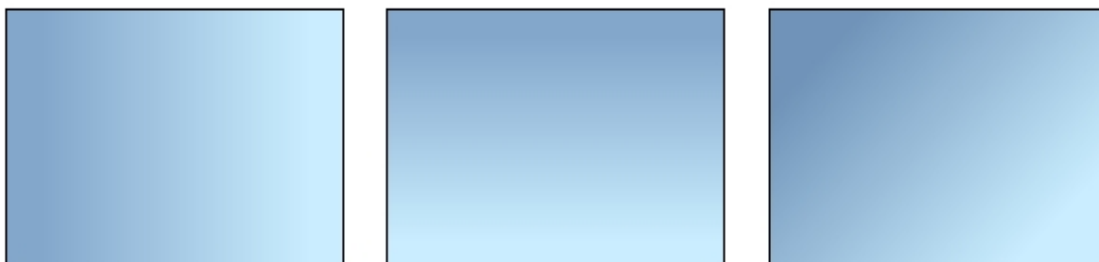
Om reliefen ska gå utåt eller inåt påverkas av attributet `relief`. Up innebär att reliefen går utåt, dvs den övre kanten färgas ljus och den undre mörk. Vid Down skiftas detta så att den över färgas mörk och den under ljus, vilket ger effekt av en inåtgående relief. Störst relief effekt får man om reliefen ritas med samma färg som bakgrunden.



### Fig Relief Up till vänster och Down till höger

### Gradient

Med de tre gradient knapparna kan man sätta en gradient på rektangeln, dvs en färgförskjutning från ljusare till mörkare färgton.



### Fig Rektanglar med olika gradient

I gradient menyn finns ytterligare 17 varianter av gradient att välja mellan. Nedan visas Horizontal Tube2 som lämpare sig för cylindriska föremål.



**Fig Rektangel med gradient Horizontal Tube2**

Kontrasten på gradienten kan påverkas med `gradient_contrast` attributet i objekts-editorn. Kontrasten kan ha ett värde mellan 0 och 10.

Gradients med två olika färger kan skapas genom att använda objektets bakgrundsfärg.

- Rita en rektangel med fyllnadsfärg.
- Sätt gradient på rektangeln.
- Öppna objektseditorn och set `bgcolor_gradient` till 1.
- Set bakgrundsfärg på objektet genom att välja ut rektangeln och klicka med Shift+Ctrl MB1 på en färg i färgpaletten.

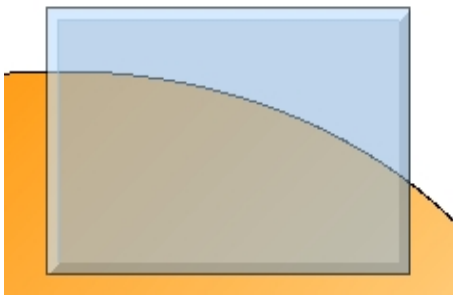
Nu ritas gradienten från fyllnadsfärgen till bakgrundsfärgen på objektet.



**Fig Gradient mellan två färger**

### Sätt transparens

Transparensen på ett objekt sätts från objektseditorn. Transparency är ett värde mellan 0 och 1, där 0 är ingen transparens, och 1 är full transparens.



**Fig Rektangel med transparens satt till 0.5**

### Rita en kvadrat

För att rita en kvadrat trycker man Ctrl+E (Scale equal) innan man ritat rektangeln. Återställ Equal scale genom att högerklicka i arbetsarean.

### Rita flera rektanglar

Om man ska rita flera rektanglar kan man hålla ner Shift-tangenten samtidigt som man klickar på rektangel-knappen i vertygspanelen. Då kan man rita flera rektanglar utan att klicka på

rektangel-knappen för varje ny rektangel. Funktionen återställs genom att högerklicka i arbetsarean.

#### 4.1.2 Linje

Man ritar en linje genom att aktivera linje knappen i verktygspanelen, trycka ner höger musknapp där linjen ska starta, dra med musknappen intryckt till slutpositionen och där släppa upp musknappen.



**Fig Linje**

##### Linjebredd

Linjebredden kan ändras genom att välja ut linjen och välja bredden i Linewidth menyn.



**Fig Linewidth satt till 8**

##### Färg

Färgen ändras genom att välja ut linjen, klicka på Border i färg-paletten och klicka på önskad färg.



**Fig Färgad linje**

##### Linjetyp

Det finns ett antal olika linjetyper att välja mellan. Välj ut linjen och aktivera önskad typ i Linetype menyn



**Fig Olika linjetyper**

##### Horisontella och vertikala linjer

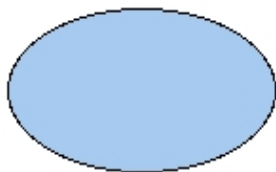
För att rita en horisontell linje trycker man Ctrl+H (MoveRestrictions Horizontal) innan man ritar linjen. En vertikal linje ritas genom att först trycka Ctrl+G (MoveRestrictions Vertical). MoveRestrictions återställs genom att högerklicka i arbetsarean.

##### Rita flera linjer

Om man ska rita flera linjer kan man hålla ner Shift-tangenten samtidigt som man klickar på linje-knappen i vertygspanelen. Då kan man rita flera linjer utan att klicka på linje-knappen för varje ny linje. Funktionen återställs genom att högerklicka i arbetsarean.

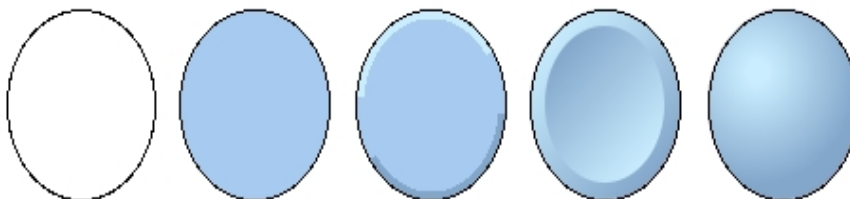
### 4.1.3 Ellips och cirkel

Rita en ellips genom att välja ellips-knappen i vertygspanelen, trycka ner vänstra musknappen, dra med knappen intryckt och slutligen släppa upp knappen.



**Fig Ellips**

Liksom för rektangeln kan man sätta egenskaper för kant, fyllnad, kantfärg, fyllnadsfärg, 3D och gradient. Några exempel visas nedan.



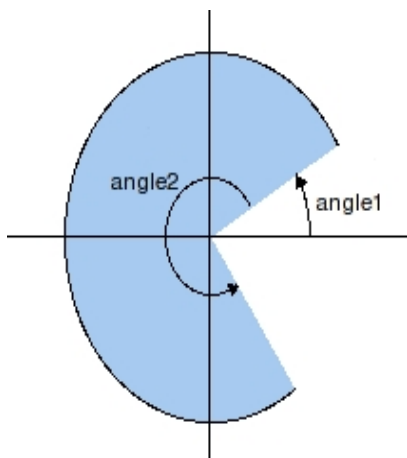
**Fig Cirkel med olika kombinationer av kant, fyllnad, 3D och gradient.**

#### Cirkelbåge

Man kan även rita en cirkelbåge, genom att i objekts-editorn ange angle1 och angle2. angle1 är vinkeln från den horisontella axeln till segmentets början, och angle2 är vinkeln för segmentets utbredning.

#### Rita flera ellipser

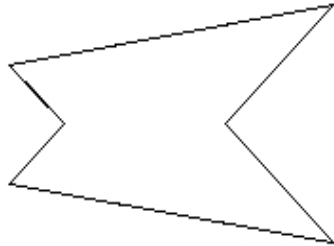
Genom att hålla ner Shift-tangenten samtidigt som man klickar på ellips-knappen i vertygspanelen kan man rita flera ellipser utan att klicka på ellips-knappen för varje ny ellips. Funktionen återställs genom att högerklicka i arbetsarean.



**Fig Cirkel-segment med angle1=30 and angle2=270**

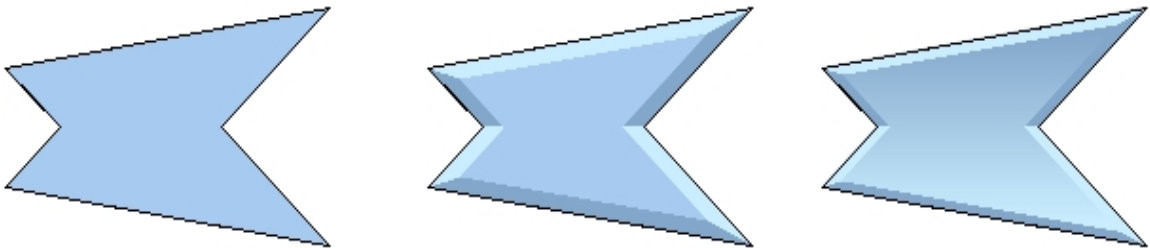
#### 4.1.4 Polylinje och polygon

En polylinje ritas genom att klicka på polyline-knappen in verktögspanelen, trycka ner vänster musknapp vid startpunkten, hålla musknappen intryckt och dra till nästa brytpunkt, släppa upp musknappen, trycka ner igen och dra till nästa brytpunkt osv. När sista linjen är ritad högerklickar man i arbetsarean för att avsluta polylinjen.



**Fig Polylinje**

Olika kombinationer av kant, fyllnad, 3D, gradient, linjebredd och färg kan användas för polylinjer



**Fig Några kombinationer av fyllnad, 3D och gradient**

##### 3D

För att kunna använd 3D effekten på polylinjer bör polylinjen vara sluten, dvs startpunkten sammanfaller med slutpunkten. För att rita en sluten polylinje aktiverar man Snap to grid i verktögspanelen innan man ritat polylinjen.

##### Ändra en polylinje

För att flytta en brytpunkt i en polylinje väljer man ut polylinjen och aktiverar sedan Edit/Edit Polyline in menyn. Brytpunkterna och ändpunkterna blir nu känsliga man kan flytta på dem genom att dra med vänstra musknappen.

##### Vertikala och horisontella linjer

Ofta vill man enbart ha vertikala och horisontella linjer i polylinjen. Genom att trycka Ctrl+H (MoveRestrictions horizontal) så blir den första linjen horisontell. Sedan ändras riktningen för varje ny del-linje, så att den andra blir vertikal, den tredje horisontal osv. Trycker man Ctrl+G (MoveRestrictions vertical) istället blir den första vertikal, den andra horisontell osv.

#### 4.1.5 Text

Texter ritas genom att aktivera text-knappen i verktögspanelen. Klicka i arbetsarean och mata



in texten i inmatningsfältet längst ner i editorn.

## Pump control

### Fig Text

#### Objekt attribut

Genom att dubbelklicka på texten öppnas objekteditorn med objektets attribut. Attributet Adjustment kan ställas in för höger-, mitten-, eller väster-justerad text.

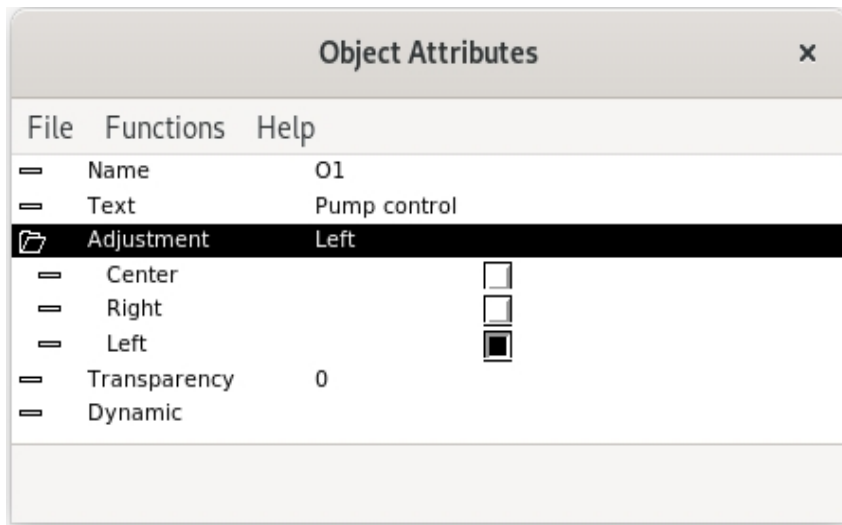


Fig Objekteditorn med val av justering

#### Text storlek

Man ändrar storlek på texten med Textsize menyn i verktygspanelen. Välj ut texten och ange lämplig storlek i menyn. Man kan även använda skalnings-funktionen för att ändra storlek.

#### Font

Fonten ställs in med font menyn in verktygspanelen. De fonter som finns a välja mellan är Helvetica, Times, New Century Schoolbook, Courier och Lucida Sans. Välj ut texten och ange lämplig font i menyn.

#### Färg

Text färg välis genom att markera Text knappen i färg-paletten och sedan klicka på en färg.

Pump control

Fig Färgad text

#### Rotera en text

En text kan roteras i steg om 90 grader genom att använda rotations-knappen i vertygspanelen. Endast texter som är mitt-justerade, dvs har Adjustment satt till Center i objekteditorn, kan roteras.

**Fig Roterad text****Ändra en text**

För att ändra en text väljer man ut texten och trycker på Ctrl+T (Change Text). Ett inmatningsfält öppnas i editorns nederkant där man kan ange en ny text.

**Bitmap fonter**

Normalt används fonter som är fullt skalbara och antialiaserade. I vissa lägen kan bitmap-fonter vara att föredra. De finns bara i vissa storlekar mellan 8 och 24, men kräver avsevärt mindre processorkraft att hantera. Bitmap-fonter väljs genom att sätta BitmapFonts i Graph Attributes till 1.

**Fontinställning i fönsterhanteraren**

Många fönsterhanterare har inställningsmöjligheter som påverkar fonterna. I Gnome hittar man inställningar under System/Inställningar/Utseende/Typsnitt. Här kan man välja mellan olika renderingar.

## 4.2 Editering

I det här avsnittet ska vi titta närmare på olika editerings-funktioner, t ex hur man flyttar, skalar, färgsätter objekt mm.

**Skapa objekt**

Hur man skapar basobjekt finns närmare beskrivet i kapitlet om basobjekt ovan. Normalt skapar man ett objekt genom att klicka på symbolen i verktygspanelen, och klicka eller dra med MB1 i arbetsarean. Subgrafer skapas genom att välja en subgraf i subgraf-paletten och klicka med MB2 i arbetsarean.

**Välja ut objekt**

Man väljer ut ett objekt med Klick MB1. Ett utvalt objekt markeras med röd färg. Om man vill välja ut fler objekt kan man klicka på dessa med Shift/Klick MB1.

För att välja ut objekt inom ett visst område kan man även dra med MB1 i arbetsarean. De objekt som ligger inom utvalsrektangeln kommer att väljas ut. Eftersom man även flyttar objekt genom att dra med MB1 gäller att att inte få träff på något objekt när man startar draget. Om det är det svårt att undvika kan man använda Shift/Drag MB1 istället som adderar objekt till utvalslistan.

Notera att när man ändrar färg på ett utvalt objekt så försvinner rödmarkeringen, eftersom man då visar den nya färgen istället. Men objektet är fortfarande utvalt trots att det inte är rödmarkerat.

Genom att klicka med MB3 i arbetsarean tömmer man utvalslistan.

## Ta bort objekt

För att ta bort ett objekt väljer man ut det och trycker på Delete knappen. Man kan även ta bort ett objekt genom att dubbelklicka med MB2 på objektet. Genom att dubbelklicka med MB2 någonstans i arbetsarean tar man bort alla utvalda objekt.

## Flytta på objekt

Man flyttar på ett objekt genom att dra med MB1. Om man vill flytta flera objekt samtidigt, väljer man ut dem och drar i ett av objekten. Alla utvalda objekt följer då med. Om man enbart vill flytta vertikalt eller horisontalt, kan man använda Functions/MoveRestrictions/Vertical (Ctrl+G) eller Functions/MoveRestrictions/Horizontal (Ctrl+H). Dessa funktioner är verksamma tills man återställer med MB3.

Om man har aktiverat Snap to grid i verktygspanelen, justeras objektets position till närmsta gridpunkt vid en förflyttning.

Man kan även flytta ett objekt med move kommandot. Det ger större precision än om man flyttar med musen. Man kan flytta genom att ange relativa eller absoluta koordinater. Kombinationer av relativa och absoluta är också tillåtet. Kommandot för att flytta det utvalda objektet till punkten (1, 1) är

```
ge> move selected /absx=1 /absy=1
```

## Objekt med fix position

Objekt som ligger i bakgrunden är det lätt att flytta på av misstag när man editerar. För vissa objekt, rektanglar, ellipser och polylinjer, kan man sätta attributet 'fixposition' som låser objektet vid en viss position.

## Kopiera objekt

Man kopierar objekt genom att välja ut dem, och aktivera Edit/Copy (Ctrl+C) och Edit/Paste (Ctrl+V) i menyn. Efter paste blir de nya objekten hängande på markören och genom att klicka med MB1 fäster man dem i arbetsarean. Vid kopiering kan man liksom vid förflyttning, välja vertikal (Ctrl+G) eller horisontal (Ctrl+H) förflyttning.

## Skala ett objekt

Man skalar objekt genom att välja ut dem, klicka på Scale knappen i verktygspanelen, och sedan dra i den fyrkant som ritas runt objekten. Avsluta med klick MB3. Om man vill att proportionerna mellan bredd och höjd ska behållas, trycker man först Ctrl+E (Scale Equal).

Det finns även en knapp för att dubblera storleken på objekten, och en för att halvera storleken.

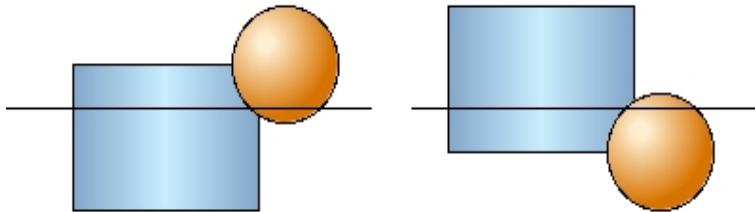
## Rotera ett objekt

Man kan rotera ett objekt med rotations-knappen i verktygspanelen. Den roterar utvalda objekt i steg om 90 grader.

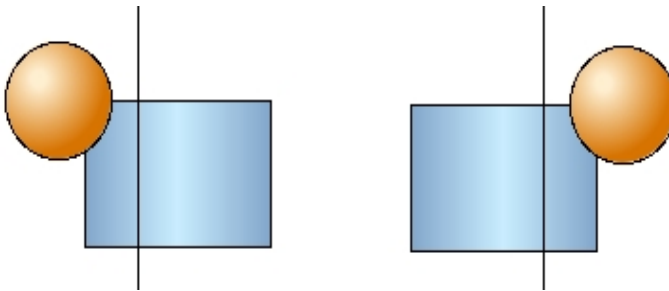
Om man vill ha en annan rotationsvinkel aktiverar man Edit/Rotate i menyn. Observera att rektanglar och ellipser endast kan roteras i steg om 90 grader, och för att texter ska kunna roteras måste attributet Adjustment för texten vara satt till Center.

## Spegling

Det finns två knappar i verktygspanelen för att spegla. En för horisontell spegling och en för vertikal spegling. Speglingen utförs på utvalda objekt.



**Fig Horisontell spegling**



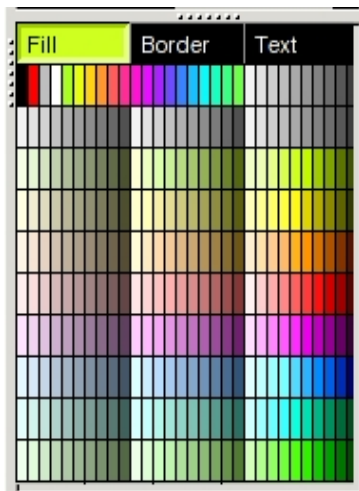
**Fig Vertikal spegling**

## Ändra färg på ett objekt

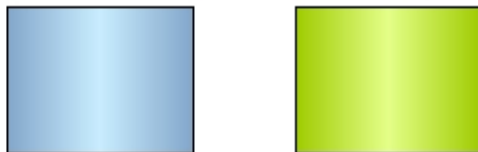
Det finns tre olika typer av färger man kan välja i färgpaletten, fyllnadsfärg, kantfärg och textfärg. I paletten finns knapparna Fill, Border och Text där man markerar vilken färgtyp som ska väljas. När man sedan väljer en färg i paletten ändras färgen på utvalda objekt. Den valda färgkombinationen sätts även på nya objekt som skapas.

### Fyllnadsfärg

För att sätta fyllnadsfärg på ett objekt, väljer man ut objektet, kontrollerar att Fill är markerat i paletten, annars klickar man på Fill. Därefter väljer man den färg man vill sätta på objektet. I exemplet nedan ändras fyllnadsfärgen från blått till grönt.



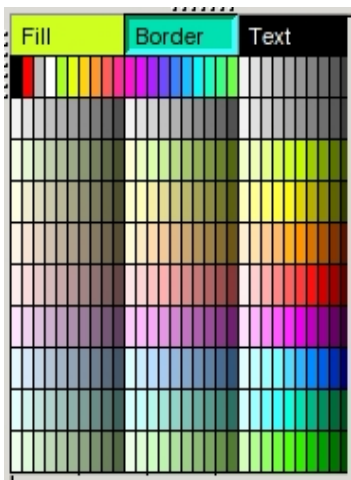
**Fig Fill är markerat och en grön nyans har valts**



**Fig Fyllnadsfärg ändras från blått till grönt**

### **Kantfärg**

Kantfärgen kan ändras på liknande sätt. Här väljer man först Border i färgpaletten, därefter väljer man en kantfärg som sätts på utvalt objekt.



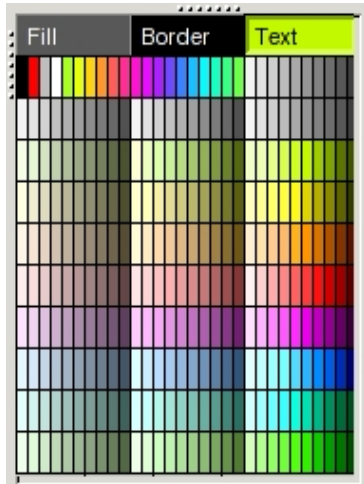
**Fig Border är markerat och en blågrön nyans har valts för kantfärgen**



**Fig Kantfärg ändras från svart till blågrönt**

### **Textfärg**

För att ändra text-färgen väljer man Text i färgpaletten, därefter väljer man en textfärg som sätts på utvalt objekt.



**Fig Text är markerat och en grön nyans har valts för textfärgen**



**Fig Textfärg ändras från svart till grönt**

## Bakgrundsfärg

Man sätter bakgrundsfärgen i bilden genom att välja en fyllnadsfärg i färg-paletten och aktivera Functions/Set Background Color i menyn.

## Grid

Det två knapparna Show grid och Snap to grid, tillsammans med Gridsize menyn hanterar grid-funktionen. Show grid visar alla gridpunkter, och med Gridsize menyn kan man ställa in avståndet mellan gridpunkterna. När Snap to grid är aktiverat, justeras koordinater för objekt som skapas och flyttas till närmaste grid-punkt.

Grid-storleken kan sättas till 1.0, 0.5, 0.25 eller 0.10 i Gridsize menyn. Med 'set gridsize' kommandot kan man även sätta andra värden på grid-storleken. För att sätta grid-storleken till 0.20 är kommandot

```
ge> set gridsize 0.20
```

## Ordna objekt

Det finns ett antal knappar i verktygspanelen för att justera positionen för en serie av objekt så att de hamnar på samma horisontella eller vertikala nivå.

Man kan justera läget så att objekten läggs på lika avstånd från varandra.

## Grupper

Man skapar en grupp av en antal objekt genom att välja ut objekten och trycka på Group knappen i verktygspanelen. Gruppen behandlas sedan som om det vore ett enskilt objekt, t ex när man väljer ut den, flyttar på den eller skalar den. Vill man ändra på ett del-objekt i en grupp måste man först lösa upp gruppen med Ungroup knappen, sedan kan man göra ändringen och gruppera igen.

Grupper har egenskaperna dynamik och aktion, dvs man kan koppla dem till signaler i databasen och t ex ändra färg på gruppen beroende på signalens status. Man kan även göra dem klick-känsliga och sätta signaler när man klickar på dem. Det här är egenskaper som man delar med subgrafer, men som basobjekten saknar. Vill man sätta dynamik på ett basobjekt, t ex en rektangel är det dock bara att skapa en grupp med enbart rektangeln, och man har möjlighet att sätta dynamik och aktion på den. På det här sättet kan man göra en indikator, tryckknapp eller stapeldiagram av rektangeln.

Object Attributes x			Object Attributes x			Object Attributes x		
File	Functions	Help	File	Functions	Help	File	Functions	Help
=	Name	Grp2_	=	Name	Grp2_	=	Name	Grp2_
=	Group	Grp2_	=	Group	Grp2_	=	Group	Grp2_
=	Transparency	0	=	Transparency	0	=	Transparency	0
	Cycle	Inherit		Cycle	Inherit		Cycle	Inherit
	DynType1	Inherit		DynType1	Inherit		DynType1	Inherit
=	Inherit	<input checked="" type="checkbox"/>		DynType2			DynType2	
=	Tone	<input type="checkbox"/>	=	DigTextColor	<input type="checkbox"/>		Action	Inherit
=	DigLowColor	<input type="checkbox"/>	=	TimeoutColor	<input type="checkbox"/>	=	Inherit	<input checked="" type="checkbox"/>
=	DigColor	<input type="checkbox"/>	=	DigFourShift	<input type="checkbox"/>	=	PopupMenu	<input type="checkbox"/>
=	AnalogColor	<input type="checkbox"/>	=	ScrollingText	<input type="checkbox"/>	=	SetDig	<input type="checkbox"/>
=	StatusColor	<input type="checkbox"/>	=	ColorThemeLightness	<input type="checkbox"/>	=	ResetDig	<input type="checkbox"/>
=	DigWarning	<input type="checkbox"/>	=	DigBackgroundColor	<input type="checkbox"/>	=	ToggleDig	<input type="checkbox"/>
=	DigError	<input type="checkbox"/>	=	DigSwap	<input type="checkbox"/>	=	StoDig	<input type="checkbox"/>
=	DigFlash	<input type="checkbox"/>	=	DigScript	<input type="checkbox"/>	=	SetValue	<input type="checkbox"/>
=	FillLevel	<input type="checkbox"/>	=	RefUpdate	<input type="checkbox"/>	=	Command	<input type="checkbox"/>
=	Invisible	<input type="checkbox"/>	=	DigTransparency	<input type="checkbox"/>	=	CommandDoubleClick	<input type="checkbox"/>
=	DigBorder	<input type="checkbox"/>	=	AnalogTransparency	<input type="checkbox"/>	=	Script	<input type="checkbox"/>
=	DigText	<input type="checkbox"/>	=	UnitConvert	<input type="checkbox"/>	=	Help	<input type="checkbox"/>
=	AnalogText	<input type="checkbox"/>		Action	Inherit	=	OpenGraph	<input type="checkbox"/>
=	Value	<input type="checkbox"/>				=	CloseGraph	<input type="checkbox"/>
=	Rotate	<input type="checkbox"/>				=	OpenURL	<input type="checkbox"/>
=	Move	<input type="checkbox"/>				=	Confirm	<input type="checkbox"/>
=	AnalogShift	<input type="checkbox"/>				=	IncrAnalog	<input type="checkbox"/>
=	DigShift	<input type="checkbox"/>				=	RadioButton	<input type="checkbox"/>
=	Animation	<input type="checkbox"/>				=	ValueInput	<input type="checkbox"/>
=	Video	<input type="checkbox"/>				=	ToolTip	<input type="checkbox"/>
=	HostObject	<input type="checkbox"/>				=	InputFocus	<input type="checkbox"/>
=	DigSound	<input type="checkbox"/>				=	PulldownMenu	<input type="checkbox"/>
=	DigCommand	<input type="checkbox"/>				=	OptionMenu	<input type="checkbox"/>
	DynType2					=	MethodPulldownMenu	<input type="checkbox"/>
	Action	Inherit				=	CatchSignal	<input type="checkbox"/>
						=	EmitSignal	<input type="checkbox"/>

Fig Objekteditorn för grupp med attribut för dynamik och aktion

## 4.3 Subgrafer

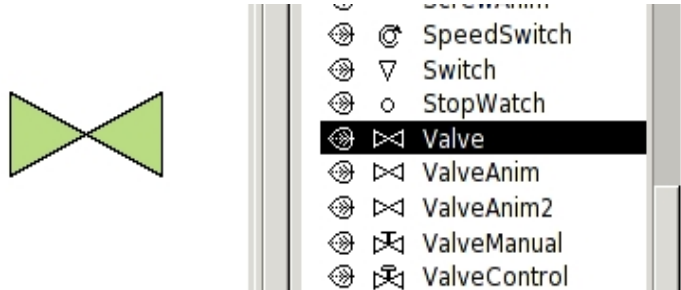
Subgraf är en grafisk komponent som är uppbyggd av bas-objekt. Ofta symboliserar den ett visst element i anläggningen, t ex en ventil eller pump, men den kan också vara en tryckknapp eller inmatningsfält.

### Skapa en subgraf

Till höger i editor finns subgrafs-paletten, och man lägger in en subgraf genom att välja



en subgraf i paletten och klicka med MB2 i arbetsarean. I figuren nedan har vi valt en Ventil under mappen Process.



**Fig En subgraf i form av en ventil**

De flesta subgrafer kan man sätta 3D och olika typer av gradienter på. Det finns dock en del subgrafer som inte är anpassade för detta.



**Fig Ventil med 3D till vänster, och 3D + gradient till höger**

## Ändra färg

Ventilen i figuren är en monokrom subgraf och dessa kan man ändra färg på genom att välja en fyllnadsfärg i färgpaletten, på samma sätt som för bas-objekt. Man kan även ändra på kantfärgen och kantbredden.



**Fig Blåfärgad ventil**

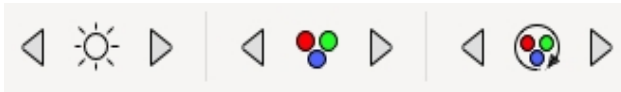
## Polykroma subgrafer

För subgrafer som innehåller olika färger ändrar man färgen med funktionen för att skifta färg eller med färgtons paletten. Låt oss skapa en brännare (Process/Burner) som innehåller färgerna gult och orange.



**Fig Polykrom subgraf**

Om vi sätter en fyllnadsfärg tappar vi färgteckningen och brännaren blir enfärgad. I stället använder vi knapparna för att skifta färg i verktygspanelen (högra knapparna i figuren nedan). Välj ut subgrafen och klicka på pilarna för att skifta färgen.



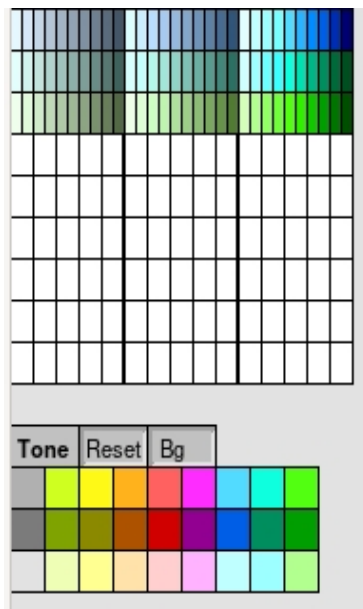
**Fig Knappar för att skifta färg**

Nu roteras färgerna på färgcirkeln så att vi fortfarande har två olika färger i subgrafen, och vi behåller den färgkontrast som fanns från början. Man kan även använda knapparna för att ändra intensitet och ljushet som ligger till vänster om färgskiftknapparna.



**Fig Exempel på brännare med skiftade färger**

Man kan även använda färgtons-paletten (nedre delen av färgpaletten i figuren nedan). Välj ut subgrafen och klicka på en färgton i paletten.



**Fig Färgtons-paletten**

Resultatet blir nu att de olika delarna har samma färgton, men ljuskontrasten mellan olika delar behålls.

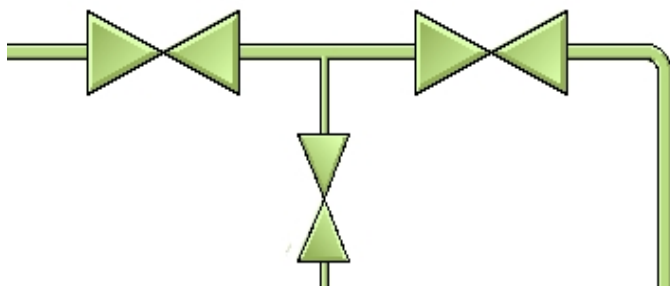


**Fig Brännare med olika färgton applicerad**

## Kopplingspunkter

Vissa subgrafer innehåller kopplingspunkter som gör att man dra kopplingar mellan subgraferna.

Man skapar en koppling genom att dra med MB2 från en kopplingspunkt i en subgraf, och släppa upp MB2 vid en kopplingspunkt i en annan subgraf. På kopplingarna kan man sätta linjebredd, färg och 3D. Släpper man en koppling i arbetsarean, skapas en knutpunkt med kopplingspunkter i olika riktningar, som man kan dra vidare till andra subgrafer.

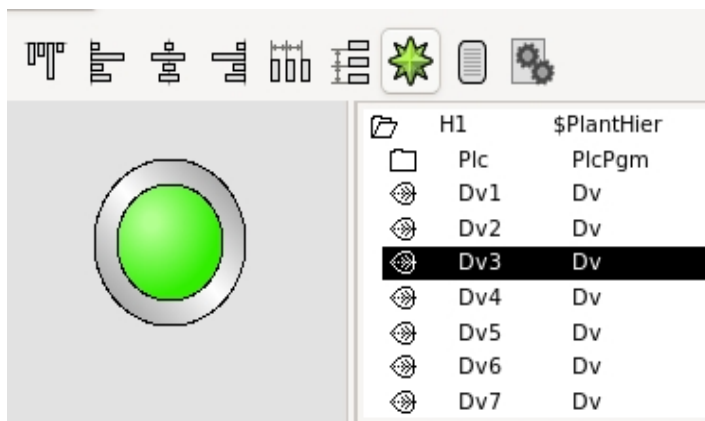


**Fig Subgrafer med kopplingar**

## Dynamik och aktion

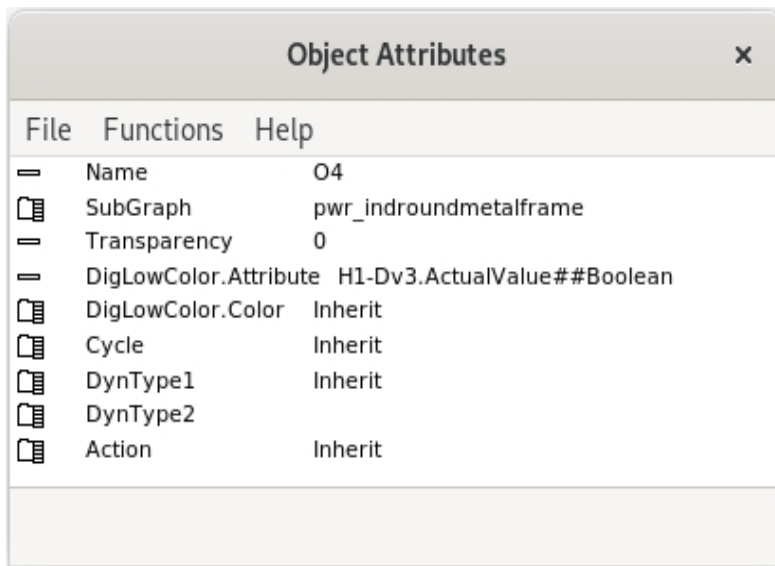
Liksom grupper har subgrafer egenskaperna dynamik och aktion, vilket gör att man kan koppla dem till signaler i databasen och ändra färg och form beroende på signalens status, eller göra dem klick-känsliga och sätta signaler när man klickar på dem. Dessutom har subgrafer ofta en förprogrammerad dynamik eller aktion. En indikator har t ex dynamiken att ändra färg som default så man behöver bara koppla en signal till den för att den ska fungera.

Låt oss skapa en indikator och se hur vi kan få den att ändra färg. Vi makerar en indikator Indicator/IndRoundMetalFrame i subgrafs paletten och klickar med MB2 i arbetsarean för att skapa indikatorn. För att koppla den till en signal i databasen klicka vi på navigator knappen i verktygspanelen. Nu visas plant hierarkin där subgrafspaletten tidigare fanns. Vi vill att indikatorn ska visa status på Dv'n H1-Dv3 och letar upp den i planthierarkin. Genom att välja ut H1-Dv3 och klicka med Ctrl+Dubbelklick MB1 på indikatorn har vi gjort kopplingen.



**Fig En signal kopplas till indikatorn med Ctrl+Dubbelklick MB1**

Genom att öppna objekteditorn för indikatorn kan vi kontrollera att kopplingen är gjord.



**Fig Objekteditorn för indikatorn**

Vi kan se att indikatorn har default dynamiken DigLowColor, som växlar mellan två färger, och DigLowColor.Attribute är satt till H1-Dv3.ActualValue.

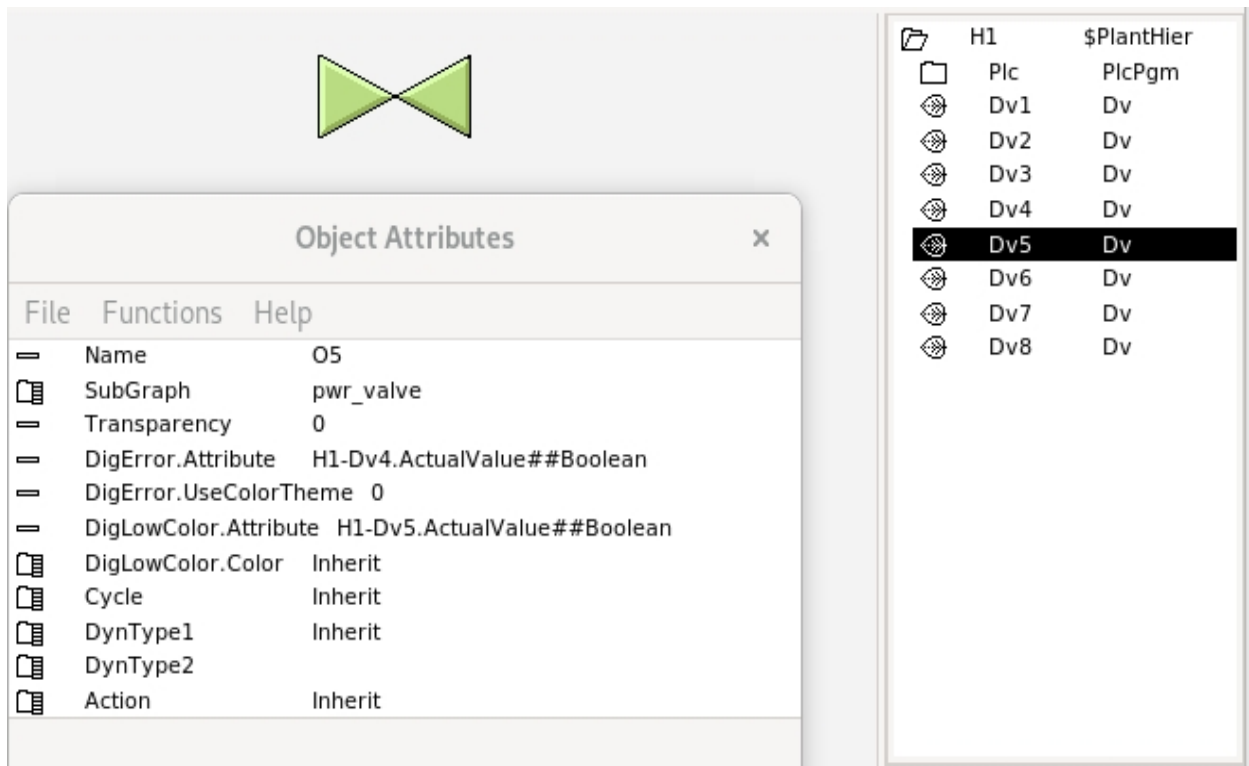
När vi öppnar grafen i rt\_xtt kan vi se hur indikatorn växlar mellan grönt och mörkgrått när Dv3 växlar mellan 1 och 0.



**Fig Indikatorn när dv'n är 1 till vänster och 0 till höger**

Om vi tittar på ventilen ovan, så har den lite mer avancerad dynamik. Den kan växla mellan tre olika färger och måste därför kopplas till två signaler. En signal färgar ventilen röd för att markera ett fel, och den andra färgar den vit för att markera stängd ventil.

Om vi öppnar objekteditorn för ventilen ser vi att DigError.Attribute ska kopplas till en signal som indikerar fel, och DigLowColor.Attribute till en signal som markerar att ventilen är öppen.



**Fig Ventil som kopplas till två signaler**

I det här fallet gör vi kopplingen i objekteditorn, och markerar H1-Dv4 i plant hierarkin. Genom att klicka med Ctrl+Dubbelclick MB1 på attributet DigError.Attribute läggs H1-Dv1 in här. På samma sätt klickar vi in H1-Dv5 på DigLowColor.Attribute. Resultatet ser vi i figuren nedan. När Dv4 är 0, dvs ventilen stäng, färgas den vit. När Dv4 går hög, dvs ventilen öppnas, färgas den grön. Om vi sätter felsignalen Dv4 färgas den röd oberoende av värdet på Dv5. Det beror på att DigError har högre prioritet än DigLowColor. I objekts-editorn är dynamiken ordnad efter prioritet så att högst prioritet ligger överst och lägst prioritet underst.



**Fig Stängd ventil till vänster, öppen ventil i mitten och felsignal till höger**

Om vi vill markera stängd ventil med mörkgrått istället för vitt, ändrar man på DigLowColor.LowColor. Defaultvärdet är Inherit vilket gör att en vit förprogrammerad färg väljs. Genom att istället välja GrayHigh9 kommer färgen när signalen är låg att bli mörkgrå. Har man svårt att identifiera färgerna kan man välja ut önskad färg som fyllnadsfärg i färg-paletten, och klicka med Ctrl+Dubbelclick MB1 på DigLowColor.LowColor attributet i objekts-editorn. Den valda färgen kommer då att läggas in på DigLowColor.LowColor.

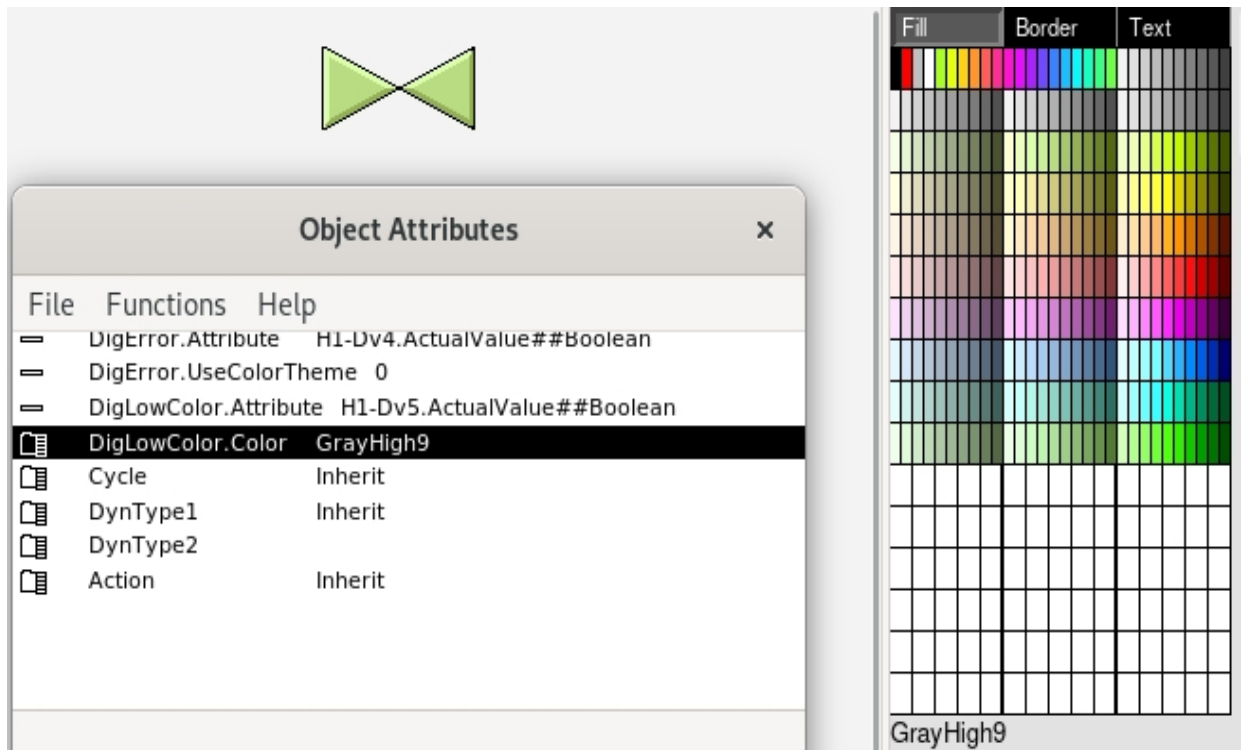


Fig LowColor ändrad från Inherit till mörkgrå



Fig Stängd ventil till vänster, öppen ventil i mitten och felsignal till höger

## 4.4 Speciella objekt

Förutom bas-objekt och subgrafer finns det en uppsättning lite mer komplexa objekt. Dessa objekt återfinns i subgraf-paletten. Objektet är

- Window
- TabbedWindow
- Slider
- Trend
- Bar
- BarArc
- Pie
- XYCurve
- FastCurve
- Axis
- Table
- PulldownMenu
- OptionsMenu
- MethodToolbar

### 4.4.1 Window

Window-objektet är en ram som visar en graf inom ett specifikt område i en annan graf.

Grafen kan visas med eller utan skroll-lister. Det är också möjligt att skifta den graf som visas i window-objektet genom att exekvera ett kommando från en trycknapp. Man kan även visa objektsbilder för olika objekt i databasen.

Här är några användningsområden för window-objektet

- man har tabell-liknande del av grafen som tar för stor plats. Denna del kan läggas i ett window objekt och genom att utnyttja window-objektets skroll-list vinner man plats.
- man vill visa information om ett databas-objekt som redan har en objektsbild.
- man kan bygga upp en graf med ett window objekt som visar olika grafer beroende på en uppsättning tryckknappar eller menyalternativ. Jämför även TabbedWindow.

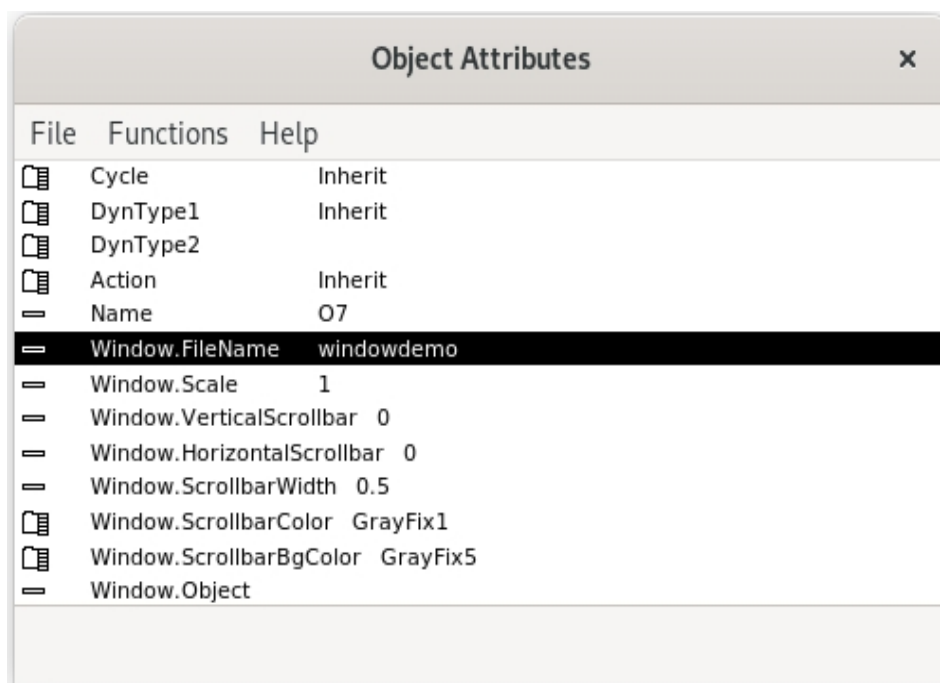
### Visa en graf i ett window-objekt

Window-objektet återfinns under Other mappen in subgraf-paletten. Skapa objektet genom att mittenklicka i arbetsarean och skala objektet till önskad storlek.



**Fig Window-objekt**

Öppna objekteditorn och ange namnet på grafen som ska visas i Window.FileName. Observera att grafen läses in från \$pwrp\_exe, och eftersom grafer in window-objekt ofta inte har något eget XttGraph objekt, kommer de inte att kopieras automatiskt från \$pwrp\_pop till \$pwrp\_exe. Kopieringen måste göras för hand eller med en make-fil.



**Fig Objekteditorn för window-objektet angivet filnamn**

När filnamnet har lagts in och filen finns på \$pwrp\_exe kommer grafen att visas i windowobjektet.



**Fig Window-objekt med inlagt filnamn**

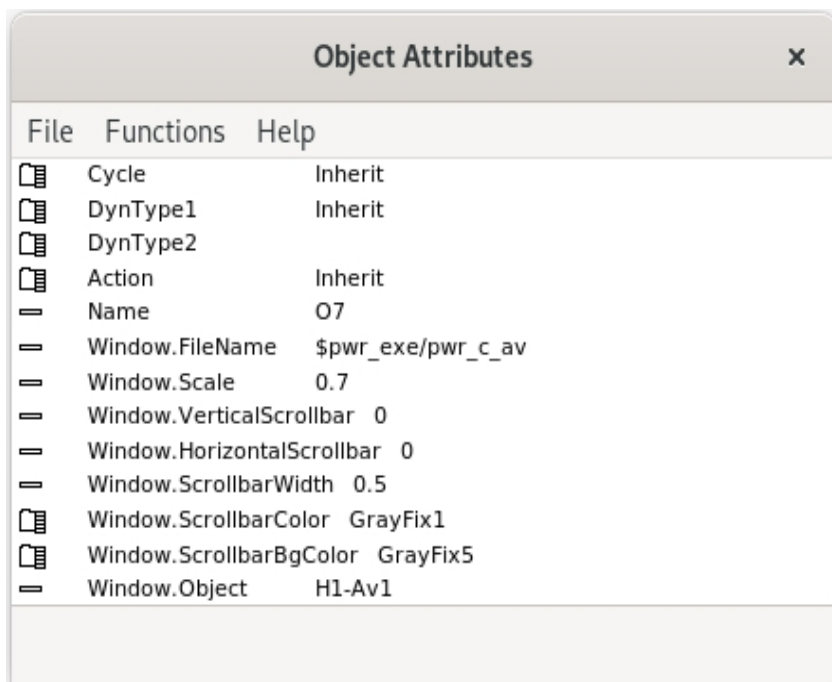
#### **Visa en objektsbild i ett window-objekt**

Många databas-objekt har en objektsbild som öppnas från 'Object Graph' i popupmenyn för objektet. Även objektsbilder kan visas window-objekt, genom att specificera namnet på graf-filen. Dessutom måste databas-objektet som bilden ska vara kopplad till anges i Window.Object.

Om klassen för objektet ingår i ProviewR's bassystem, är filnamnet för objektsbilden \$pwr\_exe/pwr\_c\_'klassnamn'.pwg. För ett Av-objekt blir det \$pwr\_exe/pwr\_c\_av.pwg. Man måste här ange även katalogen i filnamnet eftersom filen inte ligger på default katalogen \$pwrp\_exe.

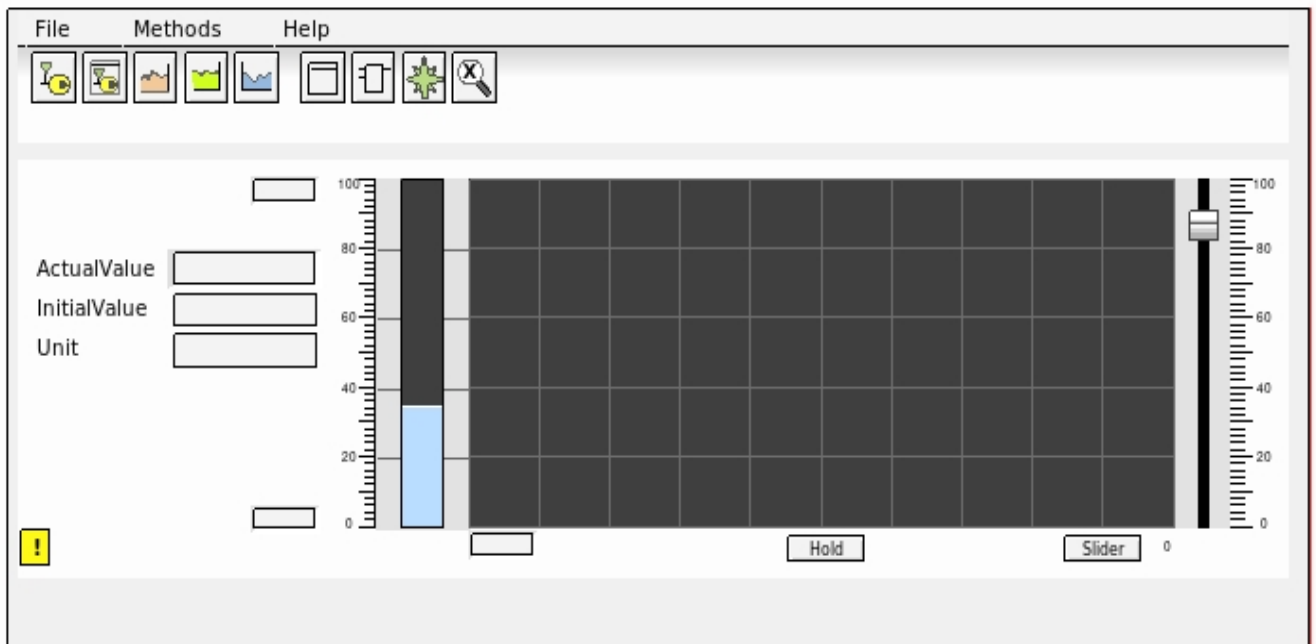
Om klassen återfinns i en klassvolym inom projektet, är filnamnet \$pwrp\_exe/'klassnamn'.pwg. Här behöver man inte ange katalogen i filnamnet.

I exemplet nedan visas objektsbilden för av-objektet H1-Av1. Bilden är nedskalad till 3/4 storleken genom att Window.Scale är satt till 0.75.



**Fig Attribut för window-objekt som visar objektsbild**





**Fig Objektsbild för en Av i window-objektet**

### Växla mellan olika grafer

Det finns ett xtt kommando som visar upp en viss graf i ett window-objekt. Genom att lägga in kommandot på en serie tryckknappar kan man växla mellan olika grafer i ett window-objekt. Funktionen motsvarar ett TabbedWindow med den skillnaden att man har en friare layout på tryckknapparna och att det inte finns någon begränsning i antalet grafer som kan visas upp.

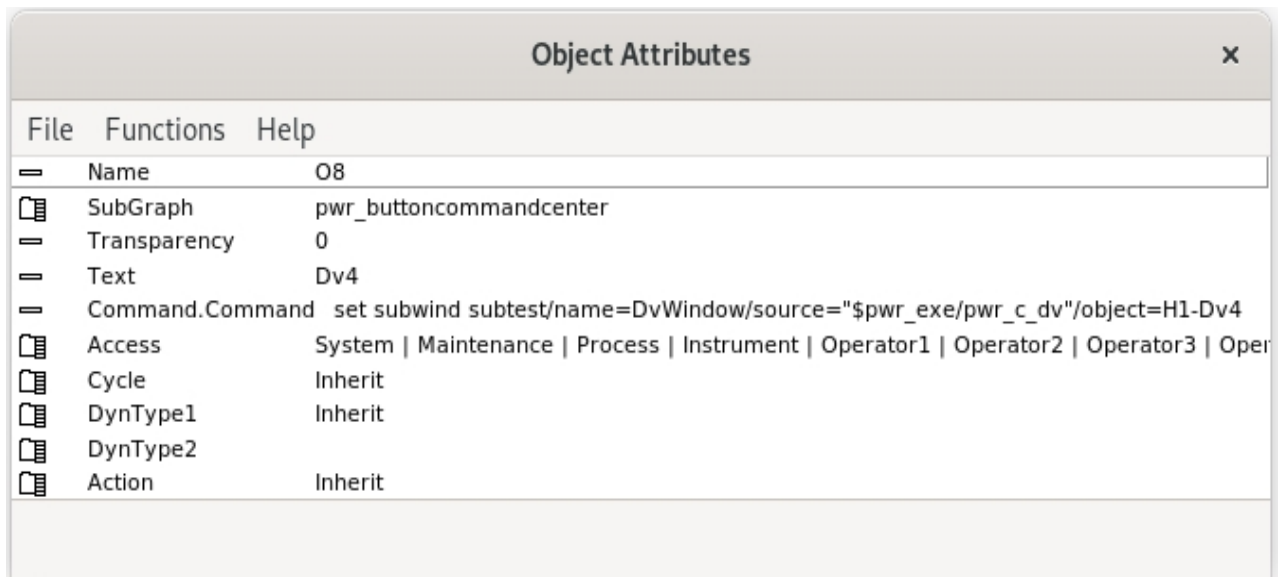
Kommandot som används är

```
set subwindow 'grafnamn' /name= /source= [/object=]
```

där grafnamn är namnet på den graf där window-objektet ligger. I /name specificeras namnet på window-objektet, och i /source anges filnamnet på den graf som ska visas. /objekt används när det är en objektsbild som ska visas, och specificerar database-objektet som bilden ska kopplas till.

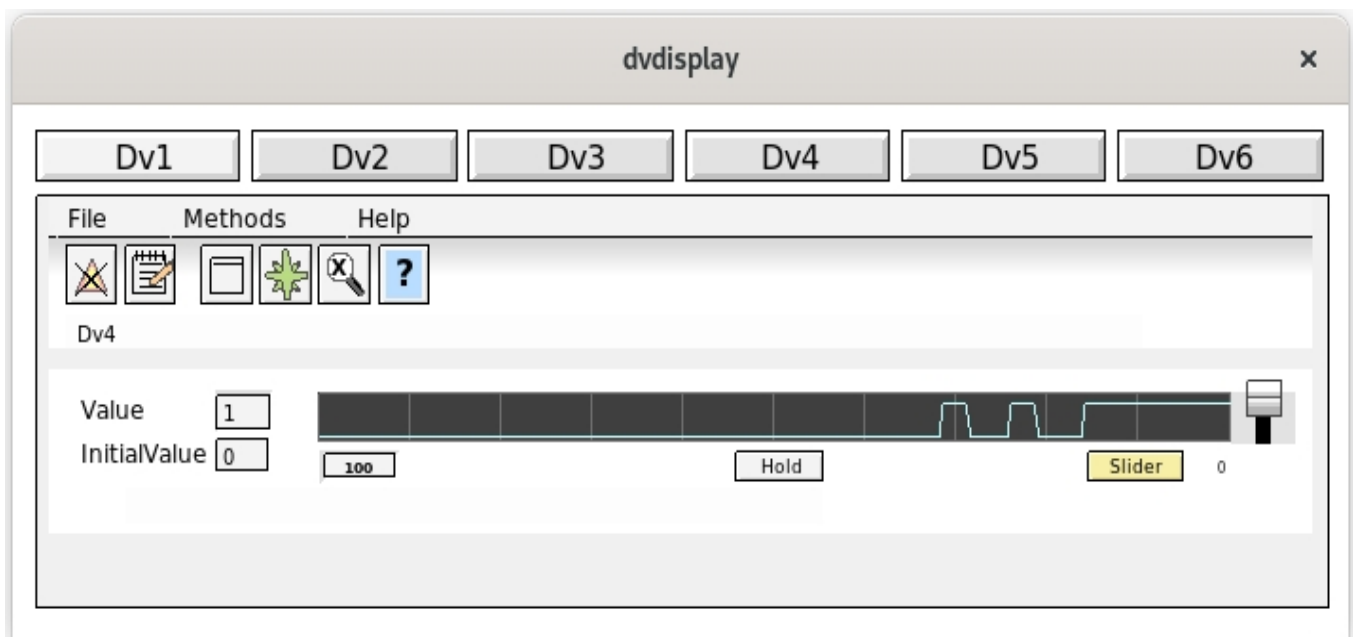
I följande exempel visas objektsbilden för ett antal Dv i ett window-objekt. I window-objektet kan man lägga in en startbild, som konfigureras på normalt sätt, men man kan även låta window-objektet vara tomt tills man har valt en graf från någon av knapparna. I det fallet bör man lägga in "\_no\_" i Window.FileName för att undvika felutskrift.

Grafen som innehåller window-objektet sparas med namnet 'dvdisplay' och innehåller förutom window-objektet 6 tryckknappar av typen CommandButtonCenter. Window-objektet namnges till DvWindow genom att aktivera Edit/Change Name i menyn. I figuren nedan visas objekteditorn för en tryckknapp med kommandot inlagt. Notera att filnamnet måste omgärdas av citations-tecken eftersom det innehåller ett slash tecken.



**Fig Kommandot 'set subwind' i en tryckknapp**

Den slutgiltiga grafen ser ut så här med Dv4 vald.

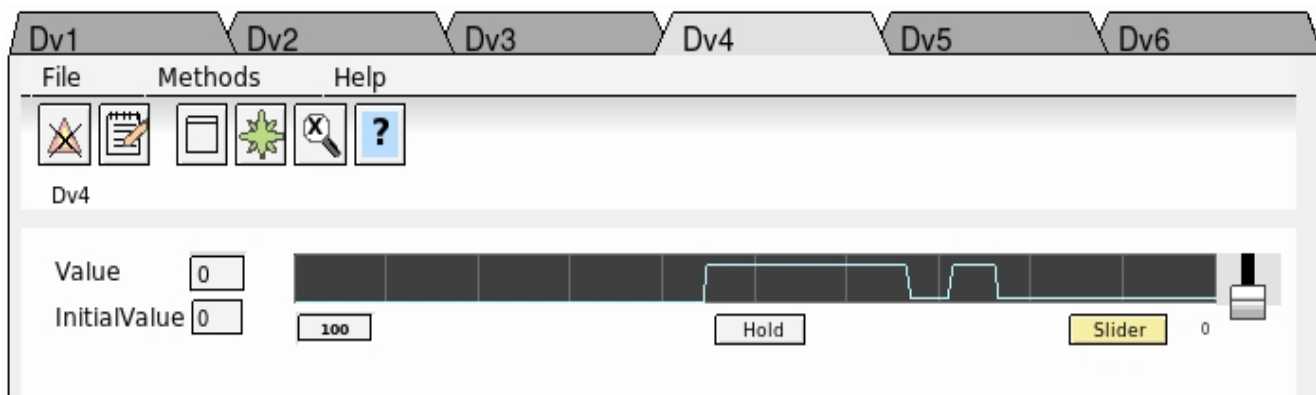


**Fig Objektsbilden för Dv4 visas efter klick på knappen Dv4.**

#### 4.4.2 TabbedWindow

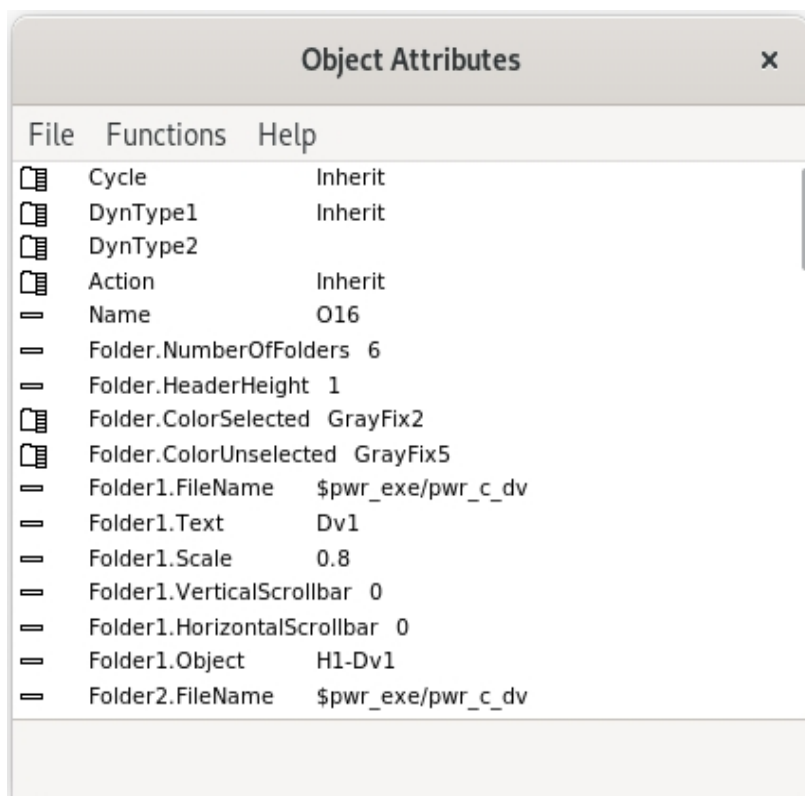
TabbedWindow är ett Window objekt med flik-register. För varje flik anges en separat graf, och när fliken aktiveras visas en angivna grafen i fönstret.

Man kan visa objekts-bilder eller andra grafer i ett tabbed window. Vi ska här titta på hur man kan visa en uppsättning objektsbilder, precis som i exemplet för window objektet ovan.



**Fig Ett flikfönster objekt**

Vi ska ha 6 stycken flikar och varje flik ska visa objektbilden för ett Dv-objekt. Efter att ha skapat TabbedWindow objektet, öppnar vi upp objekteditorn. Här sätter vi Folder.NumberOfFolders till 6. Vi ändrar också Folder.HeaderHeight till 1 för att öka storleken på flikarna. Sedan fyller vi i FileName, Text och Object för de 6 första flikarna. FileName är filnamnet på objektbilden för Dv, dvs \$pwr\_exe/pwr\_c\_dv. Text sätter vi till Dv1, Dv2 osv, och Object innehåller objektnamnet på respektive Dv, H1-Dv1, H2-Dv2 osv. Vi ökar också textstorleken i flikarna genom att välja ut window-objektet och sätta TextSize till 14 i verktygspanelen.



**Fig Objekteditorn för flikfönstret**

#### 4.4.3 Slider

En slider eller dragregel är ett flyttbart objekt som är kopplat till en analog signal i databasen. Läget på slidern bestämmer signalvärdet. Slidern flyttas antingen horisontellt eller vertikalt mellan två ändpunkter. Läget för ändpunkterna kan bestämmas med två olika

metoder, antingen genom att man lägger ett speciellt bakgrundsobjekt av typen SliderBackground, eller genom att man specificerar koordinaterna för ändpunkterna i objekteditorn.

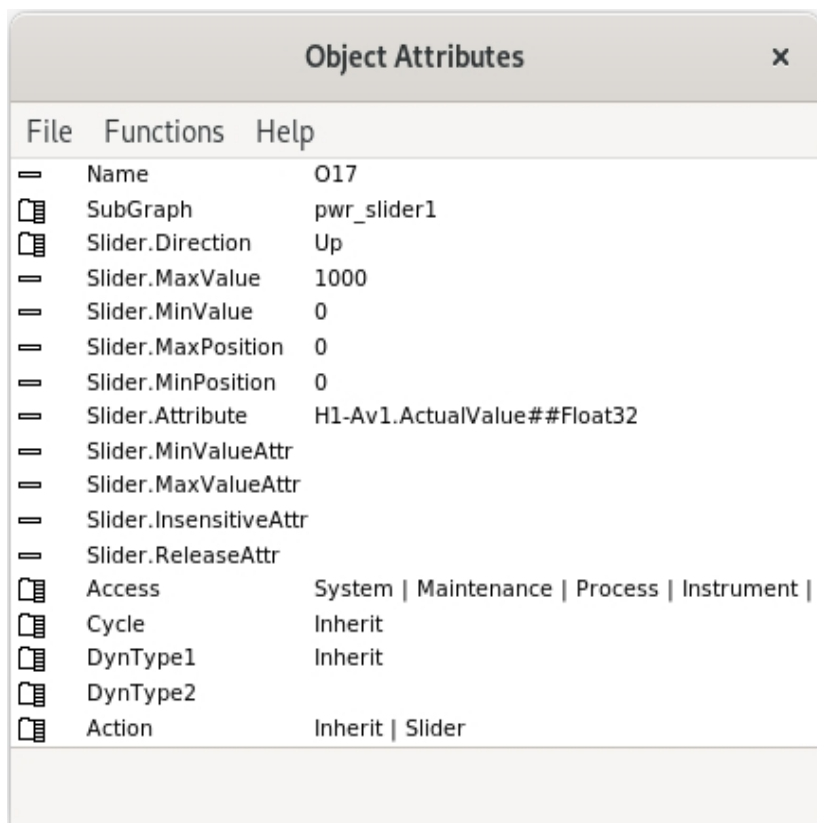
### Slider med bakgrundsobjekt

Under Slider mappen i subgraf-paletten finns slider och slider-background objekt. Vi väljer en SliderBackground1 och placerar ovanpå denna en Slider1. För att få en horisonell slider roterar vi båda objekten 270 grader.



**Fig Slider med bakgrundsobjekt**

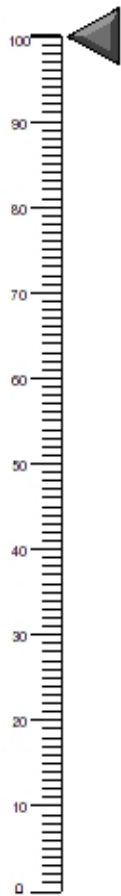
Nästa steg är att koppla slidern till en analog signal i databasen. Vi väljer Av objektet H1-Av1 i planhierarkin och klickar med Ctrl+Dubbelklick MB1 på slider-objektet. Nu återstår att ange vilket signal-område sliderns rörelse motsvarar. Signalens går mellan 0 - 1000 och vi sätter Slider.MaxValue till 1000 i objektseditorn. Notera att vi inte behöver ange min- och max-position eftersom detta ligger inbyggt i sliderbackground objektet.



**Fig Området för signalen anges i Slider.MinValue och Slider.MaxValue**

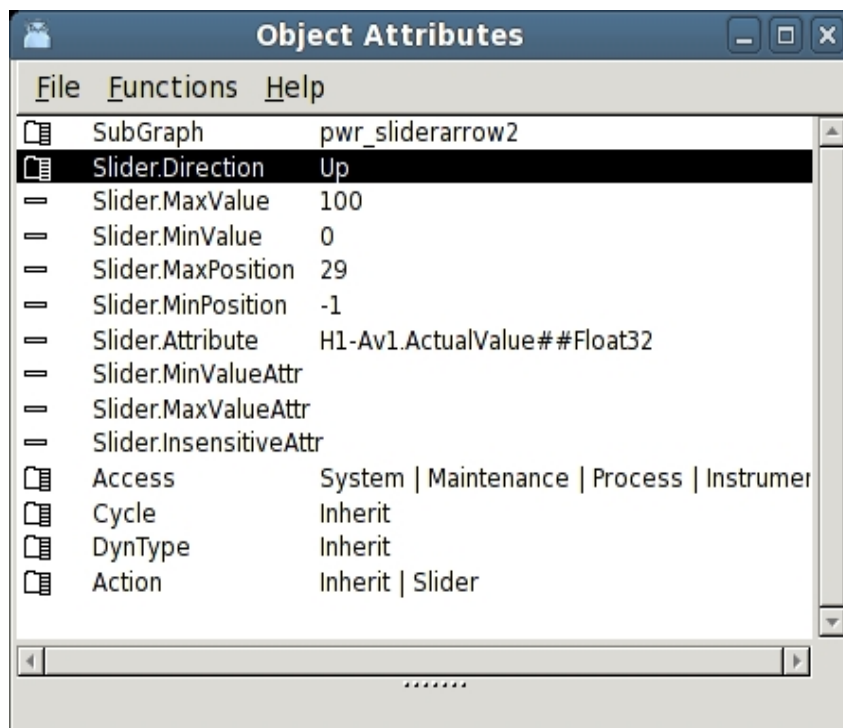
### Slider utan bakgrundsobjekt

För en slider utan bakgrunds-objekt måste man ange sliderns rörelseriktning, samt min och max-positionen för rörelsen.



**Fig Slider utan bakgrundsobjekt**

Vi skapar en slider av typ Arrow2 som ska kunna flyttas utmed en skala mellan 0 och 100. 100 ligger på y-koordinat 0 och 0 på y-koordinat 30. Slidern ska placeras vid den lägsta y-koordinaten, vilket alltså är vid 100 på skalan. Sedan ska vi ange sliderns rörelseområde, och då mäta koordinaten för slidern-objektets överkant. Det ger y-koordinaterna -1 i sliderns övre position och 29 i den undre positionen. Vi lägger in dessa i Slider.MinPosition (-1) och Slider.MaxPosition (29). Vidare sätter vi Slider.Direction till Up eftersom en rörelse uppåt i bilden ger ökande värde.



**Fig Objekteditorn för slidern**

En horisontell slider placeras på den lägsta x-koordinaten, dvs längst till vänster i bilden.  
I exemplet nedan är Slider.Direction satt till Left.

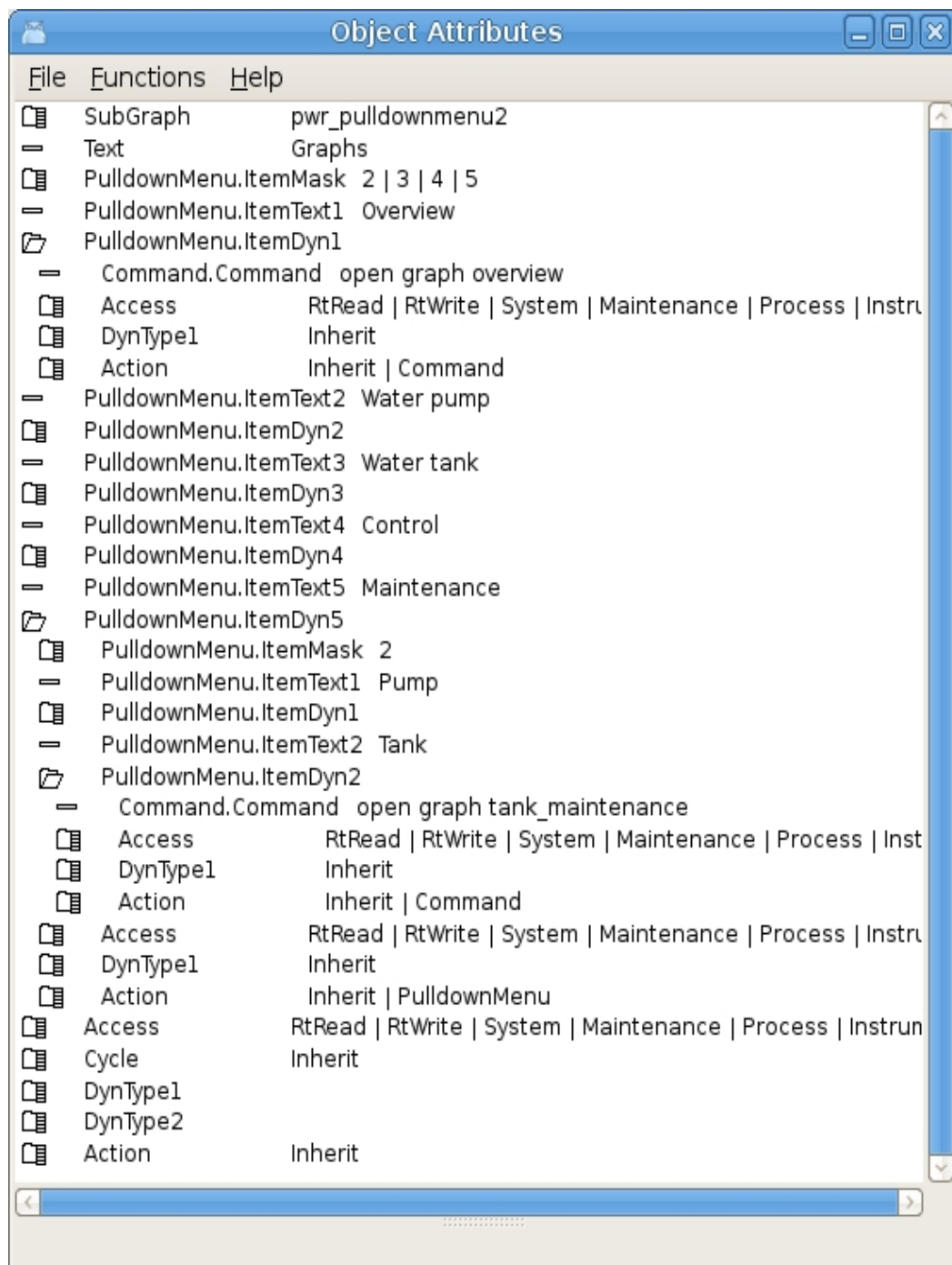


Fig Horisontell slider

#### 4.4.4 Stapel

En stapel visar värdet för en analog signal i form av en stapel. Stapeln återfinns i Analog/Bar i subgraf-paletten.

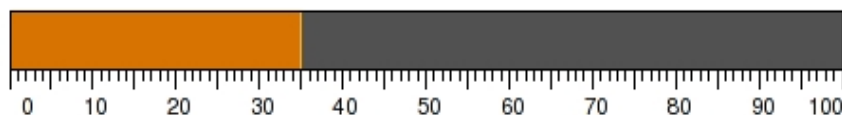
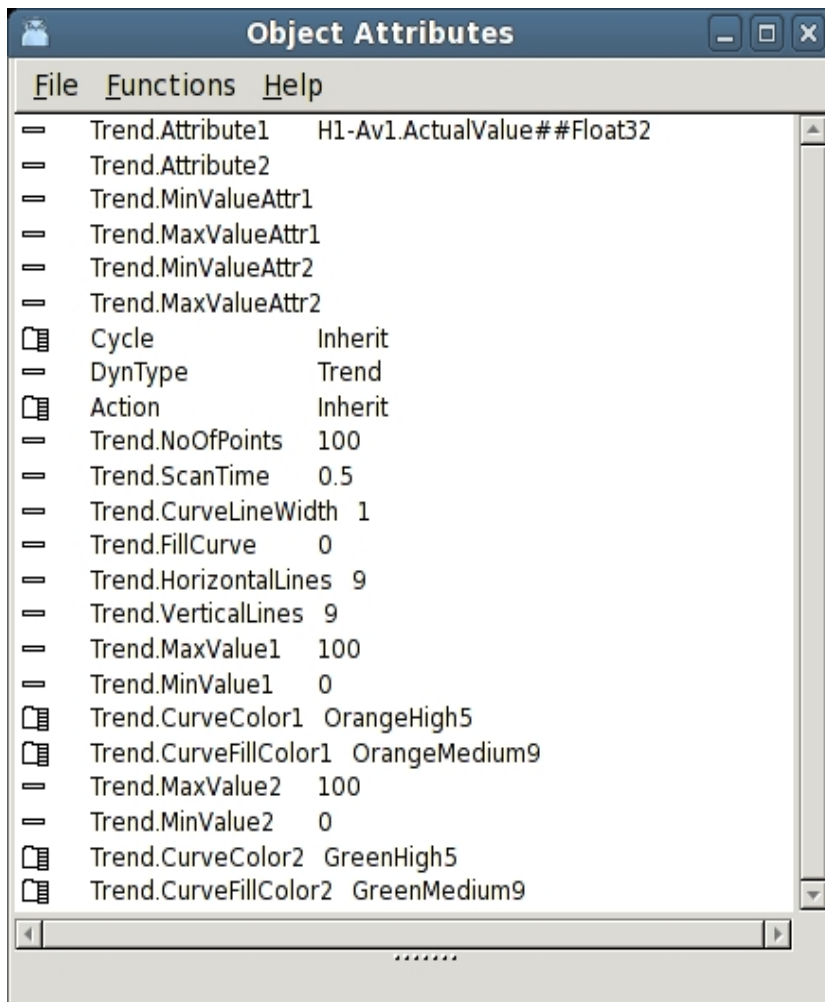


Fig Stapel

En stapel konfigureras genom att koppla den till en analog signal och lägga in området för stapeln i Bar.MinValue och Bar.MaxValue. Axis-objektet ingår inte i bar-objektet.



**Fig Attribut för ett bar-objekt**

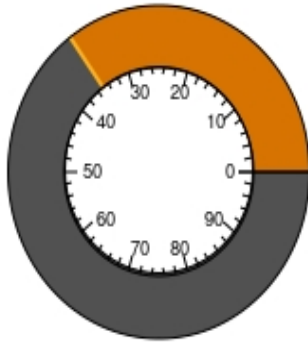
Om stapeln ingår i en objektsbild är ofta området olika för olika instanser av objekt i databasen som bilden visar. Man kan då koppla Bar.MinValueAttr och Bar.MaxValueAttr till de attribut i databas-objekt som innehåller min och max-värde för signalens område.

Notera att en stapel även kan åstadkommas med en rektangel som är grupperad och försedd med dynamiken FillLevel. Enda skillnaden är att bar-objektet har en avgränsnings-linje med separat färg i mellan stapeln och bakgrunden.

#### **4.4.5 Cirkulär stapel**

En BarArc är en stapel med cirkulär form. Den återfinns i Analog/BarArc i subgraf-paletten.





**Fig Cirkulär stapel**

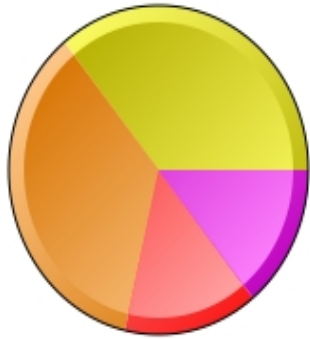
Utöver egenskaperna för en linjär stapel, kan även stapelbredd, startvinkel och utbredningsvinkel anges.

Object Attributes			×
File	Functions	Help	
Bar.Attribute		H1-Av1.ActualValue##Float32	
Bar.MinValueAttr			
Bar.MaxValueAttr			
Cycle		Inherit	
DynType1		Bar	
DynType2			
Action		Inherit	
Name		O0	
BarArc.MaxValue		100	
BarArc.MinValue		0	
BarArc.Angle1		0	
BarArc.Angle2		360	
BarArc.BarWidth		1.5	
BarArc.Direction		0	
BarArc.Value		35	
BarArc.BarColor		OrangeHigh7	
BarArc.BorderColor		OrangeHigh5	
BarArc.BorderWidth		1	
Dynamic			

**Fig Attribut för ett bararc-objekt**

#### 4.4.6 Tårtdiagram

Tårtdiagrammet är uppdelat i 12 sektorer med olika färger, där varje sektor kopplas till en analog signal som bestämmer storleken på sektorn.



**Fig Tårtdiagram**

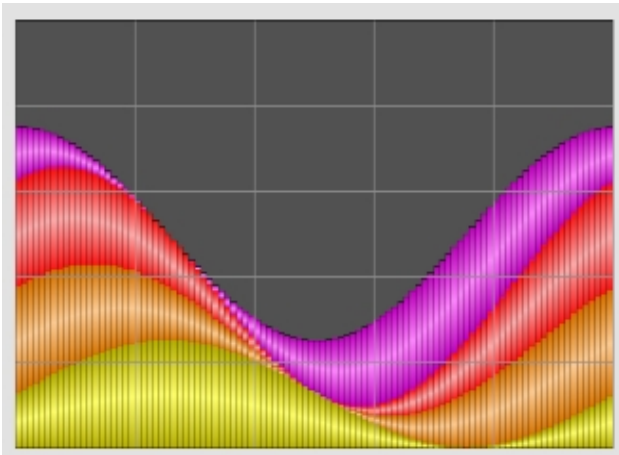
I exemplet ovan med fyra sektorer är de fyra första sektorerna kopplade till signaler, och Pie.Sectors är satt till 4.

Object Attributes			x
File	Functions	Help	
==	Pie.Attribute1	H1-Av1.ActualValue##Float32	
==	Pie.Attribute2	H1-Av2.ActualValue##Float32	
==	Pie.Attribute3	H1-Av3.ActualValue##Float32	
==	Pie.Attribute4	H1-Av4.ActualValue##Float32	
==	Pie.Attribute5		
==	Pie.Attribute6		
==	Pie.Attribute7		
==	Pie.Attribute8		
==	Pie.Attribute9		
==	Pie.Attribute10		
==	Pie.Attribute11		
==	Pie.Attribute12		
==	Pie.FixRange	0	
📁	Cycle	Inherit	
📁	DynType1		
📁	DynType2		
📁	Action	Inherit	
==	Name	O2	
==	Pie.Angle1	0	
==	Pie.Angle2	360	
==	Pie.ShadowWidth	5	
==	Pie.ShadowContrast	2	
==	Pie.GradientContrast	4	
📁	Pie.Gradient	DiagonalLowerRight	
📁	Pie.Relief	Up	
==	Pie.Sectors	4	
==	Pie.MinValue	0	
==	Pie.MaxValue	100	
📁	Pie.SectorColor1	YellowHigh5	
📁	Pie.SectorColor2	OrangeHigh5	
📁	Pie.SectorColor3	RedHigh5	
📁	Pie.SectorColor4	MagentaHigh5	
📁	Pie.SectorColor5	BlueHigh5	
📁	Pie.SectorColor6	SeaBlueHigh5	
📁	Pie.SectorColor7	GreenHigh5	
📁	Pie.SectorColor8	YellowMedium5	
📁	Pie.SectorColor9	OrangeMedium5	
📁	Pie.SectorColor10	RedMedium5	
📁	Pie.SectorColor11	MagentaMedium5	
📁	Pie.SectorColor12	BlueMedium5	

Fig Attribut för ett pie-objekt

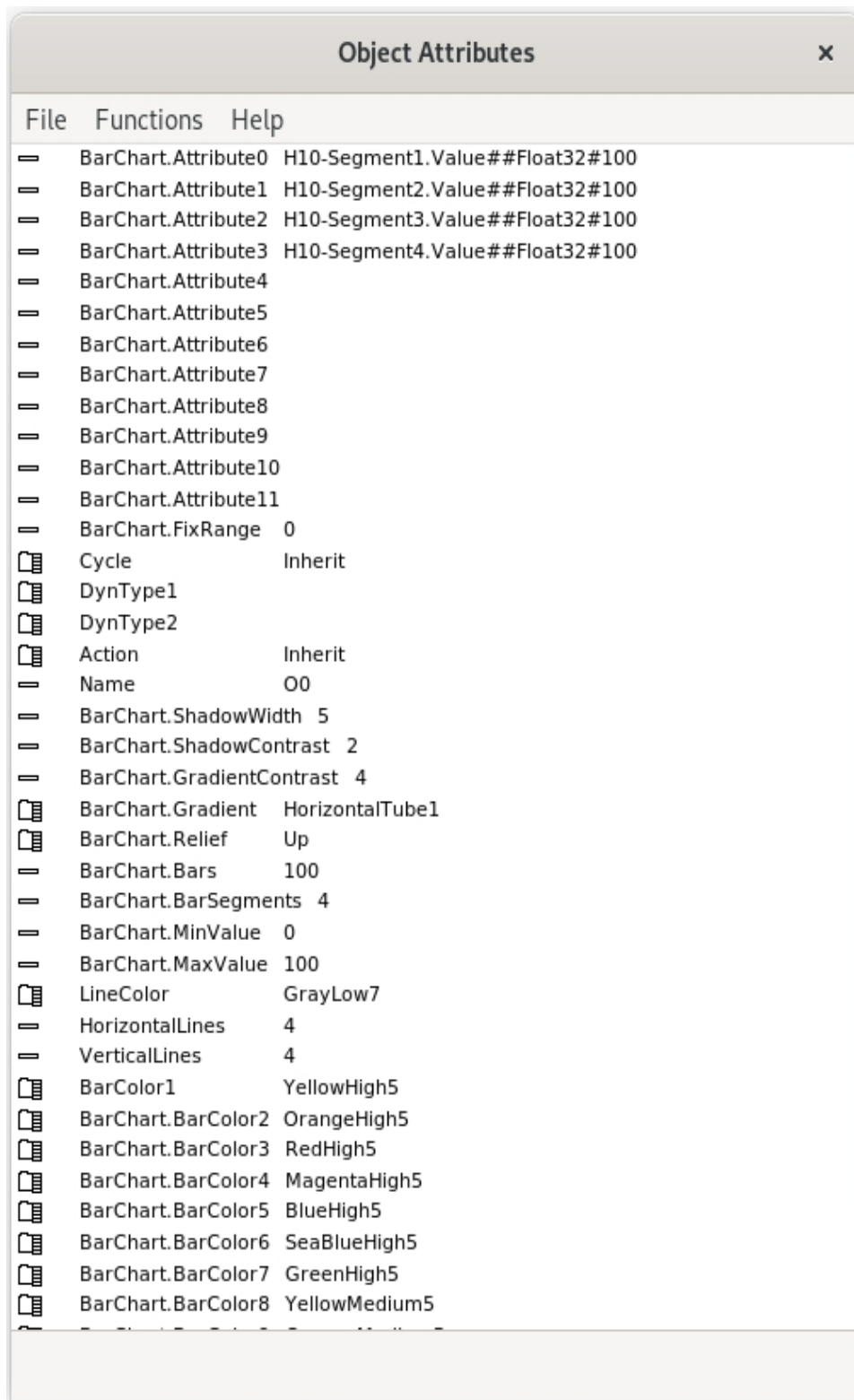
#### 4.4.7 Stapeldiagram

Stapeldiagrammet kan delas upp i 12 segment med olika färg, där varje segment kopplas till en vektor av analoga signaler som anger storleken på segmenten.



**Fig BarChart**

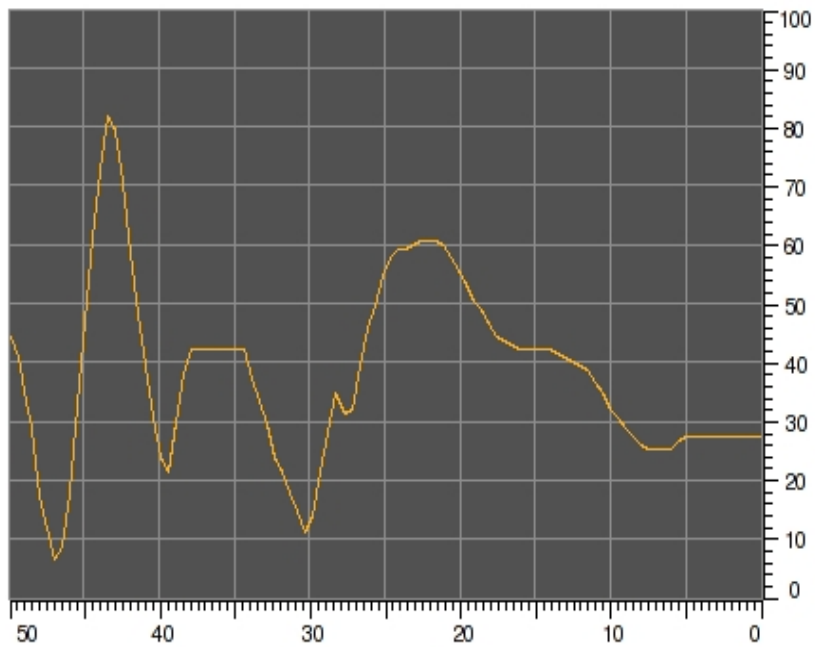
I exemplet ovan med 100 staplar och 4 segment är de första fyra segmenten kopplade till vektorer med 100 element. BarChart.Bars är satt till 100 och BarChart.Segments är satt till 4.



**Fig Pie object attributes**

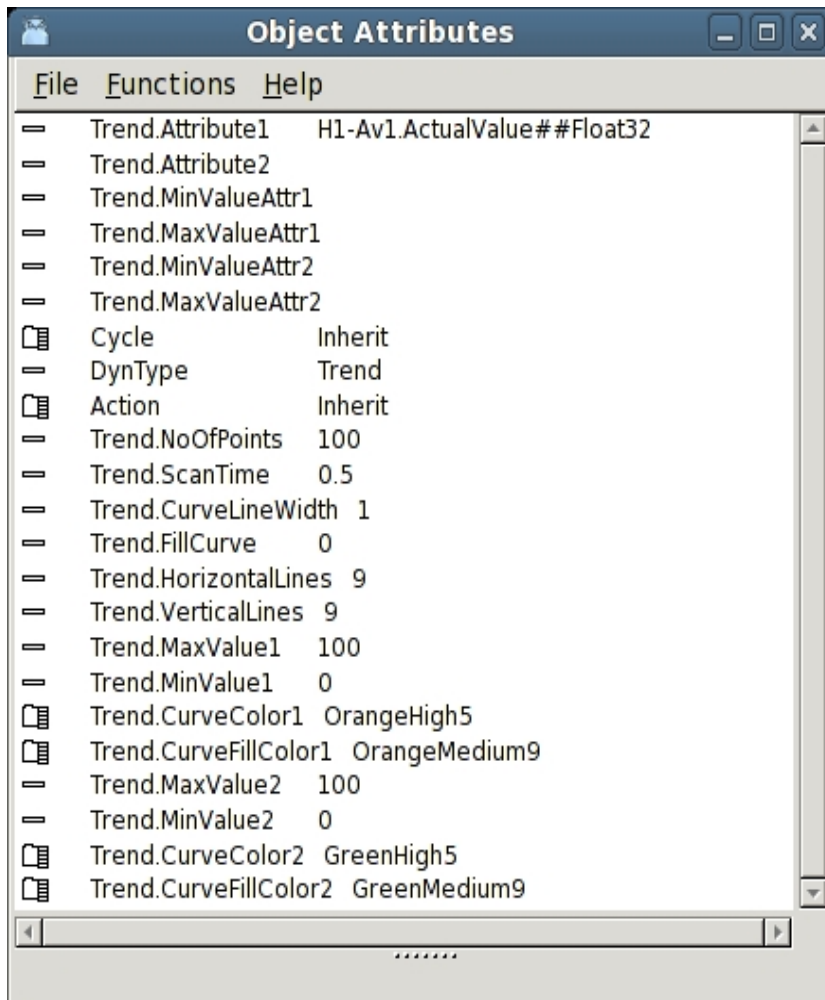
#### 4.4.8 Trend

En trendkurva visar värdet för en eller ett par analog signaler under den senaste tiden. Även heltal eller digitala signaler kan visas i trendkurvor. Trend ligger under Analog mappen in subgraf-paletten.



**Fig Trendkurva**

Ovanstående trend-objekt är konfigurerat med en analog signal. Antalet vertikala och horisontella linjer har ökats genom att sätta `Trend.HorizontalLines` och `Trend.VerticalLines` till 9. Skalorna ingår inte i trend-objektet, utan har skapats med `Axis` objekt. Vertikala axeln har området 0-100 eftersom `Trend.MinValue1` är 0 och `Trend.MaxValue1` är 100. Tidsaxelns område blir 50 s då `Trend.ScanTime` är 0.5 s och antalet lagrade punkter, `Trend.NoOfPoints`, är 100.

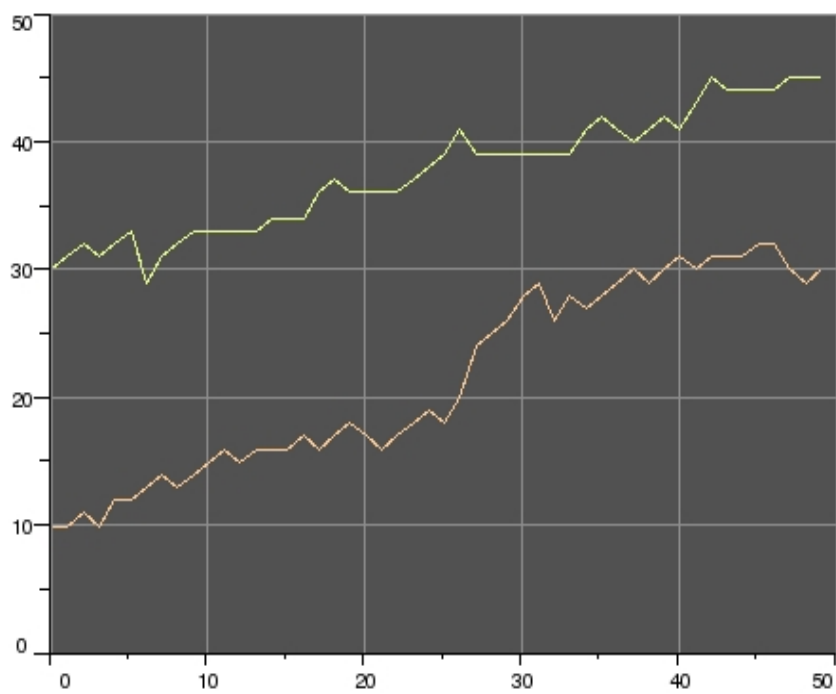


**Fig Attribut för ett trend-objekt**

Om trendkurvan ingår i en objektsbild är ofta området olika för olika instanser. Man kan då koppla Trend.MinValueAttr och Trend.MaxValueAttr till attribut i databas-objektet som innehåller min och max-värde för signalens område.

#### 4.4.9 XYCurve

XYCurve ritar en kurva som har specificerats med två vektorer som innehåller x resp y-värden för kurvan. Det finns en speciell XYCurve Klass som kan användas. Det innehåller de attribut som behövs och två vektorer med 100 element.



**Fig XYCurve**



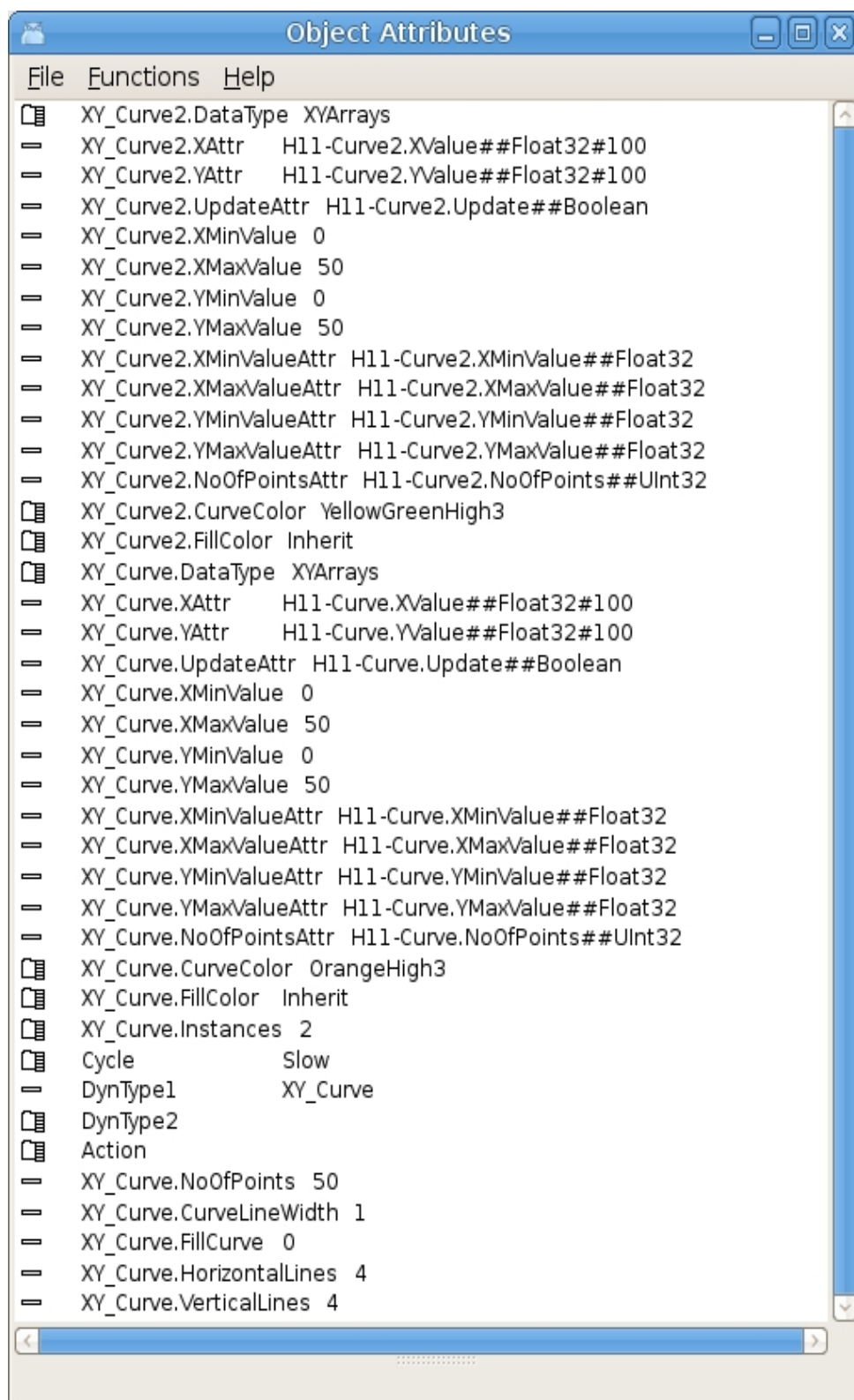
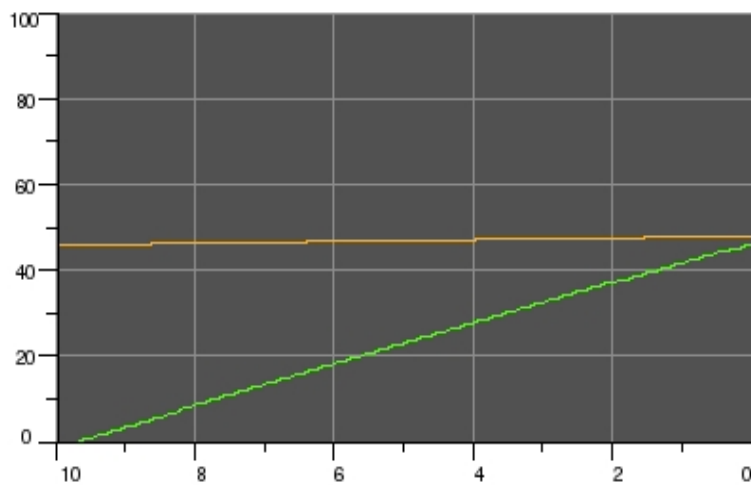


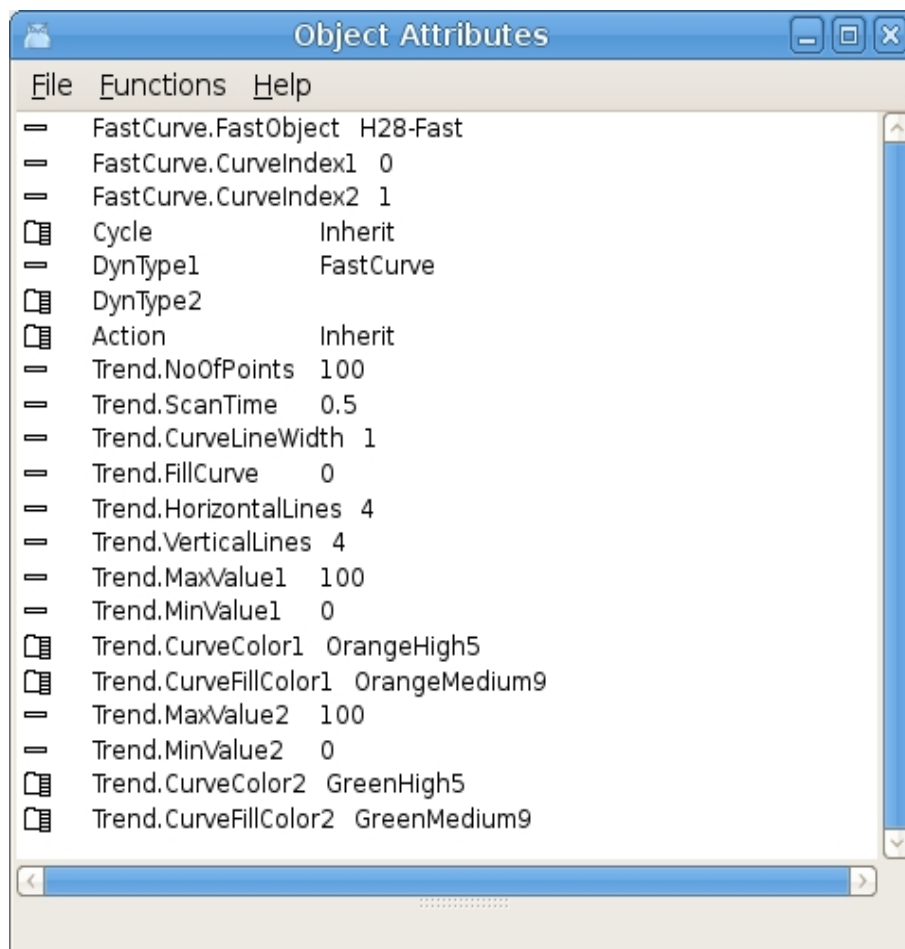
Fig Attribut för XYCurve-objektet

#### 4.4.10 FastCurve

FastCurve visar en kurva specificerad med ett DsFastCurve objekt. En DsFastCurve kan innehålla up till 10 kurvor, men FastCurve objektet kan endast visa två av dem. Vilka kurvor som visas anges i attributet FastCurve.FastIndex.



**Fig FastCurve**



**Fig Attribut för ett FastCurve-objekt**

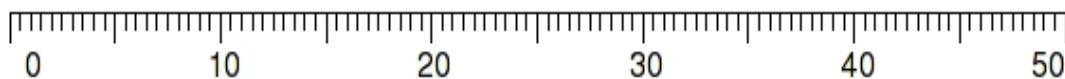
#### 4.4.11 Axis och AxisArc

Ett axis-objekt ritat ut en skala med ett visst område. Axis objekt har ingen dynamik utan skalan är fast. Det finns två varianter, en rak skala och en cirkulär eller elliptisk skala.

##### Rak skala

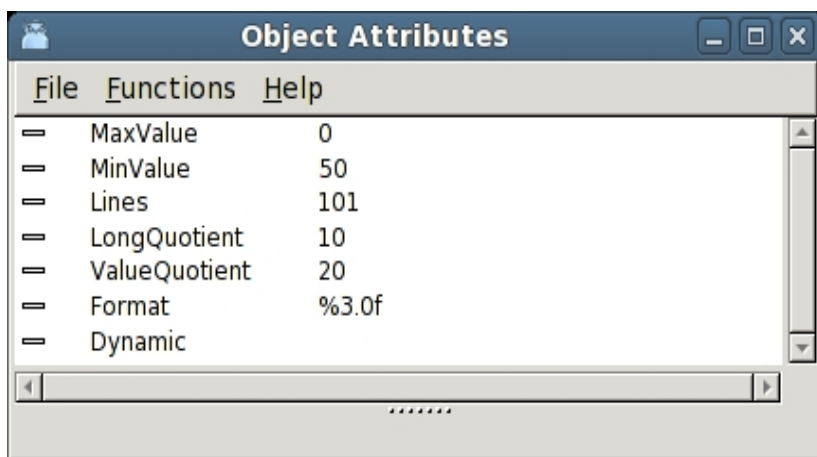
En rak skala är ett Axis-objekt som hämtas från mappen Analog och skalas till önskad storlek.

Storleken på siffrorna justeras med TextSize menyn i verktygspanelen. Skalans område bestäms av MinValue och MaxValue som i exemplet nedan är satt till 0 och 50. Vill man att skalan ska gå åt andra hållet kan man skifta värdena i MinValue och MaxValue. Det finns inte något krav att MaxValue ska vara större än MinValue.



**Fig Rak skala**

Antalen streck i skalan bestäms av attributet Lines, som är satt till 101, och eftersom var 10:de streck ska vara lite längre är LongQuotient satt till 10. På var 20:de streck ska läget markeras med ett värde, alltså sätts ValueQuotient till 20.



**Fig Attribut för ett axis-objekt**

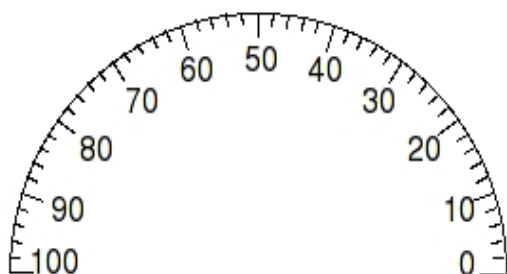
Man kan även påverka formateringen av värdena i Format, och vill man inte ha några värden utskrivna tar man bort innehållet i Format. Formatet ska vara av typen %5.1f där 5 makerar totalt antal tecken inklusive decimal-punkt, och 1 antal decimaler.

### Elliptisk skala

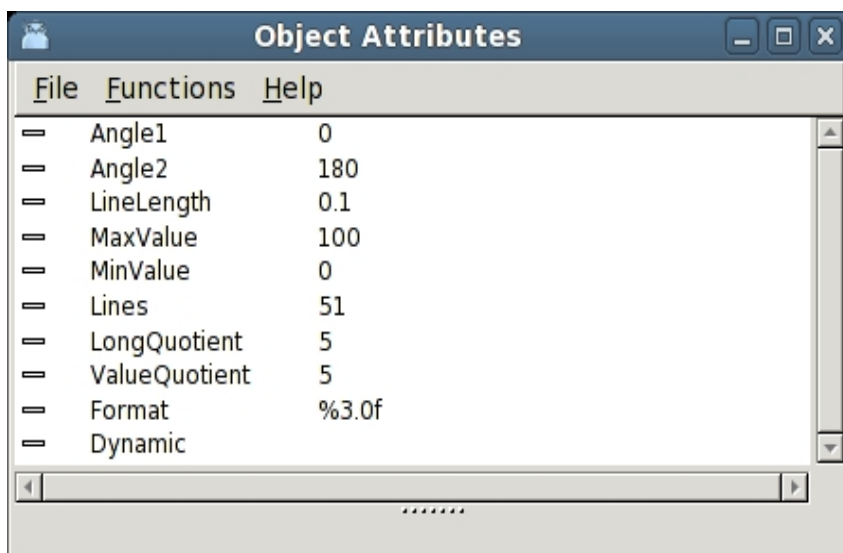
En cirkulär eller elliptisk skala utgörs av ett AxisArc objekt. Objektet skalas till lämplig storlek, och textstorleken anpassas med TextSize i vertygspanelen.

Skalan utgörs av ett cirkel-segment, genom att i objects-editorn ange Angle1 och Angle2 påverkar man segmentets läge och storlek. Angle1 är vinkeln från den horisontella axeln till segmentets början, och Angle2 är vinkeln för segmentets utbredning.

Man kan även påverka streck-längden i LineLength. Där anges längden relativt radien, så en LineLength på 0.1 ger en längd på de längre strecken på 10 % av radien.



**Fig Elliptisk skala**



**Fig Attribut för ett axisarc-objekt**

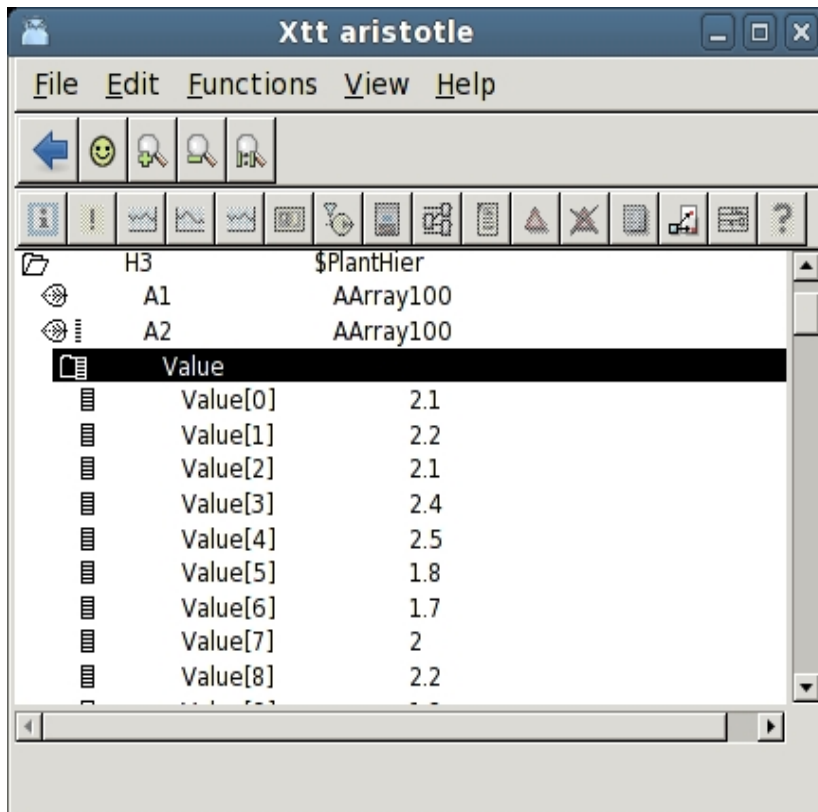
#### 4.4.12 Tabell

Tabell-objektet är en överskådlig uppställning av data, uppdelad i rader och kolumner, ofta försedd med rubrik. Tabell återfinns under Other mappen i subgrafs-paletten.

10.2	2.10
12.5	2.20
9.4	2.10
9.8	2.40
12.4	2.50
7.8	1.80
9.2	1.70
10.1	2.00
8.9	2.20
8.9	1.80

**Fig Tabell utan rubrik**

Data för en kolumn är vanligtvis samlad i ett vektor-attribut i databasen. Det finns ett antal klasser med vektorer av olika typ, t ex AArray100 och AArray500 som innehåller vektorer med 100 resp 500 element av typen Float32. Motvarande DArray, IArray och SArray objekt finns för data av typen boolean, integer och strängar. I exemplet nedan har vi samlat data i 3 st AArray100 en en SArray100.

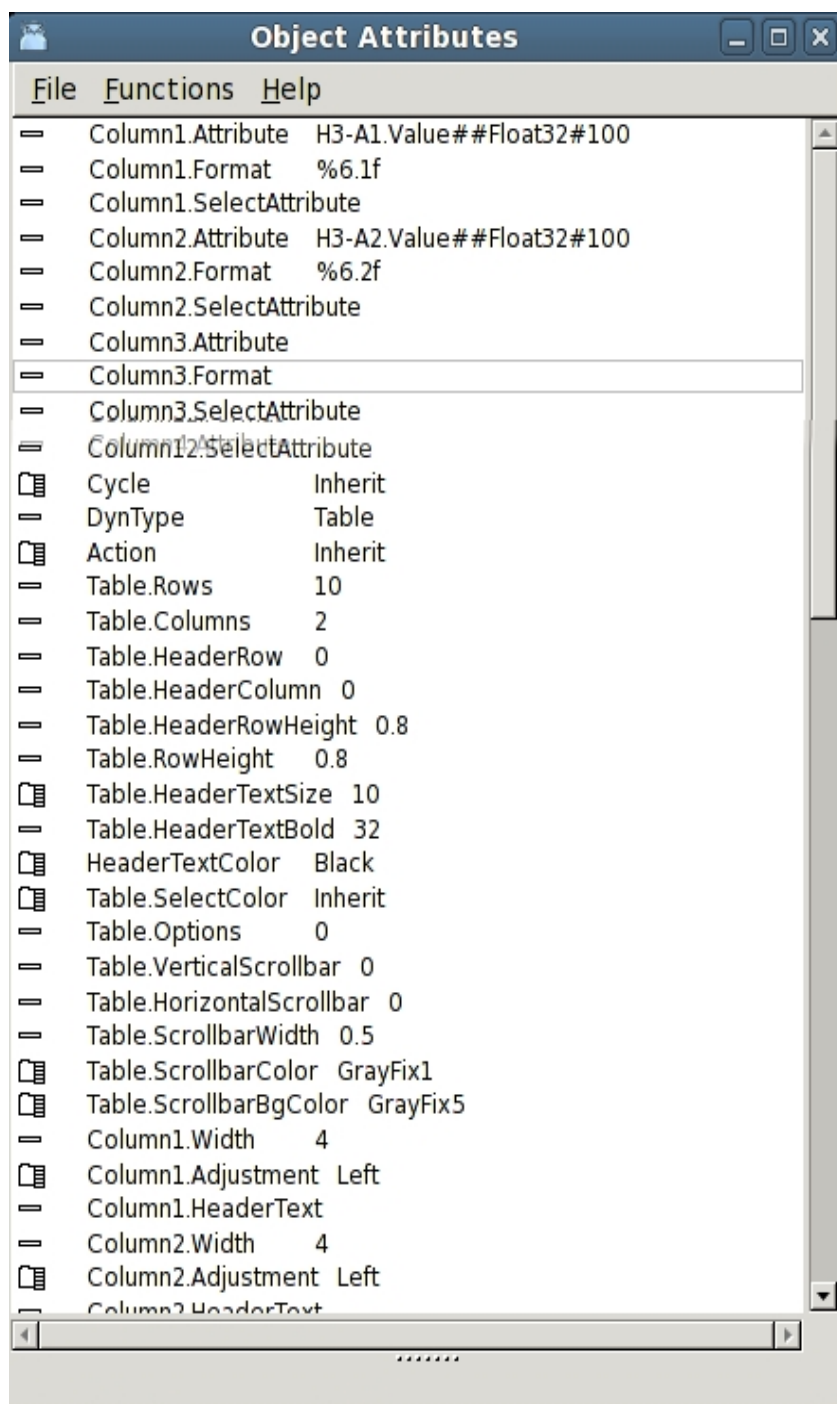


**Fig Array-objekt som innehåller tabell-data**

Om vi tittar på den enkla tabellen i Fig Tabell utan rubrik ovan, visas två kolumner. Antalet kolumner anges i Table.Columns. Column1.Attribute och Column2.Attribut är kopplade till var sitt vektor attribut i databasen som innehåller de värden som ska visas upp i respektive kolumn, H3-A1.Value och H3-A2.Value. Syntaxen är H1-A1.Value##Float32#100 vilket innebär att attributet är en array av typen Float32 med 100 element. Man måste även specificera formateringen av värdena i kolumnerna genom att fylla i Column1.Format och Column2.Format. '%6.1f' innebär ett flyttal med 6 tecken och en decimal.

En del andra detaljer är ändrade på tabellen,

- fonten är satt till Lucida Sans, genom att välja ut tabellen och ange Lucida Sans i verktygspanelen.
- Textstorleken är ökad till 12 med Textsize i verktygspanelen.
- Höjden på raderna har anpassats till den större texten genom att Table.RowHeight är 0.8.



**Fig Attribut för tabell-objekt**

Figuren nedan visar en tabell med rubrik. Här är Table.HeaderRow satt till 1. Textstorleken för rubriken, Table.HeaderTextSize är satt till 12 och Table.HeaderTextBold är satt till 0 för att undvika fet text.

Length	Width
10.2	2.10
12.5	2.20
9.4	2.10
9.8	2.40
12.4	2.50
7.8	1.80
9.2	1.70
10.1	2.00
8.9	2.20
8.9	1.80

**Fig Tabell med header row**

I tabellen nedan har en vertikal skroll-list adderats genom att sätta Table.VerticalScrollbar till 1. Notera att rubrik-raden inte skrollas utan visas hela tiden.

Tabell-objektet visar 3 kolumner, den första är kopplad till en sträng-vektor av typen SArray100.Value, och har Column1.Format satt till %s. Första kolumnen är även markerad som en header kolumn, genom att Table.HeaderColumn är satt till 1. Det innebär att gränslinjen mellan första och andra kolumnen är lite tjockare.

Object	Length	Width
N388	7.8	1.80
N324	9.2	1.70
N521	10.1	2.00
N858	8.9	2.20
N475	8.9	1.80
N549	12.7	1.60
N521	11.2	2.00
N755	14.1	2.30
N398	13.3	2.10
N497	13.9	2.50
N671	12.1	1.80

**Fig Tabell med vertikal skroll-list och header column**

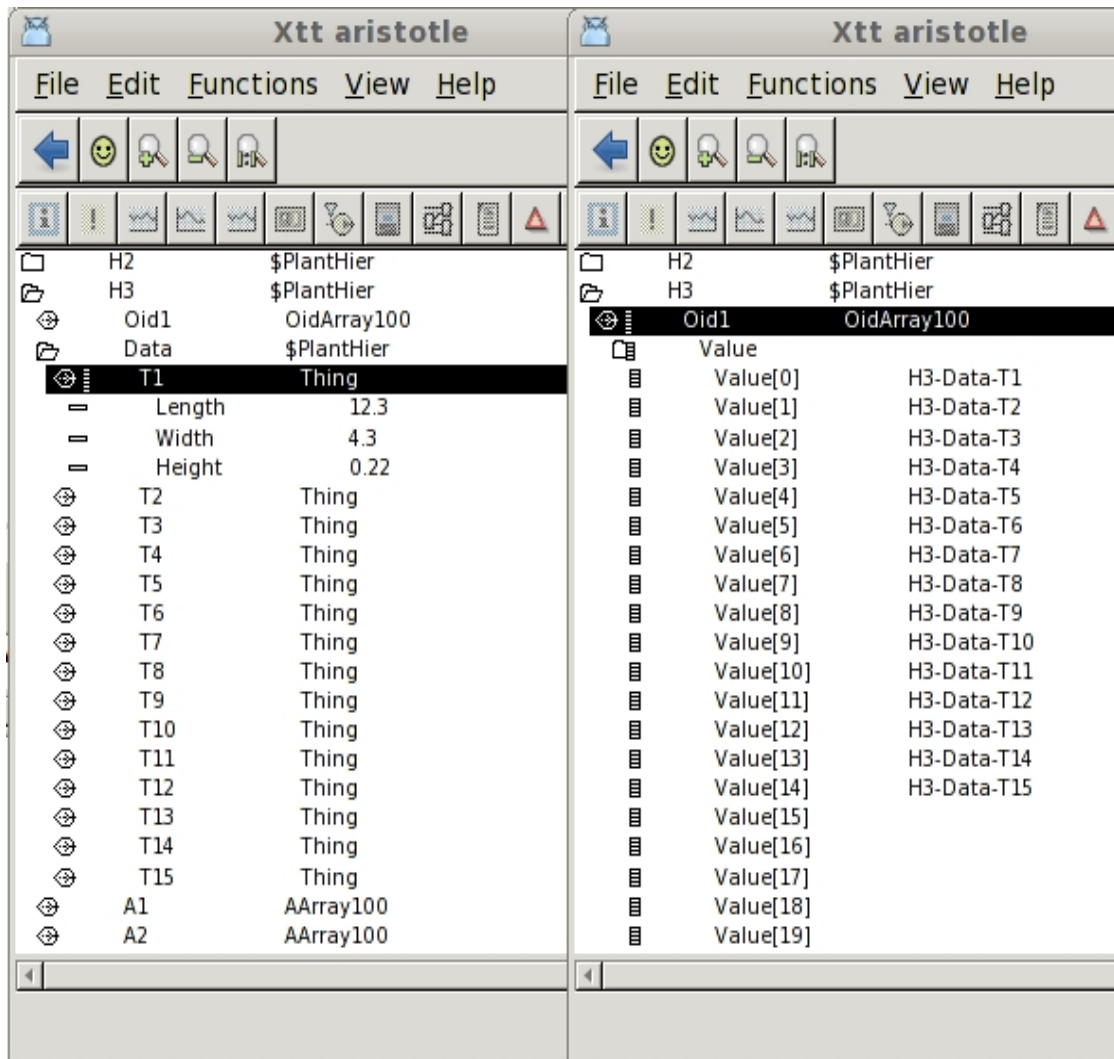
I figuren nedan visas hur en header column fungerar med en horisontell skroll-list. Den horisontella skroll-listen fås genom att sätta Table.HorizontalScrollbar till 1. Notera att skroll-listen inte omfattar den första kolumnen, utan den första kolumnen är alltid synlig.

Object	Length	Width	Height
N034	2.4	2.50	0.2
N388	7.8	1.80	0.2
N324	9.2	1.70	0.2
N521	10.1	2.00	0.2
N858	8.9	2.20	0.3
N475	8.9	1.80	0.3
N549	12.7	1.60	0.2
N521	11.2	2.00	0.2
N755	14.1	2.30	0.2
N398	13.3	2.10	0.2
N497	13.9	2.50	0.2
N671	12.1	1.80	0.2

**Fig Tabell med horisontell skroll-list och header column**

## Tabell som visar objekt av samma typ

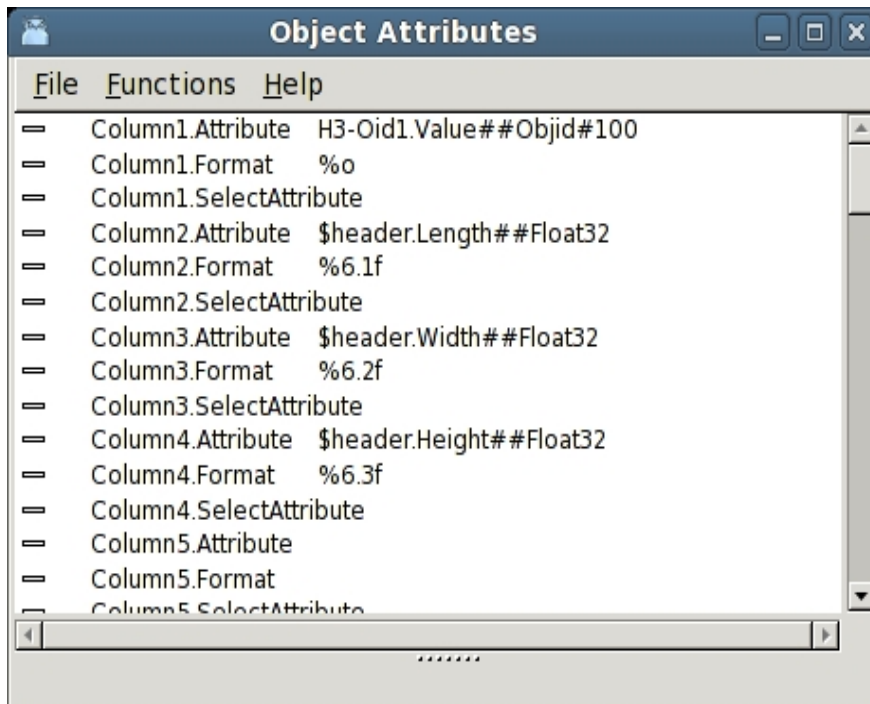
Nu ska vi titta på ett exempel där data inte ligger organiserat i vektorer, utan i ett antal objekt av samma klass. I anläggningshierarkin under H3-Data ligger ett antal objekt av klassen Thing som innehåller attributen Length, Height och Width. Vi vill visa innehållet i alla Thing-objekt i en tabell, med ett objekt på varje rad. För att kunna göra det måste vi skapa en vektor av typen Objid som innehåller identiteten på varje objekt som ska visas. Alltså skapar vi ett objekt av typen OidArray och lägger in identiteten för all Thing objekt i denna.



**Fig Objekt vars innehåll visas i tabellen till vänster, array av objid till höger**

Vi skapar en tabell med fyra kolumner. Den första kolumnen ska visa namnet på objektet, den andra attributet Length i respektive objekt, den tredje Width och den fjärde Height. Tabellen konfigureras genom att vektorn med objektsidentiteterna läggs in i Column1.Attribute. I Column2.Attribute anges \$header.Length##Float32. \$header pekar på objektet som visas i header kolumnen, dvs den första kolumnen. På samma sätt fyller vi i Column3.Attribute med \$header.Width##Float32 och Column4.Attribute med \$header.Height##Float32.





**Fig Attributen för kolumnerna i tabellen**

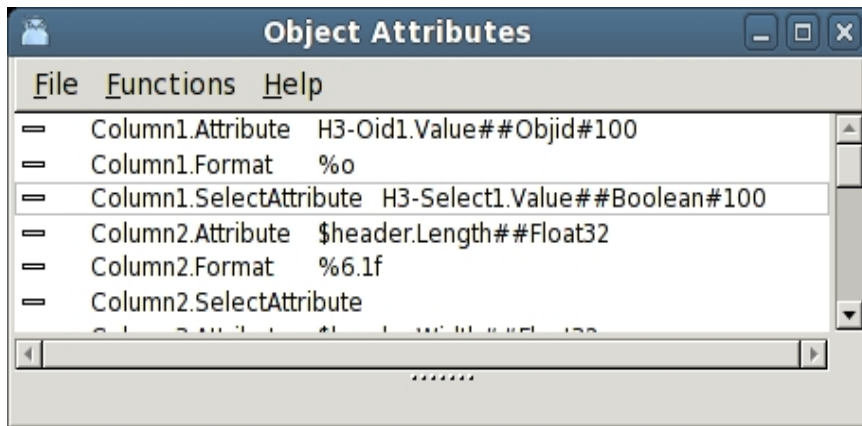
Resultatet ses i figuren nedan. I första kolumnen visas namnet på objektet och på samma rad innehållet i objektet.

Object	Length	Width	Height
T1	12.3	4.30	0.220
T2	12.2	4.50	0.220
T3	9.3	4.00	0.210
T4	11.1	3.20	0.190
T5	13.9	5.30	0.190
T6	16.4	3.90	0.220
T7	12.5	4.50	0.240
T8	11.6	3.20	0.190
T9	12.5	4.90	0.340
T10	12.0	5.10	0.210

**Fig Tabell som visar objektnamn och innehållet i objekten**

#### **Tabell med utvals funktion**

Ibland vill man kunna välja ut en cell i en tabell, och utföra någonting i plcprogrammet med utvalt data eller objekt. Det här gör man genom att koppla en vektor av boolean till attributet SelectAttribute för en kolumn. I exemplet nedan har H3-Select1.Value kopplats till Column1.SelectAttribute.



**Fig En utvals-vektor har angetts för den första kolumnen**

Den första kolumnen kommer nu att vara känslig för klick och den utvalda cellen kommer att markeras med en avvikande färg som kan anges i Table.SelectColor.

Object	Length	Width	Height
T1	12.3	4.30	0.220
T2	12.2	4.50	0.220
T3	9.3	4.00	0.210
T4	11.1	3.20	0.190
T5	13.9	5.30	0.190
T6	16.4	3.90	0.220
T7	12.5	4.50	0.240
T8	11.6	3.20	0.190
T9	12.5	4.90	0.340
T10	12.0	5.10	0.210

**Fig Fjärde raden i första kolumnen har valts ut**

Det element i vektorn som motvarar det utvalda objektet kommer att sättas till 1. I figuren ovan är fjärde raden utvald, vilket medför att fjärde elementet i utvals-vektor sätts, dvs H3-Select1.Value[3].

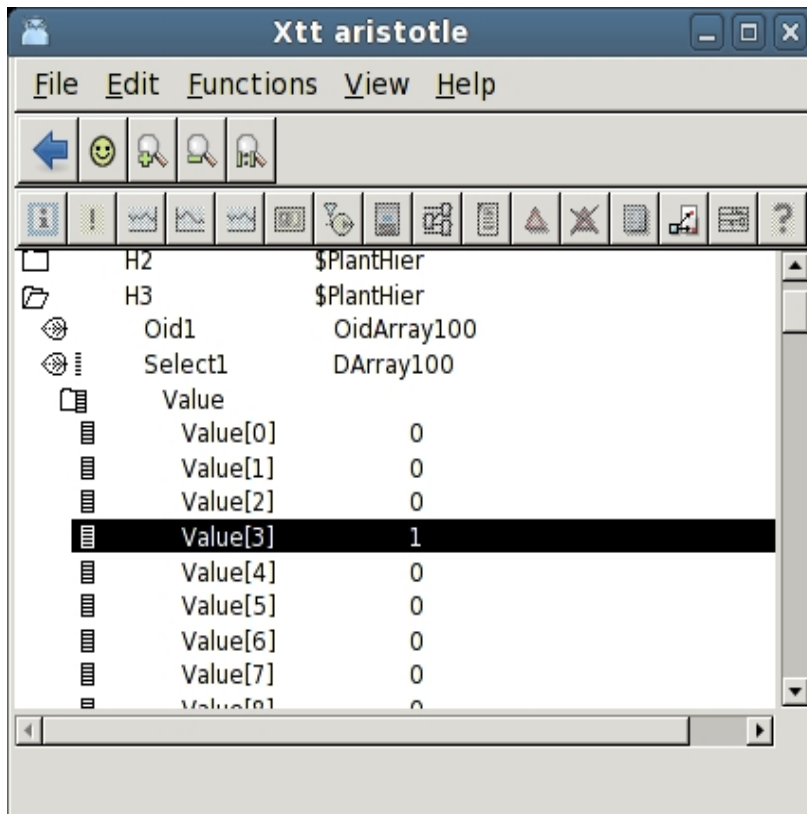


Fig Elementet som motvarar den utvalda raden markeras i utvals-vektorn

#### 4.4.13 Rullgardinsmeny

Rullgardinsmenyn innehåller ett antal menyalternativ som antingen kan vara en ny rullgardinsmeny, eller exekvera en aktion. Aktioner som kan definieras för andra tryckknappar är också tillgängliga för menyalternativ. Med aktionen PulldownMenu kan undermenyer konfigureras.

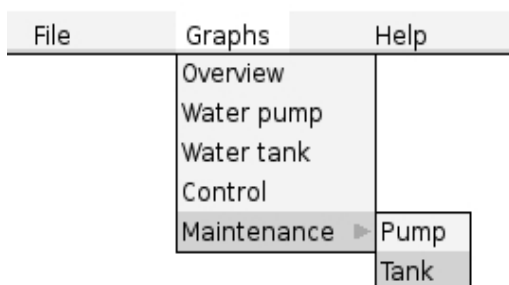
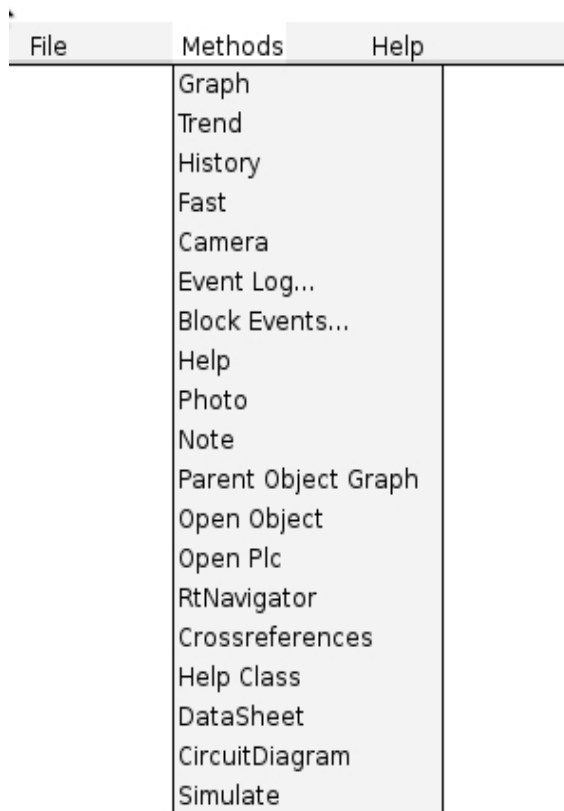


Fig Rullgardinsmeny

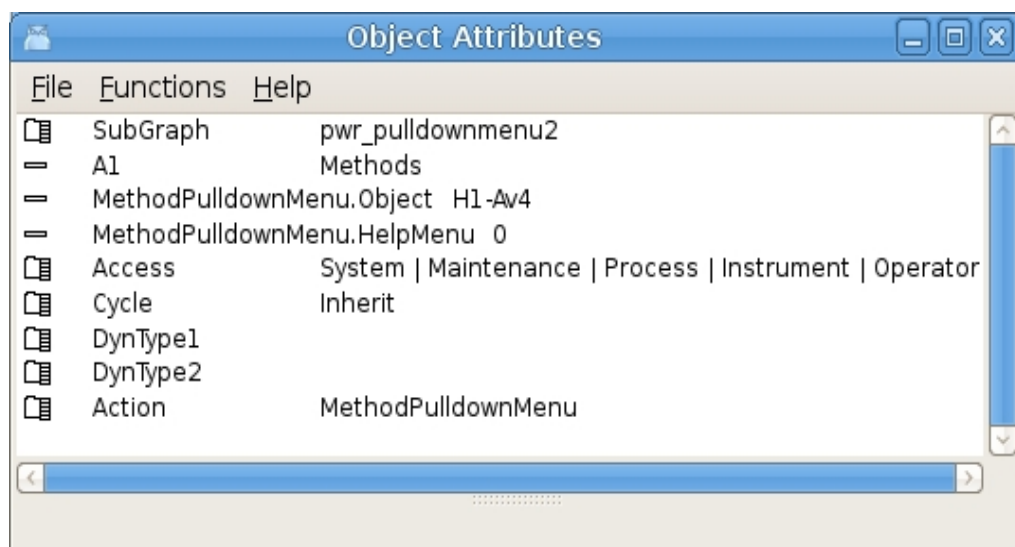


MethodPulldownMenu är en rullgardinsmeny som innehåller menyalternativ för att aktivera metoderna för ett objekt. Den används normalt i objektsbilder. Med attributet HelpMenu visas en hjälp-meny med det två hjälp-metoderna Help och HelpClass.



**Fig Rullgardinsmeny för metoder**

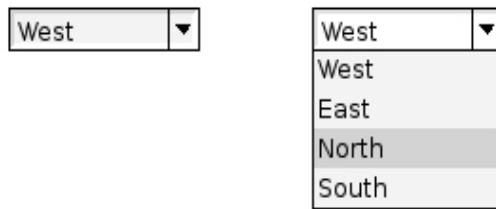
Rullgardinsmenyn för metoder konfigureras genom att sätta Action till MethodPulldownMenu.



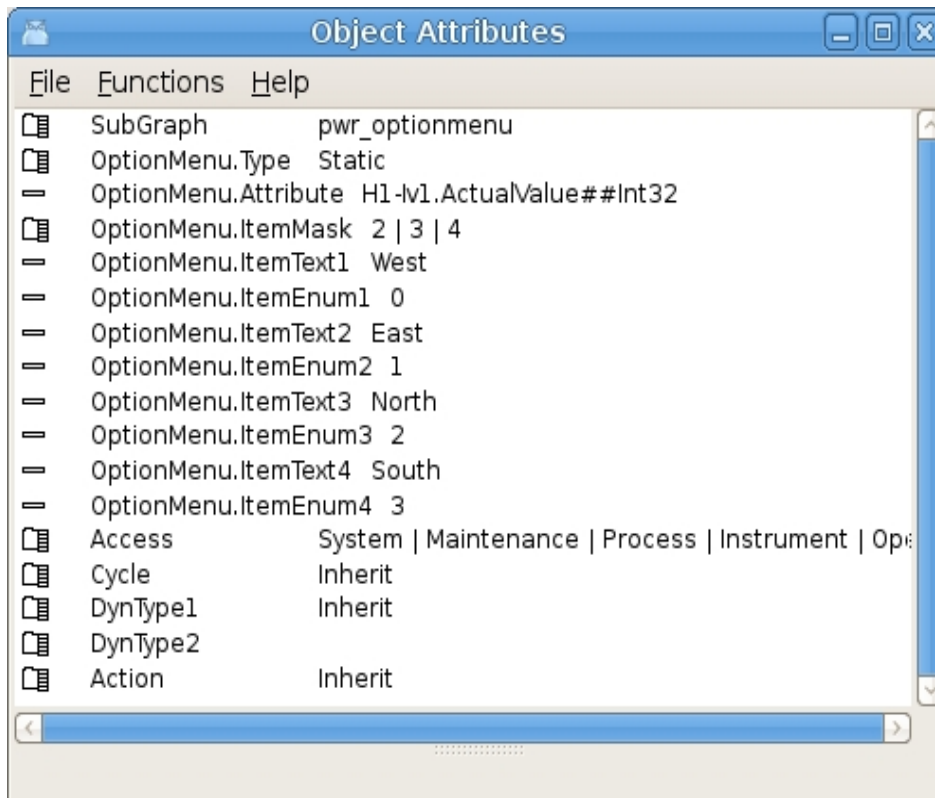
**Fig Attributen för en metod-rullgardinsmeny**

#### 4.4.14 OptionMenu

Option meny är en meny där man väljer alternativ i en lista av alternativ. Det valda alternativet visas upp i meny-komponenten. När man klicka på komponenten öppnas listan med alternativ. När man väljer ett alternativ stängs listan och det valda alternativet läggs in i komponenten.



**Fig Option meny**



**Fig Attribut för en option meny**

#### 4.4.15 MethodToolbar

MethodToolbar är en verktygspanel för metoder. Den visar en uppsättning knappar för att aktivera metoderna för ett specifikt objekt. Endast metoder som är konfigurerade för objektet visas, och knappar som den aktuella användaren inte har tillgång till dimmas.

Verktygspanelen för metoder används normalt i objektsbilder.



**Fig MethodToolbar**

## 4.5 Dynamik och action

En processbild har huvudsakligen två uppgifter, att visa läget i processen för operatoren, och att göra det möjligt för operatören att påverka processen.

För att visa läget i processen lägger man in dynamik i bilderna. Dynamiken kan vara att

- ändra färg på ett objekt
- göra objekt osynliga eller okänsliga
- skriva ut texter
- ändra form på ett objekt
- visa analoga värden
- flytta, skala eller rotera objekt
- Ljud och kommandon

För att göra det möjligt för operatören att påverka processen, lägger man in action på objekt i processbilden. Genom att lägga in action på ett objekt kan man också göra det möjligt för operatören att få mer information om objektet, från popup-menyer eller genom att öppna bilder eller kurvor mm. Action kan vara att

- sätta värden med tryckknappar
- popupmeny
- öppna bilder
- exekvera kommandon
- tooltip
- hjälptexter
- input focus

#### 4.5.1 Ändra färg på objekt

##### Skifta mellan två färger

Vi ska titta på hur man kan ändra färg på ett objekt beroende på en digital signal i databasen. Det finns några olika typer av dynamik man kan använda, DigLowColor, DigColor, DigError och DigWarning. Vi börjar med att titta på ett exempel med DigLowColor.

DigLowColor skiftar mellan två färger, den ena är den man sätter på objektet i editorn, och det är den här färgen objektet kommer att ha när signalen är hög. Den andra färgen anges i DigLowColor.LowColor. Vi kommer att rita en orange indikator, som ska vara orange när signalen är hög och mörkgrå när signalen är låg. Vi ritar alltså indikatorn orange i editorn, och sätter DigLowColor.LowColor till mörkgrå.

Vi börjar med att rita en cirkel, och fyller den med orange färg. Vi sätter också gradienten till GradientGlobe.



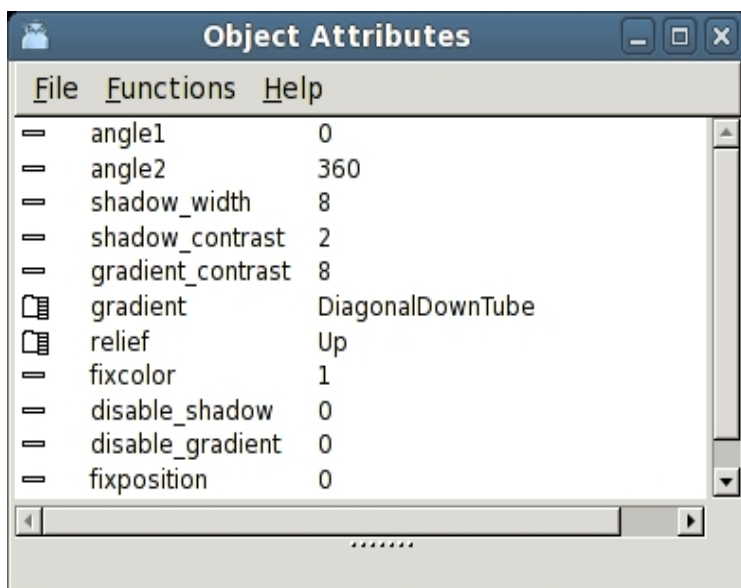
**Fig Orange indikator**

Vi skapar också en ram med metalliskt utseende till indikatorn genom att rita en lite större cirkel med grå fyllnadsfärg. På ramen sätts fixcolor i objektseditorn till 1, eftersom denna inte ska ändra färg med indikatorlampan. Vi sätter också 3D och väljer gradienten DiagonalDownTube.



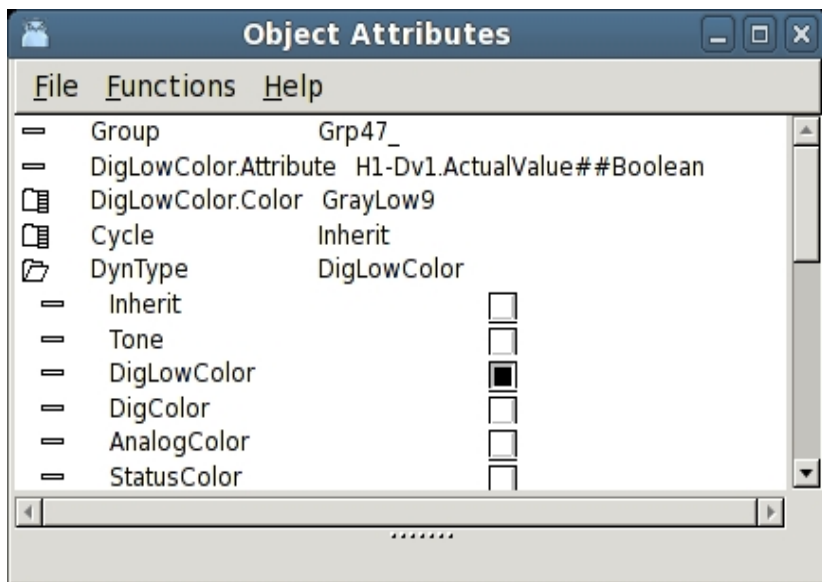
### Fig Ram till indikatorn

I objektseditorn ökas gradient\_contrast till 8 och shadow\_width till 9.



### Fig Fixcolor sätt på ramen

För att kunna sätta dynamik gör vi en grupp av de båda cirkarna. När vi öppnar objektseditorn för gruppen har vi nu attributet Dynamic och väljer DigLowColor. Vi sätter DigLowColor.Color till en mörkgrå nyans och kopplar DigLowColor.Attribute till en Dv, H1-Dv1.



### Fig Dynamik DigLowColor

Resultatet ses i figuren nedan. När signalen är hög är indikatorn orange, och signalen är låg är indikatorn mörkgrå.



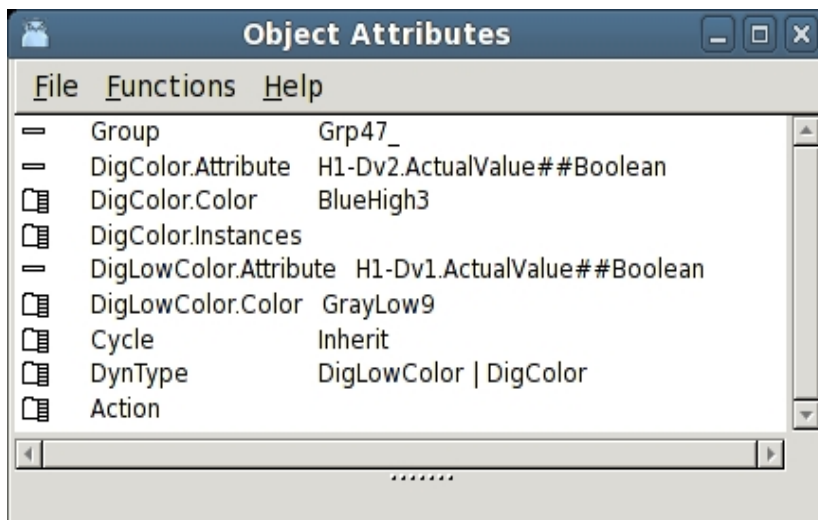
### Fig Indikatorn med låg signal till vänster och hög till höger



Dynamiktypen DigColor är snarlik DigLowColor, och man skulle också kunna ha använt denna typ här. Skillnaden är att signalens funktion är inverterad. I DigLowColor anger man den färg som objektet har när signalen är låg, medan för DigColor anger man den färg som objektet har när signalen är hög.

### Skifta mellan flera färger

Vi ska nu se hur man kan skifta mellan flera färger. Vi utgår från indikatorn i exemplet ovan och adderar dynamiken DigColor. I objektseditorn har vi nu attributen DigColor.Attribute och DigColor.Color. DigColor.Attribute kopplas mot en signal i databasen, H1-Dv2, och i DigColor.Color anges den färg som indikatorn får när signalen är hög.



**Fig En tredje färg adderad med DigColor**

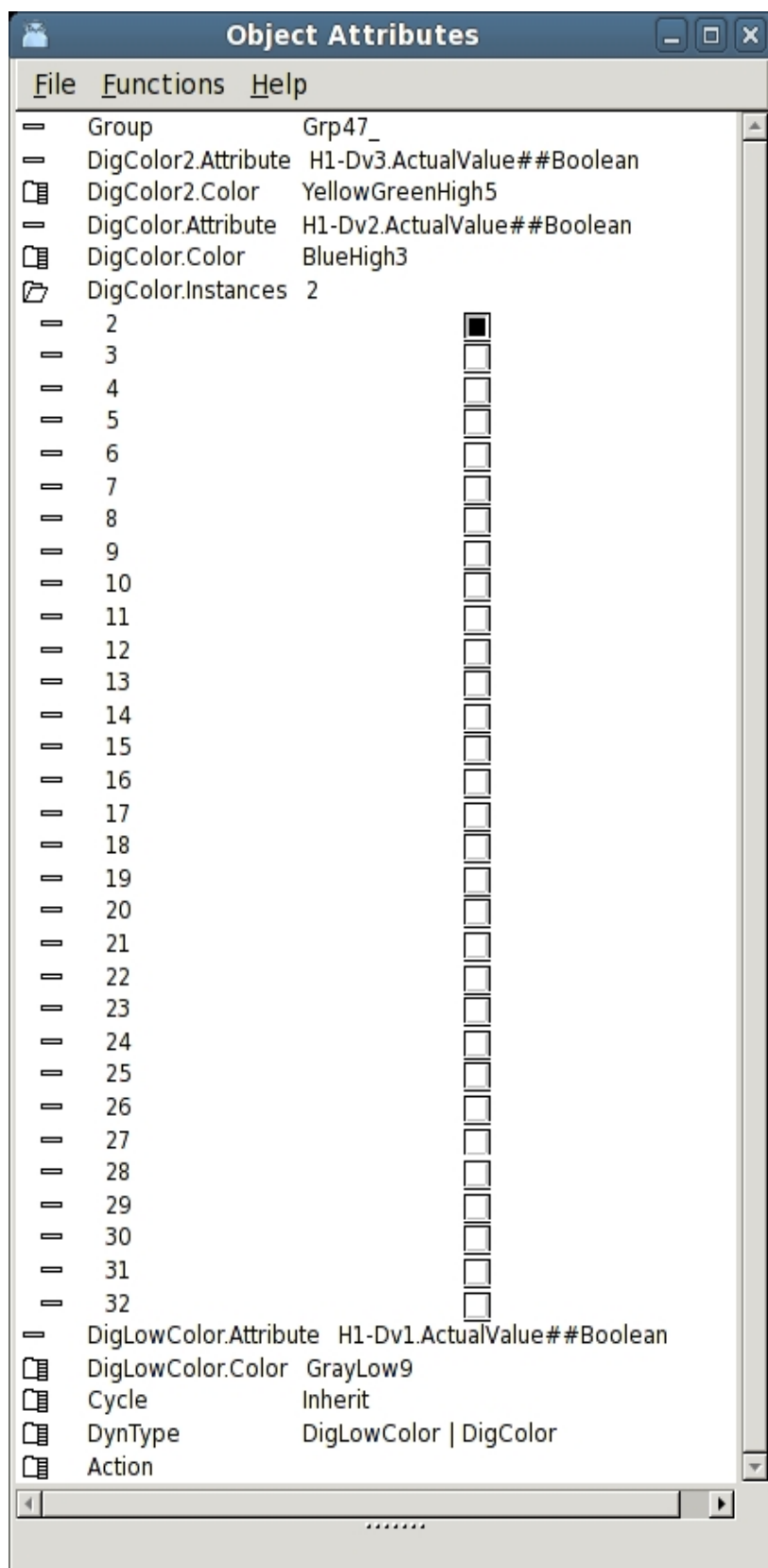
Indikatorn kan nu skifta mellan tre olika färger. DigLowColor skiftar som förut mellan mörkgrått och orange beroende på H1-Dv1, och DigColor sätter blå färg när H1-Dv2 går hög. Notera att DigColor har högre prioritet än DigLowColor. Man kan se det i objektseditorn där dynamiken ordnas i prioritetsordning med hög prioritet överst och låg prioritet underst. Den högre prioriteten på DigColor gör att när H1-Dv2 är hög, blåfärgas indikatorn oberoende av värdet på H1-Dv1.



**Fig Indikatorn färgas orange när H1-Dv1 är hög, och blå när H1-Dv2 är hög**

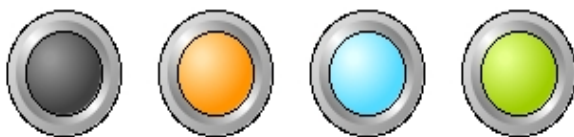
Vi ska nu lägga till ytterligare en färg och studera begreppet Instance. En del dynamik och action typer kan förekomma i flera upplagor, eller flera instanser på samma objekt. För DigColor innebär det att när man adderar ytterligare en instans, kan man sätta en färg till på objektet. Eftersom det finns upp till 32 instanser kan man sätta 32 olika färger, där varje färg kopplas till en signal i databasen.

Man adderar en DigColor instans genom att öppna DigColor.Instances och markera instans 2. Nu visas attributen för denna instans, DigColor2.Attribute och DigColor2.Color. Vi kopplar DigColor2.Attribute till signalen H1-Dv3 och anger en grön färg i DigColor2.Color. Notera att instanser med högre instans-nummer har högre prioritet, dvs H1-Dv3 kommer att färga indikatorn grön oberoende av värdet på H1-Dv1 och H1-Dv2.



**Fig En fjärde färg adderad med DigColor instans 2**

Resultatet ses i figuren nedan. Man kan nu skifta mellan fyra olika färger. Förutom de tre tidigare, mörkgrått, orange och blått, färgas indikatorn nu grön när signalen H1-Dv3 är hög.



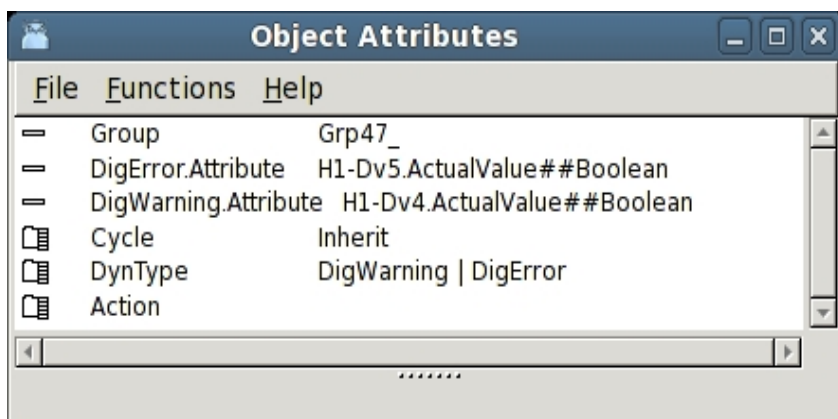
**Fig Indikatorns fyra olika färger**

### Färger för varning och fel

I ProviewR används genomgående färgerna gult och rött för varning resp fel. Det finns två typer av dynamik, DigWarning och DigError, som sätter gul resp röd färg. Fördelen jämfört med DigLowColor eller DigColor är att man inte behöver ange färgen, utan det räcker att koppla dynamiken till en digital signal i databasen.

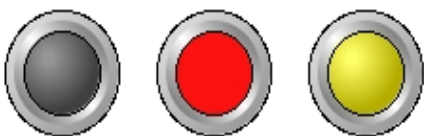
Vi använder indikatorn i ovanstående exempel, men färgar om den till mörkgrått, som blir den färg man får då varken signalen för DigWarning eller DigError är hög.

Vi kopplar DigWarning.Attribute till H1-Dv4 och DigError.Attribute till H1-Dv5.



**Fig Dynamiken DigWarning och DigError**

När ingen signal är satt har indikatorn ursprungsfärgen mörkgrått. När H1-Dv4 sätts får den varnings-färgen gul, och när H1-Dv5 sätts får den fel-färgen rött. DigError har högre prioritet än DigWarning, alltså blir den röd när H1-Dv5 är satt, oberoende av vad H1-Dv4 har för värde.

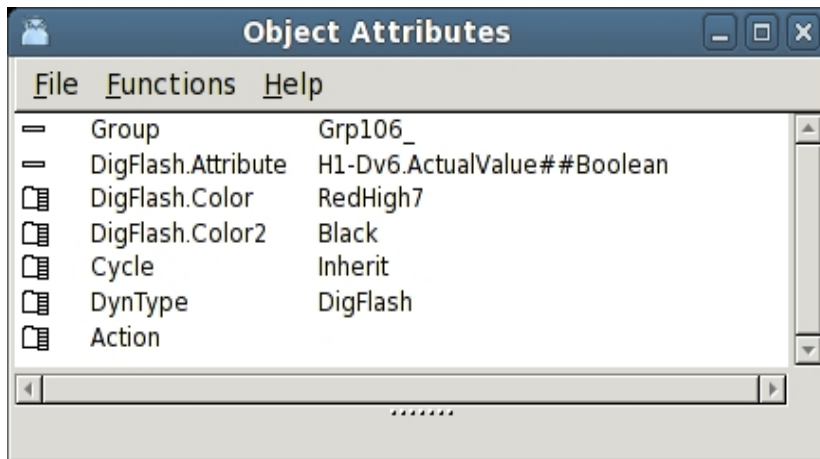


**Fig Gul färg för varning och röd för fel**

### Blinkande dynamik

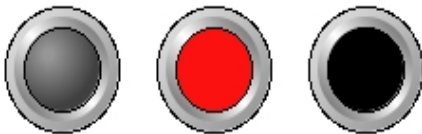
För att dra maximal uppmärksamhet till ett objekt kan man sätta DigFlash dynamik på det. DigFlash innebär att objektet kommer att blinka, dvs skifta mella två färger, när en signal är hög. Blinknings-frekvensen är lika med uppdaterings-frekvensen för objektet. Om Cycle är Slow sker blinkningen med ScanTime för grafen, och om Cycle är Fast med FastScanTime.

Vi sätter DigFlash dynamik på indikatorn och kan nu ange två färger som objektet kommer att blinka med, DigFlash.Color och DigFlash.Color2. Vi sätter DigFlash.Color till röd och DigFlash.Color2 till svart, och kopplar DigFlash.Attribute till signalen H1-Dv6.



**Fig DigFlash dynamik**

När signalen H1-Dv6 är låg har indikatorn ursprungsfärgen mörkgrå. När H1-Dv6 går hög börjar den blinka med färgerna rött och svart.



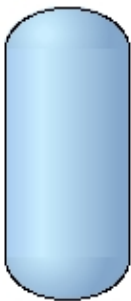
**Fig Indikatorn blinkar mellan rött och svart när signalen är hög**

### Ändra färg på en analog signal

Nu ska vi titta på hur man kan ändra färg på ett objekt beroende på värdet på en analog signal. Det finns två olika typer av dynamik, FillColor och AnalogColor, som kopplas till en analog signal. FillColor färgar en del av ett objekt och hur stor del beror på signalens värde. AnalogColor skiftar färg på objektet när signalen passerar vissa gränsvärden.

### Färga till en viss nivå

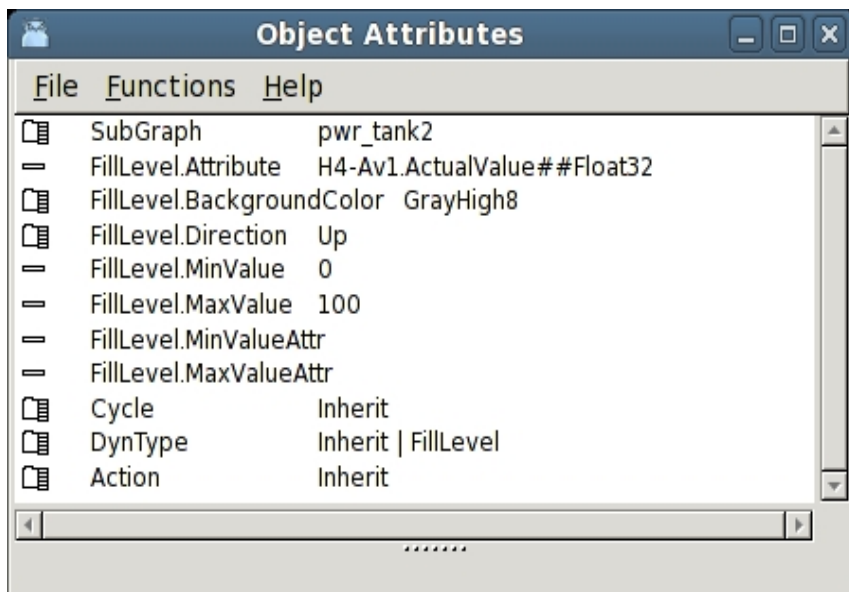
Dynamiken FillColor färgar ett objekt till en viss nivå beroende på en analog signal. Den liknar mycket stapel-objektet som vi har ett tidigare, med den skillnaden att den kan appliceras på vilken form som helst. Vi ska sätta FillColor på en tank som vi hämtar från Hydraulics/Tank2 i subgrafmenyn.



**Fig En tank**

Tanken har ingen default dynamik. Vi markerar FillColor i DynType och kopplar den till Av objektet H4-Av1. Vi sätter också en mörkgrå nyans i FillLevel.BackgroundColor. FillLevel.MinValue och FillLevel.MaxValue anger området på signalen, som default är 0-100.

När värdet är noll kommer tanken att färgas helt med bakgrundsfärgen och när värdet är 100 med den blå färg som vi sätter i editorn.



**Fig FillColor dynamik på tanken**

I figuren nedan har vi satt värdet 70 på H4-Av1 och tanken är fylld upp till denna nivå.



**Fig Tanken fylld till en viss nivå**

Man kan fylla tanken i andra rikningar genom att ändra på FillColor.Direction. Används FillColor dynamiken i en objektsbild kan också området kopplas till attribut för min och max-värde i databasen.

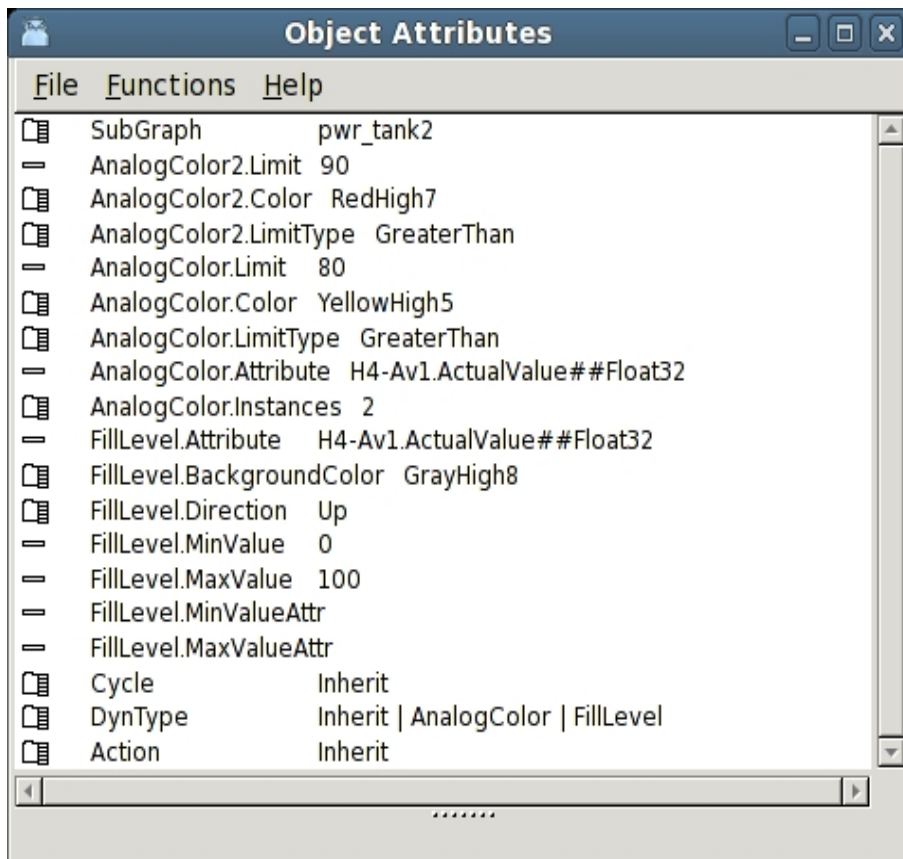
### Ändra färg över eller under en viss nivå

Ofta vill man visa att en analog signal har överskridit eller underskridit ett viss värde. I tank-exemplet ovan ska nivån normalt ligga under 80, och stiger den över 90 börjar läget bli kritiskt. Vi använder dynamiken AnalogColor för att ändra färg på tanken när värdet passerar de båda gränsvärdena.

Till att börja med vill vi färga tanken gul när signalen H4-Av1 är över 80. Vi adderar AnalogColor till DynType och kan nu sätta AnalogColor.Limit till 80. Typen av gränsvärde, AnalogColor.LimitType, är redan GreaterThan, så denna behöver vi inte ändra på. Däremot måste vi ange färgen som ska sättas när signalen överskrider 80, och vi lägger in en gul nyans i AnalogColor.Color.

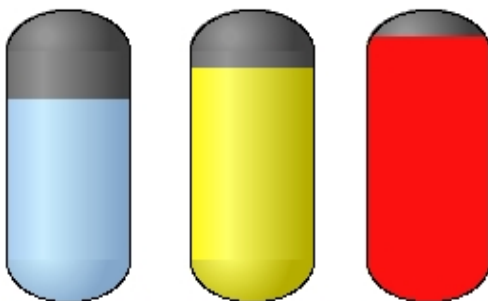
För att kunna lägga in ytterligare ett gränsvärde vid 90, lägger vi till instans 2 i AnalogColor.Instances. Nu dyker attributen för AnalogColor2 upp och vi sätter

AnalogColor2.Limit till 90 och AnalogColor2.Color till en röd nyans. Notera att man inte kopplar något ny signal till instans 2, utan alla instanser använder samma signal.



**Fig AnalogColor dynamik**

Resultatet visas figuren nedan. Så länge värdet på H4-Dv1 är under 80 är tanken blå. När värdet går över 80 blir den gul, och när 90 passeras blir den röd.

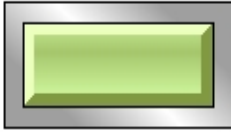


**Fig Vid 80 slår färgen om till gult, och vid 90 till rött**

Genom att sätt LimitType till LessThan kan man också lägga till gränsvärden för miniminivåer, t ex färga tanken gul om nivån underskrider 20, och röd om tanken underskrider 10.

#### 4.5.2 Göra objekt osynliga eller okänsliga

För att göra objekt osynliga eller dimmade använder man dynamiken Invisible. För att demonstrera det skapar vi en tryckknapp från Button/Button2MetalFrame i subgrafsmenyn.



**Fig En trycknapp**

### Osynlig

Trycknappen har som default attributen för DigToggle, och genom att addera Invisible till DynType ser vi även attributen för Invisible. Genom att koppla en digital signal till Invisible.Attribut kommer trycknappen att bli osynlig när signalen är hög.

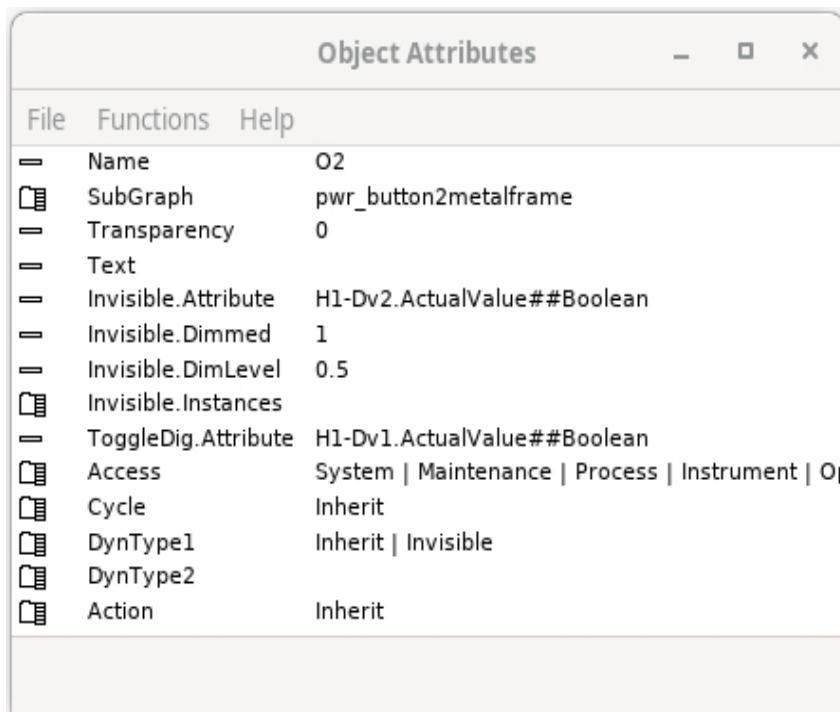
Object Attributes				
File Functions Help				
≡	Name	O2		
≡	SubGraph	pwr_button2metalframe		
≡	Transparency	0		
≡	Text			
≡	Invisible.Attribut	H1-Dv2.ActualValue##Boolean		
≡	Invisible.Dimmed	0		
≡	Invisible.DimLevel	0		
≡	Invisible.Instances			
≡	ToggleDig.Attribut	H1-Dv1.ActualValue##Boolean		
≡	Access	System   Maintenance   Process   Instrument   Op		
≡	Cycle	Inherit		
≡	DynType1	Inherit   Invisible		
≡	DynType2			
≡	Action	Inherit		

**Fig Attribut for invisible**

Vi kopplar Invisible.Attribut till H1-Dv2, och resultatet blir att när H1-Dv2 är hög blir trycknappen osynlig.

### Dimmad

Om man endast vill göra trycknappen okänslig för klick och dimmad, sätter man Invisible.Dimmed till 1.



**Fig Invisible.Dimmed och Invisible.DimLevel är konfigurerad**

När H1-Dv2 blir hög är tryckknappen inte längre känslig för musklick. Det markeras genom att den ritas dimmad. Det är också möjligt att sätta en transparens-nivå på den dimmade knappen genom att sätta Invisible.DimLevel till ett värde mellan 1 och 0.



**Fig Tryckknappen ursprungligt och dimmat läge, och med DimLevel 0.5 till höger**

### 4.5.3 Skriva ut texter

För att skriva ut en dynamisk text måste man använda en subgraf som innehåller en Annotation. Text innehåller nästan alla tryckknappar en Annotation för att man ska kunna skriva ut en text på knappen. Även value och inputvalue subgrafer innehåller en Annotation.

En annotaion är en plats i subgrafen där en text kan skrivas ut. Texten kan antingen anges i editorn, eller med hjälp av några olika typer av dynamik kopplas till attribut i databasen och förändras i runtime.

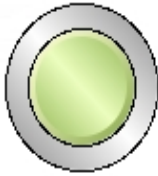
Det olika typer av text-dynamik vi ska titta på här är

- DigText, där man skiftar mellan två olika texter beroende på en digital signal.
- AnalogText, där man skiftar mellan flera olika texter beroende på en analog signal.
- Value, som kopplas till ett sträng-attribut i databasen och visar attributets innehåll.

#### **Skifta mellan två texter på en digital signal**

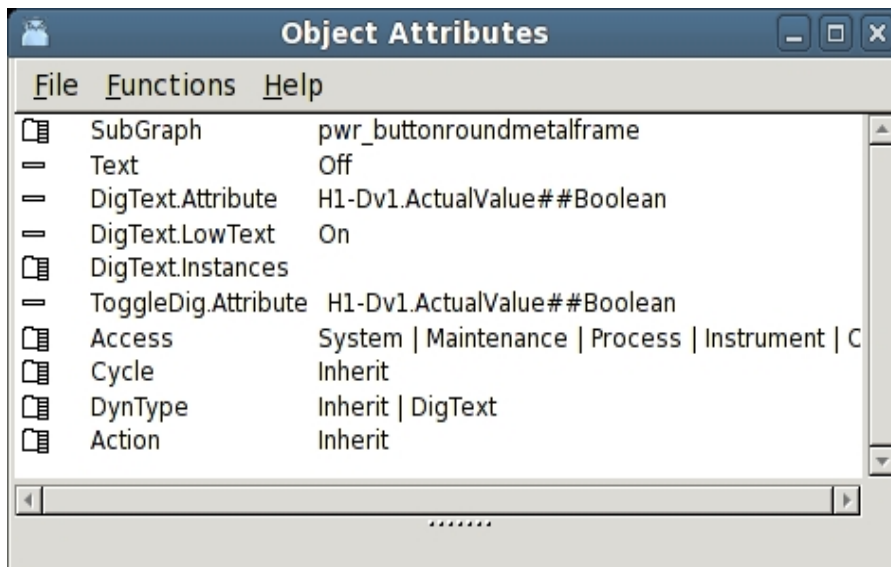
Dynamiken DigLowText skiftar mellan två texter beroende på en digital signal. Vi skapar en tryckknapp av typen ButtonRoundMetalFrame och öppnar objektseditorn.





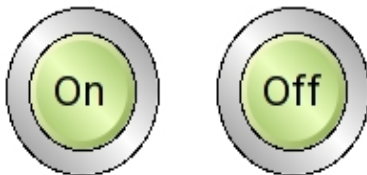
**Fig Tryckknapp med Annotation**

I Text kan man lägga in en text som visas på tryckknappen, och vi lägger in texten 'Off'.  
Genom att addera dynamiken DigText kommer texten i Text att ersättas av texten i DigText.LowText när signalen vi kopplar till DigText.Attribut är låg. Vi lägger in 'On' i DigText.LowText och kopplar DigText.Attribute till Dv'n H1-Dv1. För enkelhets skull kopplar vi även ToggleDig.Attribute till samma signal så att vi enkelt kan ändra på den genom att klicka på knappen.



**Fig Dynamik DigLowText**

Dynamiken fungerar nu så här. När H1-Dv1 är låg skrivs texten 'On' ut på knappen, dvs texten i DigText.LowText, och när signalen är hög skrivs texten 'Off' ut, den text vi skrev in i Text attributet.



**Fig Låg signal till vänster och hög signal till höger**

DigText har flera instanser, och för varje instans kan man ange en text och koppla den till en digital signal. Observera att funktionen är annorlunda för de andra instanserna. För den första anger man en LowText, dvs en text som skrivs ut när signalen är låg. På övriga instanser anger man en HighText, en text som skrivs ut när signalen är hög.

### **Skifta mellan flera texter beroende på en analog signal**

För dynamiken AnalogColor kan man ange upp till 32 olika texter, och vilken av texterna som

skrivs ut beror på värdet av den analoga signal som kopplas till dynamiken.

Vi använder en ValueLargeCenter för att demonstrera dynamiken.

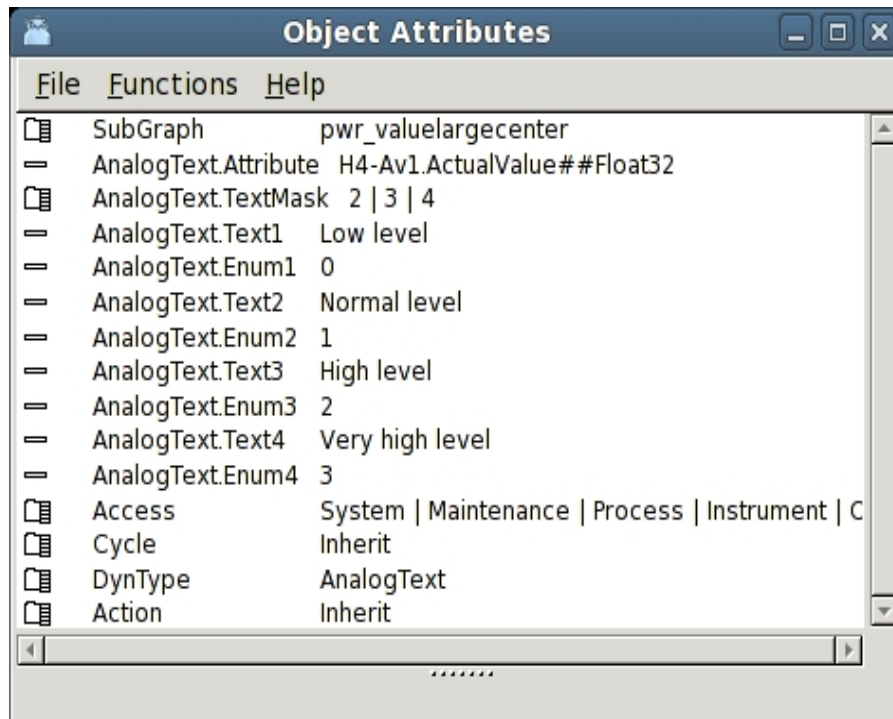


**Fig En ValueLargeCenter subgraf**

ValueLargeCenter har Value dynamik som default, så vi börjar med att ta bort den genom att ta bort Inherit in DynType, och lägger istället in AnalogText.

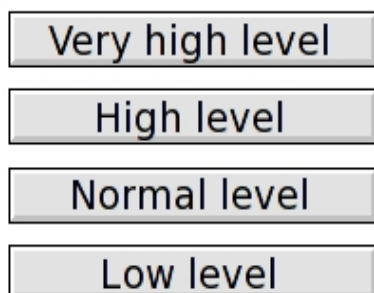
I AnalogText.TextMask markerar man de texter som ska användas. Vi kommer att skifta mellan fyra olika texter och markerar 2, 3 och 4. Texterna 'Low level', 'Normal level', 'High level' och 'Very high level' fylls i text-attributen. Vi behåller enumvärdena, vilket innebär att AnalogText.Text1 visas när signalen är 0, AnalogText.Text2 när signalen är 1 osv. I praktiken är det så att texten skiftar vid värdena 0.5, 1.5 och 2.5. Om signalen är större än 3.5 skrivs inte någon text ut alls.

Vi kopplar också Av'n H4-Av1 till AnalogText.Attribute.



**Fig Value dynamiken ersatt av AnalogText**

I figuren nedan visas value-objektet när H4-Av1 har värden 0, 1, 2 och 3, med 0 värdet längst ner.



## Fig Texten vid olika värden på H4-Av1

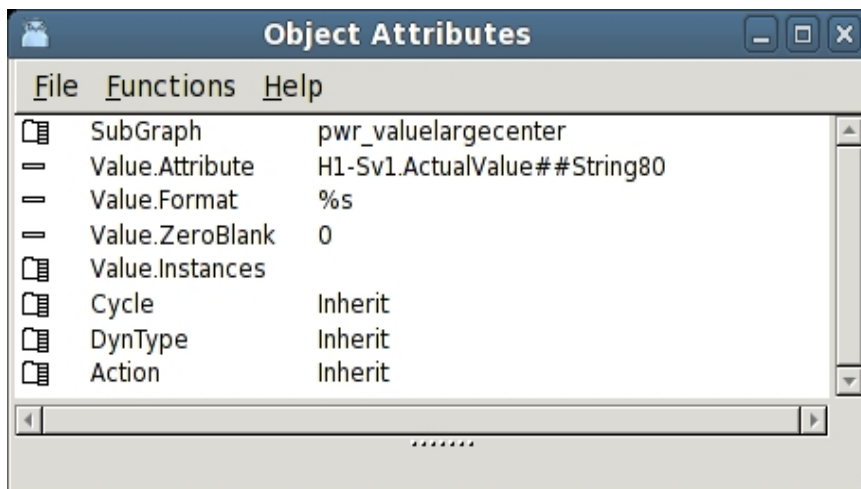
### Visa texten i ett sträng-attribut

För att visa innehållet i ett sträng-attribut i databasen använder man Value dynamiken. Som i föregående exempel skapar vi ett ValueLargeCenter objekt.



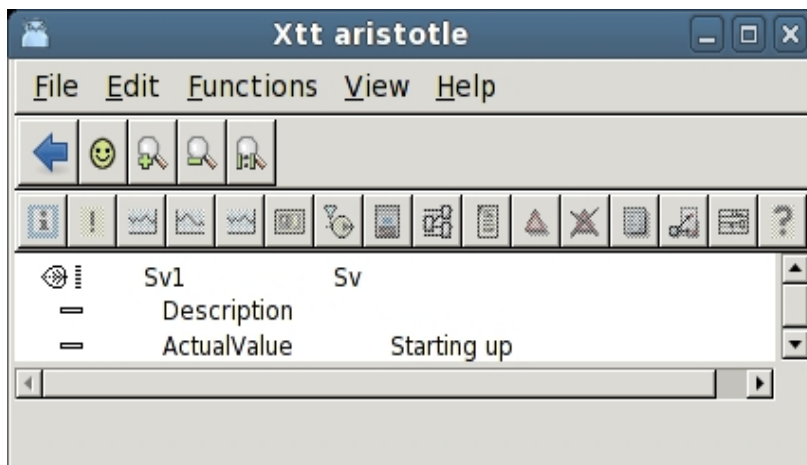
### Fig En ValueLargeCenter subgraf

Den här subgrafen har Value som default dynamik, så vi behöver bara koppla Value.Attribute till ett sträng-attribut och lägga in sträng-formatet '%s' i Value.Format. Vi kopplar Value.Attribute till Sv-objektet H1-Sv.



### Fig Value dynamik

För att visa resultatet lägger vi in texten 'Starting up' i H1-Sv1 i Xtt.



### Fig Sätt en text i Sv attributet

Texten visas i Value-objektet.



Fig Texten skrivs ut i value objektet

#### 4.5.4 Ändra form

I många fall är det bästa sättet att visa läget i processen att ändra form på ett objekt. Det kan vara en grind som visas öppen eller stängd, eller ett hänglås som är låst eller olåst. Det här kan man åstadkomma genom att använda subgrafer med två eller flera sidor.

Ett exempel på en subgraf med flera sidor är Smiley.

Subgrafer med flera sidor hanteras av dynamiktyperna DigShift, AnalogShift och Animation.

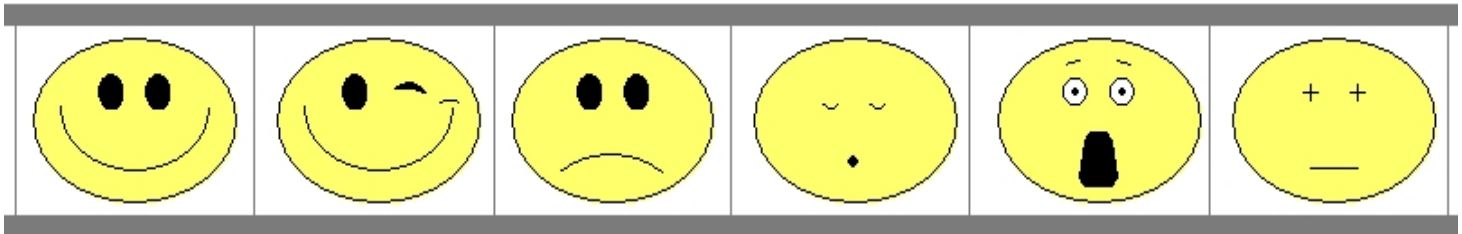


Fig Exempel på subgraf med flera sidor

- DigShift växlar mellan två olika sidor beroende på en digital signal.
- AnalogShift växlar mellan flera olika sidor beroende på en analog signal.
- Animation spelar upp ett antal sidor med en viss hastighet så att man får intryck av rörelse.

#### Växla mellan två sidor

Dynamiken DigShift kan användas på en subgraf med flera sidor. Den kopplas till en digital signal och växlar mellan första och sista sidan beroende på om signalen är hög eller låg.

Vi använder subgrafen Smiley för demonstrera DigShift. Smiley har dynamiken AnalogShift som default, och den måste vi först ta bort genom att återställa Inherit i DynType. I stället anger vi DigShift, och kopplar till signalen H1-Dv1.

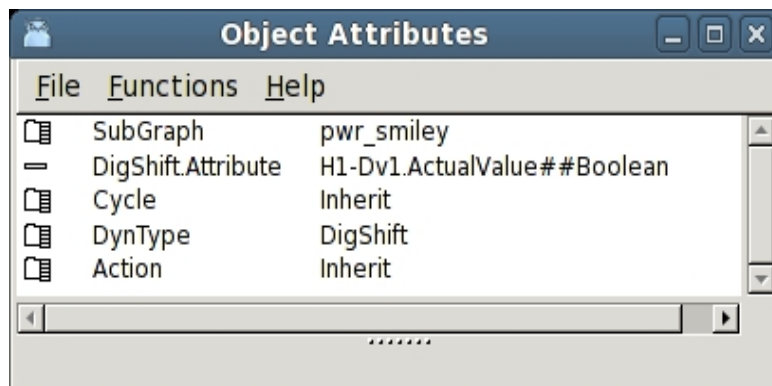


Fig Dynamiken DigShift vald

Resultatet visas i figuren nedan. När signalen är låg visas subgrafens första sida, och när signalen är hög visas subgrafens sista sida.



## Fig Låg signal till vänster och hög signal till höger

### Växla mellan flera sidor

AnalogShift kopplas till en analog signal, och kan skifta mellan flera sidor. Värdet på den analoga signalen bestämmer index för den sida som visas. När signalen är 0 visas första sidan, vid 1 visas andra sidan osv.

Subgrafen Smiley har AnalogShift som default. Det enda vi behöver göra är att koppla den till en analog signal. Vi kopplar den till H4-Av1.

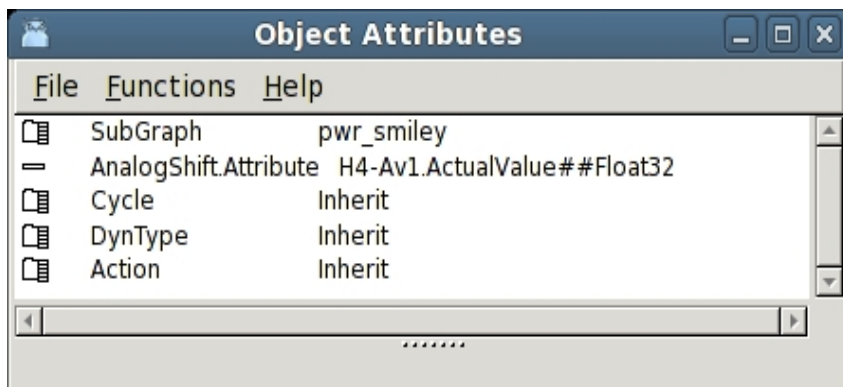


Fig Dynamik AnalogShift

Smiley innehåller 6 olika sidor. När signalen är 0 eller mindre än 0 visas första sidan, när signalen är 1 visas andra sidan osv. När signalen är 5 eller större än 5 visas sista sidan.

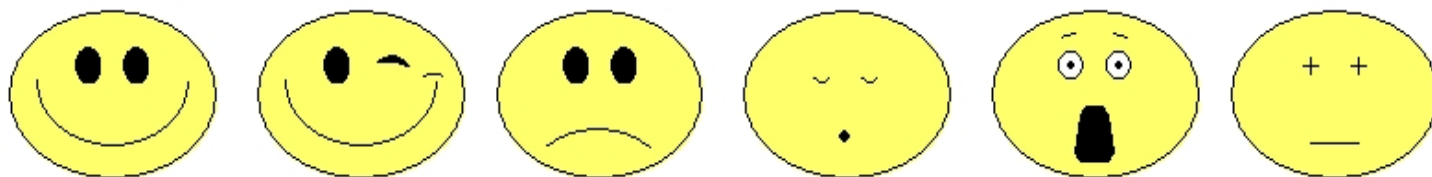


Fig Signalens värde är 0 till vänster och ökar med 1 för varje bild

### Animation

TODO...



Fig Animation för att öppna/stänga ett hänglås

## 4.5.5 Visa analoga värden

### Visa analoga värden i sifferform

För att visa analoga värden i sifferform använder man dynamiken Value. Value kopplas till en analog signal och konverteras till lämplig form med hjälp av format specifikationen.

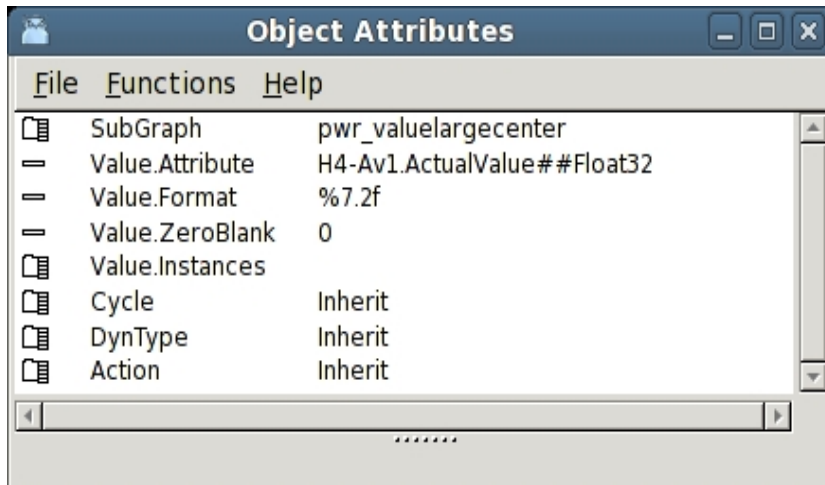
Value dynamiken kan användas på subgrafer som innehåller en annotation, dvs en plat för att

skriva ut en text. Ett exempel på en sådan subgraf är ValueLargeCenter.



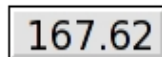
**Fig ValueLargeCenter**

Vi skapar en ValueLargeCenter och kopplar den till den analoga signalen H4-Av1. Vi måste dessutom ange formateringen i Value.Format. '%f' är formateringen för ett flyttal, och man kan även specificera hur många tecken och hur många decimaler som talet ska skrivas ut med. Med '%7.2f' anges att talet är upp till 7 tecken stort, inklusive punkten, och att två av dessa är decimaler.



**Fig Value dynamik med format specifikation**

Resultatet visas i figuren nedan. Det analoga värdet skrivs med två decimaler.



**Fig Value dynamik**

Notera att Value dynamiken kan användas för att visa innehållet i en rad olika typer av signaler, heltal, strängar, tider mm.

Genom att skapa flera instanser av Value dynamik för ett objekt kan man visa fler analoga värden på samma objekt. Det här kräver att subgrafen innehåller flera Annotations.

Analoga värden kan även visas i form av kurvor och staplar. Detta finns beskrivet i kapitlet 'Speciella Object'

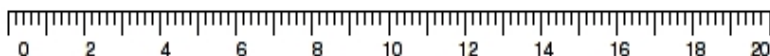
## 4.5.6 Flytta, skala och rotera objekt

### Flytta ett objekt

För att flytta och skala ett objekt används dynamiken Move. Vi ska till att börja med titta på hur man kan flytta objektet. Objektet kan flyttas i x- och y-led och för varje riktning kopplar man en analog signal som specificerar förflyttningen relativt ursprungspositionen.

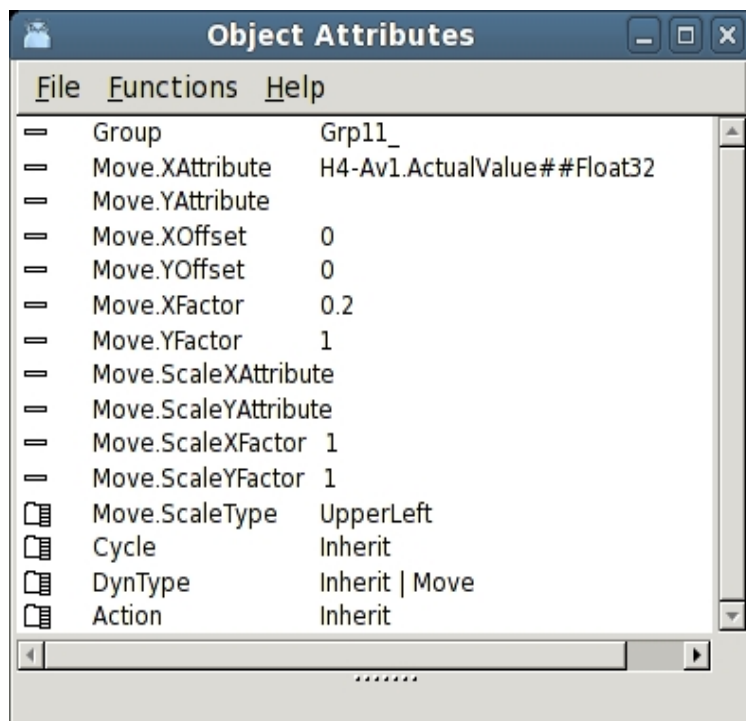
Signalvärdet kan räknas om till en förflyttning i Ge-koordinater med en faktor (Move.XFactor och Move.YFactor) och ett offset (Move.XOffset och Move.YOffset).

I det här exemplet ska vi flytta en rektangel i x-led. Vi ritar rektangeln och gör en grupp av den.



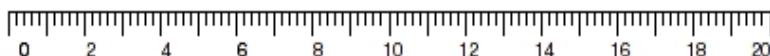
**Fig En rektangel**

När den är grupperad kan vi sätta Move i DynType och attributen för Move visas i objektseditorn. Eftersom vi ska flytta i x-led kopplar Move.XAttribute till signalen H4-Av1 i databasen. Vi mäter upp området som rektangeln ska flyttas på, och kommer fram till att när signalen varierar mellan 0-100 ska rektangeln flyttas 20 Ge-enheter i bilden. Genom att placera rektangeln på 0-punkten behöver vi inte ange något offset, men vi måste skala ner signalvärdet med  $20/100 = 0.2$  som vi lägger in i Move.XFactor. I exemplet har vi även lagt in en x-axel för att visa storleken på förflyttningen.



**Fig Move dynamik på rektangeln**

Resultatet blir att när signalen är 0 ligger rektangeln kvar på ursprungspositionen. Vid värdet 60 har rektangeln flyttats 12 enheter, se figuren nedan, och vid värdet 100 har rektangeln flyttats 20 enheter.

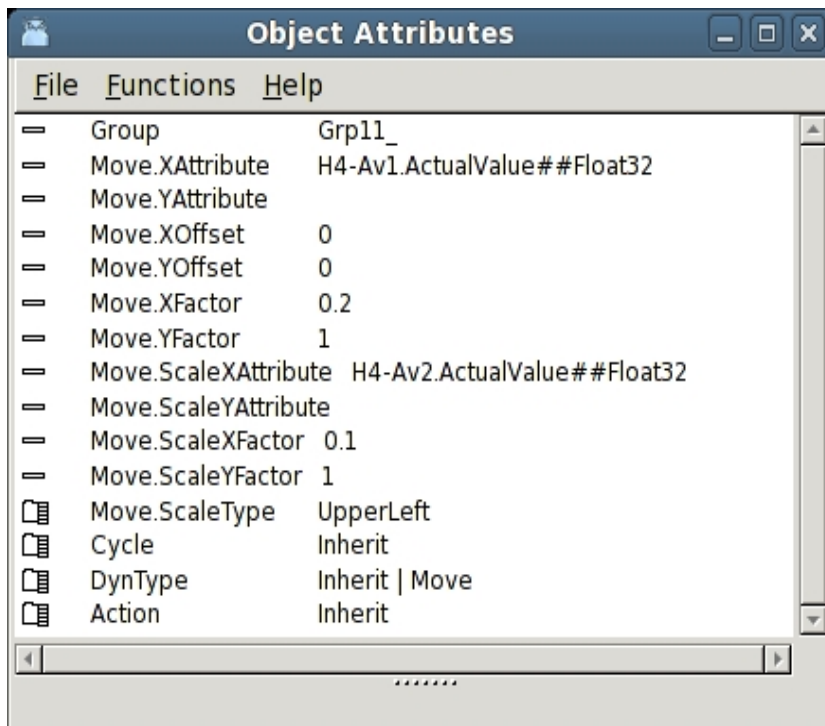


**Fig Rektangeln flyttad 12 Ge-enheter i x-led**

### Skala ett objekt

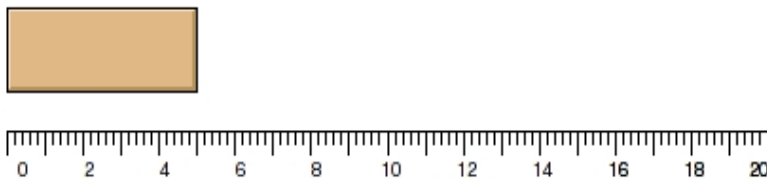
Move dynamiken kan även användas för att skala ett objekt. Skalningen kan ske i x- och y-led och för varje riktning kopplar man en analog signal som specificerar storleken på skalningen.

Vi lägger in skalning i x-led för rektangeln i exemplet ovan. Move.ScaleXAttribute kopplas till signalen H4-Av2. Vi måste också räkna ut en skalfaktor och lägga in i Move.ScaleXFactor. Rektangeln har bredden 2 enheter och med skalfaktorn 0.1 kommer ett signalvärde på 100 att ge en bredd på 20 enheter.



**Fig Signal och skalfaktor angivna**

Resultatet blir att när signalvärdet är 0, har rektangeln ingen bredd alls, utan ritas som ett streck. Vid signalvärdet 25 ser det ut som i figuren nedan, rektangeln är skalad 2.5 gånger, och när signalvärdet är 100 är rektangeln skalad 10 gånger och täcker hela x-axeln.



**Fig Signalvärdet 25 ger en skalning på 2.5 gånger**

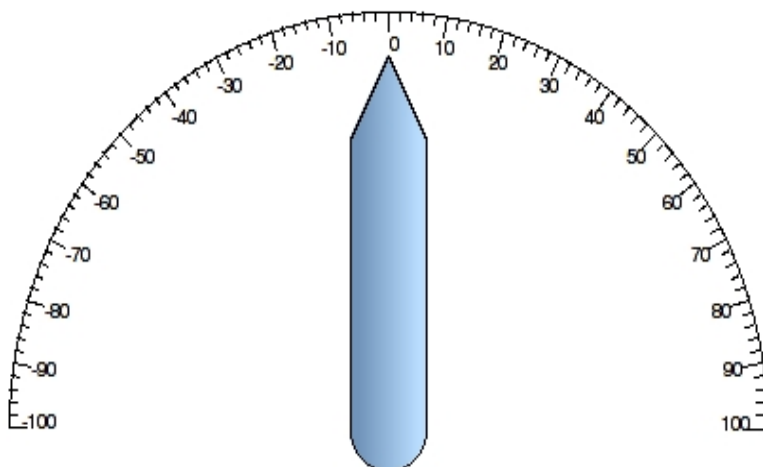
### Rotera ett objekt

För att rotera ett objekt använder man dynamiken Rotate. Rotate kopplas till en analog signal som innehåller rotationen i grader.

Rotate har vissa begränsningar eftersom rektanglar och ellipser endast kan roteras i steg med 90 grader. Därför använder man företrädesvis polylinjer, linjer och cirkelar med Rotate.



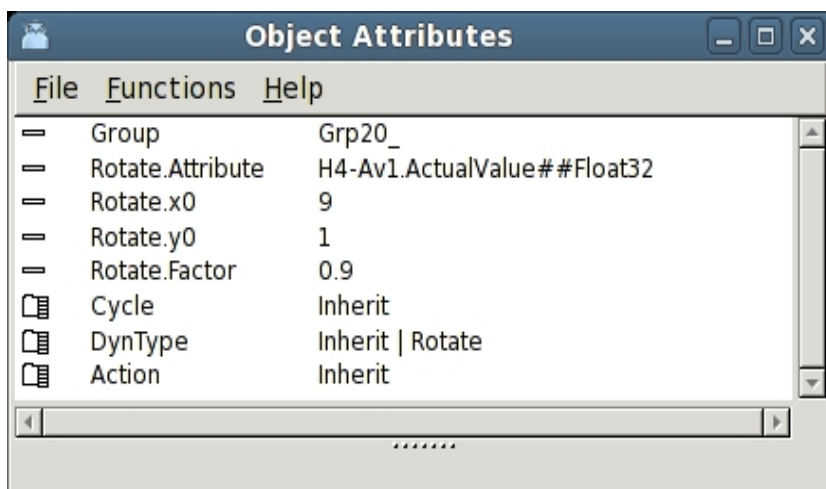
Vi ritat en enkel visare med ett cirkel-segment och en polylinje. Efter att ha grupperat dem sätter vi Rotate i DynType. Vi ritat även en cirkulär skala med ett AxisArc objekt.



**Fig Visare med rotations-dynamik**

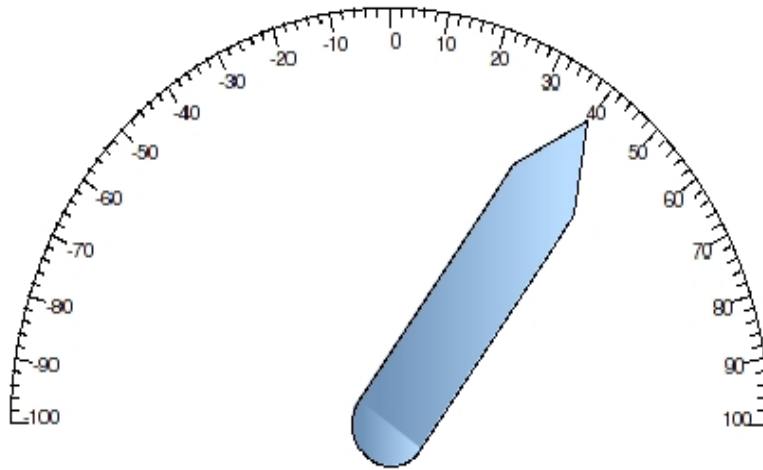
Visaren kopplas till den analoga signalen H4-Av1. Vi måste också specificera den punkt som objektet ska rotera runt. I det här fallet är det cirkelbågens mittpunkt som har x-koordinaten 9 och y-koordinaten 1. Rotationspunkten läggs in i Rotate.x0 och Rotate.y0.

Skalan som vi har ritat har området -100 - 100, vilket är området för signalen H4-Av1. Denna ska räknas om till grader och motvarande rotation är -90 - 90 grader, dvs skalfaktorn är 0.9. Vi lägger in 0.9 i Rotate.Factor.



**Fig Visaren kopplade till en analog signal och skalfaktorn satt till 0.9**

Resultatet kan ses i figuren nedan.



**Fig** Värdet 40 ger en rotation på 36 grader

#### 4.5.7 Ljud och kommandon

TODO...

#### 4.5.8 Sätta värden med tryckknappar

##### Sätta ett digitalt värde

Action SetDig sätter värdet på en digital signal till 1 när man klickar på ett objekt.

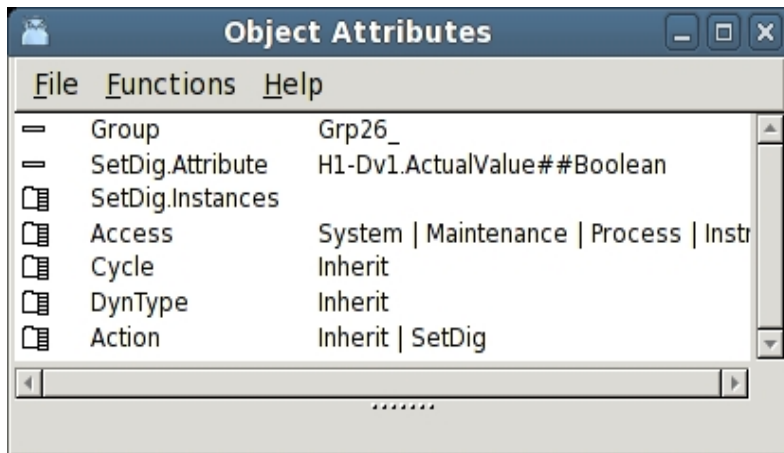
Vi börjar med att tillverka en knapp av en rektangel. Genom att sätta 3D och öka shadow\_width till 15 får vi ett karaktistiskt knapp-utseende på rektangeln. Dessutom adderar vi gradienten DiagonalLowerRight med gradient\_contrast 1 för att ytterligare förstärka 3D effekten. Slutligen gör vi en grupp av rektangeln så att vi kan sätta Action på den.



**Fig** En rektangel formad som en tryckknapp

För att sätta en digital signal till 1 när man trycker på knappen, sätter vi SetDig i Action. Det innebär att om signalens värde är 0 kommer det att sättas till 1. Om det redan är 1 blir det ingen förändring.

Vi kopplar SetDig.Attribute till signalen H1-Dv1.



**Fig Action SetDig på knappen**

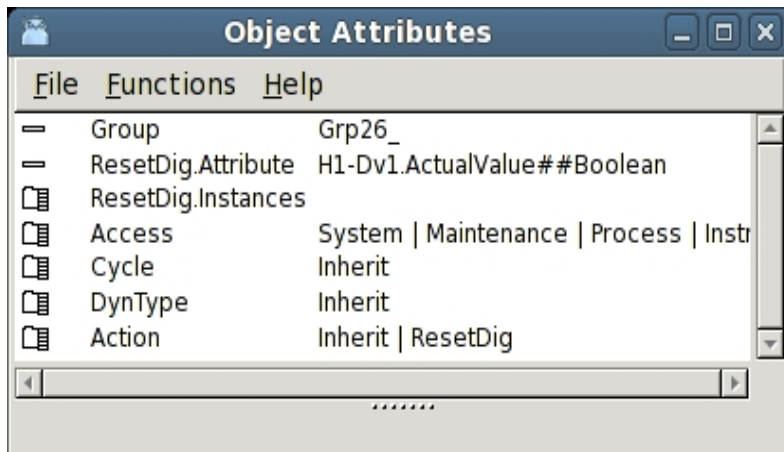
Resultatet blir att när vi klickar på knappen sätts H1-Dv1 till 1.

Det är möjligt att addera flera instanser av SetDig och för varje ny instans kopplar man en ny signal som sätts till 1 när man klickar på knappen. På så sätt kan man sätta flera signaler på en knapptryckning.

### Återställa ett digitalt värde

ResetDig fungera som SetDig med den skillnaden att den sätter värdet 0 på signalen i stället.

Vi sätter Action ResetDig och kopplar till signalen H1-Dv1.



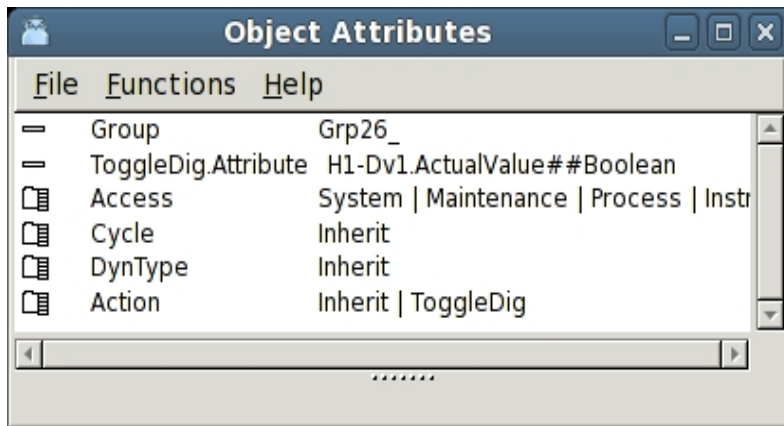
**Fig Action ResetDig på knappen**

Resultatet blir att när vi klickar på knappen sätts H1-Dv1 till 0.

### Togglar ett digitalt värde

Med Action ToggleDig ändrar man värdet på en digital signal när man klickar på objektet. Om värdet på signalen är 1, sätts det till 0, och om värdet på signalen är 0 sätts det till 1.

Vi kopplar ToggleDig.Attribute till signalen H1-Dv1.



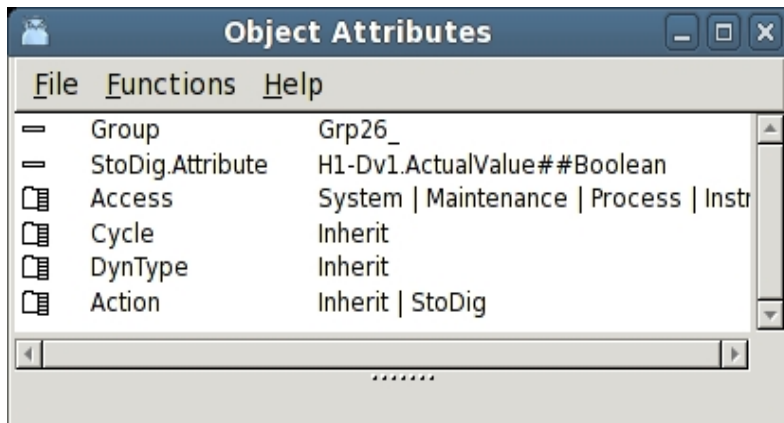
**Fig Action ToggleDig på knappen**

Resultatet blir att värdet på H1-Dv1 ändras när man klickar på knappen.

### **Sätta ett digitalt värde så länge knappen är nedtryckt**

Med Action StoDig sätter man värdet på en digital signal till 1 när knappen trycks ned, och återställer till 0 när knappen släpps.

Vi sätter Action StoDig på knappen och kopplar den till H1-Dv1.



**Fig Action StoDig**

Resultatet blir att när vi trycker på knappen, sätts H1-Dv1 till 1, och när vi släpper knappen sätts H1-Dv1 till 0.

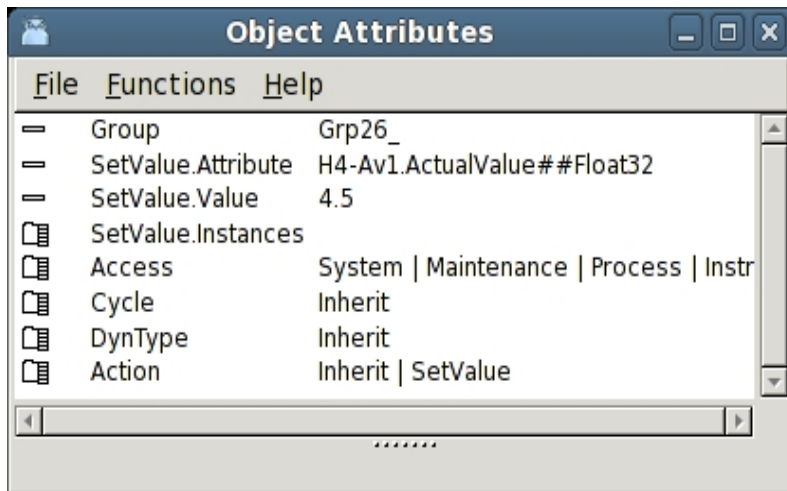
### **Tryckknapp med bekräftelse**

Om man adderar Action Confirm för en tryckknapp krävs det en bekräftelse från användaren innan actionen genomförs. När man klickar på knappen poppar det upp ett fönster där man antingen kan bekräfta eller avbryta. Texten i fönstret anges i Confirm.Text.

### **Sätta ett analog värde**

För att sätta ett värde i en analog signal använder man Action SetValue. SetValue kopplas till en analog signal, och när man klickar på objektet sätts ett specificerat värde i signalen.

Vi sätter Action SetValue på knappen, och kopplar till Av-objektet H4-Av1. Vi vill sätta värdet 4.5 och lägger in 4.5 i SetValue.Value.



**Fig SetValue**

Resultatet blir att när vi klickar på knappen läggs värdet 4.5 in i H4-Av1.

SetValue tillåter flera instanser, och för varje instans anger man ett värde och kopplar instansen till en analog signal. Men knapptryckningen kan man då sätta olika värden i ett antal analoga signaler.

### Öka/Minska knappar

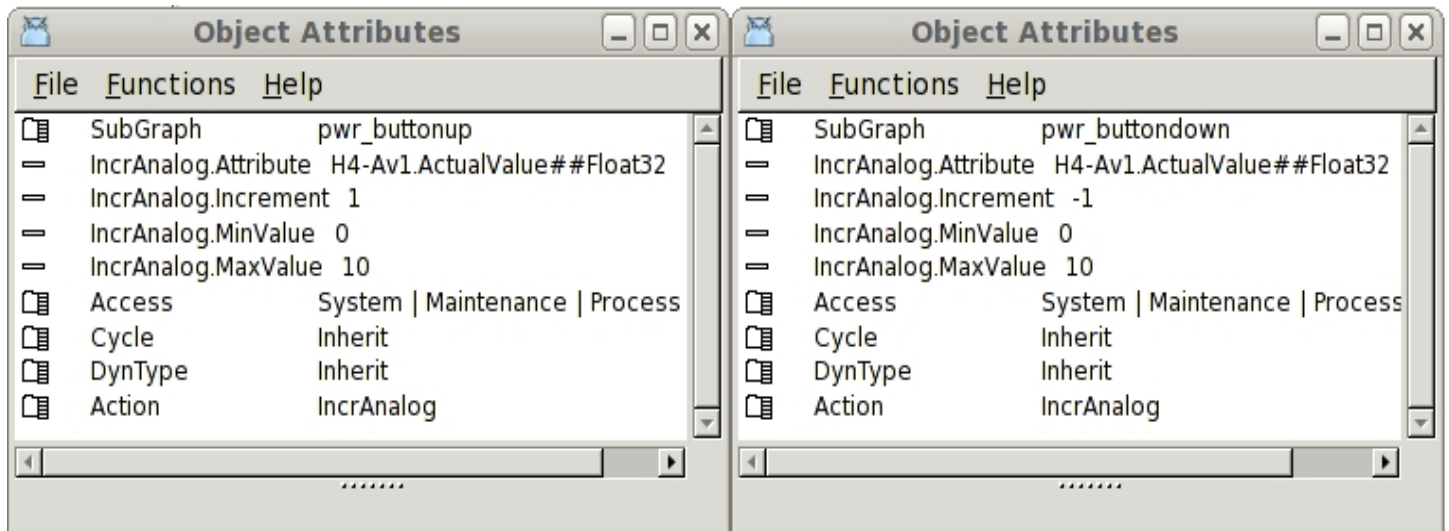
Med action IncrAnalog kan man öka eller minska värdet på en analog signal med en knapptryckning. IncrAnalog kopplas till en analog signal och i IncrAnalog.Increment anger man hur mycket värdet ska öka eller minska vid varje klick. Man kan också lägga in min och max-värde för signalen.

Vi lägger ut en ButtonUp och en ButtonDown.



**Fig ButtonUp och ButtonDown**

Båda har SetDig som default. Vi tar bort SetDig genom att avmarkera Inherit i Action, och markerar istället IncrAnalog. Vi kopplar båda knapparna till signalen H4-Av1. ButtonUp knappen ska öka värdet med 1 för varje klick, så vi sätter IncrAnalog.Increment till 1. ButtonDown knappen ska minska värdet med 1, så vi sätter IncrAnalog.Increment till -1. Vi lägger också in min och maxvärden och sätter IncrAnalog.MaxValue till 10.



**Fig ButtonUp till vänster och ButtonDown till höger**

Resultatet blir att när vi klickar på ButtonUp knappen ökar värdet på H4-Av1 med 1 för varje klick tills värdet har nått max-värdet 10. När vi klickar på ButtonDown minskar värdet med 1 för varje klick men går inte under min-värdet 0.

### **Radioknappar**

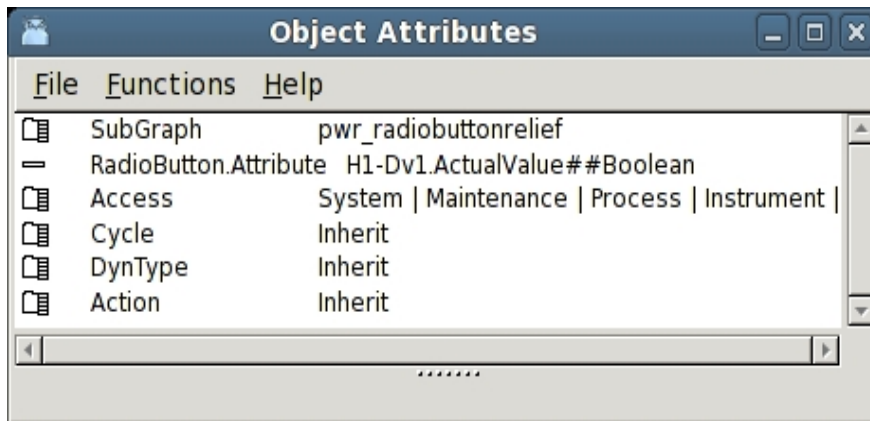
Radioknappar används när man vill välja ut ett alternativ av ett antal alternativ. För varje alternativ skapar man en radioknapp och kopplar den till en digital signal. Därefter grupperar man radio-knapparna. Action RadioButton fungerar så att signalen för den radioknapp man klickar på sätts, medan övriga radioknappar i gruppen återställs. Det innebär att det är högst ett alternativ som är valt, dvs bara en av signalerna är 1.

Vi lägger ut fyra radioknappar av typen RadioButtonRelief.

- ☐ Very slow
- ☐ Slow
- ☐ Fast
- ☐ Very fast

**Fig Fyra radioknappar**

De har redan action RadioButton som default. Vi kopplar radioknapparna till var sin digital signal, H1-Dv1, H1-Dv2, H1-Dv3 och H1-Dv4. Därefter grupperar vi de fyra radioknapparna så att de tillhör samma grupp. Vidare lägger vi in några lämpliga texter för att beskriva de olika alternativen.



**Fig Den första radioknappen kopplad till H1-Dv1**

När vi tar upp grafen med radioknappar i operatörmiljön, kan vi välja ett av alternativen. Det utvalda markeras med ett svart prick, medan tidigare utvalda återställs. Samtidigt sätts signalen som den utvalda knappen är kopplad till, medan övriga signaler nollställs. I figuren nedan är andra alternativet markerat, dvs H1-Dv2 är 1 och Dv1, Dv3 och Dv4 0.

- ☐ Very slow
- ☒ Slow
- ☐ Fast
- ☐ Very fast

**Fig Radioknapp markerad**

### Checkbox

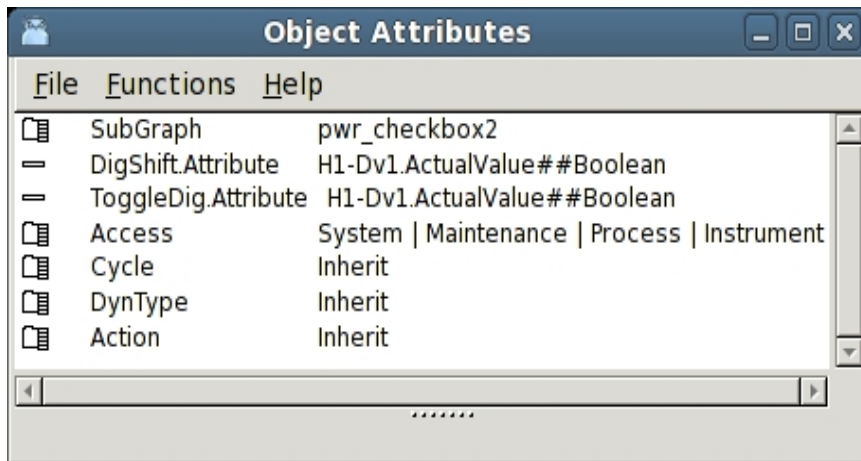
Checkboxar brukar användas för att bocka för ett antal alternativ ut en lista. Till skillnad från radioknappar kan man välja ut flera alternativ samtidigt. En checkbox har action ToggleDig kombinerad med dynamiken DigShift.

Vi skapar fyra checkboxar av typen Checkbox2, och lägger beskrivande texter intill.

- ☐ Pump 1
- ☐ Pump 2
- ☐ Pump 3
- ☐ Pump 4

**Fig Checkboxar**

Checkboxarna kopplas till signalerna H1-Dv1, H1-Dv2, H1-Dv3 och H1-Dv4. Notera att både DigShift.Attribute och ToggleDig.Attribut kopplas till samma signal.



**Fig Objektseditorn för den första checkboxen**

I figuren nedan är första och tredje checkboxen markerad, dvs H1-Dv1 och H1-Dv3 är satta till 1.

- ☒ Pump 1
- ☐ Pump 2
- ☒ Pump 3
- ☐ Pump 4

**Fig Checkboxar markerade**

#### 4.5.9 Popup-menyn

Med action PopupMenu kan man genom att högerklicka på ett objekt visa en popup-menyn. Popup-menyn är kopplad till ett objekt i databasen, och visar upp metoderna för det objektet. Vilka metoder som visas i popup-menyn beror på hur objektet är konfigurerat i databasen, och vilka metoder som finns definierade för den typen av objekt.

Låt oss börja med ett enkelt exempel där vi visar värdet i ett Av-objekt i ett Value fält. Vi skapar en ValueLargeCenter, kopplar den till H4-Av1 och lägger in formateringen %7.2f i Value.Format.

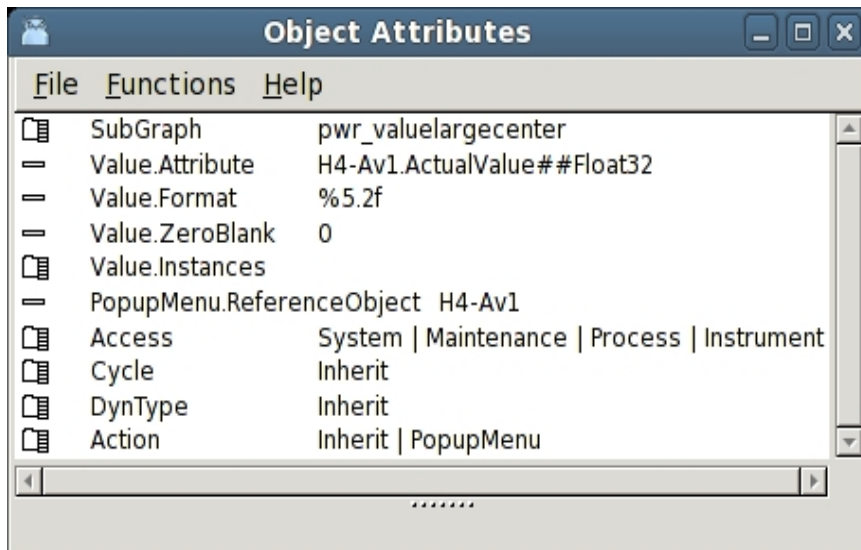


**Fig ValueLargeCenter**

Nu adderar vi även PopupMenu i Action, och lägger in Av-objektet i PopupMenu.ReferenceObject. Notera att det är hela objektet som refereras och inte ett enskilt attribut.

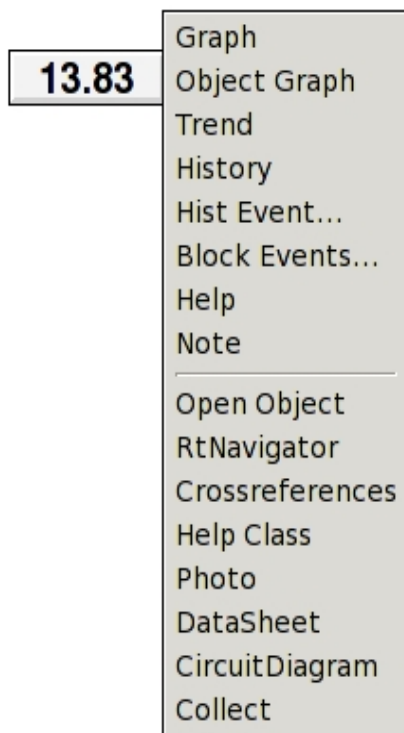
Vi måste också öppna Files/Graph Attributes från menyn och sätta MB3Action till PopupMenu.





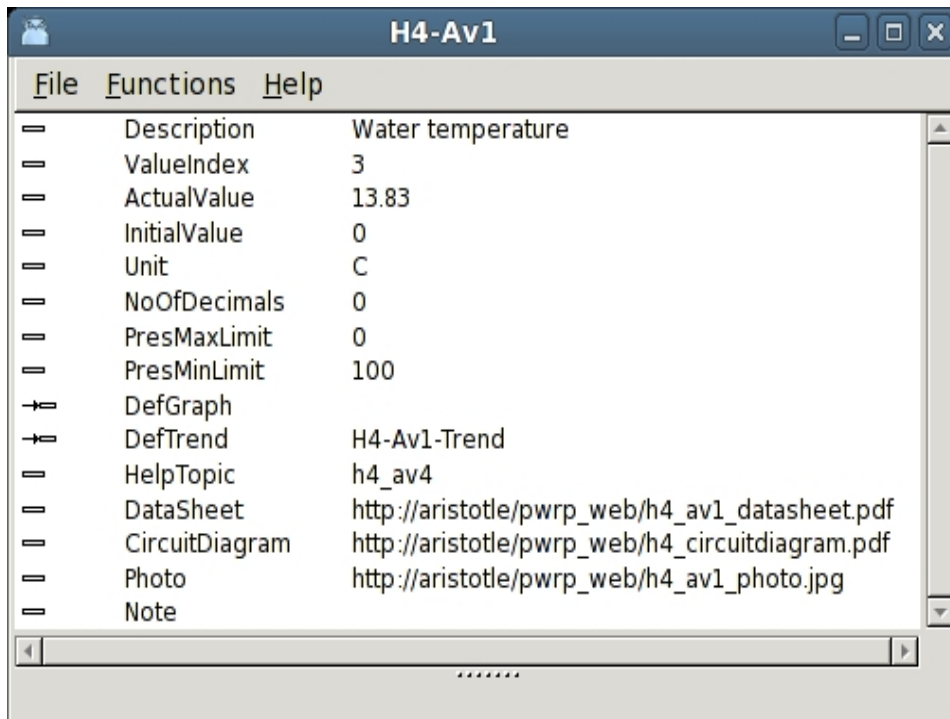
**Fig Action PopupMenu adderad**

När vi nu tar upp bilden i operatörsmiljön och högerklickar på Value fältet öppnas popup-menyn.



**Fig Popupmenyn**

Vilka metoder som finns i popmenyn beror vilken klass det refererade objektet tillhör och hur det är konfigurerat. I figuren nedan visas Av-objektets konfigurerings, och vi kan se attributen DefGraph, DefTrend, HelpTopic, DataSheet, CircuitDiagram, Photo och Note som alla har samband med olika metoder.



**Fig Konfigurereringen av H4-Av1**

Vi ska här gå igenom några av alternativen i popmenyn och se hur de är konfigurerade, eller vad de är beroende av

### **Graph**

För Graph kan man specificera en graf som visar information om objektet, eller anläggningsdelen där objektet förekommer. Konfigureringen sker genom att man lägger in ett XttGraph objekt i DefGraph. För H4-Av1 är inte DefGraph ifyllt, och då öppnas grafen för närmaste överliggande objekt som har DefGraph ifyllt. I det här fallet är det hieraki-objektet H4 som har en default graf angiven.

### **Object Graph**

Object Graph visar objektsbilden för objektet. Objektsbilden är en graf som visar information om ett objekt av en viss klass. Om det finns en objektbild tillgänglig för det aktuella klassen finns det här alternativet i menyn.

### **Trend**

Visar en trend för objektet. Det kräver att en trend är konfigurerad med ett DsTrend objekt, och att trend-objektet är inlagt in attributet DefTrend. DefTrend kan även vara ett DsFastCurve eller PlotGroup objekt.

### **History**

Visar en kurva med process historik för objektet. Det kräver att lagring av data är konfigurerat med ett SevHist objekt under objektet.

### **Hist Event**

Hist Event öppnar en dialog där man kan söka efter larm och händelser som finns lagrade för det aktuella objektet.

### **Block Events**

Block Event öppnar en dialog där man kan blockera larm av olika prioritet.

### **Help**

Help visar en hjälptext för objektet. Hjälptexter skrivs in i en hjälp-fil (\$pwrp\_cnf/xtt\_help.dat) och varje text omfattar ett ämne (topic). Genom att lägga in ett ämne i attributet HelpTopic visas texten för det här ämnet när Help alternativet i popupmenyn aktiveras.

#### **Note**

Med Note kan man skriva in en text som lagras i objektets Note-attribut. Note texten visas i objektets objektsbild.

#### **Open Object**

Open Object öppnar objektet och visar innehållet i objektet.

#### **RtNavigator**

RtNavigator letar upp objektet i runtime navigatören.

#### **Crossreferences**

Visar en lista med referenser till objektet i plc-kod och grafer.

#### **Help Class**

Help Class öppnar en hjälptext för objekts-klassen.

#### **Photo**

Photo visar ett fotografi på objektet i anläggningen. Det konfigureras genom att man lägger in en URL till fotografiet i attributet Photo.

#### **DataSheet**

Om man har ett datablad för objektet, kan man lägga in en URL till detta i attributet DataSheet.

#### **CircuitDiagram**

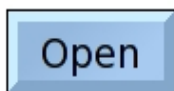
En länk till ett kretsschema kan läggas in i attributet CircuitDiagram.

### **4.5.10 Öppna en graf**

För att öppna en graf genom att klicka på ett objekt, använder man normalt action OpenGraph. Man kan även använda action Command med kommandot 'open graph'.

#### **Öppna en graf från en trycknapp**

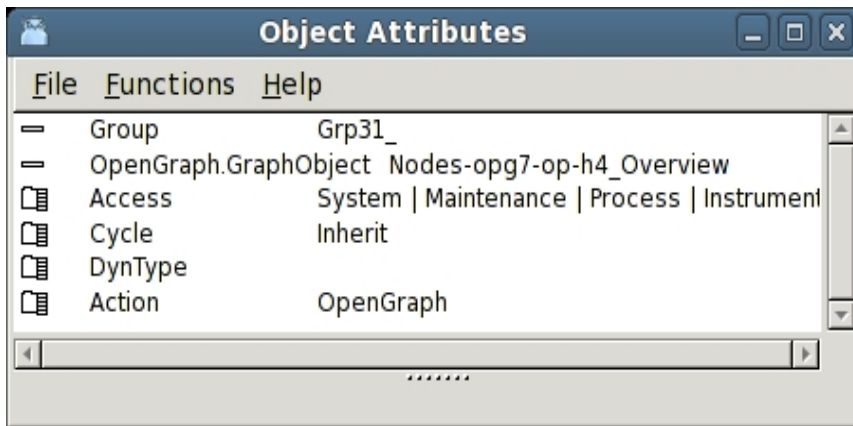
Vi tillverkar en trycknapp för att öppna en bild, med en rektangel och en text som grupperas.



**Fig Trycknapp för att öppna en bild**

#### **Action OpenGraph**

I första exemplet sätter vi action till OpenGraph. OpenGraph kräver att man har ett konfigurerat XttGraph objekt i databasen. XttGraph objektet läggs in i OpenGraph.GraphObject.



**Fig OpenGraph action**

Om grafen ska kunna öppnas på flera olika noder, och man föredrar att använda lokala XtGraph-objekt, kan man ersätta nod-objektet med '\$node'.

### **\$node-op-h4\_Overview**

\$node kommer att ersättas med namnet på nod-objektet för den aktuella noden. Man måste givetvis se till att XtGraph-objekt med motsvarande namn finns på alla aktuella noder.

### **Action Command**

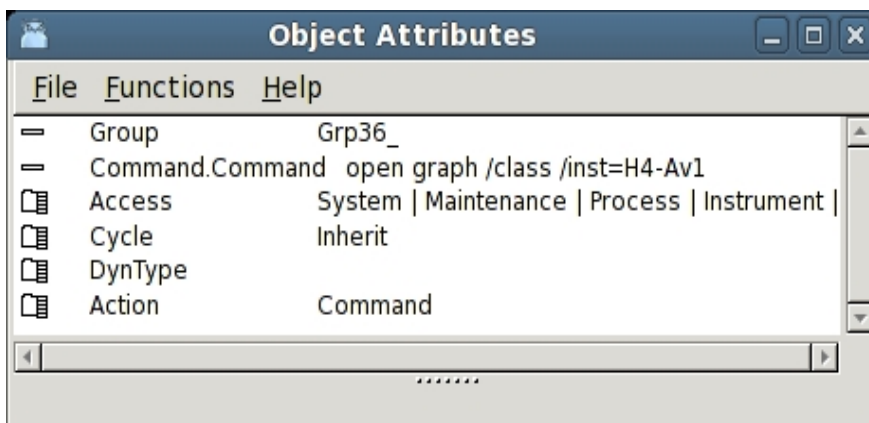
Man kan även öppna en graf med action Command och kommandot

```
open graph 'graphname' [/width=] [/height=]
```

eller en objektsbild med kommandot

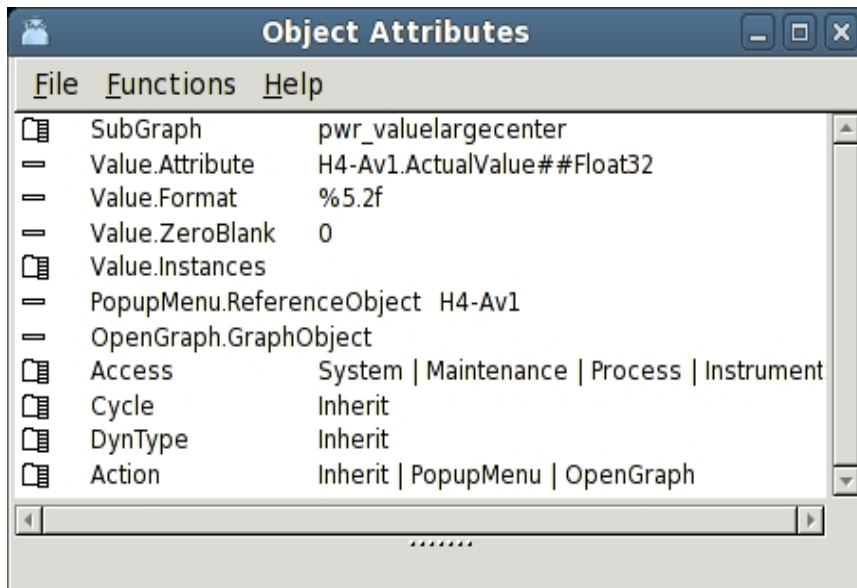
```
open graph /class /instance=
```

I det här exemplet har vi satt Command i Action och lagt ett kommando för att öppna objektsbilden för Av-objektet H4-Av1.



**Fig Öppna en objektsbild**

Om man redan har action PopupMenu och dessutom vill öppna objektsbilden när man klickar på objektet, då räcker det med att addera OpenGraph till action. Man behöver inte fylla i något objekt, utan det objekt som är angivet i PopupMenu.ReferenceObject väljs som default.



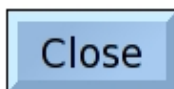
**Fig Öppna objektsbilden när PopupMenu är konfigurerat**

Om man vill öppna ett visst inmatningsfält i bilden, i och med att bilden öppnas använder man kommandot 'open graph /focus'. Här måste man även skicka med namnet på det inmatnings-objekt som ska öppnas (namnet kan man sätta med Edit/Change name i menyn).

```
open graph /object=*-Graphs-SomeXttGraph /focus="TempSetValue"
```

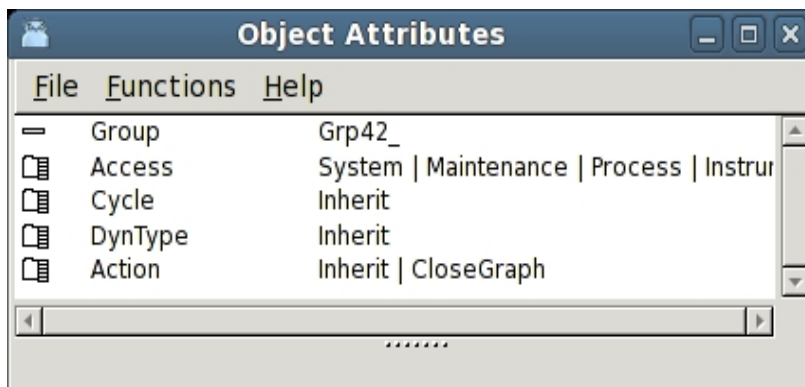
### Stänga en graf

För att stänga en graf används action CloseGraph.



**Fig Trycknapp för att stänga en graf**

Vi skapar en trycknapp med texten 'Close' och sätter Action till CloseGraph.



**Fig Action CloseGraph**

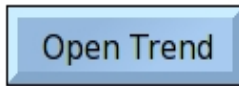
## 4.5.11 Exekvera kommandon

Med action Command kan man ange ett xtt-kommando som exekveras när man klickar på ett objekt. Det finns en rad xtt-kommandon för att öppna bilder, visa kurvor, anropa metoder etc. Xtt

kommandona finns beskrivna i Operator's Guide. Man kan även exekvera script med Command.

I det här exemplet ska vi öppna en trend-kurva från en trycknapp.

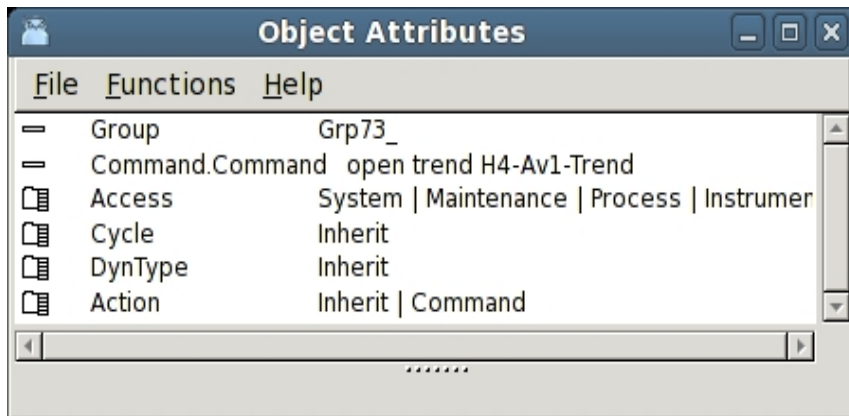
Vi skapar en rektangel, lägger en lämplig text ovanpå och grupperar rektangeln och texten.



**Fig Kommando trycknapp**

I Action markeras Command och i Command.Command läggs xtt-kommandot in. Vi vill öppna en trend-kurva som är konfigurerad med DsTrend objektet H4-Av1-Trend, och kommandot för att öppna kurvan är

```
open trend H4-Av1-Trend
```



**Fig Kommando för att öppna en trend**

### Exekvera ett script

Med ett script kan man utföra flera kommandon samtidigt, men man kan även läsa attribut i databasen, test på olika villkor, och sätta attribut.

Ett script exekveras med ett xtt-kommando genom att ange filnamnet för scriptet med ett '@' tecken framför, t ex

```
@$pwrp_exe/my_script
```

kommer att exekvera scriptet my\_script.rtt\_com på \$pwrp\_exe.

Här är ett exempel på ett script som öppnar grafen g1 om H1-Dv1 är satt, annars öppnas grafen g2.

```
main()  
  int sts;  
  
  if ( GetAttribute( "H1-Dv1.ActualValue", sts))  
    open graph g1  
  else  
    open graph g2  
  endif  
endmain
```

#### 4.5.12 Hjälp och info

Det är naturligtvis viktigt att en operatör enkelt och snabbt kan få hjälp om hur man ska hantera bilder och processer. Ett sätt att lösa det är att lägga in hjälp och info-knappar i bilden. Man kan t ex använda subgraferna ButtonHelp eller ButtonInfo. Action-typen för dessa är Help, och man anger ämne i Help.Topic, samt eventuellt bokmärke i Help.Bookmark. Help-funktionen hämtar upp hjälp-texter som skrivs in i filen \$pwrp\_cnf/xtt\_help.dat, och med ett bokmärke kan man positionera på ett visst ställe i texten.

#### 4.5.13 Navigera med tangentbordet

För mata in data och aktivera objekt utan att använda musen, använder man action-typen InputFocus. Om ett objekt har input focus skickas tangenttryckningar till objektet. För ett ValueInput objekt betyder att man kan mata in ett värde, för en tryckknapp att man kan aktivera den genom att trycka på Return.

Focus input flyttas mellan objekten m h a Tab och piltangenterna. Hur input fokus ska flyttas, konfigureras i attributen NextHorizontal, NextVertical och NextTab. Man lägger här in objektsnamnen på det objekt input fokus ska flyttas till med pil höger, pil ner resp. Tab tangenterna.

Att ett objekt har input fokus markeras normalt med att en relief ritas runt objektet.

### 4.6 Gif och JPEG-bilder

Man lägger in en gif eller JPEG-bild på följande sätt. Kopiera bild-filen till \$pwrp\_pop. Den dyker då upp under mappen Local/Images i subgraf-paletten. Välj ut bilden och klicka med MB2 i arbetsarean.

#### Ändra färg

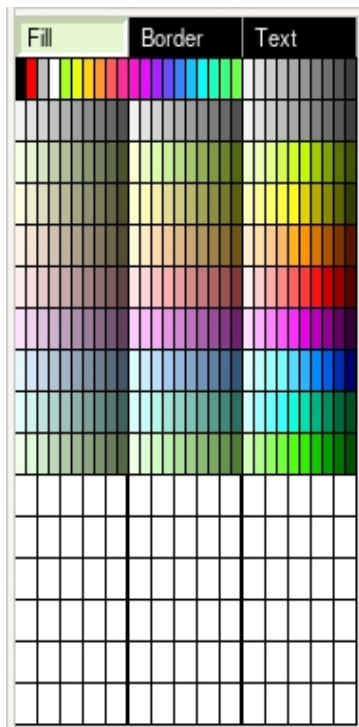
Funktionerna för att ändra färg på subgrafer, dvs färgton, ljushet, intensitet och färgskiftning, fungerar även på bild-objekt.

#### Dynamik

Man kan inte sätta någon dynamik direkt på bild-objekten. Däremot kan man låta dem ingå i en grupp eller subgraf, och ange dynamik för gruppen eller subgrafen. Om man dynamiskt vill ändra färg på bild-objektet ska man använda en dynamik-typ som ändrar färg-tonen, t ex DigTone.

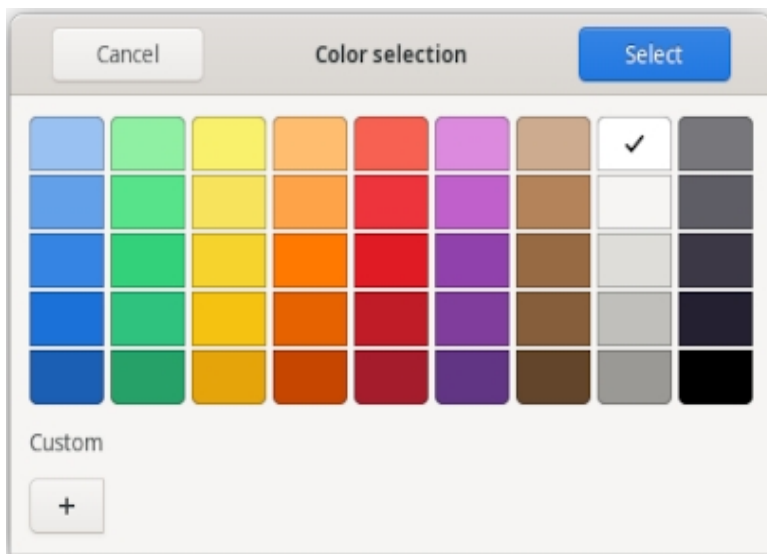
### 4.7 Custom colors

I färgpaletten finns 300 färger men ibland går det inte att hitta den färgton man söker där. Då kan man använda custom colors. De återfinns under standardpaletten och visas genom att öka storleken på färgpalettfönstret, eller genom att skrolla ner.



**Fig Custom color palett**

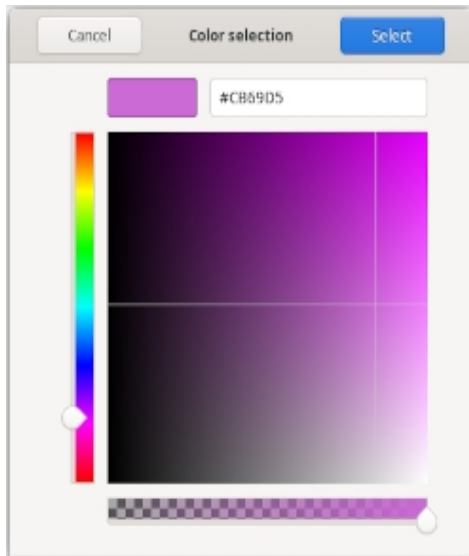
Skapa en custom color genom att dubbelklicka på ett tomt fält i custom color-paletten.



**Fig Val av färg**

Nu öppnas färgvals dialogen. Antingen kan man välja en av de fördefinierade färgerna, eller kan man trycka på '+' knappen för att välja en mer specifik färg.





**Fig Val av färg**

Välj valör med dragregeln till vänster, och klicka i färgytan för att välja ljushet och mättnad. Den undre dragregeln för transparens är inte implementerad ännu. Tryck på Select-knappen för att lägga in för lägga in färgen i paletten. Färgen kan nu användas som fyllnads, kant eller text-färg.

## 4.8 Färgteman

Operatören kan välja ett färgtema i operatörmiljön, och genom att rita en graf med färgtema-färger kommer även processbilder att anpassas till det valda färgtemat. Färgtemat laddar en uppsättning färger i custom color paletten, och varje färg används för ett specifikt ändamål. Den första färgen ska till exempel användas som bakgrundsfärg. För att följa färgtemat ska man även använda subgraferna under ColorTheme i subgrafspaletten. Dessa är ritade med färger från färgtemapaletten.

Här följer en beskrivning av hur man skapar en enkel graf med färgtema.

Sätt ColorThem i File/Graph Attributes till \$default.

Välj ett färgtema i File/ColorTheme/Select, till exempel Sand. Nu fylls custom color paletten med färger från det valda färgtemat.



**Fig Färger för Sand färgtema laddade**

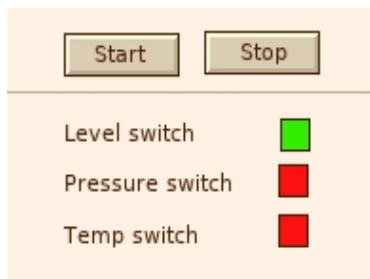
Den första custom color färgen ska vara bakgrundsfärg. Genom att sätta markören över en färg i paletten skrivs ändamålet för färgen under custom color paletten. Klicka på färgen för att sätta fyllnadsfärg, och aktivera Functions/Set background color.

För att skapa en tryck-knapp, välj en knapp under ColorTheme, till exempel ColorTheme/Pushbuttons/ButtonSmallToggle. Koppla och lätt in en text, t ex 'Start' eller 'Stopp'.

Skapa indikatorer från ColorTheme/Indicators, t ex IndSquareGreen eller IndSquareRed.

Skriv texter genom att sätta den femte färgen i custom color paletten, 'Text/Lines on background' som textfärg. Skapa sedan ett textobjekt.

Rita avgränsningslinjer mellan tryck-knapparna och indikatorerna genom att sätta den fjärde färgen, 'Delimiter line', som kantfärg, och rita linjen.



**Fig Enkel graf med färgtema**

Med File/ColorTheme/Next (Ctrl+Alt+T) byts färgtema och man kan se hur grafen fungerar för andra färgteman.

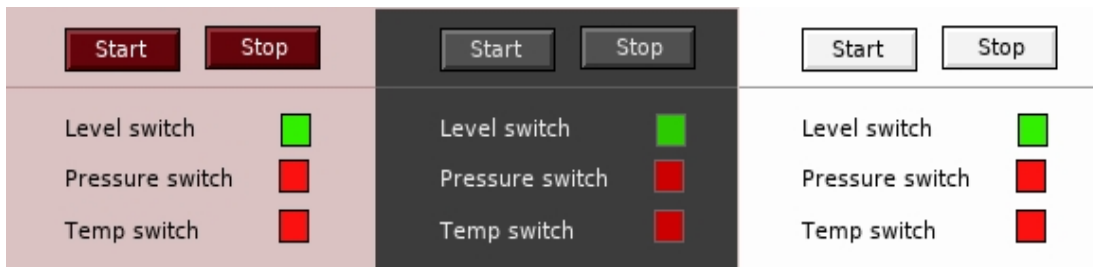


Fig Enkel graf med olika teman

## 4.9 Objektträd

Objektträdet visar de grafiska objekten i en graf i en trädstruktur. Det öppnas från View/View Object Tree. Objekt kan väljas ut med klick MB1 eller med Pil upp och ner tangenterna. Egenskaperna för ett objekt öppnas genom att klicka på lövet eller med pil höger tangenten. Det här är ofta ett enkelt sätt att sätta egenskaper för objekt, särskilt om det är en uppsättning linkande objekt, eller för objekt i en grupp.

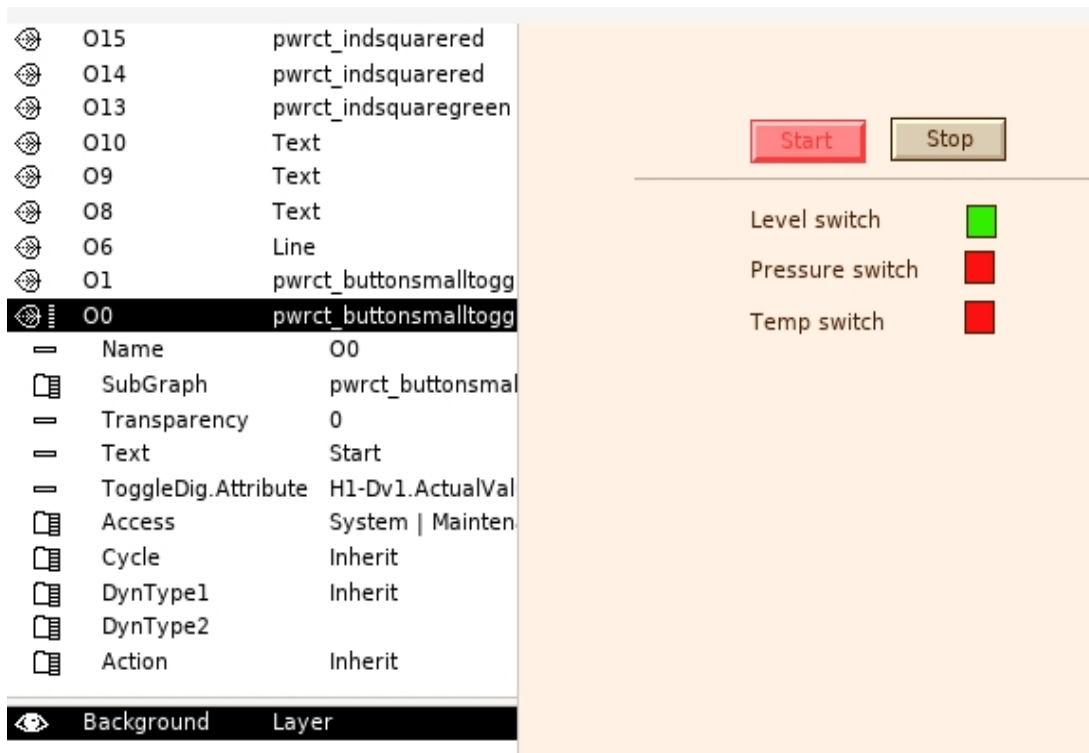


Fig Objektsträd

## 4.10 Lager

De grafiska objekten kan skapas i lager. Det är ett sätt att strukturera grafen och att förenkla editeringen. Det är endast möjligt att editera objekt i det lager som för närvarande är aktivt. Därigenom finns det ingen risk att objekt i andra lager av misstag flyttas eller ändras.

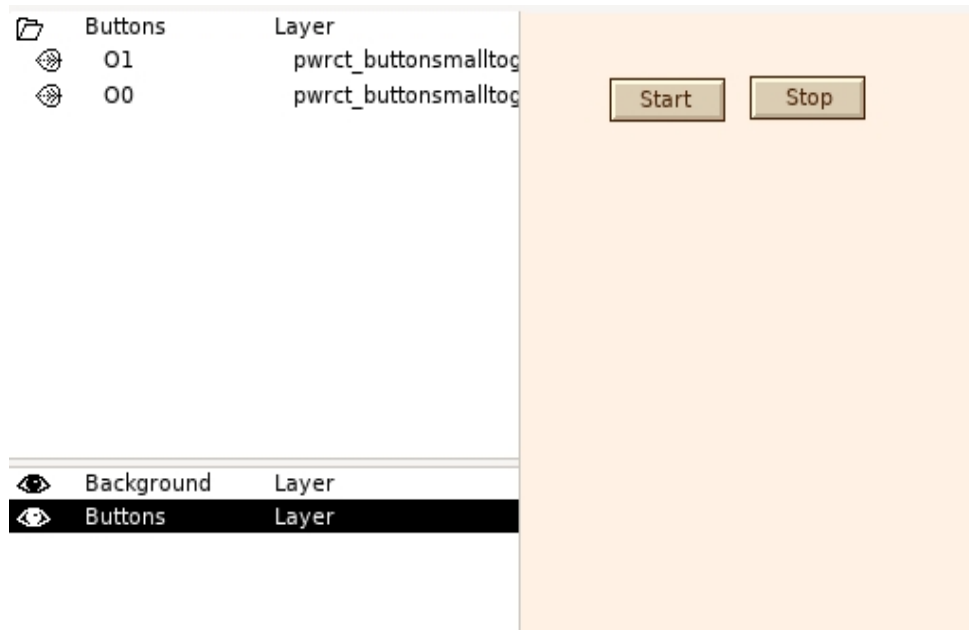
Ett lager skapas från 'Layers/Create layer' i menyn. När ett lager har skapats visas det i lagerlistan under objektträdet. När man vill skapa eller editera objekt i ett lager, gör man lagret aktivt genom att välja ut det i lagerlistan. Genom att klicka på ögon-symbolen kan lagret göras

osynligt eller synligt.

Lagret visas också i objektträdet. Det kommer att vara förälder till all objekt i lagret. Även egenskaper för lagret kan visas och editeras i objektträdet. Genom att klicka på lövet eller mappen, eller med Pil höger tangenten, visas egenskaperna för lagret. Genom att klicka med Shift + Klick MB1 på mappen, visas objekten i lagret.

Som exempel kommer vi att skapa den enkla grafen ovan med lager, och lägga knappar, texter och indikatorer i olika lager.

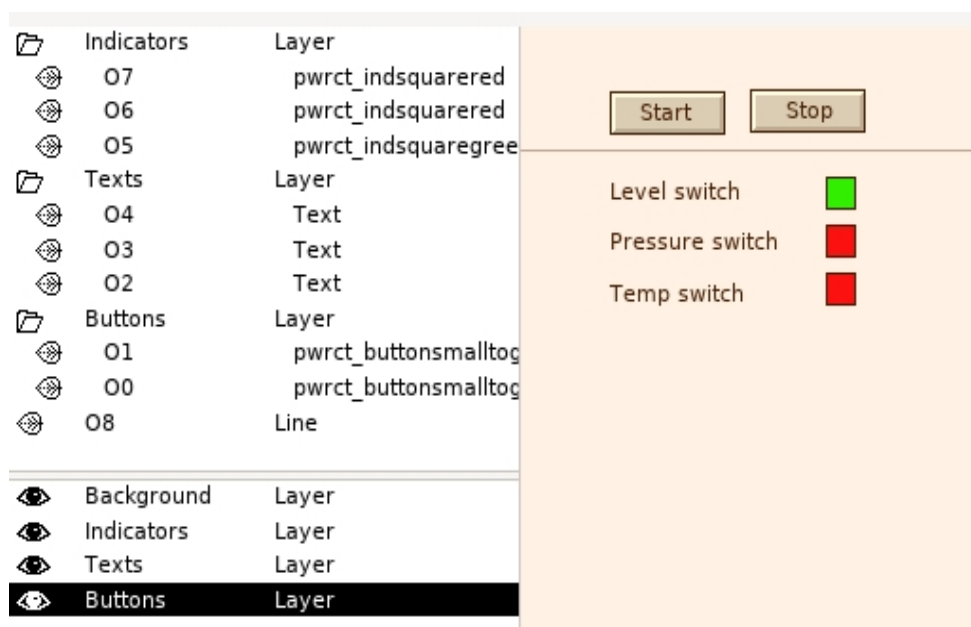
Vi skapar ett lager från 'Layers/Create layer' i menyn, öppnar egenskaperna för lagret i objektträdet, och ändrar namnet till 'Buttons'. Sedan väljer vi ut Buttons-lagret i lagerlistan för att göra det aktivt. De två knapparna vi nu skapar kommer att läggas i Button-lagret.



**Fig Buttons layer**

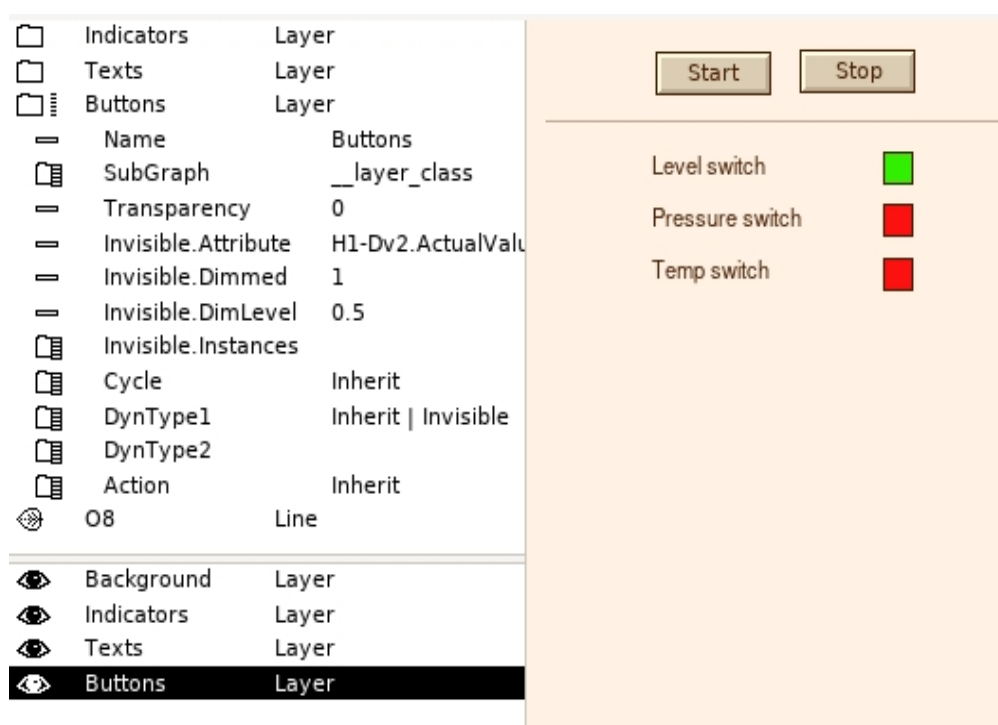
På samma sätt skapar vi Text-lagret med texter, och Indicator-lagret med indikatorer. Notera hur objekt som inte tillhör det aktiva lagret är okänsliga, och hur lager kan göras synliga resp osynliga med ögon-symbolen i lagerlistan.

Ävgränsningslinjen är skapad in bakgrundslagret. Objekt i bakgrundslagret visas på översta nivån i objektträdet.



**Fig Lager**

Det är också möjligt att sätta dynamik på ett lager. Genom att sätta Invisible dynamik med Dimmed på Buttons-lagret, blir alla knapparna in lagret dimmade och okänsliga när den kopplade signalen är satt.



**Fig Lager med dynamik**

## 4.11 Rita en subgraf

En subgraf tillverkas genom att man ritar basobjekt i arbetsarean och sedan sparar dessa som en subgraf. Man kan även lägga in andra subgrafer i arbetsarean, men dessa kan inte ha någon

egen dynamik, utan följer den dynamik som gäller för huvud-subgrafen. Det finns två speciella objekt som kan användas i subgrafer: kopplingspunkter och annotationer.

Kopplingspunkter gör det möjligt att dra kopplings-linjer mellan objekt. De skapas genom att aktivera kopplingspunkt i verktygspanelen och klicka med MB1 i arbetsarean. I objekts-editorn för kopplingspunkten kan man ställa in riktning (Direction) för punkten, dvs den riktning som en koppling ut från punkten kommer att få. Kopplingspunkter på vänstersidan av ett objekt, bör ha Direction Left, ovensidan Up, högersidan Right och undersidan Down.

Annotation är plats för en text, som är olika för olika instanser av subgrafen. Den skapas genom att välja An i verktygspanelen och klicka med MB1 i arbetsarean. När man sedan skapar instanser av subgrafen kommer annotationerna att visas som attribut i objektseditorn för subgrafs-objekten som A1, A2 osv, och man kan härigenom lägga in texter i annotationerna.

När subgrafen är färdigritad tar man upp File/GraphAttributes och lägger in en etta i Subgraph. Genom att stänga fönstret och ta upp det igen, kan man lägga in attribut för subgrafen. Bl a kan man lägga in default-värden för dynamik-typ, DigLowColor mm. Därefter sparar man den under ett lämpligt namn.

Man skapar instanser av subgrafen genom att väja ut den under mappen Local/Subgraphs i subgraf-paletten, och klicka med MB2 i arbetsarean.

### **Subgraf med flera sidor**

Vissa typer av dynamik, t ex animationer, bygger på att en subgraf har flera sidor. Vid animering gör man flera sidor med små förändringar mellan varje sida, och sidorna genomlöps sedan och ger illusionen av rörelse. Dynamik-typerna DigShift och AnalogShift använder också olika sidor, och skiftar mellan olika sidor, beroende på värdet av en digital resp analog signal.

När man ska rita en subgraf med flera sidor, gör man den första sidan på normalt sätt. När man har sparat den som en subgraf aktiverar man File/Page/Create next page (Shift+Ctrl/N). Nu skapas den andra sidan vilket markeras i titeln på fönstret. När förändringarna i sida nummer två är gjorda, sparar man denna och aktiverar återigen Create next page för att skapa den tredje sidan osv.

Man kan skifta mellan olika sidor med File/Page/Next page (Ctrl/N) och File/Page/Previous page (Ctrl+J). Ritar man animationer, är detta ett värdefullt sätt att se skillnader mellan de olika sidorna, och utröna hur animationen fungerar.

### **Slider**

En slider konstrueras med två separata subgrafer, en bakgrund, och en rörlig del. Slidern ska alltid ritas stående, dvs så att rörelsen sker vertikalt.

För bakgrunds-subgrafan mäter man y-koordinaten för de punkter som slidern ska kunna röra sig mellan, och lägger in i attributen y0 och y1 i Graph attributes. Som DynamicType lägger man in SliderBackground.

För den rörliga subgrafan mäter man y-koordinaten för den punkt som ska ställas in mot signalvärdet, och lägger in i y0. Attributet Slider sätts till 1.

### **FillLevel**

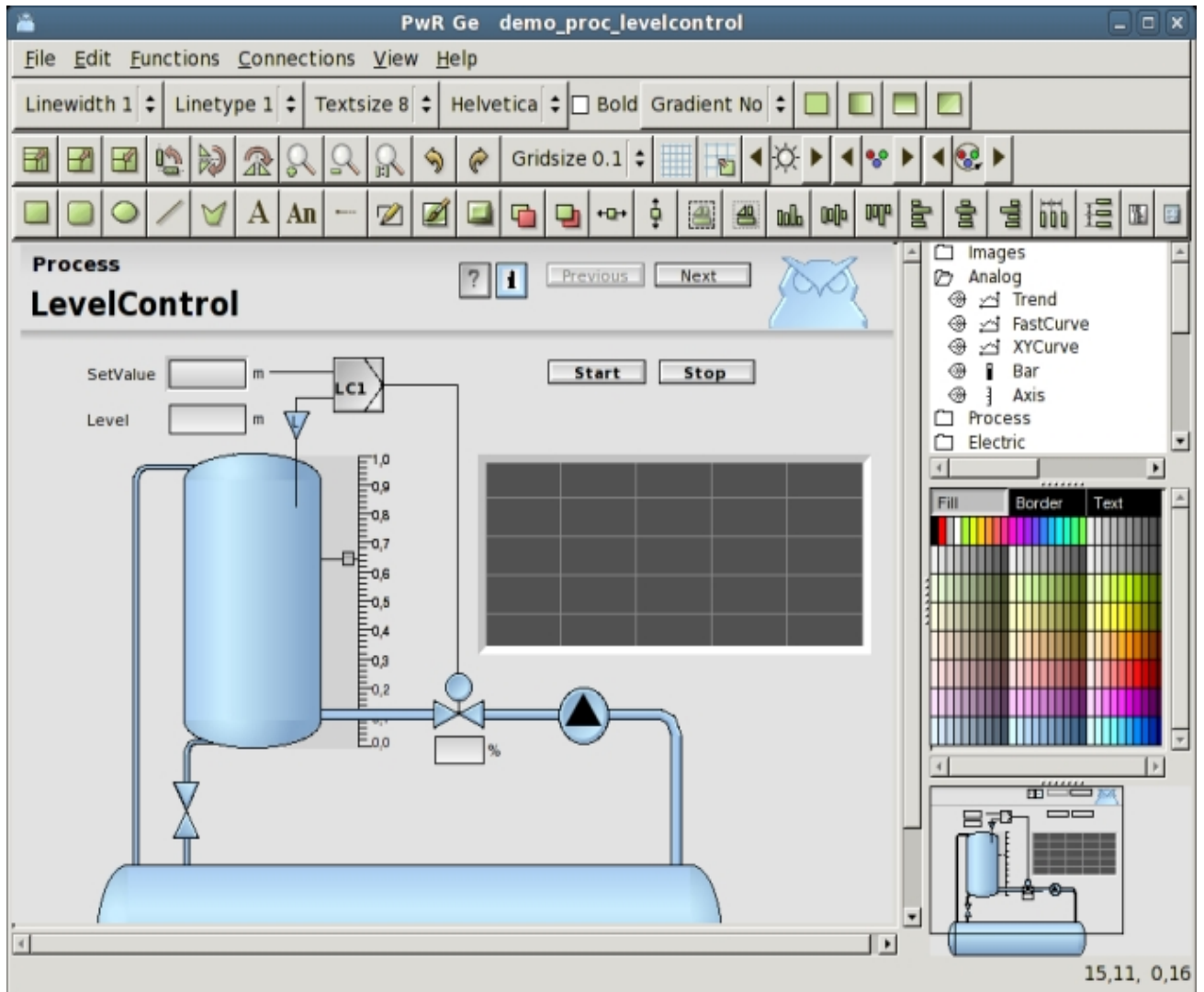
För en subgraf med dynamiken FillLevel, t ex en termometer, vill man ofta sätta en min och en max-nivå, mellan vilka nivån kan ändras. Genom att mäta max och min-nivån och lägga in dem i attributen y0 resp y1 i Graph Attributes får man den här funktionen. Subgrafan måste alltid ritas stående så att nivån ändras nedifrån och upp.

## **Animation**

För subgraf med dynamiken Animation, väljer man även en lämplig AnimationSequence i Graph Attributes. AnimationSequence kan vara Cyclic, Dig eller ForwBack.

# 5 Fönster

I ge finns ett antal paletter och andra fönster för att välja objekt, subgrafer, färger mm.



## Arbetsarean

I arbetsarean editeras bilden genom att klicka och dra med musen.

### Musklick

Klick MB1  
Shift/Klick MB1  
Press MB1  
Shift/Press MB1  
DubbelKlick MB1  
Ctrl/DubbelKlick MB1

### Funktion

Välj ut ett objekt, eller skapa ett basobjekt.  
Addera ett objekt till utvalslistan.  
Välj ut objekt inom ett område, eller flytta på objekt.  
Addera objekt inom ett område till utvalslistan.  
Öppna objektseditorn för ett objekt.  
Koppla första attributet i ett subgraf-objekt till ett rtdb-objekt.



Shift/Ctrl/DubbelKlick MB1

Koppla andra attributet i ett subgraf-objekt till ett rtdb-objekt.

Shift/DubbelKlick MB1

Lägg in vald fyllnadsfärg i LowColor i ett subgrafs-objekt.

Klick MB2

Skapa ett subgrafs-objekt.

Press MB2

Skapa en koppling

## Navigationsfönster

Längst ner till vänster finns ett navigations-fönster som innehåller in mini-kopia av hela arbetsarean. Från navigationsfönstret kan man förflytta sig (MB1) och zooma (MB2) bilden i arbetsarean.

## Subgraf-paletten

Subgraf-paletten består av ett antal mappar med subgrafer. Under Local/Subgraphs-mappen finns subgrafer som har skapats inom projektet. Övriga mappar innehåller subgrafer som följer med ProviewR.

## Färg-paletten

Färg-paletten innehåller 300 färger i 10 rader med 30 färger i varje kolumn. I första rader finns svart, vit och intensiva färger, och en samling gråtoner. Sedan följer en rad med gråskala, och efter den 8 rader med olika färgton, gulgrön, gul, orange, violett, blå, blågrön och grön. Inom varje färggrupp har de 10 första låg mättnad, de 10 mellersta medium mättnad, och de 10 sista hög mättnad.

Det tre kanpparna över paletten visar de nuvarande fyllnads, kant och text-färgerna. Nya objekt kommer att skapas med dessa färger. För att ange en ny fyllnadsfärg, sätt paletten i fyllnadsfärgs-mod genom att trycka på Fill-knappen och välj en färg i paletten. För att sätta fyllnadsfärg på ett existerande objekt, välj ut objektet, se till att paletten är i fyllnadsfärgs-mod, och välj önskad färg. På samma sätt kan paletten sättas i kantfärgs eller textfärgs-mod genom att trycka på Border resp Text knappen, och kant och text-färger kan hanteras.

Objekt har även en bakgrundsfärg, och den hanteras på ett sätt som avviker från övriga färger. För att sätta bakgrundsfärg på ett objekt, välj ut objektet och klicka med Ctrl+Shift MB1 på den önskade färgen i paletten.

För att sätta bakgrundsfärg på grafen, sätt den önskade färgen som nuvarande fyllnadsfärg, och aktivare 'Functions/Set background color' i menyn.

## Custom colors

Utöver de fördefinierade färgerna finns även utrymme för 90 användardefinierade färger (custom colors) under de fördefinierade färgerna. För att sätta en custom color, öppna färg-väljaren genom att dubbelklicka på ett fält i custom color paletten. Tryck på '+' knappen för att ange en mer specifik färg än de fördefinierade. Välj valör med dragregeln till vänster, och klicka i färgytan för att välja ljushet och mättnad. Den undre dragregeln för transparens är inte implementerad ännu. Tryck på Select-knappen för att lägga in för lägga in färgen i paletten. Färgen kan nu användas som fyllnads, kant eller text-färg.



**Fig Färgväljaren**

För att exportera custom color paletten till en annan graf, spara paletten från 'File/CustomColors/Save' i meny. Paletten sparas på en pwgc-fil in \$pwrp\_pop. Ladda in filen i en annan graf genom att aktivera 'File/CustomColor/Load' och välj ut filen.

Det är också möjligt att använda en gemensam färg-fil för flera grafer i ett projekt. Om en färg-fil anges i ColorTheme i Graph attributes, kommer färgfilen att laddas automatiskt när grafen öppnas. När en färg ändras och sparas, kommer den att påverka alla objekt som är ritade med färgen i alla grafer med denna färg-fil. På detta sätt kan färger i graferna enkelt ändras.

För grafer ritade med färgteman, används custom color paletten för tema färger och kan inte användas som användardefinierade färger.

## Color tones

I färgpaletten finns plats för 300 olika färg-nyanser. Här väljer man fyllnadsfärg, kantfärg och text-färg. Fyllnadsfärgen väljs med MB1 och visas i det vänstra fältet överst i paletten, kantfärg väljs med MB2 och visas i mitten fältet, och textfärgen väljs med Shift MB1 och visas i det högra fältet.

Paletten är uppdelad rader med 30 färger i varje. Den första raden består av svart, vitt, starka signalfärger samt en uppsättning gråtoner. Sedan följer en rad med gråskalan, och där efter 8 rader med olika färgton: gulgrön, gul, orange, röd, violett, blå, blågrön och grön. Inom varje färggrupp har de 10 första låg mättnad, de 10 mitten-färgerna lite högre mättnad, och de 10 sista hög mättnad.

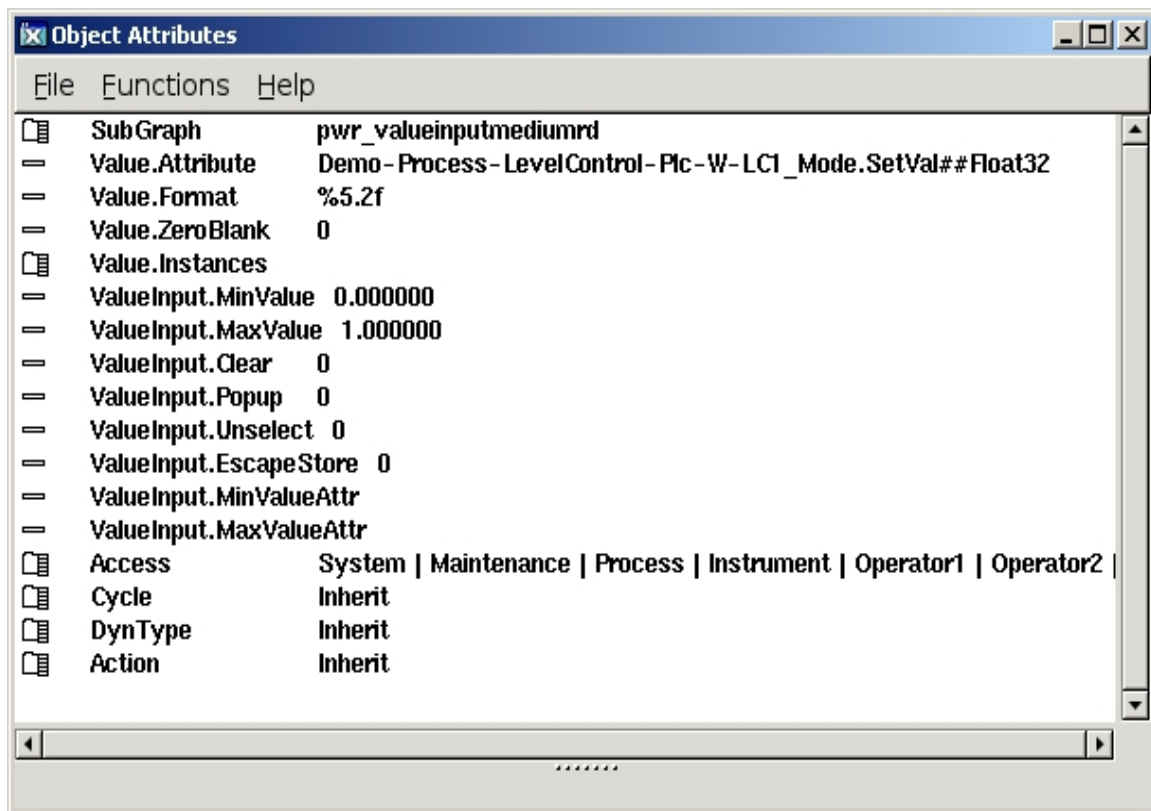
Under färg-paletten finns även en färgton-palett med vilka man kan välja färgton för subgrafs-objekt.

Här finns även en reset-knapp med vilken man kan återgå till objektets ursprungliga färg.

## Verktygs-panel

Verktygs-panelen består av ett antal tryckknappar för att skapa rektanglar, ellipser, linjer etc. Där finns även funktioner för att rotera och skala, sätta linjebredd, fontstorlek, modifiera färger på subgrafer mm.

## Objekts och attribut-editorn



Objekts-och attribut-editorn gör det möjligt att ange olika egenskaper på objekt, grafer och subgrafer.

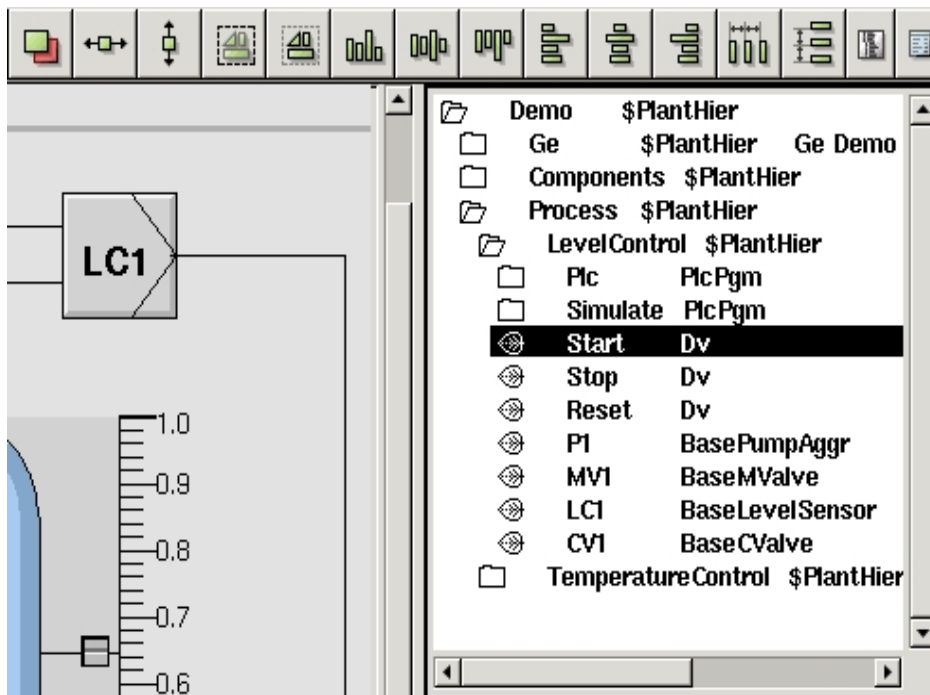
För ett objekt tas den upp genom att man dubbelklickar på objektet, för graf eller subgrafer tas den upp från menyn under File/Graph attributes.

Värden i objektseditorn ändras genom att man markerar attributet och trycker på PilHöger eller från menyn med Functions/Change value.

Bitmaskar, enum-attribut och vektorer öppnas med PilHöger eller DubbelKlick MB1. Checkboxar ändras med PilHöger eller genom att klicka i checkboxen.

Det snabbaste sättet är att välja ut attribut med PilUpp och PilNer, att använda PilHöger för att öppna och ändra attribut och PilVänster för att stänga attribut.

## Plant hierarki



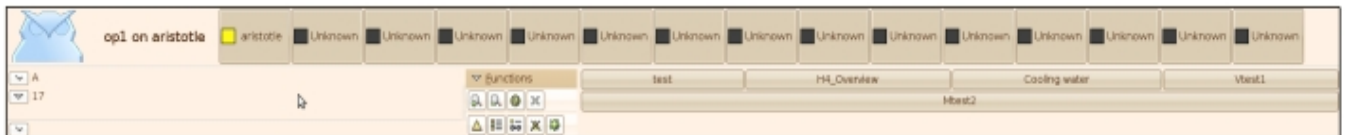
Om Ge startas från Navigatören visas planthierarkin i ett fönster under färgpaletten. Det används för att koppla dynamiska objekt till signaler i rtdb. Man kopplar ett subgraf-objekt till en signal genom att välja ut objektet (eller ett attribut i objektet) i planthierarkin, och klicka med Ctrl/DubbelKlick MB1 på subgraf-objektet. Då läggs signalnamet in i subgrafs-objektets Attribute-attribut. Finns det ett andra attribut som ska kopplas till en signal, kopplas detta på samma sätt med Shift/Ctrl/DubbelKlick MB1.

## 6 Färgteman

Färgteman tillhandahåller ett antal färgkombinationer för operatörmiljön, för att tillgodose operatörernas preferenser och kunna anpassa ljusheten av miljön till omgivningen. Operatören kan välja ett färgtema från menyn i operatörsfönstret, vilket kommer att påverka utseendet på alla vertyg och grafer i ProviewR's bassystem.



**Fig Val av färgtema**



**Fig Operatörsfönstret i färgtema Sand.**

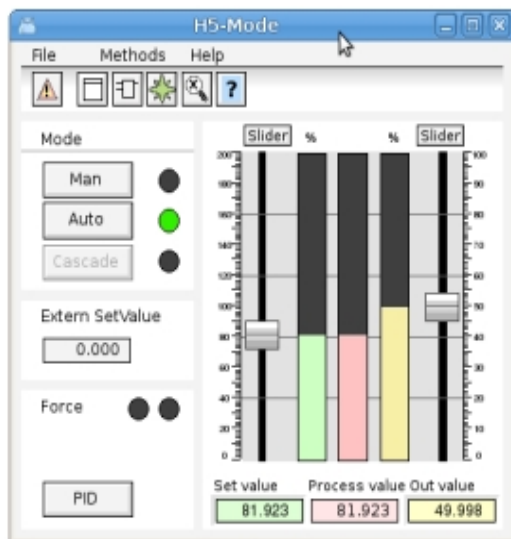


Fig Färgteman Standard och Sand

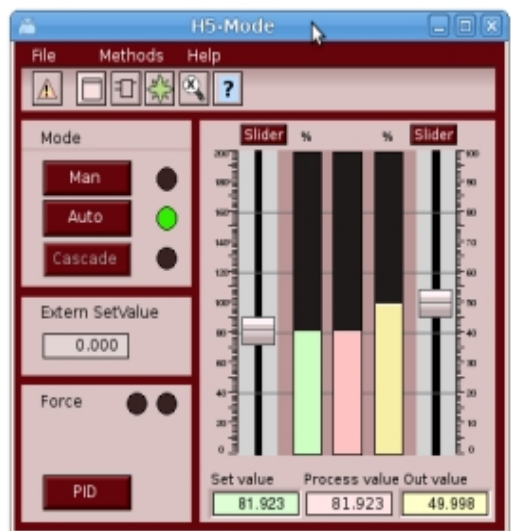
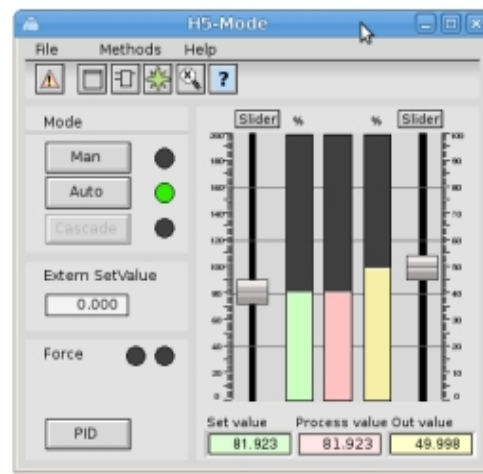


Fig Färgteman Marron och Sienna



Fig Färgteman DarkBlue och Classic



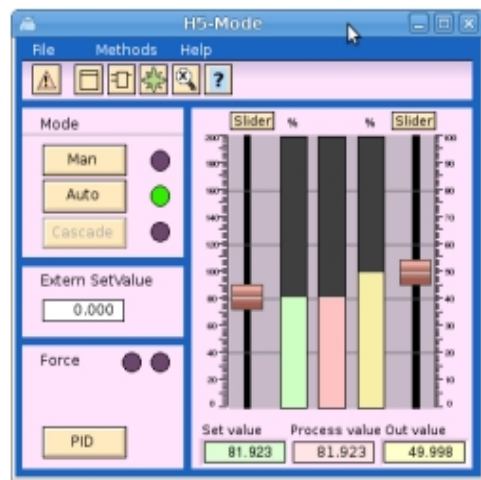


Fig Färgteman Midnight och Playroom

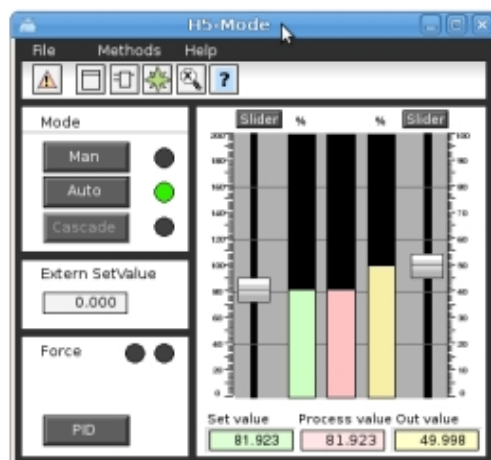
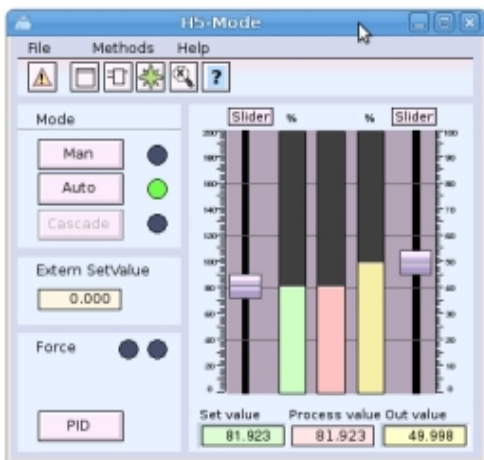


Fig Färgteman NordicLight och Contrast

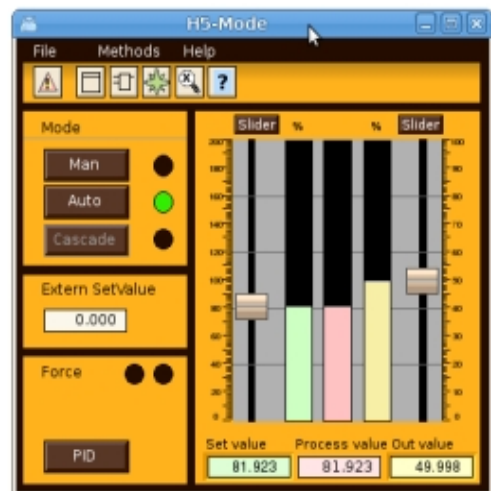
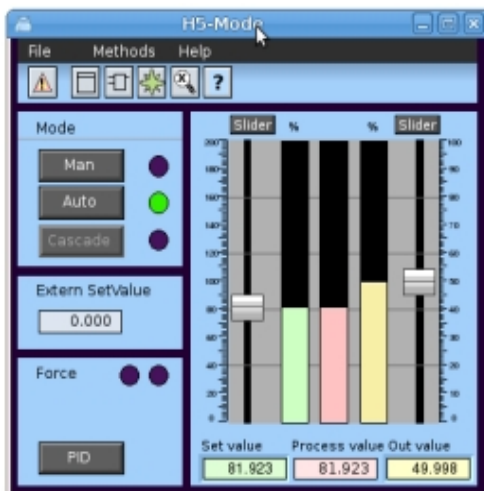
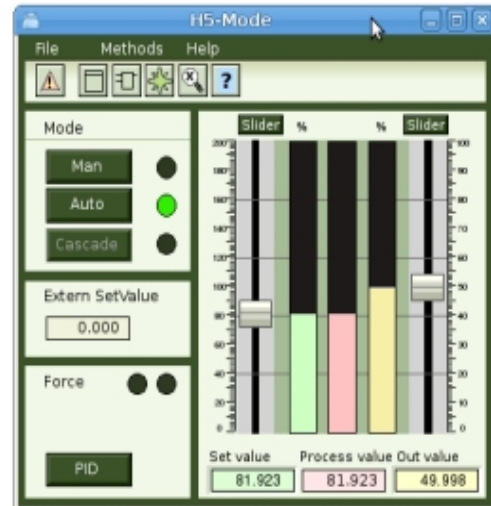


Fig Färgteman AzurContrast och OchreContrast



**Fig Färgteman Chesterfield och TerraVerte**



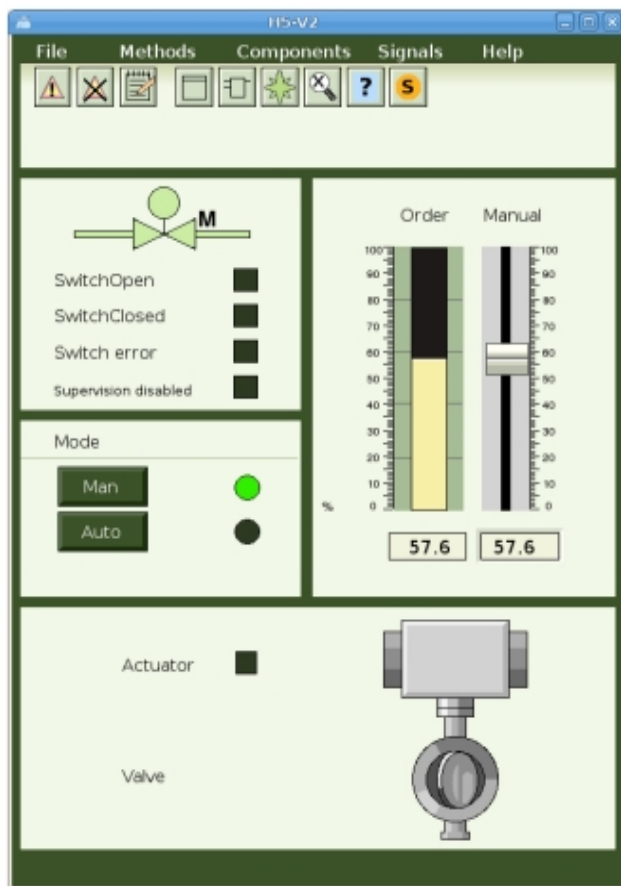
**Fig Färgtema Polar**

Färgtema kan även användas för att rita grafer i ett projekt. Dessa grafer kommer att följa utseendet för ett färgtema som operatören har valt, men det krävs att speciella färger i custom color paletten används när de olika objekten i bilden ritas. Ett färgtema laddas från File/ColorTheme/Select i Ge menyn. Färgerna för demat läggs då in i custom color paletten. Varje färg i custom color paletten är anpassad för ett specifikt ändamål. Den första färgen t ex är bakgrundsfärgen för grafen, och den femte färgen ska användas för texter på bakgrunden. När man placerar markören på en färg i paletten skrivs ändamålet för färgen ut i ett fält under paletten.

Färgtema stödjer färgsättningen av specifika element i en graf. I exemplet nedan är bakgrundsfärgen ljusgrön. De mörkgröna linjerna kallas avgränsningsfält, 'delimiter area'. I det här fallet är de rektanglar som indelar grafen i olika sektioner. De kan även användas för större områden med text och indikatorer. Längs upp finns en meny, och under den en gradient för makera en skugga under menyn. Objektbilder bör innehålla en 'method toolbar' placerad under menyn. Andra element är staplar, skalor, knappar, indikatorer, texter på bakgrunden och texter på avgränsningsfält, tabeller, diagram etc.

För större och mer komplexa element som inte är anpassade för en förändring av fyllnadsfärgen, som ställdonet och ventilen nedan, kan dynamiken ColorThemeLightness användas för att anpassa ljusheten av dessa element till ljusheten hos färgtemat.



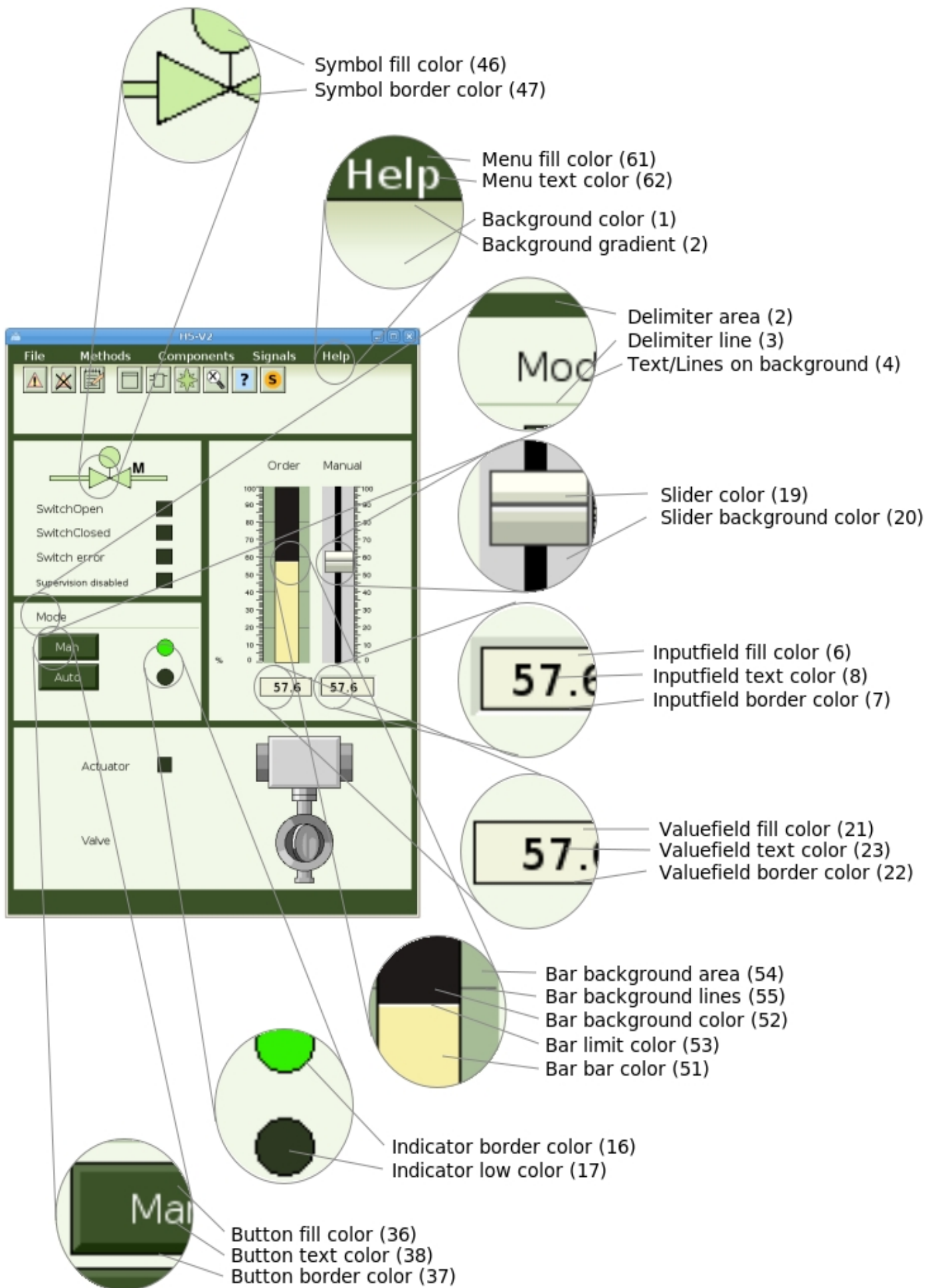


**Fig Graf ritade med färger från färgtemat**

Beskrivning av färgtema färger

Namn	Beskrivning
1 Background	Ska sättas som bakgrundsfärg för grafen. Ljusgrön i exemplet ovan.
2 Background gradient	Kan användas som gradient till bakgrundsfärgen, t ex gradienten under menyn i exemplet ovan.
3 Delimiter area	Färg på avgränsningytor. De mörkgröna områdena.
4 Delimiter lines	Färg på linjer som delar upp bakgrunden, t ex linjen under Mode ovan.
5 Text/Lines on background	Färg på texter och linjer på bakgrunden.
6 Input field fill color	Fyllnadsfärg på ValueInput-objekt.
7 Input field border color	Kantfärg på ValueInput-objekt.
8 Input field text color	Textfärg på ValueInput-objekt.
16 Indicator border color	Ska sättas som kantfärg på indikatorer.
17 Indicator low color	Färg när signalen är låg för indikatorer.
18 Indicator on delimiter low color	Färg när signalen är låg för en indikator på en avgränsningsyta.
19 Slider color	Färg på dragreglar.
20 Slider background color	Färg på bakgrund för dragreglar.
21 Value field fill color	Fyllnadsfärg för Value-objekt.
22 Value field border color	Kantfärg för Value-objekt.
23 Value field text color	Textfärg för Value-objekt.
31 Limit switch high color	Färg för gränsläge när signalen är hög.
32 Limit switch low color	Färg för gränsläge när signalen är låg.

33	Limit switch border color	Kantfärg för gränsläge.
34	Text/Lines on delimiter	Färg på texter eller linjer på avgränsningsytor.
35	Button active color	Färg för att markera att en knapp är aktiv.
36	Button fill color	Fyllnadsfärg för knappar.
37	Button border color	Kantfärg för knappar.
38	Button text color	Textfärg för knappar.
39	Button insensitive text color	Textfärg när en knapp är satt okänslig av Invisible-dynamik.
40	Button insensitive border color	Kantfärg när en knapp är satt känslig av Invisible-dynamik.
46	Symbol fill color	Fyllnadsfärg för komponenters symboler.
47	Symbol border color	Kantfärg för symboler.
48	Symbol low color	Färg när symbolen är låg.
49	Symbol empty color	Färg för att markera att en ventil är stängd eller tom.
50	-	
51	Bar bar color	Färg för stapel-delen i ett stapel-objekt.
52	Bar background color	Färg för bakgrunds-delen in ett stapel-objekt.
53	Bar bar limit color	Färg för kanten i ett stapel-objekt.
54	Bar background area	Färg för omgivande ytor för stapel-objekt.
	Color used for surrounding area for bar objects.	
55	Bar background lines	Mörkgrönt i exemplet ovan. Färg för linjer på omgivande ytor för stapel-objekt.
61	Menu fill color	Fyllnadsfärg för menyer.
62	Menu text color	Textfärg för menyer.
63	Toolbar fill color	Fyllnadsfärg för metod-panelen i objektbilder.
64	Toolbar border color	Kantfärg för metod-panelen.
65	Toolbar text color	Textfärg för metod-panelen.
66	Diagram fill color	Fyllnadsfärg för diagram.
67	Diagram border color	Kantfärg för diagram.
68	Diagram curve color	Default kurvfärg för diagram.
69	Axis border color	Färg för Axis-objekt.



## **Fig Exempel på temafärger**

### **Custom color theme**

Färgtema-väljaren har ett Custom alternativ. För att skapa ett eget färgtema, ladda ett färgtema i custom color paletten, dubbelklicka på färger för att öppna färgväljaren och ändra färgerna. När de önskade färgerna är modifierade, spara temat från /FileCustomColors/Save med namnet 'pwr\_colortheme100'. Färgtemat sparas i filen \$pwrp\_pop/pwr\_colortheme100.pwgc. Kopiera den här filen till \$pwrp\_exe och se också till att den distribueras till operatörsstationen.

För att välja Custom temat i operatörsmiljön, öppna färgtemaväljaren från 'Functions/View/Color theme' i operatörsfönstrets meny, och välj 'Custom'.

# 7 Objekt

Objekt skapas genom att man väljer objektet i verktygspanelen och ritar det i arbetsarean med MB1. Objektet ritas med de aktuella värdena för fyllnad, fyllnadsfärg, kant, kantfärg och linjebredd. Om egenskaperna för ett existerande objekt ska ändras, markerar man det och markerar den färg, linjebredd eller dylikt som ska gälla. För ett objekt kan attribut-editorn öppnas genom att dubbelklicka på objektet. Nedan anges under rubriken egenskaper de attribut som kan ändra från verktygspanelen eller färg-paletten, och under rubriken attribut de attribut som finns i objektseditorn.

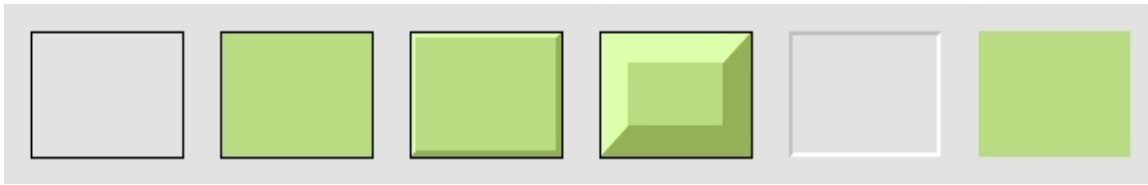
## Basobjekt

- Rektangel
- Rektangel med rundade hörn
- Ellips
- Linje
- Polylinje
- Text
- Annotation
- Kopplingspunkt

## Sammansatta objekt

- Stapel
- Stapeldiagram
- Cirkeldiagram
- Trendkurva
- Fastkurva
- DsTrend
- DsTrendCurve
- Axis
- AxisArc
- Window
- TabbedWindow
- Table
- XYCurve

## 7.1 Rektangel



En rektangel skapas genom att välja rektangel i verktygs-panelen och dra med MB1 i arbetsarean. Om man först aktiverar Functions/ScaleEqual ritas en kvadrat.

### Egenskaper

- Fyllnad
- Fyllnadsfärg
- Kant
- Kantfärg
- Kantbredd
- 3D
- Gradient
- Transparens
- Bakgrundsfärg

### Attribut

#### Attribut

shadow\_width  
shadow\_contrast  
gradient\_contrast  
gradient  
transparency  
invisible

fill\_eq\_background

fixcolor

relief  
disable\_shadow

disable\_gradient

bgcolor\_gradient  
fixposition  
Dynamic

#### Beskrivning

Bredden av 3D-skuggan i % av längd eller bredd (den minsta).

Kontrast på 3D-skuggan. Ett värde mellan 1 - 3.

Kontrast på gradienten. Ett värde mellan 0 - 10.

Typ av gradient.

Grad av transparens. Ett värde mellan 0.0 - 1.0.

Objektet är osynligt men känsligt för musklick. Används för subgrafer för att den känsliga arean för subgrafen.

Markerar att fyllnadsfärgen ritas med angiven bakgrundsfärg för objektet istället för aktuell fyllnadsfärg.

Färgen är fix. Används när objektet är den del av en subgraph, och färgen inte ska påverkas av färgsättningen av subgrafen.

Up ger skugga på undersidan och Down ger skugga på ovansidan

Eliminerar 3D-funktionen. Används när objektet ingår i en subgraf och inte ska ritas med 3D när 3D är valt för subgrafs-instansen.

Eliminerar gradient-funktionen. Används när objektet ingår i en subgraf och inte ska ritas med gradient när gradient är valt för subgrafs-instansen.

Färgen i gradienten går mellan bakgrundsfärgen och fyllnadsfärgen.

Objektet kan inte flyttas.

Ej implementerat

## 7.2 RundadRektangel



En rektangel med rundade hörn skapas genom att välja rundad rektangel i verktygs-panelen och dra med MB1 i arbetsarean. Om man först aktiverar Functions/ScaleEqual ritas en kvadrat.

### Egenskaper

- Fyllnad
- Fyllnadsfärg
- Kant
- Kantfärg
- Kantbredd
- 3D
- Gradient
- Transparens

### Attribut

#### Attribut

round\_amount  
shadow\_width  
shadow\_contrast  
relief  
gradient\_contrast  
gradient  
transparency  
disable\_shadow  
  
disable\_gradient  
  
fixposition  
Dynamic

#### Beskrivning

Storlek på rundingen i % av längd eller bredd (den minsta).  
Bredden av 3D-skuggan i % av längd eller bredd (den minsta).  
Kontrast på 3D-skuggan.  
Up ger skugga på undersidan och Down ger skugga på ovansidan  
Kontrast på gradienten. Ett värde mellan 0 - 10.  
Typ av gradient.  
Grad av transparens. Ett värde mellan 0.0 - 1.0.  
Eliminerar 3D-funktionen. Används när objektet ingår i en subgraf och inte ska ritas med 3D när 3D är valt för subgrafs-instansen.  
Eliminerar gradient-funktionen. Används när objektet ingår i en subgraf och inte ska ritas med gradient när gradient är valt för subgrafs-instansen.  
Objektet kan inte flyttas.  
Ej implementerat

## 7.3 Ellips



En ellips skapas genom att välja ellips i verktygs-panelen och dra med MB1 i arbetsarean. Mha angle1 och angle2 kan segment eller pajbitar av ellipsen ritas. Om man först aktiverar Functions/ScaleEqual ritas en cirkel.

### Egenskaper

- Fyllnad
- Fyllnadsfärg
- Kant
- Kantfärg
- Kantbredd
- 3D
- Gradient
- Transparens
- Bakgrundsfärg

### Attribut

#### Attribut

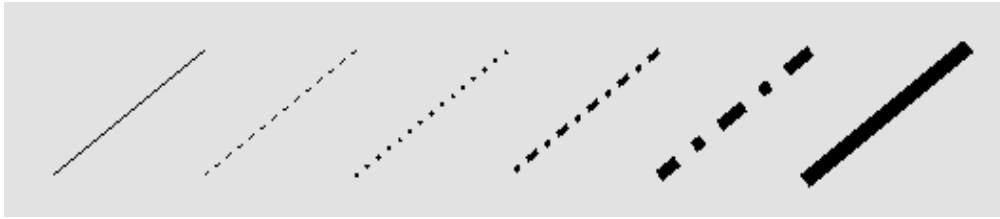
angle1  
angle2  
shadow\_width  
shadow\_contrast  
gradient\_contrast  
gradient  
transparency  
relief  
fixcolor  
  
disable\_shadow  
  
disable\_gradient  
  
fixposition  
fill\_eq\_background  
  
Dynamic

#### Beskrivning

Vinkel från x-axeln i grader till början av segmentet.  
Vinkel i grader som anger storleken på segmentet.  
Bredden av 3D-skuggan i % av längd eller bredd (den minsta).  
Kontrast på 3D-skuggan.  
Kontrast på gradienten. Ett värde mellan 0 - 10.  
Typ av gradient.  
Grad av transparens. Ett värde mellan 0.0 - 1.0.  
Up ger skugga på undersidan och Down ger skugga på ovansidan  
Färgen är fix. Används när objektet är den del av en subgraph, och färgen inte ska påverkas av färgsättningen av subgrafen.  
Eliminerar 3D-funktionen. Används när objektet ingår i en subgraf och inte ska ritas med 3D när 3D är valt för subgrafs-instansen.  
Eliminerar gradient-funktionen. Används när objektet ingår i en subgraf och inte ska ritas med gradient när gradient är valt för subgrafs-instansen.  
Objektet kan inte flyttas.  
Markerar att fyllnadsfärgen ritas med angiven bakgrundsfärg för objektet istället för aktuell fyllnadsfärg.  
Ej implementerat



## 7.4 Linje



En linje skapas genom att välja linje i verktygs-panelen och dra med MB1 i arbetsarean. Genom att aktivera Functions/MoveRestrictions/Horizontal eller Vertical ritas man horisontella resp vertikala linjer.

### Egenskaper

- Kant
- Kantfärg
- Kantbredd
- Linjetyp
- Transparens

### Attribut

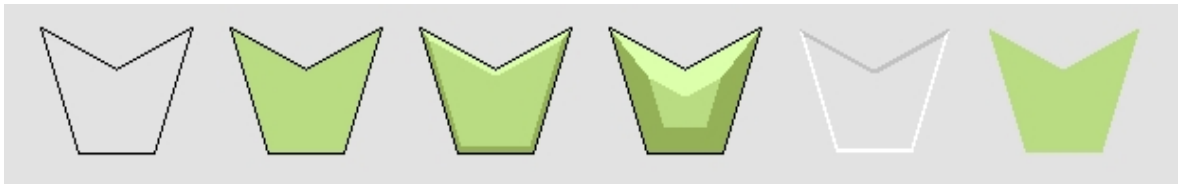
#### Attribut

transparency  
Dynamic

#### Beskrivning

Grad av transparens. Ett värde mellan 0.0 - 1.0.  
Ej implementerat

## 7.5 Polylinje



En polylinje skapas genom att välja polylinje i verktygs-panelen och dra med MB1 i arbetsarean. När MB1 släpps skapas första del-linjen. Genom att på nytt dra med MB1 skapas flera del-linjer. När alla linjer är ritade avslutas med MB3.

Om man vill flytta på enskilda bryt-punkter i en polylinje, markerar man polylinjen och väljer Edit/Edit polyline i menyn. Editeringen avslutas med MB3.

Genom att aktivera Functions/MoveRestrictions/Horizontal eller Vertical ritas man horisontella resp vertikala linjer. Horizontal eller Vertical anger riktningen på den första del-linjen. Övriga ritas vinkelräta mot närmast föregående del-linje. MoveRestrictions kan avbrytas i en polyline med Functions/MoveRestrictions/No.

### Egenskaper

- Fyllnad
- Fyllnadsfärg
- Kant
- Kantfärg
- Kantbredd
- 3D
- Gradient
- Transparens
- Bakgrundsfärg

### Attribut

#### Attribut

shadow\_width  
shadow\_contrast  
relief  
gradient\_contrast  
gradient  
transparency  
disable\_shadow

disable\_gradient

fill\_eq\_border

fill\_eq\_background

fill\_eq\_light

#### Beskrivning

Bredden av 3D-skuggan i % av längd eller bredd (den minsta).

Kontrast på 3D-skuggan.

Up ger skugga på undersidan och Down ger skugga på ovensidan

Kontrast på gradienten. Ett värde mellan 0 - 10.

Typ av gradient.

Grad av transparens. Ett värde mellan 0.0 - 1.0.

Eliminerar 3D-funktionen. Används när objektet ingår i en subgraf och inte ska ritas med 3D, när 3D är valt för subgrafs-instansen.

Eliminerar gradient-funktionen. Används när objektet ingår i en subgraf och inte ska ritas med gradient när gradient är valt för subgrafs-instansen.

Markerar att fyllnadsfärgen ritas med angiven kantfärg istället för aktuell fyllnadsfärg. Detta kan användas i submodeller där vissa markeringar, pilar o dyl, mer hör ihop med kant-färgen än fyllnadsfärgen.

Markerar att fyllnadsfärgen ritas med angiven bakgrundsfärg för objektet istället för aktuell fyllnadsfärg.

Om objektet ingår i en subgraf, ges det samma färg som en 3D-dager,

	när 3D är valt får subgrafs-instansen.
fill_eq_shadow	Om objektet ingår i en subgraf, ges det samma färg som en 3D-skugga, när 3D är valt för subgrafs-instansen.
fill_eq_bglight	Om objektet ingår i en subgraf, ritas det med dager-tonen för bakgrundsfärgen, när 3D är valt för subgrafs-instansen.
fill_eq_bgshadow	Om objektet ingår i en subgraf, ritas det med skuggtonen för bakgrundsfärgen, när 3D är valt för subgrafs-instansen.
fixcolor	Färgen är fix. Används när objektet är den del av en subgraph, och färgen inte ska påverkas av färgsättningen av subgrafen.
fixposition	Objektet kan inte flyttas.
Dynamic	Ej implementerat

## 7.6 Text



En text skapas genom att välja text i verktygs-panelen och klicka med MB1 i arbetsarean. Då öppnar sig ett inmatningsfält och texten kan matas in. Texten kan ändras med Edit/Change text i menyn eller från objekts-editorn.

För närvarande finns endast fonterna Helvetica, normal och fet i svart. Max antal tecken i textsträngen är 79.

### Egenskaper

- Fontstorlek
- Normal, fet text
- Text färg
- Justering
- Transparens

### Attribut

Attribut	Beskrivning
Text	Aktuell text, max 79 tecken.
Adjustment	Justering av texten, vänster, höger eller mitt.
Transparency	Grad av transparens. Ett värde mellan 0.0 - 1.0.
Dynamic	Ej implementerat

## 7.7 Annotation

Annotation är en text i en subgraf som är unik för varje instans av subgraf. Det kan t ex vara texten på en tryck-knapp, eller visning av ett analogt värde. Vid editering av subgraf. markerar man platsen för texten. Själva texten matas in i objektseditorn för instansen av subgraf (eller hanteras av vissa dynamiska funktioner).

En annotation skapas genom att välja annotation (An) i verktygs-panelen och klicka MB1 i arbetsarean.

Annotation kan endast skapas i en subgraf. Det finns plats för 10 annotations i en subgraf. Varje annotation ska ges ett nummer som är unikt inom subgraf.

Observera att vissa typer av dynamik (Value och SetDig etc) kräver speciella nummer för annotation.

### Egenskaper

- Normal, fet
- Textfärg
- Justering

### Attribut

Attribut	Beskrivning
TextSize	Textstorlek 0-7
Number	Numrering av annotations inom en subgraf.
Adjustment	Justering av texten, vänster, höger eller mitt.
Font	Typsnitt.
Type	Enradig eller flerradig.
Protected	Texten visas ej utan markeras med asterisk tecken. Används för inmatning av passerord.

## 7.8 Kopplingspunkt

En kopplingspunkt i en subgraf gör det möjligt att dra kopplingar mellan instanser av subgrafen. I subgrafen anges kopplingspunktens läge, samt den riktning från kopplingspunkten en koppling ska ritas.

En kopplingspunkt skapas genom att välja kopplingspunkt i verktygs-panelen och klicka med MB1 i arbetsarean.

Kopplingspunkter kan endast skapas i subgrafer.

Kopplingspunkten markeras med en cirkel vid editeringen av subgrafen men kommer ej att synas i instanserna av subgrafen.

Om kopplingpunkten ska kopplas till kopplingar av grafcet eller routed typ, måste riktning anges. Denna kan ställas in i meny eller anges i objektseditorn.

Varje kopplingspunkt har ett nummer som måste vara unikt inom subgrafen.

### Egenskaper

- Riktning

### Attribut

Attribut	Beskrivning
Number	Numrering av kopplingspunkter inom en subgraf
Direction	Riktning: upp, ner vänster eller höger.
Dynamic	Ej implementerat

## **7.9 Sammansatta objekt**

## 7.9.1 Stapel



Stapel är ett sätt av visa värdet på en analog signal.

Fyllnad och fyllnadsfärg, kantfärg mm anger egenskaper för stapelns bakgrundsrektangel.

Egenskaper för själva stapeln anges i objektseditorn.

En stapel skapas genom att välja stapel i subgrafs-paletten, 'Analog-Bar', och klicka med MB2 i arbetsarean.

### Egenskaper

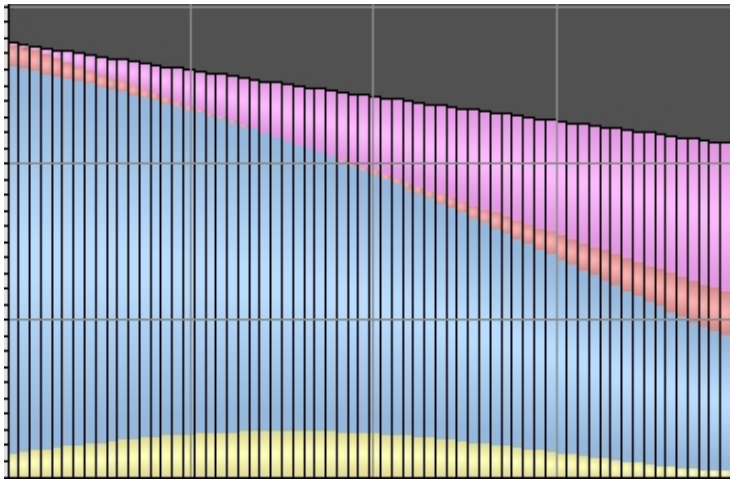
- Fyllnad (för bakgrundsrektangel)
- Fyllnadsfärg (för bakgrundsrektangel)
- Kant (för bakgrundsrektangel)
- Kantfärg (för bakgrundsrektangel)
- Kantbredd (för bakgrundsrektangel).

### Attribut

Attribut	Beskrivning
Bar.Attribute	Namn på signal i rtdb.
Bar.MaxValue	Signalvärde som motsvarar full längd på stapeln.
Bar.MinValue	Signalvärde som motsvarar noll längd på stapeln
Bar.Value	Kan utnyttjas som testvärde i editorn.
Bar.BarColor	Fyllnadsfärg på stapeln (defaultvärdet Inherit innebär att kantfärgen för bakgrundrektangeln väljs)
Bar.BorderColor	Kantfärg på kantlinjen mellan staplen och bakgrundsrektangeln (defaultvärdet Inherit innebär att kantfärgen för bakgrundrektangeln väljs)
Bar.BorderWidth	Kantlinjens linjebredd (1-8)
Dynamic	Ej implementerat



## 7.9.2 Stapeldiagram



Ett stapeldiagram visar ett antal staplar i ett diagram. Varje stapel kan indelas i segment med olika färg. Man kan ha upp till 12 segment. Storleken på varje segment anges av en vektor av float, där elementen i vektorn anger storleken på segmentet i de olika staplarna.

Egenskaperna `fillnad`, `fillnadsfärg`, `kantfärg` mm, anger egenskaper för bakgrunds-rektangeln. Egenskaperna för staplar och segment anges i objekts-editorn.

Ett stapeldiagram skapas genom att välja `BarChart` i subgrafs-paletten, '`Analog/BarChart`', och klicka med MB2 i arbetsarean.

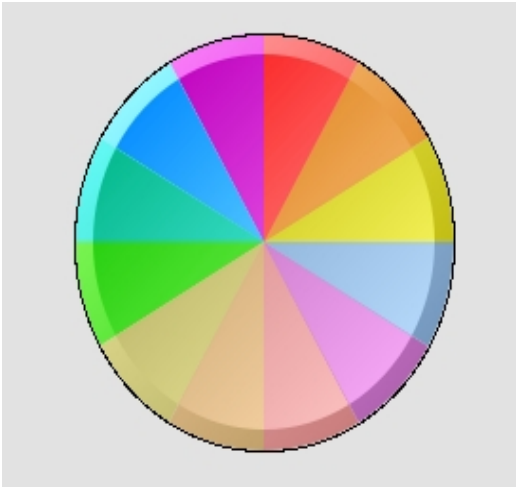
### Egenskaper

- `Fill` (för bakgrundsrektangel)
- `FillColor` (för bakgrundsrektangel)
- `Border` (för bakgrundsrektangel)
- `BorderColor` (för bakgrundsrektangel)
- `Linewidth` (för bakgrundsrektangel)

### Attribut

Attribut	Beskrivning
<code>BarChart.Attribute0</code>	Namnet på en array i databasen. Värdena i arrayen anger storleken på det första segmentet i de olika staplarna.
<code>BarChart.Attribute1</code>	Array för andra segmentet.
...	
<code>BarChart.Attribute11</code>	Array för det tolfte segmentet..
<code>BarChart.Bars</code>	Antal staplar.
<code>BarChart.BarSegments</code>	Antal segment i varje stapel.
<code>BarChart.MaxValue</code>	Värde som motsvarar full längd på en stapel.
<code>BarChart.MinValue</code>	Värde som motsvarar noll längd på en stapel.
<code>BarChart.BarColor1</code>	Fyllnadsfärg för det första segmentet.
<code>BarChart.BarColor2</code>	Fyllnadsfärg för det andra segmentet.
...	
<code>BarChart.BarColor12</code>	Fyllnadsfärg för det tolfte segmentet.
<code>Dynamic</code>	Ej implementerat.

## 7.9.3 Cirkeldiagram



Ett cirkeldiagram med ett antal sektorer i olika färger. Diagrammet kan innehålla upp till 12 sektorer. Storleken på varje sektor anges av ett signalvärde.

Egenskaperna fyllnad, fyllnadsfärg, kantfärg mm, anger egenskaper för bakgrunds-cirkeln.

Egenskaperna för sektorerna anges i objekts-editorn.

Ett cirkeldiagram skapas genom att välja Pie i subgrafs-paletten, 'Analog/Pie', och klicka med MB2 i arbetsarean.

### Egenskaper

- Fill (för bakgrundscirkeln)
- Fillcolor (för bakgrundscirkeln)
- Border (för bakgrundscirkeln)
- Bordercolor (för bakgrundscirkeln)
- Linewidth (för bakgrundscirkeln)

### Attribut

Attribut

Pie.Attribute0

Pie.Attribute1

...

Pie.Attribute11

Pie.Angle1

Pie.Angle2

Pie.Sectors

Pie.MaxValue

Pie.MinValue

Pie.SectorColor1

Pie.SectorColor2

...

Pie.SectorColor12

Dynamic

Beskrivning

Databas attribut av typen Float som anger storleken på den första

Databas attribut för den andra sektorn.

Databas attribut för den tolfte sektorn.

Startvinkel i grader från x-axeln.

Vinkel i grader som anger diagrammets utbredning.

Antal sektorer.

Maxvärde för diagrammets signalvärdesområde.

Minvärde för diagrammets signalvärdesområde.

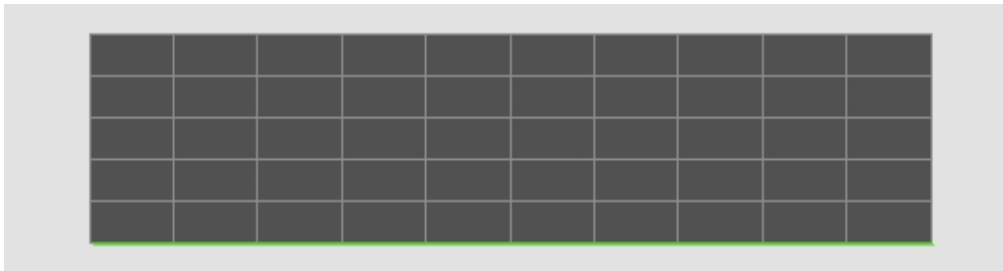
Fyllnadsfärg för den första sektorn.

Fyllnadsfärg för den andra sektorn.

Fyllnadsfärg för den tolfte sektorn.

Ej implementerad.

## 7.9.4 Trendkurva



Trendkurva är ett sätt av visa värdet på en eller ett par analoga signaler.

Kurvorna kan rita med eller utan fyllnad, där fyllnad innebär att arean mellan kurvan och tidsaxeln fylls. Kurvorna ritas med andra rtdb-parameter överst. Om kurvan för denna helt täcker den andra kurva markeras värdet på rtdb-parameter nr 1 med kantlinjen.

I Trendkurvan kan ritas ett antal horisontella och vertikala linjer med kantfärgen för bakgrundsrektangeln. Antalet linjer konfigureras i objektseditorn.

Fyllnad och fyllnadsfärg, kantfärg mm anger egenskaper för kurvans bakgrundsrektangel. Egenskaper för själva kurvan anges i objektseditorn.

En trendkurva skapas genom att välja trend i subgrafs-paletten, 'Analog-Trend', och klicka med MB2 i arbetsarean.

### Egenskaper

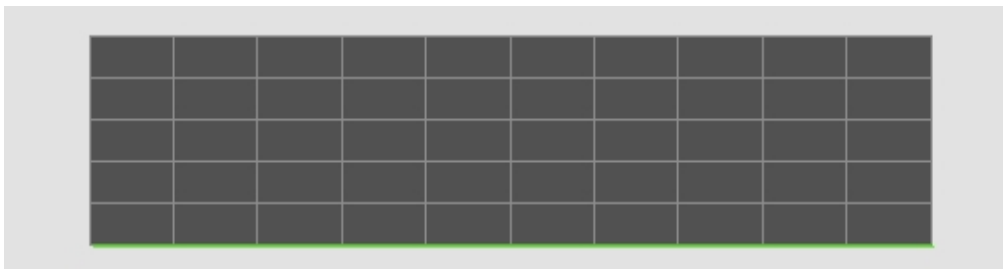
- Fyllnad (för bakgrundsrektangel)
- Fyllnadsfärg (för bakgrundsrektangel)
- Kant (för bakgrundsrektangel)
- Kantfärg (för bakgrundsrektangel och vertikala och horisontella linjer)
- Kantbredd (för bakgrundsrektangel)

### Attribut

Attribut	Beskrivning
Trend.Attribute1	Signal för kurva nr 1.
Trend.Attribute2	Signal för kurva nr 2.
Trend.MinValueAttr1	Signal för minvärde kurva nr 1. Används när minvärdet är dynamiskt. För statisk värde används Trend.MinValue1.
Trend.MaxValueAttr1	Signal för maxvärde kurva nr 1. Används när maxvärdet är dynamiskt. För statisk värde används Trend.MaxValue1.
Trend.MinValueAttr2	Signal för minvärde kurva nr 2. Används när minvärdet är dynamiskt. För statisk värde används Trend.MinValue2.
Trend.MaxValueAttr2	Signal för maxvärde kurva nr 2. Används när maxvärdet är dynamiskt. För statisk värde används Trend.MaxValue2.
Trend.HoldAttr	Signal för att frysa kurvan.
Trend.TimeRangeAttr	Signal av typen Float32 för kurvans tidskala.
Trend.Mark1Attr	Signal för positionen av markeringslinje nr 1.
Trend.Mark2Attr	Signal för positionen av markeringslinje nr 2.
Trend.Mark1Color	Färg på markeringslinje nr 1.

Trend.Mark2Color	Färg på markeringslinje nr 2.
Trend.NoOfPoints	Antal punkter på kurvan
Trend.ScanTime	Tidsintervall mellan två punkter.
Trend.CurveLineWidth	Linjebredd på kurvan (1-8)
Trend.FillCurve	Anger att området mellan kurvan och tidsaxeln ska fyllas
Trend.HorizontalLines	Antal horisontella linjer
Trend.VerticalLines	Antal vertikala linjer
Trend.MaxValue1	Maxvärde för Attribute1
Trend.MinValue1	Minvärde för Attribute1
Trend.CurveColor1	Kantfärg på kurvan för Attribute1
Trend.CurveFillColor1	Fyllnadsfärg på kurvan för Attribute1
Trend.MaxValue2	Maxvärde för Attribute2
Trend.MinValue2	Minvärde för Attribute2
Trend.CurveColor2	Kantfärg på kurvan för Attribute2
Trend.CurveFillColor2	Fyllnadsfärg på kurvan för Attribute2
Trend.Dynamic	Ej implementerat

## 7.9.5 FastCurve



FastCurve visar en eller två kurvor konfigurerade med ett DsFastCurve objekt. En FastCurve är mätning av en signal under en viss tid. Mätningen triggas av någon händelse, och när mätningen är slutförd visas kurvan i FastCurve objektet.

### Attribut

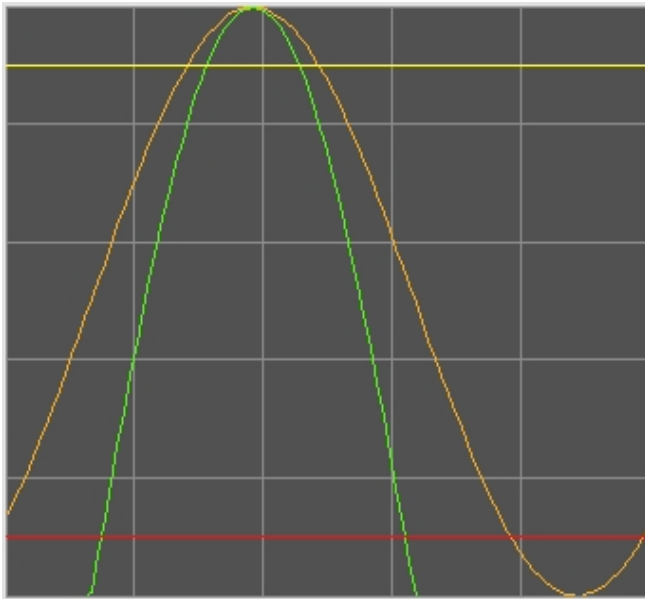
#### Attribut

FastCurve.FastObject  
FastCurve.CurveIndex1  
FastCurve.CurveIndex2  
FastCurve.NoOfPoints  
FastCurve.CurveLineWidth  
FastCurve.FillCurve  
FastCurve.HorizontalLines  
FastCurve.VerticalLines  
FastCurve.MaxValue1  
FastCurve.MinValue1  
FastCurve.CurveColor1  
FastCurve.CurveFillColor1  
FastCurve.MaxValue2  
FastCurve.MinValue2  
FastCurve.CurveColor2  
FastCurve.CurveFillColor2

#### Beskrivning

Databas-objekt av typen DsFastCurve.  
Index i DsFastCurve objektet för den första kurvan.  
Index i DsFastCurve objektet för den andra kurvan.  
Antal punkter i kurvan.  
Linjebredd.  
Areal mellan kurvan och x-axeln fylls.  
Antal horisonella linjer.  
Antal vertikala linjer.  
Maxvärde för signalområdet för kurva 1.  
Minvärde för signalområdet för kurva 1.  
Färg på kurva 1.  
Fyllnadsfärg på kurva 1.  
Maxvärde för signalområdet för kurva 2.  
Minvärde för signalområdet för kurva 2.  
Färg på kurva 2.  
Fyllnadsfärg på kurva 2.

## 7.9.6 DsTrend



DsTrend visar en eller två kurvor konfigurerade med DsTrend-objekt.

### Attribut

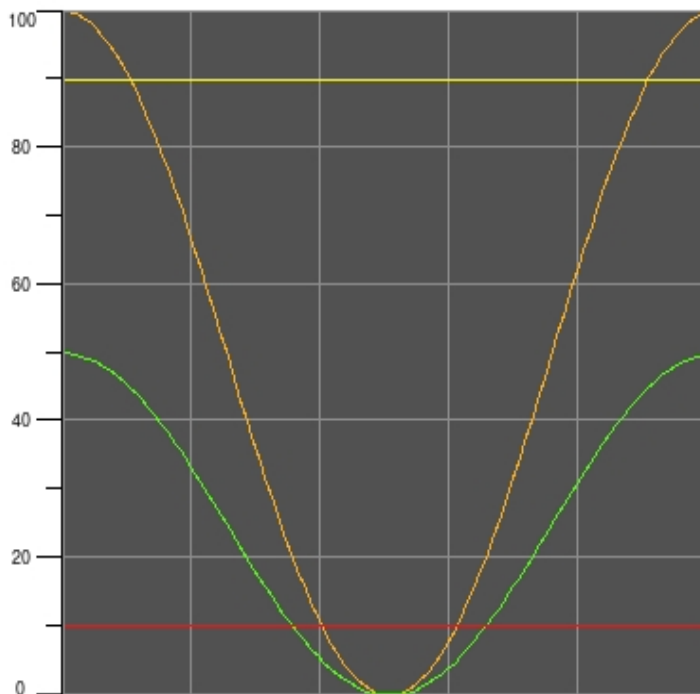
#### Attribut

DsTrend.Object1  
DsTrend.Object2  
DsTrend.MinValueAttr1  
DsTrend.MaxValueAttr1  
DsTrend.MinValueAttr2  
DsTrend.MaxValueAttr2  
DsTrend.HoldAttr  
DsTrend.Mark1Attr  
DsTrend.Mark2Attr  
DsTrend.Mark1Color  
DsTrend.Mark2Color  
DsTrend.NoOfPoints  
DsTrend.CurveLineWidth  
DsTrend.FillCurve  
DsTrend.HorizontalLines  
DsTrend.VerticalLines  
DsTrend.MaxValue1  
DsTrend.MinValue1  
DsTrend.CurveColor1  
DsTrend.CurveFillColor1  
DsTrend.MaxValue2  
DsTrend.MinValue2  
DsTrend.CurveColor2  
DsTrend.CurveFillColor2  
DsTrend.Direction

#### Beskrivning

Databas-objekt av class DsTrend för kurva 1.  
Databas-objekt av class DsTrend för kurva 2. Optional.  
Attribut för minvärde för kurva 1. Optional.  
Attribut för maxvärde för kurva 1. Optional.  
Attribut för minvärde för kurva 2. Optional.  
Attribut för maxvärde för kurva 2. Optional.  
Attribut för att stoppa uppdateringen av kurvan. Optional.  
Attribut för markör 1. Optional.  
Attribut för markör 2. Optional.  
Färg på markör 1.  
Färg på markör 2.  
Antal punkter i kurvan.  
Linjebredd.  
Fyll ytan mellan kurvan och x-axeln.  
Antal horisontella linjer i diagrammet.  
Antal vertikala linjer.  
Maxvärde för området för kurva 1.  
Minvärde för området för kurva 1.  
Färg på kurva 1.  
Fyllnadsfärg för kurva 1.  
Maxvärde för området för kurva 2.  
Minvärde för området för kurva 2.  
Färg på kurva 2.  
Fyllnadsfärg för kurva 2.  
Riktning för kurvans rörelse, höger eller vänster.

## 7.9.7 DsTrendCurve



DsTrendCurve visar en eller två kurvor konfigurerade med ett DsTrendCurve-objekt.

### Attribut

#### Attribut

DsTrendCurve.Object  
DsTrendCurve.MinValueAttr1  
DsTrendCurve.MaxValueAttr1  
DsTrendCurve.MinValueAttr2  
DsTrendCurve.MaxValueAttr2  
DsTrendCurve.HoldAttr  
DsTrendCurve.Mark1Attr  
DsTrendCurve.Mark2Attr  
DsTrendCurve.Mark1Color  
DsTrendCurve.Mark2Color  
DsTrendCurve.NoOfPoints  
DsTrendCurve.CurveLineWidth  
DsTrendCurve.FillCurve  
DsTrendCurve.HorizontalLines  
DsTrendCurve.VerticalLines  
DsTrendCurve.MaxValue1  
DsTrendCurve.MinValue1  
DsTrendCurve.CurveColor1  
DsTrendCurve.CurveFillColor1  
DsTrendCurve.MaxValue2  
DsTrendCurve.MinValue2  
DsTrendCurve.CurveColor2

#### Beskrivning

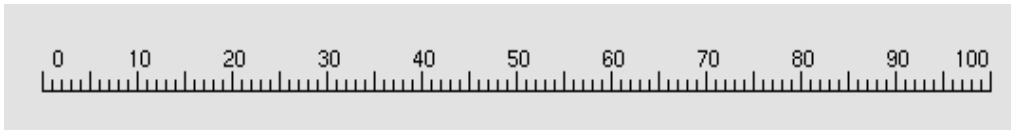
Databas-objekt av klass DsTrendCurve.  
Attribute för minvärde för kurva 1. Optional.  
Attribute for maxvärde för kurva 1. Optional.  
Attribute for minvärde för kurva 2. Optional.  
Attribute for maxvärde för kurva 2. Optional.  
Attribut för att stoppa uppdateringen av kurvan. Optional.  
Attribut för markör 1. Optional.  
Attribut för markör 2. Optional.  
Färg på markör 1.  
Färg på markör 2.  
Antal punkter i kurvan.  
Linjebredd.  
Fyll området mellan kurvan och x-axeln.  
Antal horisonetlla linjer i diagrammet.  
Antal vertikala linjer.  
Maxvärde för området för kurva 1.  
Minvärde för området för kurva 1.  
Färg på kurva 1.  
Fyllnadsfärg på kurva 1.  
Maxvärde för området för kurva 2.  
Minvärde för området för kurva 2.  
Färg på kurva 2.

DsTrendCurve.CurveFillColor2  
DsTrendCurve.Direction

Fyllnadsfärg för kurva 2.  
Rikting för kurvans rörelse, höger eller vänster.



## 7.9.8 Axis



Axis visar skalområdet på kurva eller stapel i x- eller y-led.  
Axis finns i två varianter, en med fasta min och max värden, och en med dynamiska min och max värden, där min och max kopplas till signaler i databasen. För ett dynamiskt Axis objekt beräknas värden på Lines, LongQuotient, ValueQuotient och Format automatiskt i runtime.

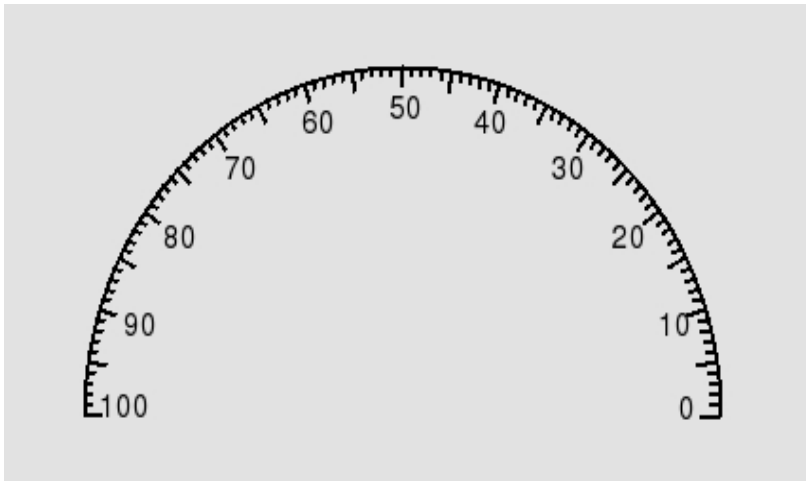
### Attribut

Attribut	Beskrivning
MaxValue	Övre värde på skalområdet.
MinValue	Undre värde på skalområdet.
Lines	Antal linjer vinkelräta mot axeln.
LongQuotient	Hur stor del av linjerna som ska vara långa. T ex 4 innebär att var 4:e linje är längre
ValueQuotient	Hur ofta värden ska skrivas ut. T ex 4 innebär att ett värde skrivs ut vid var 4:e linje.
Format	Format i c-syntax för utskrivna värden.
Dynamic	Ej implementerat.

### Dynamisk Axis

Axis.MinValueAttr	Signal för minvärde av typ Float32 eller Int32.
Axis.MaxValueAttr	Signal för maxvärde av typ Float32 eller Int32.
Axis.KeepSettings	Om 0 beräknas nya värden på Lines, LongQuotient, ValueQuotient och Format när min eller maxvärdet har ändrats dynamiskt. Om KeepSettings är 1 behålls de ursprungliga värdena.

## 7.9.9 AxisArc



AxisArc är en cirkelformat skala som visar skalområdet för en visare. Som för Axis finns det två varianter av AxisArc, en med fasta min och max värden, och en med dynamiska min och max värden, där min och max kopplas till signaler i databasen. För ett dynamiskt AxisArc objekt beräknas värden på Lines, LongQuotient, ValueQuotient och Format automatiskt i runtime.

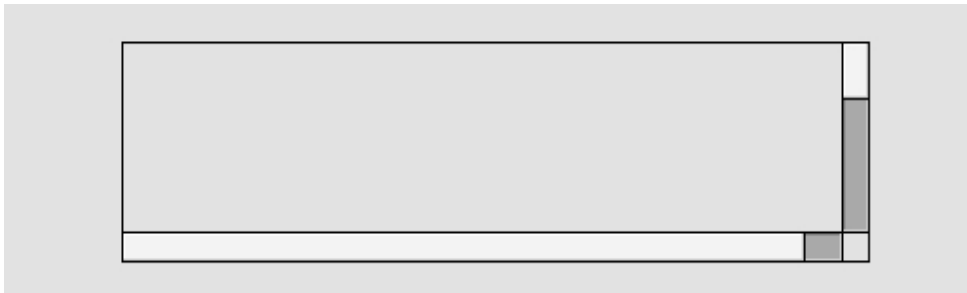
### Attribut

Attribut	Beskrivning
Angle1	Startvinkel i grader från x-axeln.
Angle2	Vinkel i grader som anger skalans utbredning.
MaxValue	Övre värde på skalområdet.
MinValue	Undre värde på skalområdet.
Lines	Antal linjer vinkelräta mot axeln.
LongQuotient	Hur stor del av linjerna som ska vara långa. T ex 4 innebär att var 4:e linje är längre
ValueQuotient	Hur ofta värden ska skrivas ut. T ex 4 innebär att ett värde skrivs ut vid var 4:e linje.
Format	Format i c-syntax för utskrivna värden.
Dynamic	Ej implementerat.

### Dynamisk AxisArc

Axis.MinValueAttr	Signal för minvärde av typ Float32 eller Int32.
Axis.MaxValueAttr	Signal för maxvärde av typ Float32 eller Int32.
Axis.KeepSettings	Om 0 beräknas nya värden på Lines, LongQuotient, ValueQuotient och Format när min eller maxvärdet har ändrats dynamiskt. Om KeepSettings är 1 behålls de ursprungliga värdena.

## 7.9.10 Window



Window visar en separat graf inom ett specificerat område i en annan graf. Grafen kan visas med eller utan skroll-lister.

### Begränsningar

Kopplingar i fönstret skalas inte på rätt sätt.

### Attribut

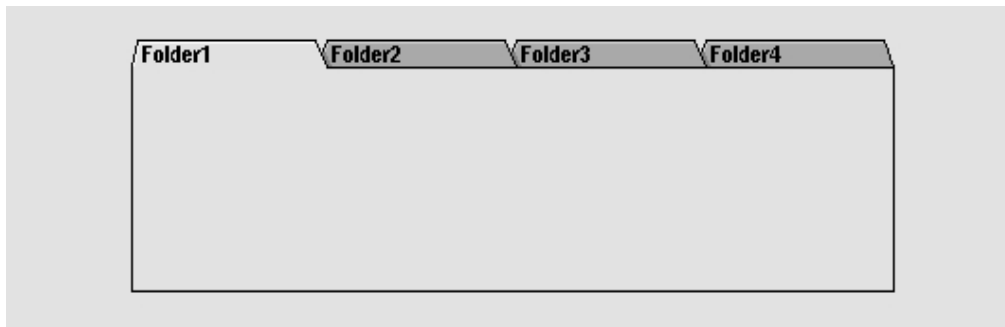
#### Attribut

Window.FileName  
Window.Scale  
Window.VerticalScrollbar  
Window.HorizontalScrollbar  
Window.ScrollbarWidth  
Window.ScrollbarColor  
Window.ScrollbarBgColor

#### Beskrivning

Namn på den graf som ska visas i fönstret.  
Skalning av grafen.  
I fönstret visas en vertical skroll-list.  
I fönstret visas en horisontell skroll-list.  
Bredd på skroll-list.  
Fyllnadsfärg på skroll-listens rörliga del.  
Fyllnadsfärg på skroll-listens fasta del.

## 7.9.11 TabbedWindow



TabbedWindow är ett Window objekt med flik-register. För varje flik anges en separat graf, och när fliken aktiveras visas en angivna grafen i fönstret.

### Begränsningar

Kopplingar i fönstret skalas inte på rätt sätt.

### Attribut

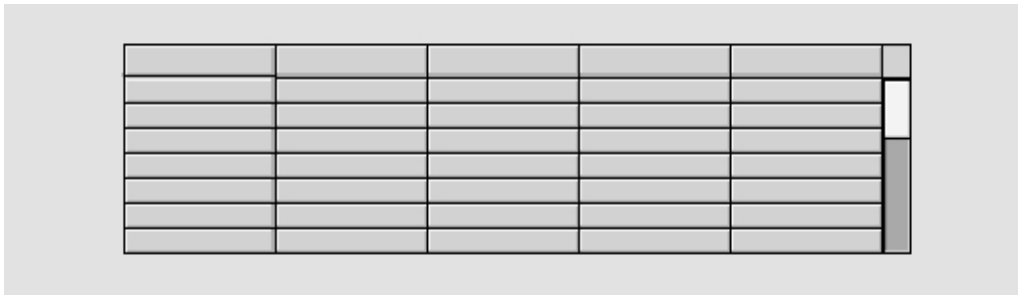
#### Attribut

Folder.NumberOfFolders  
Folder.HeaderHeight  
Folder.ColorSelected  
Folder.ColorUnselected  
Folder.ScrollbarWidth  
Folder.ScrollbarColor  
Folder.ScrollbarBgColor  
Folder1.FileName  
  
Folder1.Text  
Folder1.Scale  
Folder1.VerticalScrollbar  
Folder1.HorizontalScrollbar

#### Beskrivning

Antalet flikar.  
Höjd på flikarna.  
Färg på den flik som visas för tillfället.  
Färg på flik som ej visas för tillfället.  
Bredd på skroll-list.  
Fyllnadsfärg på skroll-listens rörliga del.  
Fyllnadsfärg på skroll-listens fasta del.  
Namn på den graf som ska visas i den första fliken.  
Text på den första fliken  
Skalning på grafen i flik nr 1.  
För flik nr 1 visas en vertical skroll-list.  
För flik nr 1 visas en horisontell skroll-list.

## 7.9.12 Table



Ett table-objekt visar en tabell med ett visst antal rader och kolumner. Varje kolumn kopplas till ett vektor-attribut i databasen, och värdet för elementen i vektorn visas i kolumnen.

En cell i tabellen kan väljas ut genom att man klickar på den eller med pil-tangenterna (om aktions-typen InputFocus är konfigurerad för tabellen). Till varje kolumn kan man koppla ett urvals-attribut, i form av ett vektor-attribut i databasen av typen Boolean. När en cell väljs ut, sätts motsvarande element i urvals-attributet, och det tidigare utvalda återställs.

Om det kopplade vektor-attributet till en kolumn är av typen Objid, kan man öppna en popup-meny med metoderna för objekten som visas i kolumnen.

Vektor-attributen ska anges med datatyp och storlek, t ex VWX-P1-Table.TabVect##Float32#100, där Float32 är datatypen och 100 antalet element i vektorn.

Tabell-objektet består av följande komponenter:

- Vertikal och horisontal skroll-list.
- En header rad med rubrik för varje kolumn. Header raden omfattas ej av den vertikala skroll-listen.
- En header kolumn. Kolumnen längst till vänster kan konfigureras som en header kolumn, vilket innebär att den inte omfattas av den horisontella skroll-listen.

### Attribut

#### Attribut

Table.Rows

Table.Columns

Table.HeaderRow

Table.HeaderColumn

Table.RowHeight

Table.HeaderTextSize

Table.HeaderTextBold

Table.HeaderTextColor

Table.Options

Table.VerticalScrollbar

Table.HorizontalScrollbar

#### Beskrivning

Antalet rader i tabellen (header raden ej inräknad).

Antalen kolumner i tabellen (ev header kolumn inräknad).

Header rad med rubrik för varje kolumn ritas.

Den vänstra kolumnen omfattas inte av den horisontella scroll-listen.

Rad höjd.

Text-storlek på texten i header raden.

Texten i header raden ska vara fet.

Färg på texten i header raden.

Bitmask med tillvals-funktioner.

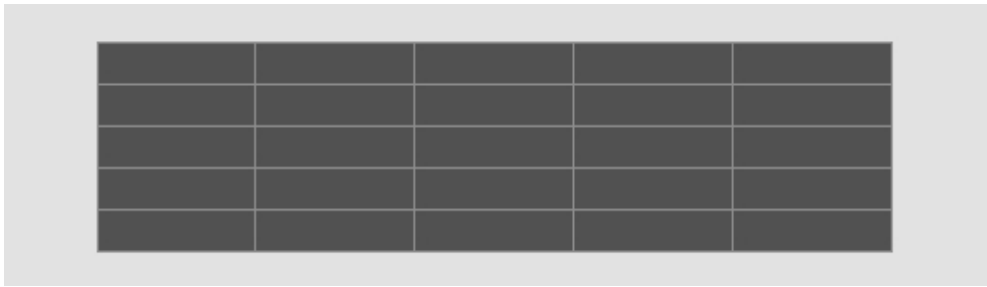
1: Om cellen i kolumnen längst till vänster i en rad är tom, ritas alla celler på raden tomma.

Vertikal skroll-list i tabellen.

Horisontal skroll-list i tabellen.

Table.ScrollbarWidth	Bredd på skroll-list.
Table.ScrollbarColor	Fyllnadsfärg på skroll-listens rörliga del.
Table.ScrollbarBgColor	Fyllnadsfärg på skroll-listens fasta del.
Column1.Attribute	Vektor-attribut. Värdet på varje element visas på motsvarande rad i kolumnen. Attributet ska anges med datatyp och storlek, se ovan.
Column1.Format	Format i c syntax.
Column1.SelectAttribute	Vektor-attribut av typen Boolean. När en cell i kolumnen väljs ut, sätt motsvarande element i vektorn. Attributet ska anges med datatyp och storlek, se ovan.
Column1.Width	Bredd på kolumnen.
Column1.HeaderText	Rubrik på kolumnen. Rubriken skrivs i header-raden.

## 7.9.13 XYCurve



XYCurve visar kurvor definierade med x och y koordinater. 10 kurvor kan visas i ett XYCurve objekt. För varje kurva kopplas objektet till två vektorer som innehåller x och y koordinaterna för kurvan. Ett digital signal kopplas till UpdateAttr som indikerar när koordinaterna har uppdaterats och kurvan ska ritas om. Skalområdena i x och y riktning kan anges med fixa värden, eller kopplas till attribut i databasen. Antal punkter i kurvan kan också definieras statistiskt eller med en koppling till ett databasattribut.

XYCurve kan kopplas till godtyckliga attribut i databasen, men det finns en speciell klass, XyCurve, som kan användas för att lagra en kurva med 100 punkter. Den innehåller två vektorer med 100 element för koordinaterna, samt attributa för skalområden, antal punkter och uppdatering.

### Attribut

#### Attribut

XY\_Curve.XAttr  
XY\_Curve.YAttr  
XY\_Curve.UpdateAttr  
  
XY\_Curve.XMinValue  
XY\_Curve.XMaxValue  
XY\_Curve.YMinValue  
XY\_Curve.YMaxValue  
XY\_Curve.XMinValueAttr  
  
XY\_Curve.XMaxValueAttr  
  
XY\_Curve.YMinValueAttr  
  
XY\_Curve.YMaxValueAttr  
  
XY\_Curve.NoOfPointsAttr  
XY\_Curve.CurveColor  
XY\_Curve.FillColor  
XY\_Curve.Instances  
  
XY\_Curve.CurveLineWidth  
XY\_Curve.FillCurve  
XY\_Curve.HorizontalLines

#### Beskrivning

Vektor i databasen som innehåller x-koordinater för kurvan.  
Vektor i databasen som innehåller y-koordinater för kurvan.  
Databas attribut av typen Boolean som indikerar att kurvan ska uppdateras.  
Minvärde för skalområde i x-riktning.  
Maxvärde för skalområde i x-riktning.  
Minvärde för skalområde i y-riktning.  
Maxvärde för skalområde i y-riktning.  
Databas attribut som innehåller minvärde för skalområde i x-riktning.  
Databas attribut som innehåller maxvärde för skalområde i x-riktningen.  
Databas attribut som innehåller minvärde för skalområde i y-riktning.  
Databas attribut som innehåller maxvärde för skalområde i y-riktning.  
Databas attribut av typ integer med antal punkter i kurvan.  
Kurvans färg.  
Kurvans fyllnadsfärg.  
Genom att addera flera instanser kan man visa flera kurvor i samma diagram. Max 10 kurvor kan visas.  
Kurvans linjebredd.  
Om 1 ritas området mellan kurva och x-axeln med fyllnadsfärg.  
Antal horisontella linjer i diagrammet.

XY_Curve.VerticalLines	Antal vertikala linjer i diagrammet.
XY_Curve.HorizontalPadding	Kurvan utanför x-området komma att ritas med horisontella linjer från första och sista punkt.
XY_Curve.XMark1Attr	Signal för x positionen för den första vertikala markeringslinjen.
XY_Curve.XMark2Attr	Signal för x positionen för den andra vertikala markeringslinjen.
XY_Curve.YMark1Attr	Signal för y positionen för den första horisontella markeringslinjen.
XY_Curve.YMark2Attr	Signal för y positionen för den andra horisontella markeringslinjen.
XY_Curve.Mark1Color	Färg för markeringslinje nr 1 (vertikal och horisontell).
XY_Curve.Mark2Color	Färg för markeringslinje nr 2 (vertikal och horisontell).



# 8 Subgrafer

En subgraf är en klass eller mall som byggs upp av ett antal basobjekt.

En instans av en subgraf skapas genom välja ut subgrafen i subgraf-paletten och klicka med MB2 i arbetsarean. Instanserna hanteras i editorn som andra objekt och man kan sätta fyllnadsfärg, kantfärg och linjebredd på dem.

## Färg

På ett subgraf-objekt kan man sätta fyllnads- och kant-färg på samma sätt som på bas-objekten.

Men det innebär att man ger dessa egenskaper till samtliga delobjekt i subgraf. Sätter man fyllnadsfärg på en subgraf som har ritats med skuggor och dagrar för att ge 3d effekt, försvinner denna effekt och objektet blir helt platt. För att kunna modifiera färgen på den här typen av subgrafer kan man använda funktionerna för ljushet, färgintensitet och färgton i verktygs-panelen.

### Ljushet

Funktionen för ljushet finns i verktygs-panelen och markeras med en sol. Här finns öka och minska-knappar som gör fyllnadsfärgen i objektet ljusare resp mörkare. Ljusheten kan modifieras i sju steg.

### Intensitet

Funktionen för färgintensitet finns i verktygs-panelen och markeras av tre färgpunkter. Intensiteten kan modifieras i tre steg: stark intensitet, medel intensitet och grå.

### Skift

Skiftning av färg innebär att varje färg i subgrafs-objektet roteras på färgcirkeln. Ett objekt som innehåller blå och gröna färger kommer t ex efter fyra stegs skiftning att innehålla orange och rött. Färgerna behåller sin ursprungliga ljushet och intensitet, och även sin inbördes färgkontrast.

Funktionen finns i verktygs-panelen och markeras av tre färgpunkter omgärdade av en cirkulär pil. Färgerna på första raden i färgpaletten påverkas inte av skift-funktionen.

### Ton

Att tona ett objekt innebär att alla färger i subgrafs-objekt får samma färgton. De behåller sin ljushet och intensitet. Det finns nio olika färgtoner: grå, gul, guld, orange, röd, magenta, blå, blågrön och grön.

Funktionen för ton finns längst ner i färgpaletten. Här finns även en tryckknapp för att återställa ett objekt till ursprungsfärgerna.

## Animering och skiftning

### Subgrafer med flera sidor

Animering och Shift är typer av dynamik där man inte bara ändra färgen på subgraf, utan även formen. Detta kan man åstadkomma genom att skapa en subgraf med flera sidor.

För att skapa fler sidor till en subgraf aktiverar man File/Page/Create next page. Denna funktion skapar en ny subgraf med suffixet '\_\_\_px', där x är sidnumret, och länkar denna till föregående sida genom att lägga in namnet i attributet NextSubgraph i graf-attributen. För den sista sidan är NextSubgraph tom. När man har skapat och sparat en sida kan man snabbt skifta till nästa eller föregående sida med funktionerna File/Page/NextPage resp File/Page/PreviousPage.

Man bör spara de olika sidorna med samma zoom-faktor, för att undvika att riskera att få en pixels förskjutning mellan olika sidor i runtime.

## Skift

Skiftning mellan olika sidor i runtime kan ske med dynamik-typerna DigShift, DigFourShift och AnalogShift. DigShift växlar mellan första och andra sidan beroende på ett digitalt värde. DigFourShift växlar mellan fyra sidor, beroende på värdet av tre digitala signaler. AnalogShift kopplas till en analog signal där värdet på den motsvara index för den sidan som ska visas (första sidan har index 0).

## Animering

Vid animering spelas sidorna upp efter varandra och ger intryck av rörelse hos subgrafen.

Tiden mellan varje sidbyte måste oftast vara betydligt kortare än den normala uppdateringen av dynamiska objekt i bilden. För en graf kan man ange animationstiden i File/Graph attributes i menyn. Observera att tiden anges för grafen som helhet och inte kan anges för varje subgraf individuellt. Ett lämpligt värde på animations-tiden är 0.2 sekunder.

Om en animation ska vila på samma sida under flera animations-cykler, kan man lägga in antalet cykler i AnimationCount i Graph attributes för den aktuella sidan.

Animation hanterar av dynamiktypen Animation. Den kopplas till en digital signal, som kontrollerar start och stopp av animationen. Det har också attributet Sequence som talar om i vilken ordning, och villkoret för att animationen ska köras. Sequence kan ha värdet Cyclic, Dig eller ForwBack.

Om Sequence är Cyclic, visas den första sidan när värdet på signalen är 0. När värdet på signalen är 1 är animationen aktiv. Det startar på andra sidan och försätter till sista sidan, varefter den hoppar till andra sidan igen och försätter till sista sidan osv. Detta pågår tills signalvärdet blir 0 igen.

Sequence ForwBack är liknande, men här går animationen kontinuerligt fram och tillbaka mellan första och sista sidan, så länge signalvärdet är 1.

Om Sequence är Dig genomlöps animationen endast vid omslag av det digitala värdet. Vid omslag till 1 animeras från första till sista sidan, och vid omslag till 0 animeras tillbaka till första sidan.

## 8.1 Koppling till databas

Subgrafer kopplas till attribut i databasen vilket gör det möjligt att ändra färg och på objekt när attributvärden ändras. Till exempel har dynamiken DigColor egenskapen Attribute där det attribut i databasen som ska påverka färgen specificeras.

Specifikationen av ett attribut består av attributets namn och typ, till exempel

```
H1-H2-Start.ActualValue##Boolean
```

Vanligtvis läggs attributet in genom att välja ut attributet i navigatören för databasen, och klicka på attribut-fältet i objektseditorn med Ctrl+dubbeltklick. Om det endast finns ett attribut i subgrafen kan man klicka direkt på subgrafen med Ctrl+dubbeltklick.

### **Invertera en signal**

En signal inverteras genom att ett utropstecken placeras framför attributnamnet. Endast attribut av typen Boolean kan inverteras. Invertering kan användas i dynamik men inte i aktioner.

```
!H1-H2-Start.ActualValue##Boolean
```

Utropstecknet i exemplet ovan anger att signalvärdet kommer att inverteras innan det används i dynamiken.

### **Objekt- och hierarkigrafer**

När man ritar objektgrafer anges instans-objektet med strängen \$object. Om till exempel attributet

```
$object.ActualValue##Float32
```

förekommer i en objektsgraf, och objektgrafen öppnas för objektet H1-H2-Temperature, kommer \$object att ersättas av instansobjektet och kopplingen kommer att göras till

```
H1-H2-Temperature.ActualValue##Float32
```

\$object används även i hierarkigrafer på motsvarande sätt.

```
$object-Temperature.ActualValue##Float32
```

med hierarki-objektet H1-H2 kommer att ge

```
H1-H2-Temperature.ActualValue##Float32
```

### **Förälderobjekt**

Förälderobjektet till \$object kan refereras med '<' eller '-<'. Dessa operatorer kommer att ta bort sista attributet eller sista objektet till närmast föregående '.' eller '-'. H1-H2-<.Description kommer att bli H1.Description. Om man har en hierarkigraf för H1-H2 kan beskrivningen för förälderobjektet H1 visas med

```
$object-<.Description##String80
```

För attributobjekt, där attributnamn tas bort istället, används '<.'. H1-Plates.Data.<.Temperature kommer att bli H1-Plates.Temperature. I objektgrafen för Data kan Temperatur-attributet i förälderobjektet Plates visas med

```
$object.<.Temperature##Float32
```

### **Attributreferenser**

Om ett objekt innehåller attribut av typen AttrRef eller Objid kan värden in de objekt som dessa attribut pekar på visas med &() syntax.

```
&(H1-H2-Motor.Object).ActualValue##Float32
```

betyder att H1-H2-Motor.Object innehåller en pekare till ett objekt, och det är till ActualValue attributet i detta objekt kopplingen kommer att göras.

Kopplingen görs när grafen öppnas, och om pekaren in H1-H2-Motor.Object ändras, kommer kopplingen inte automatiskt att ändras. Dynamiken DigSwap kan dock användas för att koppla om alla kopplingar i en graf.

### **Variabelt vektorindex**

Vektorindex kan specificeras med attribut av typen Int32 eller UInt32. I

```
H1-H2-Array[&(amp;(H1-H2-CurrentIndex.ActualValue))].Value##Float32
```

är H1-H2-CurrentIndex.ActualValue ett integer-attribut, och värdet av detta attribut kommer att hämtas och läggas in som vektorindex.

Indexet evalueras när grafen öppnas och om indexvärdet ändras kommer inte kopplingen automatiskt att ändras. Dynamiken DigSwap kan dock användas för att koppla om alla kopplingar i en graf.

### **Nodobjekt**

\$node kommer att bytas ut mot nodobjektet. Detta kan till exempel användas för att referera XttGraph-objekt där samma graf används i olika noder. Till exempel

```
$node-Graphs-Overview
```

kommer att fungera för alla noder som har ett XttGraph objekt med namnet Overview under en Graph hierarki under nodobjektet.

### **Lokal databas**

Lokala variabler i en graf kan användas för att kommunicera mellan objekt i en graf. De specificeras med prefixet \$local. följt av variabelnamn och variabeltyp, t ex

```
$local.HoldTrend##Boolean
```

En lokal variabel kan till exempel användas för att frysa en trendkurva genom att denna anges i Trend.HoldAttr. Med en toggle-knapp som har samma lokala variabel i ToggleDig.Attribute, kan frysningen slås på och av genom att trycka på knappen.

Lokala variabler är dock inte implementerat för alla typer av dynamik och aktioner.

### **Vektorer**

En del dynamiktyper som Table och XYCurve kopplas till hela vektorer i databasen. Syntaxen för att koppla en hel vektor är 'attributename'##'type'##'length', till exempel

```
H1-H2-A1.Value##Float32#100
```

där 100 är antalet element i vektorn.

### **Vektorelement**

För att koppla ett element i en vektor, är syntaxen 'attributename'##'type'##'length'['index'], t ex

```
H1-H2-A1.Value##Float32#100[2]
```

### **Attribut typ**

Typen för att attribut specificeras efter attributnamnet. Följande typer supportas

##Boolean  
##Float32  
##Float64  
##Char  
##Int8  
##Int16  
##Int32  
##Int64  
##UInt8  
##UInt16  
##UInt32  
##UInt64  
##Objid  
##Time  
##DeltaTime  
##Status  
##NetStatus  
##Enum  
##Mask  
##DataRef  
##String  
##Bit

### String

String ska föjas av storleken på strängen, t ex ##String80

### Bit

Bit-typen gör det möjligt att koppla en individuell bit i en mask eller ett heltalsattribut.

Syntaxen är ##Bit#size['bitnumber'] eller ##Bit['bitnumber'].

size är antalet bitar i masken, och bitnumber är bitnumret som räknas från 0 för första biten.

Exempel

```
H1-H2-Mode.ActualValue##Bit[2]
```

## 8.2 Dynamik

Man kan definiera ett antal olika typer av dynamik för subgrafer. Dynamiken beskriver hur signaler i rtdb ska påverka subgrafen, t ex ändra färg eller skriva ut ett analogt värde.

Det finns ett antal fördefinierade typer av dynamik, t ex skifta mellan två färger eller mellan två texter. När man ritar subgrafan anger man typen av dynamik som ska gälla för subgrafan, men detta är endast ett defaultvärde som varje instans får, om inte konstruktören väljer en annan typ av dynamik.

Dynamik	Signal typ	Beskrivning
Inherit	-	Dynamiken för instansen ärvs från subgrafan.
Tone	-	Anger att dynamik som ändrar färg på komponenten ska ändra färgton i stället för fyllnadsfärgen.
DigLowColor	Boolean	Sätter angiven fyllnadsfärg när signalen är låg.

DigColor	Boolean	Sätter angiven fyllnadsfärg när signalen är hög.
AnalogColor	Float32 Int32	Sätter angiven fyllnadsfärg när signalen över- eller underskrider ett angivet värde.
StatusColor	Status	Sätter fyllnadsfärg efter ett status värde.
DigError	Boolean	Sätter röd fyllnadsfärg vid hög signal.
DigWarning	Boolean	Sätter gul fyllnadsfärg vid hög signal.
DigFlash	Boolean	Blinkar med angiven färg vid hög signal.
FillLevel	Float32	Ändrar fyllnadsfärgen på en del av komponenten. Nivån som avgränsar den färgade delen bestäms av värdet på signalen.
Invisible	Boolean String	Gör komponenten osynlig vid hög signal, eller vid tom sträng.
DigBorder	Boolean	Sätter angiven kantfärgen vid låg signal.
DigBackgroundColor	Boolean	Sätter angiven bakgrunds färg när signalen är hög.
TimeoutColor	Boolean	Sätter angiven färg vid timeout på prenumerationen.
DigTextColor	Boolean	Sätter angiven textfärg vid hög signal.
DigText	Boolean	Sätter angiven text vid låg signal.
AnalogText	Float32 Int32	Sätter upp till 32 olika text beroende på värdet av en analog signal.
Value	Valfri	Skriver ut värdet av ett attribut.
Rotate	Float32 Int32	Roterar komponenten.
Move	Float32 Int32	Flyttar och skalar komponenten i x- och y-led.
DigShift	Boolean	Skiftar mellan två sidor i subgrafen (första och sista sidan).
DigFourShift	Boolean	Skiftar mellan fyra sidor i subgrafen.
AnalogShift	Float32 Int32	Skiftar mellan olika sidor i subgrafen. Värdet på signalen bestämmer sidnr.
Animation	Boolean	Olika typer av animering.
Video	-	Kontinuerlig uppdatering av en blidfil.
SliderBackground	-	Markerar att subgrafen är bakgrund till en slider-komponent.
DigCommand	Boolean	Exekverar ett kommando när signalen blir hög.
DigScript	Boolean	Exekverar ett script när signalen blir hög.
ScrollingText	Boolean	Visar en rullande text.
DigSound	Boolean	Spela upp ett ljud.
DigTransparency	Boolean	Skiftar transparens beroende på en digital signal.
AnalogTransparency	Float32	Skiftar transparens beroende på en analog signal.
ColorThemeLightness	-	Anpassa ljusheten hos en komponent till färgtemat.
DigSwap	Boolean	Starta om alla prenumerationer i en graf.
UnitConvert	Float32	Konvertera värdets enhet före visning.
HostObject	Object	Dynamik för komplexa komponenter.

## Inherit

För en instans av en subgraph betyder det att dynamiken ärvs från subgrafs-klassen, dvs den dynamik som angetts som default när subgrafen editrades. Om även default-dynamiken är Inherit innebär det att subgrafen saknar dynamik.

Default-dynamiken för en instans ser man enklast genom att öppna objektseditorn för instansen och där öppna subgraph-mappen.

## Xtt-kommanon

Det finns en rad tryckknappar som exekverar xtt-kommandon. Det vanligaste är att man öppnar en annan graf, men man kan även öppna trace med ett visst objekt utvalt. Här följer några exempel på användbara xtt-kommandon.

```
open graph GraphName [/width=][/height=][/scrollbar][/navigator]
```

```
open trace WindowName [/center=]
```

## Access

All dynamik som gör det möjligt att påverka en signal i rtdb innehåller attributet Access. Access talar om de privilegier som krävs för att en användare ska tillåtas att ändra ett värde. Det finns 15 olika privilegier som är intressanta i operatörsbilder, 10 för olika slags operatörer, och 4 för olika yrkesgrupper. Dessutom finns RtRead, dvs läsrättighet i runtime, som även en användare som inte lyckats logga in får. En användare ges ett eller flera privilegier och om något av hans privilegier finns med i Access har han rätt att påverka objektet.

Default-access är alla privileger utom RtRead. Vissas operationer som att öppna andra bilder bör man även tillåta RtRead att utföra.

Privilegier i runtime	Beskrivning
RtRead	Behörig att läsa.
System	Systemansvarig.
Maintenance	Underhåll.
Process	Process tekniker.
Instrument	Instrumentare.
Operator1	Olika operatörer eller operatörsplatser.
Operator2	"
Operator3	"
Operator4	"
Operator5	"
Operator6	"
Operator7	"
Operator8	"
Operator9	"
Operator10	"

## 8.2.1 DigLowColor

Sätt angiven fyllnadsfärg när signalen är låg.

Objektet kopplas till en digital signal i rtdb. Om värdet är 1 ritas objektet med den normala fyllnadsfärgen, om värdet är 0 ritas det med DigLowColor.Color som anges i objektseditorn.

Jämför DigColor nedan som sätter angiven färg vid hög signal istället för låg. Om man ska använda DigLowColor eller DigColor avgörs närmast av hur man vill att komponenten ska ritas i editorn: så som den kommer att se ut vid hög eller vid låg signal, dvs om man vill rita bilden så det kommer att se ut när anläggningen är i drift, eller avstängd.

### Attribut

DigLowColor.Attribute

DigLowColor.Color

### Beskrivning

Signal i rtdb av typen Boolean som ska påverka komponenten.

Fyllnadsfärg på komponenten om signalen är 0.



## 8.2.2 DigColor

Sätt angiven fyllnadsfärg när signalen är hög.

Objektet kopplas till en digital signal i rtdb. Om värdet är 1 ritas objektet med DigColor.Color som anges i objektseditorn, om värdet är 0 ritas det med den normala fyllnadsfärgen.

DigColor kan finnas i flera instanser, vilket gör det möjligt att växla mellan flera färger (upp till 32 stycken). Instanser med högre nummer har högre prioritet, dvs om signalen för en högre instans är hög, bestämmer denna färgen oavsett värdet på signaler kopplade till lägre instanser.

### Attribut

DigColor.Attribute

DigColor.Color

DigColor.Instance

DigColor2.Attribute

DigColor2.Color

DigColor3.Attribute

DigColor3.Color

### Beskrivning

Signal i rtdb av typen Boolean som ska påverka komponenten.

Fyllnadsfärg på komponenten om signalen är 1.

Anger vilka instanser som ska skapas.

Signal för instans 2.

Fyllnadsfärg för instans 2.

Signal för instans 3.

Fyllnadsfärg för instans 3.

## 8.2.3 AnalogColor

AnalogColor kopplas till en analog signal, och sätter angiven fyllnadsfärg när signalen under- eller överskrider ett gränsvärde. Gränsvärdet anges i AnalogColor.Limit och AnalogColor.LimitType avgör om det är ett undre eller övre gränsvärde. Om LimitType är GreaterThan sätts fyllnadsfärgen när signalens värde är större än gränsvärdet, och när LimitType är LessThan sätts fyllnadsfärgen när signalvärdet är mindre än gränsvärdet. Andra gränsvärdestyper är GreaterEqual (större eller lika med), LessEqual (mindre eller lika med) och Equal (lika med).

AnalogColor kan finnas i flera instanser, vilket gör det möjligt att sätta flera gränsvärden och skifta mellan flera olika färger (upp till 32 stycken).

OBS ! Om det förekommer flera instanser med samma LimitType, måste en högre instans ha ett större gränsvärde vid LimitType GreaterThan och GreaterEqual. Vid LimitType LessThan och LessEqual måste en högre instans ha ett lägre gränsvärde.

### Attribut

AnalogColor.Attribute

AnalogColor.Color

AnalogColor.Border

AnalogColor.CommonAttribute

AnalogColor.Limit

AnalogColor.LimitType

AnalogColor.Instances

AnalogColor2.Color

AnalogColor2.Limit

AnalogColor2.LimitType

AnalogColor3.Color

### Beskrivning

Signal i rtdb av typen Float32 eller Int32. Värdet jämförs med gränsvärdet och om detta

över/underskrids ändras färgen på komponenten.

Fyllnadsfärg på komponenten om signalens värde har passerat gränsvärdet.

Sätt kantfärg istället för fyllnadsfärg.

Samma attribut används av alla instanser.

Annars har varje instans sitt eget attribut.

Gränsvärde.

Typ av gränsvärde. Kan vara GreaterThan, GreaterEqual, LessThan, LessEqual eller Equal.

Mask för instanser, där instanser kan skapas och tas bort.

Fyllnadsfärg för instans 2.

Gränsvärde för instans 2.

Typ av gränsvärde för instans 2.

Fyllnadsfärg för instans 3.

## 8.2.4 StatusColor

StatusColor kopplas till ett attribut av typen Status. En status definieras av en text, och tillhör en av följande fem kategorier: Success, Info, Warning, Error, Fatal. Statusvärdet kan även vara 0, NoStatus. StatusColor sätter fyllnadsfärgen utgående från vilken kategori statusvärdet tillhör.

Success, Info	Ursprunglig färg
Warning	Gul
Error	Röd
Fatal	Blinkande röd.

### Attribut

StatusColor.Attribute

StatusColor.NoStatusColor

StatusColor.UseColorTheme

### Beskrivning

Signal i rtdb av typen Status. Färgen på komponenten sätt utgående från den kategori statusvärdet tillhör.

Färg om statusvärdet är 0.

Använd färgtemats färger.

## 8.2.5 DigWarning

Sätter fyllnadsfärgen till gul vid hög signal.

Objektet kopplas till en digital signal i rtdb. Om värdet är 1 ritas objektet med gul fyllnadsfärg, om värdet är 0 ritas det med den normala fyllnadsfärgen.

Attribut	Beskrivning
DigWarning.Attribute	Signal i rtdb av typen Boolean som ska påverka objektet.
DigWarning.UseColorTheme	Använd färgtemats färger.

## 8.2.6 DigError

Sätter fyllnadsfärgen till röd vid hög signal.

Objektet kopplas till en digital signal i rtdb. Om värdet är 1 ritas objektet med röd fyllnadsfärg, om värdet är 0 ritas det med den normala fyllnadsfärgen.

Attribut	Beskrivning
DigError.Attribute	Signal i rtdb av typen Boolean som ska påverka objektet.
DigError.UseColorTheme	Använd färgtemats färger.

## 8.2.7 DigFlash

Blinkar med angivna färger vid hög signal.

DigFlash ändrar färg på objektet varje cykel. Färgen skiftas mellan de två angivna färgerna i Color och Color2.

Blink-frekvensen bestäms av cykeltiden för den cykel som objektet går på.

### Attribut

DigFlash.Attribute

DigFlash.Color

DigFlash.Color2

### Beskrivning

Signal i rtdb av typen Boolean som ska påverka objektet.

Första blinkande färg.

Andra blinkande färg.

## 8.2.8 FillLevel

Rita objektet med en fyllnadsfärg, upp till en viss nivå.

FillLevel ritar ursprungsfärgen (eller den färg som bestäms av övrig dynamik) upp till en viss gränslinje. På andra sidan gränslinjen ritas objektet med angiven bakgrundsfärg. Gränslinjens läge bestäms av värdet på den analoga signalen. Om värdet är mindre än MinValue ritas hela objektet med bakgrundsfärg. Om värdet är större än MaxValue ritas hela objektet med ursprungsfärgen.

Direction avgör vilken riktning gränslinjen ska flytta sig. Vid Direction Up flyttar sig gränslinjen uppåt med ökande värde, vid Down nedåt, vid Left åt vänster och vid right åt höger.

### Attribut

FillLevel.Attribute

FillLevel.BackgroundColor

FillLevel.Direction

FillLevel.MinValue

FillLevel.MaxValue

FillLevel.MinValueAttr

FillLevel.MaxValueAttr

### Beskrivning

Signal i rtdb av typ Float32.

Bakgrundsfärg.

Riktning Up, Down, Left eller Right.

Minvärde. När värdet på Attribute är lika med MinValue, ritas hela objektet med bakgrundsfärgen.

Maxvärde. När värdet på Attribute är lika med MaxValue, ritas hela objektet med ursprungsfärgen.

Signal för minvärde.

Signal för maxvärde.

## 8.2.9 Invisible

Gör objektet osynligt (eller dimmar) vid hög signal.

Attributet kopplas normalt till en digital signal, men kan även kopplas till en sträng. I detta fall testas på om strängen är tom eller inte. Objektet blir osynligt vid tom sträng.

Invisible kan också användas med xtt-kommandot 'check' för att testa om en metod är definierad för ett objekt. I Attribute skriver man då

`$cmd('command') , t ex`

`$cmd(check method/method=Note/object=VK-Valve)`

### Attribut

Invisible.Attribute

Invisible.Dimmed

Invisible.DimLevel

Invisible.Instances

### Beskrivning

Signal i rtdb av typen Boolean som ska påverka objektet.

1 dimmar objektet, 0 gör objektet osynligt.

Ett transparensvärde mellan 0 och 1 för att addera transparens till det dimmade utseendet.

Mask för instanser, där instanser kan skapas och tas bort.



## 8.2.10 DigBorder

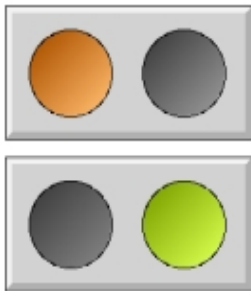
Sätt angiven kantfärg när signalen är låg.

Objektet kopplas till en digital signal i rtdb. Om värdet är 1 ritas objektet med den normala kantfärgen, om värdet är 0 ritas det med DigBorder.Color som anges i objektseditorn.

Attribut	Beskrivning
DigBorder.Attribute	Signal i rtdb av typen Boolean som ska påverka komponenten.
DigBorder.LowColor	Kantfärg på komponenten om signalen är 0.

## 8.2.11 DigBackgroundColor

DigBackgroundColor gör det möjligt att sätta dynamik på ett objekts bakgrundsfärg. DigBackgroundColor fungerar analogt med DigColor, men kommer att ändra bakgrundsfärgen istället för fyllnadsfärgen. DigBackgroundColor har flera instanser och kan därmed kopplas till flera digitala signaler, och alternera mellan flera färger. Med DigBackgroundColor kan man modifiera färgen på två olika delar av en subgraf eller grupp oberoende av varandra. I exemplet nedan är den grå rektangeln ritad med en fix grå färg, den vänstra cirkeln med fyllnadsfärgen, och den högra med bakgrundsfärgen. Att rita en cirkel med bakgrundsfärgen åstadkoms genom att sätta `fill_eq_background` i egenskaperna för cirkeln.



**Fig Dynamik med bakgrundsfärg**

När man sätter DigColor och DigBackgroundColor dynamik på gruppen, kommer DigColor dynamiken att påverka den vänstra cirkeln ritad med fyllnadsfärg, och DigBackgroundColor dynamiken den högra cirkeln ritad med bakgrundsfärg.

### Attribute

DigBackgroundColor.Attribute  
DigBackgroundColor.Color  
DigBackgroundColor.Instances

### Description

Signal i databasen av typ Boolean.  
Bakgrundsfärg när signalen är 1.  
Mask för instanser, där instanser kan skapas och tas bort.

## 8.2.12 TimeoutColor

Sätter angiven färg vid timeout på prenumerationen.

Kontrollerar tiden för prenumerationen, och sätter fyllnadsfärgen till den angivna färgen om timeout tiden har överskridits.

TimeoutColor måste kombineras med någon annan dynamik som anger attributet för prenumerationen. Om det finns flera typer av dynamik, övervakas prenumerationen för dynamiken med högst prioritet.

### Attribut

TimeoutColor.Time

TimeoutColor.Color

### Beskrivning

Timeout tid i sekunder.

Fyllnadsfärg för komponenten vid timeout.

## 8.2.13 DigTextColor

Sätter angiven färg när signalen är hög.

Objektet kopplas till en digital signal i rtdb. Om värdet är 0 ritas objektet med den normala textfärgen, om värdet är 1 ritas det med DigTextColor.Color som anges i objektseditorn.

### Attribut

DigTextColor.Attribute

DigTextColor.Color

### Beskrivning

Signal i databasen av typen Boolean som ändrar textfärg på komponenten.

Färg som sätts på texter när signalen är 1.

## 8.2.14 DigText

Skifta mellan två texter.

Objektet kopplas till en digital signal i rtdb. Om värdet är 1 skrivs original-texten ut, om värdet är 0 skrivs texten i LowText ut.

Med flera instanser kan objektet kopplas till fler signaler, och fler texter kan visas. För instans 2 och uppåt, visas den angivna HighText när motsvarande signal är hög.

DigText opererar på textfält 1.

### Attribut

DigText.Attribute

DigText.LowText

DigText.Instances

DigText2.Attribute

DigText2.HighText

### Beskrivning

Signal i rtdb av typen Boolean som skiftar text.

Text som skrivs ut när signalen är låg.

Mask för instanser, där instanser kan skapas och tas bort.

Signal i rtdb av typen Boolean.

Text som skrivs ut när signalen är hög.

## 8.2.15 AnalogText

Skifta mellan flera texter beroende på värdet på en analog signal.

Objektet kopplas till att databas attribut av typen float eller int. Till varje text anger man ett Enum värde. Texten vars Enum värde är lika med databas attributets värde kommer att visas i komponenten.

### Attribut

AnalogText.Attribute

AnalogText.TextMask

AnalogText.Text1

AnalogText.Enum1

AnalogText.Text2

AnalogText.Enum2

### Beskrivning

Databas attribut av typen Float32, Float64, Int32, UInt32, Int16, UInt16, Int8 eller UInt8. Den text vars Enum motsvarar värdet på databas attributet visas.

Mask som bestämmer antalet texter.

Text nr 1.

Uppräkningsvärdet för första texten.

Text nr 2.

Uppräkningsvärdet för andra texten.

## 8.2.16 Value

Skriver ut värdet på en signal. Värdet skrivs ut i ett textfält.

Objektet kopplas till ett attribut i rtdb av valfri typ. I Format anges format på utskriften med c-syntax.

Det är möjligt att skapa fler instanser. Detta kräver att subgrafan har flera fält att skriva ut värden på. Värdet för en instans skrivs ut i fältet med motsvarande nummer (annotation number).

### Attribut

Value.Attribute

Value.Format

Value.ZeroBlank

Value.DecimalsAttr

Value.DecimalsDecrease

Value.Instances

### Beskrivning

Signal i rtdb av av valfri typ, som ska visas i textfältet.

Format i c-printf syntax.

När värdet är noll är textfältet blankt.

Heltalssignal som anger antal decimaler.

Antalet decimaler kommer att minskas med angivet värde.

Mask för instanser, där instanser kan skapas och tas bort.

### Format

För följande attribut typer finns ytterligare format definierade.

#### pwr\_tTime

%t	Datum och tid. Ex 24-MAY-2007 11:33:43.91
%1t	Enbart tid, inga hundradelar. Ex 11:33:43
%2t	Enbart tid med hundradelar. Ex 11:33:43:91
%3t	Komprimerad datum och tid. Ex 07-05-24 11:33:43
%4t	Enbart datum. Ex 24-MAY-2007
%5t	Komprimerat datum. Ex 07-05-24
%6t	Tid och datum. Ex 01:00:00 30/01/87

#### pwr\_tDeltaTime

%t	Tid med hundradelar. Ex 1:23:45.43
%1t	Tid utan hundradelar. Ex 1:23:45

#### pwr\_tEnum

%d	Uppräkningsvärdet visas som en UInt32.
%s	Uppräkningsvärdet konverteras till motsvarande sträng.

#### pwr\_tObjid

%o	Objektsnamn (sista segmentet).
%1o	Hierarki och objektsnamn.
%2o	Volym, hierarki och objektsnamn.

#### pwr\_tAttrRef och pwr\_tDataRef

%o	Objektsnamn (sista segmentet) och attributnamn.
%1o	Hierarki, objektsnamn och attributnamn.
%2o	Volym, hierarki, objektsnamn och attributnamn .

**pwr\_tStatus och pwr\_tNetStatus**

%m	Statusvärdet konverteras till motsvarande sträng. Ex "GDH-E-NOSUCHOBJ, No such object".
%1m	Enbart textdelen av strängen visas. Ex "No such object".

**pwr\_tMask**

%d	Masken visas som en UInt32.
%b	Masken visas som ett 32-bitars binärt värde.
%16b	Masken visas som ett 16-bitars binärt värde. Ex 0011111100011001.



## 8.2.17 Rotate

Roterar ett objekt.

Rotationen sker med utgångspunkt från den position objektet ges i editorn. Värdet i Attribute multipliceras med Factor och ger rotationen i grader.

x0 och y0 anger rotations-punkten. Om x0 och y0 är noll, används origo i subgrafens koordinater som rotationspunkt. Subgrafer med Rotate som standard-dynamik är oftast ritade så att rotations-punkten inte behöver anges.

### Attribut

Rotate.Attribute

Rotate.x0

Rotate.y0

Rotate.Factor

Rotate.Offset

Rotate.MinAngle

Rotate.MaxAngle

### Beskrivning

Signal i rtdb av typen Float32. Ger rotation av objektet.

x-koordinat för rotationspunkten.

y-koordinat för rotationspunkten.

Factor multiplicerat med signalens värde ger rotationen i grader.

Offset för vinkeln.

Min-värde för vinkeln.

Max-värde för vinkeln.

## 8.2.18 Move

Flyttar och skalar ett objekt. Kan kopplas till fyra analoga attribut som flyttar objektet i x- resp y-led, och skalar det i x- resp y-led.

Förflyttningen sker med utgångspunkt från den position objektet ges i editorn. Omräkningen från värdet i XAttribute och YAttribute till Ge -koordinater kan påverkas av XOffset, YOffset och Factor. XOffset och YOffset är det värden som motsvarar utgångspunkten i x- resp y-led. Factor anger omräknings-faktor från XAttribute och YAttribute-värden till Ge -koordinater. Omräkningen sker enligt formeln:

$$dx = (XAttribute - XOffset) * Factor$$
$$dy = (YAttribute - YOffset) * Factor$$

Skalningen i x- och y-led sker med ledning av värdet av ScaleXAttribute resp ScaleYAttribute. Dessa multipliceras med ScaleFactor och ger skalfaktorn, där 1.0 ger oförändrad skalning, värden större än 1 ger förstoring och värden mindre än 1 ger förminskning.

Attribut	Beskrivning
Move.XAttribute	Signal i rtdb av typen Float32. Ger förflyttning i x-led.
Move.YAttribute	Signal i rtdb av typen Float32. Ger förflyttning i y-led.
Move.XFactor	Omräkningsfaktor från storhet för XAttribute till storhet för Ge-koordinater i x-led.
Move.YFactor	Omräkningsfaktor från storhet för YAttribute till storhet för Ge-koordinater i y-led.
Move.XOffset	Värde för XAttribute som motsvarar utgångspositionen för objektet.
Move.YOffset	Värde för YAttribute som motsvarar utgångspositionen för objektet.
Move.ScaleXAttribute	Signal i rtdb av typen Float32. Ger skalning i x-led.
Move.ScaleYAttribute	Signal i rtdb av typen Float32. Ger skalning i y-led.
Move.ScaleXFactor	Omräkningsfaktor från storhet för ScaleXAttribute till skalfaktor i x-led.
Move.ScaleYFactor	Omräkningsfaktor från storhet för ScaleYAttribute till skalfaktor i y-led.

## 8.2.19 AnalogShift

Skifta flera sidor i subgrafen. Subgrafen måste innehålla minst två sidor.

Objektet kopplas till en analog signal i rtdb. Värdet på det analoga attributet bestämmer index för den sida som ska ritas. Vid 0 ritas den första sidan, vid 1 den andra, osv.

Attribut	Beskrivning
AnalogShift.Attribute	Signal i rtdb av typen Float32 eller Int32 som skiftar sida på subgrafen. Värdet bestämmer vilken sida som ska ritas. Vid 0 ritas första sidan.

## 8.2.20 DigShift

Skifta två sidor i subgrafén. Subgrafén måste innehålla minst två sidor.

Objektet kopplas till en digital signal i rtdb. Om värdet är 0 ritas objektet med första sidan, om värdet är 1 ritas det med sista sidan.

Attribut	Beskrivning
DigShift.Attribute	Signal i rtdb av typen Boolean som skiftar sida. Vid 0 visas första sidan, vid 1 visas sista sidan.

## 8.2.21 DigFourShift

Shifta mellan fyra sidor i subgrafén. Subgrafén måste innehålla minst fyra sidor.

Objektet kopplas till tre digitala signaler i databasen. Om värdet är 1 på Attribute1, ritas den första sidan, om Attribute2 är 1 ritas den andra sidan, och om Attribute3 är 1 ritas den tredje sidan. Om värdet är 0 på alla signaler ritas den sista sidan.

Om flera signaler är 1 samtidigt har högre sidnummer högre prioritet.

### Attribut

DigFourShift.Attribute1

DigFourShift.Attribute2

DigFourShift.Attribute3

### Beskrivning

Signal i databasen som skiftar sida. Aktiverar ritning av första sidan.

Signal i databasen som skiftar sida. Aktiverar ritning av andra sidan.

Signal i databasen som skiftar sida. Aktiverar ritning av tredje sidan.

## 8.2.22 ScrollingText

Visa en rullande text.

Objektet kopplas till ett sträng-attribut i databasen, och texten i attributet visas.

Attribut	Beskrivning
ScrollingText.Attribute	Attribut i databasen av typ String.
ScrollingText.Direction	Rullningsriktning, vänster, höger, upp eller ner.
ScrollingText.Speed.	Hastighet på rullningen.

## 8.2.23 Animation

Dynamik för animerade subgrafer, dvs subgrafer med flera sidor. Objektet kopplas till en digital signal som aktiverar animationen.

Animations-sekvensen bestäms av Sequence, som kan vara Cyclic, ForwBack eller Dig.

### Cyclic

Skifta viloläge och cyklisk animering. Animationen sker från 2:a till sista sidan och börjar sedan om på 2:a sidan osv.

Om signalens värde är 0 ritas objektet med första sidan, om värdet är 1 startar cyklisk animering mellan 2:a sidan och sista sidan. Vid animeringen genomlöps sidorna från 2:a till sista sidan, därefter hoppar man till 2:a sidan och animerar upp till sista sidan igen, osv.

### ForwBack

Skifta viloläge och cyklisk animering. Animationen sker fram och tillbaka mellan 2:a och sista sidan. Om signalens värde är 0 ritas objektet med första sidan, om värdet är 1 startar cyklisk animering fram och tillbaka mellan första och sista sidan.

### Dig

Skiftar två vilolägen. Vid omslag sker animering mellan vilolägena.

Om signalens värde är 0 ritas objektet med första sidan, om värdet är 1 ritas sista sidan. Vid omslag från 0 till 1 sker animering från första till sista sidan, vid omslag från 1 till 0 sker animering från sista till första sidan.

### Attribut

Animation.Attribute

Animation.Sequence

### Beskrivning

Signal i rtdb av typen Boolean som aktiverar animationen.

Animering sekvens. Kan vara Cyclic, ForwBack eller Dig.

## 8.2.24 Video

Visar en gif-fil som kontinuerligt uppdateras av en Web-kamera.  
Video har inga attribut.



## 8.2.25 DigCommand

Exekvera ett kommando när signalen går från låg till hög, eller när signalen är hög.

Objektet kopplas till en digital signal i rtdb. Kommandot exekveras antingen när signalvärdet ändras från 0 till 1, eller cykliskt när signalvärdet är högt.

DigCommand kan finnas i flera instanser, vilket gör det möjligt att exekvera flera kommandon.

### Attribut

DigCommand.Attribute

DigCommand.Level

DigCommand.Command

DigCommand.Instances

DigCommand2.Attribute

DigCommand2.Level

DigCommand2.Command

### Beskrivning

Signal i databasen av typ boolean som ska påverka komponenten

Om Level är 0, exekveras kommandot när signalen ändras från 0 till 1.

Om level är 1, exekveras kommandot cykliskt när signalen är hög.

Xtt kommando som exekveras när signalen går från 0 till 1.

Mask för instanser, där instanser kan skapas och tas bort.

Signal för instans nr 2.

Level for instans nr 2.

Kommando för instans nr 2.

## 8.2.26 DigScript

Exekvera ett script när signalen går från låg till hög, eller när signalen är hög.

Objektet kopplas till en digital signal i rtdb. Scriptet exekveras antingen när signalvärdet ändras från 0 till 1, eller cykliskt när signalvärdet är högt.

Attribut	Beskrivning
DigScript.Attribute	Signal i databasen av typ boolean som ska påverka komponenten
DigScript.Level	Om Level är 0, exekveras scriptet när signalen ändras från 0 till 1. Om level är 1, exekveras scriptet cykliskt när signalen är hög.
DigScript.Arguments	Argument som kan användas som p1-p9 i scriptet. Argumenten kommer att översättas om ett kommando (se action Command), till ersätts \$object med aktuell instans för en objektbild.
DigScript.Script	Xtt script som exekveras när signalen går från 0 till 1.

## 8.2.27 DigTransparency

Ändra transparans beroende på en digital signal. När signalen är låg sätts transparensen till det värdet som är angivet i LowValue, och när signalen är hög till värdet i HighValue.

### Attribut

DigTransparency.Attribute  
DigTransparency.LowValue

DigTransparency.HighValue

DigTransparency.SmoothTransition

### Beskrivning

Signal i databasen av typ boolean som ska påverka komponenten

Transparans som sätts när signalen är låg. Ett värde mellan 0 och 1.

Transparans som sätts när signalen är hög. Ett värde mellan 0 och 1.

Gradvis övergång mellan högt och lågt värde.

## 8.2.28 AnalogTransparency

Transparansen för ett objekt kopplas till en analog signal, och transparansen bestäms av värdet på den analoga signalen.

### Attribut

AnalogTransparency.Attribute  
AnalogTransparency.MinValue  
AnalogTransparency.MaxValue

### Beskrivning

Signal i databasen av typ Float32.  
Signalvärde som motsvarar minimal transparans.  
Signalvärde som motsvarar maximal transparans.

## 8.3 Action

Action anger vad som ska hända när man aktiverar ett objekt t ex genom att klicka på det, Man kan sätta eller återställa signaler i rtdb, ta upp en annan graf, visa en hjälptext eller en tooltip text mm.

Action	Signal typ	Beskrivning
Inherit	-	Action för instansen ärvs från subgrafen.
PopupMenu	Objekt	En popupmeny kan öppnas från komponenten med MB3.
SetDig	Boolean	Sätter värdet på en signal till true vid Klick MB1.
ResetDig	Boolean	Sätter värdet på en signal till false vid Klick MB1.
ToggleDig	Boolean	Ändrar värdet på en signal vid Klick MB1.
StoDig	Boolean	Sätter värdet på en signal så länge knappen är intryckt.
SetValue	Float, Int, String	Sätter vardet på en signal till ett angivet värde.
Command	-	Exekverar ett xtt-kommando vid Klick MB1.
CommandDoubleClick	-	Exekverar ett xtt-kommando vid Dubbelklick MB1.
Script	-	Exekverar ett xtt-script vid Klick MB1, eller när grafen öppnas eller
Confirm	-	Verifiering innan aktionen utförs.
IncrAnalog	Float32	Öka eller minska ett analogt värde.
RadioButton	Boolean	Radio-knapps uppförande hos komponenten.
Slider	Float32 Int32	Reserverad för slider subgrafer.
ValueInput	Valfri	Inmatning.
ToolTip	-	Visning av tooltip text.
Help	-	Visar en hjälptext vid Klick MB1.
OpenGraph	-	Öppnar en graf vid Klick MB1.
OpenURL	-	Öppnar en URL i en lämplig web-läsare vid Klick MB1.
PulldownMenu	-	Aktion för rullgardings menyer.
OptionsMenu	-	Aktion för option menyer.

## 8.3.1 PopupMenu

En popup-meny öppnas med MB3. Popup-menyn innehåller metoderna för det objekt som anges i RefObject (se avsnittet Metoder i Xtt manualen).

Attribut	Beskrivning
PopupMenu.RefObject	Objekt i rtdb, vars metoder aktiveras från popup-menyn.

## 8.3.2 SetDig

Sätter värdet på en digital signal till ett vid Klick MB1 på objektet.

SetDig kan finnas i flera instanser, och sätter då värdet på flera signaler.

Attribut	Beskrivning
SetDig.Attribute	Signal i rtdb av typen Boolean som sätts till 1 vid Klick MB1 på objektet.
SetDig.Instances	Mask för instanser, där instanser kan skapas och tas bort.

## 8.3.3 ResetDig

Sätter värdet på en digital signal till 0 vid Klick MB1 på objektet.

ResetDig kan finnas i flera instanser, och sätter då värdet på flera signaler till 0.

Attribut	Beskrivning
ResetDig.Attribute	Signal i rtdb av typen Boolean som sätts till 0 vid Klick MB1 på objektet.
ResetDig.Instances	Mask för instanser, där instanser kan skapas och tas bort.



## 8.3.4 ToggleDig

Togglar värdet på en digital signal vid Klick MB1 på objektet.

Attribut	Beskrivning
ToggleDig.Attribute	Signal i rtdb av typen Boolean som togglas vid Klick MB1 på objektet.

## 8.3.5 StoDig

Sätter värdet på en digital signal till ett så länge knappen är intryckt.

### **Attribut**

StoDig.Attribute

### **Beskrivning**

Signal i rtdb av typen Boolean som sätts vid Klick MB1 på objektet.

## 8.3.6 SetValue

Sätter värdet på en signal till ett angivet värde.

Genom att använda flera instanser kan man sätta värden i flera attribute på samma knapp.

### Attribut

SetValue.Attribute

SetValue.Value

SetValue.Instances

### Beskrivning

Signal i rtdb av typen Float32, Int32, UInt32, Int16, UInt16, Int8, UInt8 eller String, som sätt vid Klick MB1 på objektet.

Värdet som sätts.

Mask för instanser, där instanser kan skapas och tas bort.

## 8.3.7 Command

Exekverar ett xtt-kommando vid Klick MB1 på objektet.

### Attribut

Command.Command

### Beskrivning

Xtt-kommando som exekveras vid mus-klick på objektet.

Tillägg till xtt-kommando syntaxen

### \$object

\$object kommer att ersättas av det aktuella objektet i en objekts bild, eller av den aktuella hierarkin i en hierarki-bild.

### \$hostobject

En speciell syntax för HostObject dynamik där \$hostobject kommer att ersättas av det aktuella host objektet.

### &(attribut-referens)

Strängen för attribut-referenser av typen '&(attributnamn)' kommer att ersättas av innehållet i attributreferensen. Antag att det finns en vektor, H1-ObjectList.ObjArray, med attributreferenser som pekar på objekt för vilka objektsbilden ska visas. Med kommandot

```
> open graph /class /instance=&(H1-ObjectList.ObjArray[0])
```

öppnas objektgrafan för objektet i första elementet. Man kan även ange dubbla led av attributreferenser med syntaxen &(&(attribut-referens)).

### Array index replacement

Ett Int32 eller UInt32 attribut kan användas för vektor index med syntaxen [&(integer-attribute)]. Med kommandot

```
> open graph /class /instance=H1-Plates[&(H1-CurrentIndex.ActualValue)]
```

kommer element indexet för H1-Plates att hämtas från det aktuella värdet av H1-CurrentIndex.ActualValue.

### Parent object references

Med .< tas det föregående attributet bort i ett attributnamn. H1-Plates.Data.<.CurrentIndex kommer att resultera i H1-Plates.CurrentIndex. Detta kan användas i objektsbilder i kombination med \$object. Till exempel

```
> open graph /class /instance=$object.<
```

kommer att öppna objektsbilden för objektet som innehåller det aktuella objektet.

-< syntaxen är likartad och används för objektshierarkier.

H1-Pumps-P1<- kommer att resultera i H1-Pumps. Det gör det möjligt att adressera förälderobjekt i hierarkibilder. Till exempel

```
> open graph /class /instance=$object-<
```

kommer att öppna objektsbilden för förälderobjektet.

## 8.3.8 CommandDoubleClick

Exekverar ett xtt-kommando vid Dubbelklick MB1 på objektet.

Attribut	Beskrivning
CommandDoubleClick.Command	Xtt-kommando som exekveras vid mus-klick på objektet.

## 8.3.9 Script

Exekverar ett xtt-script vid Klick MB1 på objektet, eller när grafen öppnas eller stängs.

Attribut	Beskrivning
Script.TriggerEvent	Händelse som startar scriptet. ClickMB1, vid Klick MB1. Open, när grafen öppnas. Close, när grafen stängs.
Script.Arguments	Argument som kan användas som p1-p9 i scriptet. Argumenten kommer att översättas om ett kommando (se action Command), till ersätts \$object med aktuell instans för en objektbild.
Script.Script	Xtt-script som exekveras vid mus-klick på objektet.

Script kan använda externa variabler för att lagra statusvärden och för interaktion mellan script. De externa variablerna kan deklarerars i ett script som triggas av Open, och tas bort i ett script triggat av Close.

Externa variabler kan även användas i objektbilder, där flera objektbilder kan vara öppna samtidigt. Sammanblandning av variablerna undviks genom att sätta namnrymden för de externa variablerna till objektnamnet. Ange instans-objektet som argument och sätt namnrymd med `set_namespace(p1)`.

## 8.3.10 Help

Öppnar hjälpfönstret med angivet ämne vid Klick MB1 på objektet.

Attribut	Beskrivning
Help.Topic	Ämne vars hjälptext ska visas.
Help.Bookmark	Bokmärke i hjälptexten. Kan utelämnas.



## 8.3.11 OpenGraph

Öppnar en graf via ett XttGraph-objekt vid Klick MB1.

Attribut	Beskrivning
OpenGraph.GraphObject	Objekt av klassen XttGraph.

## 8.3.12 OpenURL

Öppnar en URL i lämplig webbläsare Klick MB1.

Attribut	Beskrivning
OpenURL.URL	En URL

## 8.3.13 Confirm

Confirm innebär bekräftelse krävs innan en action genomförs. En dialogruta med angiven text resp Ja och Nej knappar visas, innan övriga action-element exekveras.

### Attribut

Confirm.Text

Confirm.OnSet

Confirm.OnReset

### Beskrivning

Text som visas i verifierings fönstret.

Används med action ToggleDig när bekräftelse enbart krävs när v

När värdet sätts till 0 sker det utan bekräftelse.

Används med action ToggleDig när bekräftelse enbart krävs när v

När värdet sätts till 1 sker det utan bekräftelse.

## 8.3.14 IncrAnalog

Öka eller minska värde på ett anlogt attribut.

Objektet kopplas till ett analog attribut. Vid Klick MB1 på objektet ökas värdet på attributet med värdet angivet i Increment. Min och Max-värde för attributet kan anges.

Attribut	Beskrivning
IncrAnalog.Attribute	Signal i rtdb av typen Float32. Vid Klick MB1 ökas värdet av attributet med värdet i Increment.
IncrAnalog.Increment	Värde som attributet i Attribute ökas/minska med.
IncrAnalog.MinValue	Minvärde för attributet.
IncrAnalog.MaxValue	Maxvärde för attributet.

## 8.3.15 RadioButton

Sätter värdet på ett digitalt attribut vid klick och återställer värdet för övriga RadioButton-objekt i gruppen.

Objektet måste ingå i en grupp med andra objekt av typen RadioButton.

### Attribut

RadioButton.Attribute

### Beskrivning

Signal i rtdb av typen Boolean. Vid Klick MB1 sätts värdet av attributet och övriga RadioButton objekt i samma grupp återställs

## 8.3.16 ValueInput

Inmatningsfält. Ger möjlighet att mata in ett värde på en signal, i en komponent som har Value dynamik. Värdet kan ändras genom att man klickar med MB1 på objektet.

Signalen anges i Value.Attribute.

### Attribut

ValueInput.MinValule  
ValueInput.MaxValue  
ValueInput.Clear  
ValueInput.Popup  
ValueInput.Unselect  
ValueInput.EscapeStore

ValueInput.MinValueAttr  
ValueInput.MaxValueAttr  
ValueInput.KeyboardType

### Beskrivning

Minimivärde vid inmatning.  
Maximivärde vid inmatning  
Fältet töms när det öppnas för inmatning.  
Vid inmatning öppnas ett inmatningsfönster.  
Texten i fältet markeras inte när fältet öppnas för inmatning.  
Om fältet är öppet när grafen stängs, lagras det imatade värdet i databasen. Det lagras även av xtt kommandot 'set graph /escapestore'.  
Signal för minivärde vid inmatning.  
Signal för maxivärde vid inmatning.  
Typ av virtuellt tangentbord som öppnas när fältet har input focus. Kräver att VirtualKeyboard är angivet för operatörsplatsen.

## 8.3.17 ToolTip

Visar ToolTip text för en komponent. ToolTip är en vit textruta som visas om markören vilar på en komponent en viss tid. När markören lämnar komponenten tas textrutan bort.

I 'Text' anges normal tooltip-texten. Om texten ska vara ändringsbar, kan man även koppla till ett sträng-attribut genom att ange sträng-attributet i 'Text' med '&' som prefix, t ex

```
&A1-B1- Sv1.ActualValue##String80
```

Attribut	Beskrivning
ToolTip.Text	ToolTip text.

## 8.3.18 InputFocus

Input fokus är en funktion gör att man kan påverka komponenter i bilden från tangentbordet istället för med musen. Det flesta funktioner som utförs med pekare och mus kan även utföras från tangentbordet.

En komponent väljs ut mha pil- eller Tab-tangenterna, (eller genom att klicka på den), så att den har input fokus.

Det innebär att all inmatning från tangentbordet kanaliseras till komponenten. En tryckknapp aktiveras t ex genom att man trycker på Return, en optionmeny genom att man väljer ut önskat alternativ med piltangenterna och trycker på Return, en ValueInput genom att man matar in önskat värdet och trycket på Return.

Endast komponenter med actiontypen InputFocus kan erhålla input focus, undantaget är ValueInput som även kan få input fokus genom att man klickar på dem.

Man skiftar input fokus mellan komponenterna med pil-tangenterna eller Tab-tangenten. Komponenternas ordning konfigureras genom att man länkar dem i tre listor, en horisontell, en vertikal och en Tab-lista. I den horisontella listan förflyttar man sig med Pil Vänster och Pil Höger, i den vertikala med Pil Upp och Pil Ner, i Tab listan med Tab tangenten. Det hela konfigureras med attributen InitialFocus, NextHorizontal, NextVertical och NextTab. I InitialFocus kan man ange att ett objekt ligger först eller sista i en lista. Man kan även ange att en komponent ska ha input fokus när bilden tas upp. I NextHorizontal, NextVertical och NextTab lägger man in objektsnamnet på nästa komponent i respektive lista.

Man brukar dela in komponenterna i Tab grupper. Mellan komponenterna inom en Tab-grupp navigerar man med pil-tangenterna, och mellan Tab grupperna navigerar man med Tab-tangenten. Detta åstadkommes genom att samtliga komponenter inom en Tab-grupp har första objektet i nästa Tab-grupp som NextTab.

### Attribut

InputFocus.InitialFocus

### Beskrivning

Bitmask med följande bitar.

- InitialFocus Komponenten har input fokus när grafen öppnas
- FirstHorizontal Komponenten är första objekt i horisontella listan.
- FirstVertical Komponenten är första objekt i vertikala listan.
- FirstTab Komponenten är första objekt i Tab-listan.
- LastHorizontal Komponenten är sista objekt i horisontella listan.
- LastVertical Komponenten är sista objekt i vertikala listan.

InputFocus.NextHorizontal

Objektsnamn på nästa komponent i horisontella listan.

InputFocus.NextVertical

Objektsnamn på nästa komponent i vertikala listan.

InputFocus.NextTab

Objektsnamn på nästa komponent i Tab listan.



## 8.3.19 PulldownMenu

Action för en rullgardinsmeny, eller för en submeny i en rullgardinsmeny.

En rullgardinsmeny består av ett antal meny alternativ som antingen är tryckknappar eller submenyer.

Antalet alternativ i menyn konfigureras i ItemMask, en meny kan ha upp till 32 menyalternativ.

För varje menyalternativ väljer man en actiontyp. För tryckknappar kan man välja mellan SetDig, ResetDig, ToggleDig, Command, Help, OpenGraph, CloseGraph, OpenURL och IncrAnalog.

Submenyer ska ha actiontyp PulldownMenu, och för denna konfigurerar man antalet menyalternativ och actiontyp på samma sätt som för rullgardinsmenyn. Submenyer kan byggas i obegränsat antal nivåer.

Attribut	Beskrivning
Text	Text för rullgardins menyn.
PulldownMenu.ItemMask	Mask som bestämmer antalet menyalternativ.
PulldownMenu.ItemText1	Text för första menyalternativet.
PulldownMenu.ItemDyn1	Dynamik för första menyalternativet. Innehåller action typ, samt eventuell access, för menyalternativet.
PulldownMenu.ItemText2	Text för andra menyalternativet.
PulldownMenu.ItemDyn2	Dynamik för andra menyalternativet.

## 8.3.20 OptionMenu

Option meny är en meny där man väljer alternativ i en lista av alternativ. Det valda alternativet visas upp i meny-komponenten. När man klicka på komponenten öppnas listan med alternativ. När man väljer ett alternativ stängs listan och det valda alternativet läggs in i komponenten.

Optionmeny kopplas till ett databas-attribut av typ float eller int. Varje alternativ i listan motsvarar ett uppräkningsvärde, och när alternativet väljs läggs värdet in i databas-attributet. Komponentens läser kontinuerligt av värdet på databasattributet, och skriver ut texten för det alternativ som motsvarar värdet, i sitt text-fält.

Optionmenyn kan vara statisk eller dynamisk. För en statisk meny anger man texter och uppräkningsvärde i olika item, för en dynamisk meny hämtas meny-texterna från ett attribut i databasen av typen array of String80. Antalet texter i menyn hämtas från ett attribut av typen Int32, och dessutom ska man ange ett attribut som indikerar att nya texter ska läsas in. Det här attributet ska vara av typen Boolean och inläsningen av nya texter sker när värdet ändras från 0 till 1.

### Attribut

OptionMenu.Type

### Beskrivning

Typ av optionmeny. Statisk eller dynamisk.

### Statisk meny

OptionMenu.Attribute

Databas attribut av typen Float32, Float64, Int32, UInt32, Int16, UInt16, Int8 eller UInt8. Värdet i ItemEnum för valt alternativ sätts i databas attributet.

OptionMenu.ItemMask

Mask som bestämmer antalet menyalternativ.

OptionMenu.ItemText1

Text för första menyalternativet.

OptionMenu.ItemEnum1

Uppräkningsvärdet för första menyalternativet.

OptionMenu.ItemText2

Text för andra menyalternativet.

OptionMenu.ItemEnum2

Uppräkningsvärdet för andra menyalternativet.

### Dynamisk meny

OptionMenu.Attribute

Databas attribut av typen Float32, Float64, Int32, UInt32, Int16, UInt16, Int8 eller UInt8. Index för valt alternativ sätts i databas attributet.

OptionMenu.TextAttribute

Databas attribut av typen vektor med String80, som innehåller de texter som visas i menyn.

OptionMenu.SizeAttribute

Databas attribut av typen Int32, som anger antalet alternativ i menyn.

OptionMenu.UpdateAttribute

Databas attribut av typen Boolean, som anger att texterna i menyn ska uppdateras. Uppdateringen sker när värdet ändras från 0 till 1.

## 8.3.21 MethodPulldownMenu

Action för en rullgardinsmeny med menyalternativ för ett objekts metoder.  
Endast konfigurerade metoder visas i menyn.

### Attribut

MethodPulldownMenu.Object  
MethodPulldownMenu.HelpMenu

### Beskrivning

Objekt för vilket metoderna ska visas.  
Menyn är en hjälpmeny och endast hjälp-metoderna  
Help och Help Class visas i menyn.

## 8.3.22 EmitSignal

Aktion för att sända en signal.

### Attribut

EmitSignal.SignalName

EmitSignal.Global

### Beskrivning

Signalnamn.

Om global är satt, sänds signalen till all öppna grafer och multiviewer. Annars vara till den nuvarande grafen.

## 8.3.23 CatchSignal

Aktion för att fånga en signal. När signalen är fångad, exekveras aktionerna för mus-klick för det aktuella objektet.

Attribut	Beskrivning
CatchSignal.SignalName	Signalnamn.

## 8.4 Skapa en subgraf

Man skapar en subgraf genom att rita bas-objekt i arbetsarean. Man har här även tillgång till kopplingspunkter och annotations.

Markera att det är en subgraf genom att öppna File/Graph attributes och sätt en etta på subgraph.

Stäng attribut-editorn och öppna den igen och editera subgraf attributen.

Attribut	Beskrivning
DynType	Typ av dynamik. Instanserna kommer att ärva denna som default-dynamik.
Action	Typ av action. Instanserna kommer att ärva denna som default-action.
Color1	Första färg eller färgton som instanserna kommer att ärva som default.
Color2	Andra färg eller färgton som instanserna kommer att ärva som default.
Color3	Tredje färg eller färgton som instanserna kommer att ärva som default.
Color4	Fjärde färg eller färgton som instanserna kommer att ärva som default.
AnimSequence	Typ av animation då DynType är Animation.
NoConObstacle	Anger att kopplingar kan ignorera en instans av subgraf. Kopplingar av typen 'routed' undviker normalt ett subgraf-objekt, men om NoConObstacle är satt kan kopplingarna ritas över objektet.
Slider	Markerar att subgraf är en slider.
Subgraph	Ska vara ett för en subgraf.
NextSubgraph	Namn på subgrafens nästa sida.
AnimationCount	Används vid animation. Anger antal cykler animeringen ska uppehålla sig vid den här sidan.
JavaName	Vid export som java-bean kommer subgraf att exportera som en java-klass med det här namnet.
Cycle	Default-värde för cykel, långsam eller snabb, som instanserna kommer att ärva som default.
x0	I (x0, y0) och (x1, y1) kan man ange max-storleken på en subgraf i de fall när subgraf innehåller sidor med olika utbredning.
y0	Se x0. Används även av vissa subgrafer med DynType Slider, Sliderbackground och FillLevel.
x1	Se x0.
y1	Se x0. Används även av vissa subgrafer med DynType Sliderbackground och FillLevel.
InputFocusMark	Hur objekt markeras när de har input fokus.
RecursiveDynamic	Används för dynamiken HostObject när objekt i subgraf har individuell dynamik.
Dynamic	Ej implementerat.

Spara subgraf med 'File/SaveAs...'. Kopiera den .pwsig fil som skapats till \$pwrp\_exe. Subgraf ska nu synas under Local/Subgraphs-mappen i subgraf-paletten.

Det är lämpligt att rita subgraf i närheten av origo i arbetsarean eftersom när man skapar instanser motsvaras positionen för musklicket av origo. Kopplingspunkter läggs på grid-punkter, om rektangulära kopplingar ska användas för att knyta ihop instanserna. Då undviker man onödiga hack i kopplingslinjerna.

### Externa och interna subgrafer

När man första gången i en bild skapar en instans av en viss subgraf, laddas denna in från

.pws-gfilen. Den är då intern vilket innebär att subgraf-n sparas tillsammans med bilden, och pws-gfilen behövs i princip inte mer. Man behöver t ex inte tänka på att kopiera den till operatörsnoder och processnoder. Detta fungerar bra ända tills man vill ändra i subgraf-n. Så länge subgraf-n är intern kommer inte dessa ändringar med. För att få med en ändring måste subgraf-n vara extern. När bilden då sparas kommer inte subgraf-n att lagras i tillsammans i bilden, och nästa gång man läser bilden, laddas subgraf-n från från .pws-gfilen. När den ändrade versionen är inladdad kan man åter ställa om den till intern.

För att ställa om en subgraf från intern till extern och vice versa väljer man 'Loaded Subgraps' under 'File' i menyn. Här får man en lista på alla laddade subgrafer och kan sätta dem externa eller interna.

## Slider

Slider eller dragregel är en speciell typ av subgraf. En slider ska kunna flyttas horisontellt eller vertikalt mellan två ändpunkter, och läget ska överföras, via skalning, till ett analogt värde i rtdb.

### Med bakgrund

Det enklaste sättet att konfigurera en slider, är att lägga ett SliderBackground-objekt bakom slidern. Det enda man behöver göra är att koppla slider-objektet till en analog signal.

### Utan bakgrund

Om man absolut inte vill ha någon bakgrund till slidern måste man lägga in riktning, samt min och max-position för rörelse-området. En horisontal slider konfigureras på följande sätt (vertikal slider inom parentes).

Placera slidern på den lägsta y (x) koordinaten som slidern ska kunna röra sig över. Tänk på att origo är placerat i övre vänstra hörnet.

Mät upp min och max-värde för sliderns rörelse relativt sliderns övre (vänstra) sida genom att placera markören där. Markörens position kan avläsas i nedre högra hörnet. Mata in dess värden i objektseditorn för subgraf-objektet, attributen MaxPosition resp MinPosition. Ange riktning (Direction). Up innebär en vertikal slider med ökande värde uppåt. Right innebär en horisontal slider med ökande värde åt höger etc.

Attribut	Beskrivning
Attribute	Signal i rtdb av typen Float32 eller Int32 som ändras när slidern dras.
Access	Privilegier som krävs för att få ändra värdet på signalen.
Direction	Behöver endast anges om man inte har något SliderBackground-objekt. Riktning på slidern. Up innebär en vertikal slider med ökande värde uppåt. Right innebär en horisontal slider med ökande värde åt höger.
MaxValue	Värde på signalen som motsvarar sliderns max-position.
MinValue	Värde på signalen som motsvarar sliderns min-position.
MaxPosition	Behöver endast anges om man inte har något SliderBackground-objekt. x eller y koordinat för sliderns max-position (se ovan).
MinPosition	Behöver endast anges om man inte har något SliderBackground-objekt. x eller y koordinat för sliderns min-position (se ovan).

## 9 Grupper

En grupp är ett antal objekt som uppträder som ett enstaka objekt vid förflyttning, skalning, rotation mm. En grupp skapas genom att man väljer ut de objekt som ska tillhöra gruppen, och sedan aktiverar 'Functions/Group' i menyn. Om man vill lösa upp en grupp, väljer man ut gruppen och aktiverar 'Functions/Ungroup' i menyn.

Det finns vissa begränsningar på objekt som kan tillhöra grupper, kopplingar är t ex inte tillåtna i en grupp, och slider objekt kommer att förlora sin känslighet.

### Dynamik

En egenskap hos grupper är att man kan sätta dynamik på dem. Alla de typer av dynamik och aktion som finns för subgrafer är också tillgängliga för grupper.

De typer av dynamik som innebär en förändring av färg eller färg-ton, kommer att påverka all objekt i gruppen som inte har egen dynamik. För subgrafer som ska följa gruppens färgsättning bör man sätta No som dynamiktyp (xtt och applets fungerar här lite olika). Subgrafer och sub-grupper inom gruppen som har egen dynamik kommer att bestämma sina färger själva.

När man löser upp en grupp som man har satt någon typ av dynamik på, går data för dynamiken förlorad. För att förenkla i de situationer när man tillfälligt löser upp en grupp för att ändra någon komponent i gruppen, finns det en funktion som försöker återskapa dynamiken. När man löser upp gruppen sparas dynamik-data i en recall-buffer med grupp-namnet som nyckel. Vidare sparas gruppnamnet hos alla objekt som tillhörde gruppen. När man sedan grupperar objekten igen, letar man rätt på den grupp som de flesta objekt tillhört, och försöker hitta dynamiken för den gruppen i recall-bufferten. Om det hittas, läggs den in som dynamik för den nya gruppen. Det finns fall när återskapandet inte fungerar, därför bör man vara uppmärksam på att man verkligen har fått rätt dynamik efter en tillfällig upplösning av en grupp. Om det har gått snett, kan man normalt hitta dynamiken i recall-bufferten, som man hanterar med 'Functions/Recall' och 'Functions/Recall previous' i objektseditorn.



# 10 GIF och JPEG bilder

GIF- och JPEG-bilder går här under benämningen image.

## Skapa en image

Man skapar en image på samma sätt som en subgraf. GIF och JPEG-bilder finns under Images-mappen, och under Local/Images mappen. Vill man läsa in en speciell GIF eller JPEG-fil kopierar man den till \$pwrp\_pop. Filen dyker då upp under Local/Images mappen. Genom att välja ut den och klicka med MB2 i arbetsarean skapas image-objektet.

En image kan skalas och roteras i steg om 90 grader. Funktionerna för färgton, ljushet, intensitet och färgskiftning fungerar också.

## Dynamik

För att kunna lägga dynamik bakom en image måste man göra en subgraf av den först, eller låta image-objektet ingå i en grupp. Vill man ändra färg dynamisk ska man använda någon dynamiktyp som ändrar färgtonen, t ex DigTone. Ändring av fyllnadsfärg på en image ger ingen effekt.

## Transparans

Om man har en image på en symbol som inte är rektangulär, t ex en rund indikatorlampa, så vill man att området utanför indikatorn ska vara transparent.

GIF-formatet stöder transparens. Genom att välja transparens-färgen (R: 255, G: 0, B: 255) kommer transparensen även att fungera i Ge.

Java stöder tyvärr inte transparens för GIF-filer. Ge's image-klass lägger däremot in transparens i de pixel som har (R:255, G:255, B:255) dvs vitt. Detta kan dock leda till att högdagrar i bilden visas som transparenta.

# 11 Kopplingar

Subgrafs-objekt som innehåller kopplingspunkter kan kopplas ihop med kopplingar. Kopplingar ritas genom att man drar med MB2 från det ena objektet till det andra. Den kopplingspunkt som ligger närmast markören när man trycker ner resp släpper musen väljs.

Kopplingar karakteriseras av typ, linjebredd, 3D och färg.

## Typ

Typen väljs i menyn under 'Connections'.

### Straight

Koppling i form av en rak linje mellan kopplingspunkterna.

### StraightOneArrow

Koppling i form av en rak linje med en pil.

### Routed

Koppling i form av horisontella och vertikala linjer. Kopplingens väg och brytpunkternas placering beräknas så att kollision med övriga subgrafs-objekt undviks (de som ej har attributet NoConObstacle satt).

### Grafcet

Det finns ett antal kopplingar som används vid ritning av grafcet-sekvenser: StepDivergence, StepConvergence, TransDivergence och TransConvergence. Dessa kan användas tillsammans med subgraferna under mappen Grafcet i subgrafs-paletten om man vill visa en grafcet-sekvens i en bild.

### Runda hörn

På kopplingar av typen routed kan man välja att ha runda hörn, och även ange radien på rundningen. Detta väljs i menyn under 'Connections/Corners' och 'Connectins/CornerRoundAmount'.

## Färg

Färgen på kopplingen väljs som fyllnadsfärg, dvs med MB1 i färg-paletten.

## Kant

Border gör att svarta kantlinjer ritas på kopplingen.

## 3D

Vid 3D ritas en ljusare skugga på ovensidan och en mörkare på undersidan. Endast implementerat på kopplingar med typ Routed och runda hörn.

## Tjocklek

Tjockleken väljs i verktygspanelen pss som för linjer.

## **Förgreningar**

Om en koppling dras ut från en komponent och släpps i arbetsarean skapas en komponent av typen ConGlue i ändan på kopplingen. ConGlue har fyra kopplingspunkter och fungerar som antingen en avslutning, eller en förgrening av kopplingen, eller som en möjlighet att kunna påverka kopplingens väg. Genom att dra nya kopplingar från ConGlue komponentens kopplingspunkter skapas nya kopplingar, och ConGlue objektet anpassar sin form och färg efter de anslutna kopplingarna.

ConGlue finns även i subgrafs paletten under Other.

# 12 Editering

Ge innehåller en rad funktioner för att skapa objekt, modifiera dem, ordna dem mm. Funktionerna hittar man i verktygs-panelen, i menyn. En del aktiveras med mus-klick.

## Skapa objekt

Basobjekt skapas genom att man väljer en objekt-typ i verktygspanelen och sedan drar eller klickar i arbetsarean med MB1. Om man håller ner Shift tangenten samtidigt som objekts-typen väljs i verktygspanelen, kan man skapa flera objekt, utan välja ny objekts-typ. En mer ingående beskrivning av hur varje objekts-typ skapas finns i kapitlet Objekt.

En instans av en subgraf eller ett komplext objekt, skapas genom att man väljer ut subgrafen i subgrafspaletten och klickar i arbetsarean med MB2.

## Skapa kopplingar

Kopplingar skapas genom att man drar med MB2 mellan två subgrafs-objekt som innehåller kopplingspunkter.

Först ska man ha ställt in lämplig kopplings-typ under Connections i menyn.

## Återställ

Genom att klicka med MB3 återställer man de flesta funktioner eller avslutar funktioner: tömmer listan på utvalda objekt, avslutar ritning av en polylinje, avslutar skalning, avslutar editering av polylinje.

## Välja ut objekt

Ett objekt eller en koppling som är utvalt markeras genom att rödfärgas. Det här gäller dock inte när man har ändrat färg på de utvalda objekten, då de ritas med de nya färgerna för att man ska kunna se effekten av färgförändringen.

Objekt väljs ut på följande sätt

- Klicka på objektet med MB1. Tidigare utvalda objekt tas bort ur listan på utvalda objekt. Om objektet redan är utvalt töms listan på utvalda objekt.
- Klicka på objektet med Shift MB1. Objektet adderas till listan på utvalda objekt. Om objektet redan är utvalt tas det bort ur listan på utvalda objekt.
- Dra med MB1. En rektangel ritas ut och objekt som ligger innanför rektangeln med hela objekt-kroppen blir utvalda. Tidigare utvalda objekt tas bort ur listan på utvalda objekt. Man måste se till att markören inte ligger på något objekt efter som man även flyttar objekt med MB1. Om man har ett objekt som täcker bakgrunden får man använda Shift MB1 istället (se nedan).
- Dra med Shift MB1. Objekt som ligger innanför rektangeln adderas till listan på utvalda objekt.

- Select all objects under Edit i menyn väljer ut samtliga objekt.
- Select all connections väljer ut samtliga kopplingar.

Listan på utvalda objekt nollställs genom att klicka med MB1 på ett tomt område i arbetsarean, eller genom att klicka med MB3 i arbetsarean.

## **Flytta objekt**

Ett objekt flyttas med MB1. Vill man flytta på flera objekt samtidigt, väljer man ut objekten. Om man nu flyttar på ett av de utvalda objekt följer samtliga utvalda objekt med.

Om man vill flytta objekt vertikalt eller horisontalt väljer man först Move Restrictions under Edit i menyn. Move Restrictions återställs med MB3.

## **Cut, copy och paste**

Man kopierar objekt genom att välja ut de objekt som ska kopieras och kopiera dem till paste-bufferten med Ctrl+C. Med Ctrl+V kopieras de in i arbetsarean, och blir 'hängande' på markören (dvs följer markörens rörelser) tills man klickar med MB1 och låser fast dem. Klippa fungerar på samma sätt med Ctrl+X.

## **Grid**

Grid och gridstorlek kan väljas från verktygspanelen.

## **Skalning**

Skalnings-funktionen aktiveras från verktygspanelen och verkar på utvalda objekt. De utvalda objekten ramas in av en rektangel. Genom att dra i ett hörn eller en långsida skalas de utvalda objekten om. Om man drar i en långsida sker skalningen vertikalt eller horisontalt, drar man i ett hörn kan man skala objekten fritt. Genom att aktivera Functions/ScaleEqual före skalningen får man likformig skalning i x och y-led.

## **Rotera**

Utvalda objekt roteras 90 grader modsols genom att rotations-funktionen i verktygs-panelen aktiveras. 90-graders rotationen fungerar på samtliga objekt utom texter. Vill man rotera en annan vinkel kan man göra detta genom Rotate under Edit i menyn. Här kan man mata in en godtycklig vinkel. Av basobjekten är det dock endast linjer och polylinjer som klarar en godtycklig rotation. Det gäller med andra ord att bygga upp subgrafer och objekt av linjer och polylinjer om de ska kunna roteras godtyckligt.

## **Ordna objekt**

### **Vertikalt**

Objekt som är placerade vertikalt kan vänsterjusteras, högerjusteras resp centreras med Align vertical under Functions i menyn. Välj ut de objekt som ska justeras och aktivera

meny-entryt.

## **Horisontalt**

Objekt som är placerade horisontalt kan justeras med avseende på översida, undersida resp centrum med Align horizontal under Functions i menyn. Välj ut de objekt som ska justeras och aktivera meny-entryt.

## **Koncentriska cirklar**

Koncentriska cirklar åstadkommer man genom att först aktivera Align vertical center och sedan Align horizontal center.

## **Lika avstånd mellan objekt**

Objekt som är placerade vertikalt eller horisontalt ska ofta ha lika avstånd mellan dem. Detta åstadkommer man med Equidistance under Functions i menyn. Equidistance funktionen för horisontella objekt, behåller positionen på objektet längst till höger och längst till vänster, men placerar mellanliggande objekt så att avståndet mellan objekten blir lika. Avståndet kan räknas från objektens högerkant, vänsterkant eller centrum. På motsvarande sätt kan vertikala objekt ordnas med 'Equidistance/vertical'.

## **Under och över**

Om objekt överlappar kan man flytta dem över eller under med Pop och Push under Functions i menyn. Pop lägger ett objekt över samtliga andra objekt i bilden, push lägger objektet under alla andra objekt. Vill man lägga ett objekt mellan två andra objekt krävs det lite tankearbete för att göra pop eller push i rätt ordning.

## **Färger**

### **Bakgrundsfärg**

Bakgrundsfärgen i bilden sätts genom att välja ut en lämplig fyllnadsfärg i färgpaletten (MB1) och aktivera Set background color under Functions i menyn.

### **Färger på objekt**

Färger på objekt delas upp i fyllnadsfärg och kantfärg. Den fyllda delen av objektet ritas med fyllnadsfärgen och kanten ritas med kantfärgen. En del objekt t ex linjer ritas enbart med kantfärg. Färgerna väljs i färgpaletten, fyllnadsfärg väljs med MB1 och kantfärg med MB2. De senast valda färgerna visas i två rektanglar överst i färgpaletten, fyllnadsfärg till vänster och kantfärg till höger. När ett basobjekt eller sammansatt objekt skapas hämtas de senast valda färgerna från paletten och objektet ges dessa färger. Vill man ändra på färger på existerande objekt väljer man ut dem och väljer en ny fyllnads- eller kant-färg i paletten. Observera att objekten efter denna operation ritas med de aktuella färgerna trots att de är utvalda, dvs de förlorar sin rödmarkering. Det är därför bra klicka med MB3 efter operationen för att nollställa utvalslistan.

### **Färger på subgrafs-objekt**

Man kan ändra färg på subgrafs-objekt på samma sätt som man ändrar färg på basobjekt, genom att välja ut dem och ge dem en kant- eller fyllnadsfärg. Objekt som är ritade med skuggningar och 3D-effekter förlorar dessa om man ändrar fyllnadsfärgen. Därför finns speciella

funktioner för att ändra färg på subgraf-objekt, i termer av ljushet, ton, intensitet och skiftning. I kapitlet Subgrafer behandlas detta mer ingående.

## Koppling till rtdb

När ett subgrafs-objekt ska påverka eller påverkas av en signal i rtdb måste man ange detta i subgrafens attribut. Ett sätt är att mata in det för hand i objektseditorn för subgraf. Ett annat betydligt smidigare är att använda planthierarkin som kan hittas ovanför navigationsfönstret. Planthierarkin finns med om man har startat Ge från navigatören. I planthierarkin väljer man ut det objekt man vill koppla subgrafs-objektet till, och klickar sedan på subgrafs-objektet med Ctrl/DubbelKlick MB1. Finns det två rtdb-kopplingar i subgrafs-objektet kopplas den första på detta sätt och den andra kopplingen med Ctrl/Shift/DubbelKlick MB1. Man kan även öppna objektseditorn och klicka på raden för attributet med Ctrl/Dubbelklick MB1 för att lägga in signalen. Är det en signal man kopplar till räcker det att välja ut objektet i planthierarkin, då väljs attributet ActualValue automatisk. Om det är ett annat typ av objekt måste man ange vilket attribut man ska koppla subgraf-objektet till. Det gör man genom att öppna objektet med DubbelKlick MB1, och sedan välja ut lämpligt attribut.

Man kan även kopiera in signalen från navigatören.

## Suffix

Till namnet på signalen ska adderas ett suffix som talar om vilken typ attributet är. Dynamik där det är uppenbart vilken typ attributet har är detta inte nödvändigt, t ex varianter av Dig opererar ju enbart på Boolean. Följande typer av suffix kan anges:

```
##BOOLEAN
##FLOAT32
##FLOAT64
##CHAR
##INT8
##INT16
##INT32
##UINT8
##UINT16
##UINT32
##OBJID
##STRING
##TIME
##DELTATIME
##ATTRREF
```

Är attributet en sträng eller en vektor ska dessutom storleken anges pss som i GMS.

## Graf-attribut

Under File i menyn finns Graph attributes och här kan man mata in attribut för bilden.

Här anges bl a vilken del av arbetsarean som ska visas i runtime. Koordinaterna för övre vänstra hörnet och nedre högra hörnet mäts upp mha markören och matas in i x0, y0 resp x1, y1.

Attribut	Beskrivning
----------	-------------

subgraph	Anger att grafen är en subgraf. I så fall är inte nedanstående attribut aktuella.
x0	x-koordinat för övre vänstra hörnet på bilden.
y0	y-koordinat för övre vänstra hörnet på bilden.
x1	x-koordinat för nedre högra hörnet på bilden.
y1	y-koordinat för nedre högra hörnet på bilden.
Scantime	Cykeltid i sekunder för uppdateringar av objekt som går med den långsamma cykeln.
FastScantime	Cykeltid i sekunder för uppdateringar av objekt som går med den snabba cykeln.
AnimationScantime	Cykeltid i sekunder för animeringar i bilden.
JavaWidth	Bredd i pixel som bilden skalas till vid exportering till java.
IsJavaApplet	Bilden exporteras som en java applet vid exportering.
IsJavaFrame	Bilden exporteras som en java frame vid exportering.
BackgroundImage	Namn på gif eller jpg-fil för bakgrundsbild.
BackgroundTiled	Bakgrundsbilden skalas ej utan upprepas så att bakgrunden täcks.
DoubleBuffered	Grafen ska dubbelbuffras, vilket innebär att bilden först ritas i en pixmap i minnet och sedan kopieras till skärmen. Bör användas om bilden innehåller rörliga objekt eller animationer.
MB3Action	Anger vad som ska hända vid högerklick i grafet.
Translate	All dynamisk och statisk text översätts.
BitmapFonts	Bitmapsfonter används i stället för skalbara fonter.
HotIndication	Hur känsliga objekt markeras när makören pekar på dem.
TooltipTextsize	Storlek på text i tooltip.
AppMotion	Vad som ska ske vid drag i en Android app.

## Objekts och hierarki grafer

Det här kapitlet beskriver hur man kan göra ge-bilder som visar innehållet objekt av en viss klass, och som kan visa alla objekt av den klassen. Det beskriver också hur man kan göra gemensamma bilder för likartade hierakier i ProviewR-databasen.

## Objekts graf

För en del klasser som tillhör ProviewR's basklasser finns det sk objekts-bilder, dvs bilder som kan öppnas för varje objekt som tillhör klassen. Objekts-bilder finns bl a för klasserna PID, Mode, Av, Ai, Ao, Di, Do, Dv, och öppnas i xtt med kommandot 'open graph /instance=' där man i /instance skickar med namnet på objektet. Man kan också öppna objektbilden genom att välja ut objektet och aktivera Functions/Open object graph i xtt menyn.

Om man har gjort en egen klass, kan man göra en objektsbild även för denna klass. Man editerar bilden som en normal ge-graph. Skillnaden kommer när man ska knyta de olika dynamiska objekten till ProviewR-objektet i rtdb. Där man normalt skriver ett objekts-namn skriver man istället \$object. Data för en indikator som kopplas till attributet ActualValue i ProviewR-objektet kan se ut så här

Attribut	Värde
SubGraph	pwr_indsquare
LowColor.Attribute	\$object.ActualValue##Boolean
LowColor.Color	Inherit
Cycle	Inherit
DynType	Inherit
Action	Inherit

Grafen bör sparas med samma namn som klassen, med små bokstäver, om man vill kunna öppna den



med Functions/Open object graph i xtt-menyn.

Kommandot för att öppna objektbilden för objektet H1-H2-MyObject av klassen MyClass blir

```
open graph /classgraph /instance=H1-H2-MyObject
```

## Hierarki graf

Ofta finns det anläggningsdelar i ett system som är identiska och som ger upphov till närmast identiska hierarkier. Det kan t ex finnas 30 likadana frekvensomformare, och det finns i Ge möjlighet att, istället för att editera 30 likadana grafer, göra en generisk som klarar alla frekvensomformarna. Tillvägagångssättet bygger på samma princip som objekts-graferna ovan. Den hierarki som skickas med i /instance i open graph kommandot när grafen öppnas, kommer att ersätta alla \$object som förekommer i dynamiska kopplingar i bilden.

Data för en indikator som kopplas till Dv'n Start kan t ex se ut så här

Attribut	Värde
SubGraph	pwr_indsquare
LowColor.Attribute	\$object-Start.ActualValue##Boolean
LowColor.Color	Inherit
Cycle	Inherit
DynType	Inherit
Action	Inherit

När bilden öppnas med kommandot

```
open graph my_fo_graph /instance=H1-H2-Fo1.53
```

kommer indikatorn ovan att kopplas till attributet H1-H2-Fo1-Start.ActualValue.

Även i kommandon som exekveras av tryckknappar i bilden, kommer strängen \$object att ersättas av hierarki-namnet. Detta gör att man från tryckknappar kan öppna objekts-bilder eller trend-kurvor på objekt inom hierakin. Kommandot för att öppna objekts-bilden för ett Mode-objekt kan t ex bli

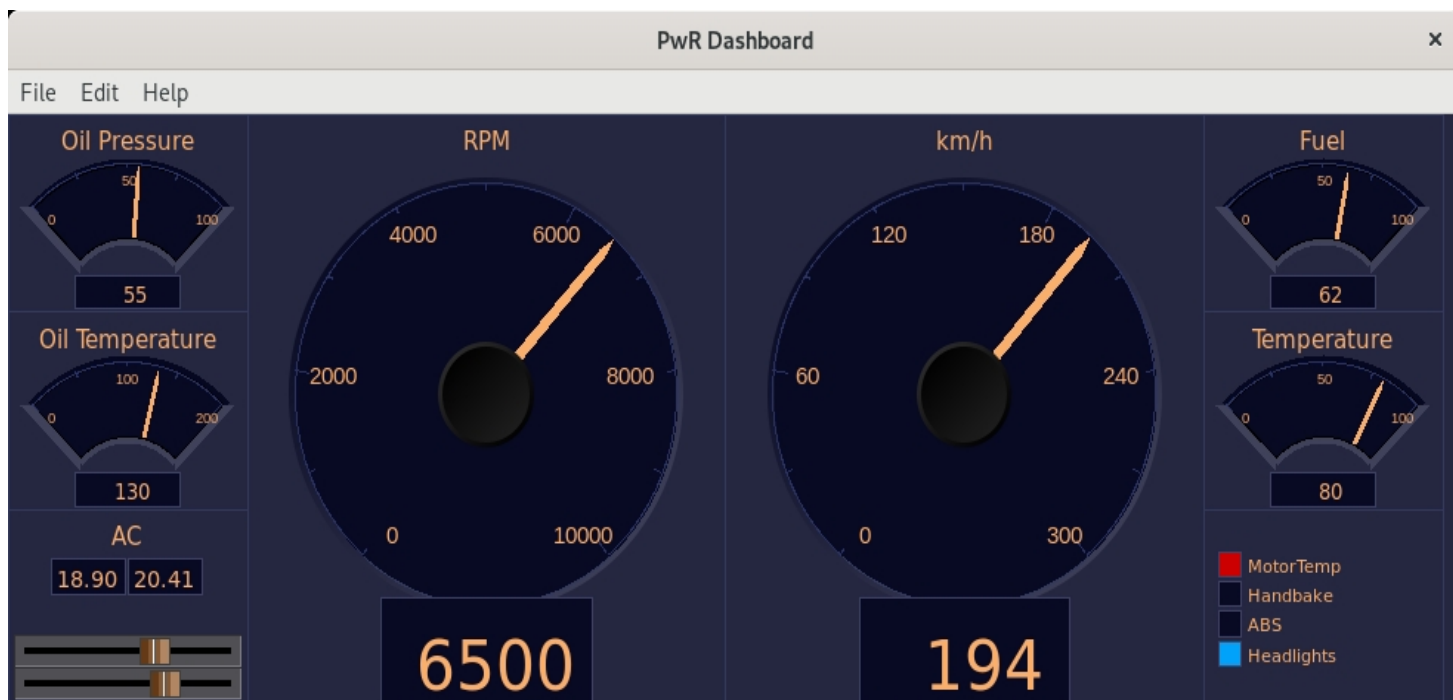
```
open graph /classgraph /instance=$object-Plc-W-Mode0/name=$object
```

Exempel på kommando för att öppna en trendkurvor från en plotgrupp inom hierakin

```
open trend $object-Plot
```

# 13 Dashboard

Ett dashboard (instrumentbräda) är en förenklad graf med ett mycket begränsat antal byggstenar. Byggstenarna kallas för celler och kan visa värden i form av indikatorer, staplar, trendkurvor, givare etc.



Ett dashboard kan skapas in Ge editorn eller i operatörmiljön. Byggstenarna och hur man skapar ett dashboard i operatörmiljön är beskrivet i Operatörshandboken.

I Ge editorn skapar man ett dashboard på följande vis.

- Open Graph Attributes and set Dashboard to 1.
- Open Graph Attributes again and set the size of the dashboard in DashRows and DashColumns.
- Dash cells are drawn with colortheme colors, thus select a colortheme from File/Colortheme/Select in the menu.
- Set the background color to the first color in the custom color palette.
- If the dashboard should follow the color theme selected by the operator, set ColorTheme in Graph Attributes to \$default.
- Create a DashCell from Other/DashCell in the subgraph palette.
- Double click on the cell to set properties and size, and connect the cell to a signal.
- Save and build the dashboard.

Ett dashboard sparas i en pwrp-fil på \$pwrp\_pop, som kopieras till \$pwrp\_exe när den byggs. Filen ska distribueras till operatörstationen.

Ett dashboard kan öppnas från Functions/Dashboard/Open i Xtt-menyn, eller med kommandot 'open dashboard'.

# 14 Testning

## Preview

Det finns möjlighet att testa bilden direkt i editorn. Förutsättningen är att man har startat näthanteraren på sin utvecklingsmaskin och att den har kontakt med de objekt som förekommer i bilden. Med View/Preview i menyn kopplar editorn upp sig mot rtdb och börjar exekvera bildens dynamik. Preview avslutas med View/Preview close. Därefter kan man fortsätta och editera bilden.

## Xtt

Genom att starta xtt i utvecklings-miljön eller på en operatörs/process-nod, kan man också enkelt testa bilden. Det enda som krävs är att .pwg filen för bilden är tillgänglig på \$pwrp\_exe. Man öppnar en ge-graf med kommandot

```
xtt> open graph 'grafnamn' [/width=][/height=][/scrollbar][/navigator]
```

Med kommandot

```
xtt> show graph
```

listas tillgängliga grafer.

# 15 Utvecklingsmiljö

Här beskrivs lite om filkataloger och filer som används av ge, samt vilka filer som genereras.

## Filtyper

Ge läser in bild-filer och sparar bild-filer på filkatalogen pwrp\_pop.

En graf sparas som en fil av typen .pwg. En subgraf sparas också som en .pwg-fil, som används när man ska editera subgrafen, men även som en .pws-g-fil. Det är .pws-g-filen som laddas in när man skapar objekt av subgrafen.

Exporterar man bilden som en java applikation skapas en .java -fil men det namn som man angivit som klassnamn.

Exporterar man bilden som en java-applet skapas en .java-fil med namnet som angivits som klassnamn med suffixet \_A, t ex Oversikt\_A.java. Dessutom skapas ett html-fil med klassnamnet som namn.

## Setup-filer

Ge använder två setup-filer, en för färg-paletten och en för subgrafs-paletten.

### Färgpalett

Om man vill modifiera färgerna i färg-paletten är detta möjligt genom att skapa en fil med namnet ge\_colors.dat.

Denna ska innehålla rgb-kod för färgerna i paletten. Det finns en mall på ...

### Subgraf-palett

Subgrafpaletten innehåller en uppsättning fasta menyer och subgrafer samt mappen Local som läser in subgraferna på pwrp\_pop. De fasta menyerna och subgraferna konfigureras i filen pwr\_ge\_setup.dat. Denna fil är generell för samtliga projekt men kan modifieras av systemansvarig.

# 16 Kommandon

Ge innehåller en kommandorad som öppnas från menyn eller med ctrl/b. Kommandona är i första hand avsedda att användas i ge-script, men kan även utföras från kommandoraden. Från kommandoraden körs även ge-script genom att skriva filnamnet med ett '@' framför. Observera att omställning av funktioner i editorn som görs via kommandon inte 'syns' i options-menyer och check-boxar. När ett script har exekverats kan inställningen vara annorlunda än den som visas.

# 17 Script

Scripthanteringen i ge finns beskriven i Ge script Användarhandledning. Här ges några enkla exempel på vad script kan användas till.

## 17.1 Rita en graf med script i Xtt

Draw a graph with script in Xtt

Förutom att rita en graf i Ge editorn, är det också möjligt att skriva ett script som ritar grafen. Scriptet kan anropa script-funktionerna i Ge för att rita grafiska element in grafen, men kan även anropa script-funktionerna i Xtt. Det gör det möjligt att få information från databasen och anpassa bilden efter det aktuella tillståndet.

Script-grafen öppnas som en ordinare graf, men graf-namnet ersätts av script-namnet med ett '@' tecken framför, t ex 'open graph @myscript'. Scripten hämtas från \$pwr\_exe eller \$pwrp\_exe om inte någon annan katalog anges i namnet, t ex 'open graph @"\$pwrp\_login/myscript".

### Exempel

Ett enkelt script som ritar en rektangel och en tryckknapp.

```
main()
{
    int id;
    float x1;
    float y1;
    float x2;
    float y2;
    float width;
    float height;

    SetDraw(0);
    SetBackgroundColor(eDrawType_Color66);

    # Draw rectangle
    x1 = 1;
    y1 = 1;
    width = 18;
    height = 5;
    id = CreateRectangle(x1, y1, width, height);
    SetObjectFillColor(id, eDrawType_Color74);
    SetObjectFill(id, 1);
    SetObjectBorder(id, 0);

    # Draw pushbutton
    x1 = 7.5;
    y1 = 2;
    x2 = 10.5;
```

```

y2 = 3.5;
id = CreateObject("pwr_buttontogglecenter", x1, y1, x2, y2);
SetObjectAttribute(id, "Text", "Toggle");
SetObjectAttribute(id, "ToggleDig.Attribute", "H1-Dv1.ActualValue##Boolean");

# Set graph size
SetGraphAttribute("x1", 20.0);
SetGraphAttribute("y1", 7.0);

SetDraw(1);
endmain

```

## 17.2 Modifiera grafer i editorn

Ett script exekveras från kommando-raden i Ge genom att sätta ett '@' före filnamnet, t ex

```
ge> @my_script
```

Kommandoraden öppnas med Functions/Command (Ctrl+B).

### Exempel

Nästa exempel visar hur man kan byta ut en subgraf i flera bilder. En subgraf är ju normalt intern, och en ändring i subgrafet slår inte igenom förrän man har genomfört proceduren att

- sätta subgrafen extern
- spara bilden
- läsa in bilden, nu med den ändrade subgrafen
- sätta subgrafen intern igen
- spara bilden.

Ändrar man ofta i sina subgrafer kan man med fördel skriva ett script som byter ut en subgraf i alla grafer i projektet.

```

!
! Replace a subgraph
!
function int process( string graph)
    int sts;
    printf( "Processing graph %s\n", graph);
    open 'graph'
    sts = SetExtern("my_subgraph");
    if ( sts)
        printf( "Changing my_subgraph\n");
        save
        open 'graph'
        SetIntern( "my_subgraph");
        save
    endif
endfunction

main()
    process( "nu4_alla_platar");
    process( "nu4_status_trp");
    process( "nu4_trp_hyl");

```

```
process( "nu4_trp_rb_ut" );  
process( "nu4_buffhog" );  
process( "nu4_status_ugn" );  
process( "nu4_trp_lul" );  
process( "nu4_trp_start" );  
process( "nu4_inlagg" );  
process( "nu4_trend" );  
process( "nu4_trp_rbl" );  
process( "nu4_ugn_start" );  
endmain
```



# 18 Web bilder

## Exportera som java-applet

Om en bild ska kunna visas i en web-bläddrare, exporterar man den som en java-applet. Först måste se till att attributet `IsJavaApplet` i `File/Graph` attributes är satt. Därefter aktiverar man `File/Export Java`, som gör att bilden exporteras som en java applet och kompileras. Dessutom skapas en html-fil för att öppna appleten. Html-filen tillsammans med ett java-arkiv med applet-klassen läggs på katalogen `$pwrp_web`. Genom att kopiera filerna på `$pwrp_web` till någon lämplig katalog under webservern (`$pwrp_websrv`), kan man öppna dem med web-bläddraren.

## Begränsningar

Java versionen av en Ge -bild skiljer sig från utseendet och funktionen i Ge och Xtt på ett antal punkter.

- Format-satserna för visning av flyttal, heltal och strängar ska vara av typen `%n.mf`, `%-n.mf`, `%d`, `%nd`, `%s` där `n` och `m` är siffror.
- Bredden på kantlinjerna i en subgraph skalas pss som subgrafen i övrigt.
- Vid förändring av ett dynamisk objekt ritas inte bakgrunden eller täckande objekt om. Dvs dynamiska objekt bör inte täckas av andra objekt, och texter som kan förändras måste ha en bakgrund i subgrafen.
- Javabilderna är inte skalbara. Skalfaktorn bestäms av attributet '`JavaWidth`'.
- Subgrafer får inte innehålla grupper. Detta ger kompileringsfel.
- Basobjekt ritas alltid under komponenter. Läger men t ex en Text ovanpå en komponent kommer inte texten att synas.

## Konfigurera en web-site

För att kunna öppna de olika process-bilderna i ett system är det lämpligt att ha en web-sida med en meny till väster och aktuell bild till höger. Med lite konfigurering kan man generera en sådan mha `Generate web` i menyn.

Konfigureringen går till så här:

Skapa ett `WebHandler` objekt under `node`-objektet i nod-hierarkin. `Webhandler` objektet gör att en server-process om förser bilderna med dynamisk information från realtidsdatabasen startas.

I objektet anger man bl a också om man ska ha möjlighet att logga in som `ProviewR`-användare. Under `WebHandler`-objektet lägger man ett `WebGraph`-objekt för varje bild som ska kunna öppnas från menyn.

Under `WebHandler`-objektet kan man även lägga `WebLink`-objekt. Varje `WebLink`-objekt ger upphov till ett meny-entry som kan kopplas till en URL, t ex dokument för arbetsinstruktioner, funktions-specifikationer o dyl.

Alla filer som krävs för web-siten samlas på filkatalogen `$pwrp_web`. Genom att kopiera dessa, tillsammans med jar-filerna `$pwr_lib/pwr_rt_client.jar` och `$pwr_lib/pwr_jop.jar`, till lämplig filkatalog under webservern på operatörsnoden blir de tillgängliga på nätet.

# 19 Runtime

När bilden är klar ska den visas på en operatörs-station, en process-station eller kanske via intranätet.

## Underhålls-bilder i xtt

Bilder av underhållsskärmar kan tas upp utan konfiguration från kommando-raden i xtt, vilket beskrivs ovan i Test avsnittet. Detta fungerar bra så länge bara man själv och andra invigda ska ha tillgång till bilderna. Ska en bredare krets kunna titta på dem kan man bygga upp ett meny-träd i xtt. Detta görs genom att lägga in kommandon i xtt's startup fil. Startup-filen `~/xtt_setup.rtt_com` exekveras varje gång man startar xtt. Här kan man definiera symboler men även lägga in kommandon för att skapa menyer. Här följer ett exempel på en symbol-fil som skapar en enkel underhålls-menü.

```
create item/text="Underhåll" /menu/dest=DataBase/before
create item/text="Översikt" /command="open graph oversikt" /menu
                        /dest=Underhåll /firstchild
create item/text="Bläster" /command="open graph blaster" /menu
                        /dest=Underhåll /firstchild
create item/text="Rullbana" /command="open graph rullbana" /menu
                        /dest=Underhåll /firstchild
```

## Operatörs-bilder i xtt

Ska bilderna vara operatörbilder bygger man upp en operatörs-plats men `OpPlace`-objekt, `User`-objekt och `XttGraph`-objekt. Xtt startas med namnet på `OpPlace`-objektet som argument.

## Java applikationer i xtt

En `ge`-bild kan exporteras som en java-applikation. Denna kan tas upp i xtt med kommandon `'open jgraph'`.

## Web-bilder

`Ge` grafer kan även visas i en webbläsare med weboperatörsmiljön. Hur denna konfigureras beskrivs i Konstruktörshanboken, kapitel Web operatörsmiljö. Webgränssnittet har dock begränsad funktionalitet för `Ge` grafer

- Dynamiktyperna `DigScript`, `Script`, `EmitSignal`, `CatchSignal` och `PopupMenu` är inte implementerade.
- Endast ett fåtal Xtt kommandon, använda i `Command` och `DigCommand`, är implementerade. Här är en lista på implementerade kommandon
  - `open graph`
  - `open url`

open trend  
open fast  
set subwindow  
help  
check  
call

# 20 Ge script och kommandon

## 20.1 Introduktion till script

I GE finns en uppsättning kommandon för att skapa grafiska element som rektanglar, texter, polylinjer och subgrafer. Dessa kommandon kan utföras direkt från kommandoraden i GE, men det vanligaste är att skriva dem i script-filer som sedan exekveras från kommandoraden.

Scriptfilerna kan dessutom innehålla ett c-liknande språk som innehåller operatorer för beräkningar, villkors-satser och for-loopar mm. Dessutom finns rutiner för att hämta information i utvecklings-databasen. Se manualen för pwr\_cmd för en utförligare beskrivning.

Ge-script kan t ex användas för att konvertera bilder från andra bildhanterings-system till ge-bilder, eller för att automatgenerera formulär eller bilder utifrån objekt i databasen.

### Skapa objekt

Basobjekt som rektanglar, linjer mm skapas med create-kommandot. Innan objektet skapas ställer man in editorn med egenskaper som objektet ska ha (fill, border, fyllnadsfärg, textstorlek etc), sedan skapar man objektet.

#### Rektangel

Exempel på en fylld rektangel

```
set fill
set linewidth 1
set bordercolor Black
set fillcolor BlueHigh7
create rect /x1=10 /y1=1 /width=4 /height=2
```

#### Cirkel

Exempel på en halvcirkel

```
set nofill
set linewidth 2
set bordercolor Black
create arc /x1=1 /y1=1 /x2=3 /y2=2 /angle1=0 /angle2=180
```

#### Linje

Exempel på en linje

```
set linewidth 1
set bordercolor Black
create line /x1=0 /y1=0 /x2=10 /y2=0
```

## Polylinje

För en polylinje skapar man den första del-linjen med 'create', och resterande dellinjer med 'add'.

### Exempel på polylinje

```
set nofill
set linewidth 1
set bordercolor Black
create polyline /x1=5 /y1=5 /x2=6 /y2=6
add polyline /x1=5 /y1=7
add polyline /x1= 6/y1=8
add polyline /x1=5 /y1=9
add polyline /x1= 6 /y1=10
```

## Text

### Exempel

```
set textsize 14
set textfont LucidaSans
set textcolor RedHigh8
set bold
create text/text="Example" /x1=3 /y1=5
```

## Subgraf objekt

Vid skapande av subgrafobjekt är gången lite annorlunda. Här skapar man först objektet, och sedan sätter man attributen på objektet.

### Exempel på ventil

```
create object /sub=pwr_valve /x1=1 /y1= 1 /x2=3 /y2=2
set current fillcolor YellowGreenMedium4
set current attr DigLowColor.Attribute "Rt-Dv1.ActualValue##Boolean"
set current attr DigLowColor.Color GrayHigh8
```

### Exempel på tryckknapp

```
create object /sub=pwr_buttonset /x1=5 /y1=1
! Change type to SetDigConfirmWithTone
set current attr Dyntype1 DigLowColor
set current attr Action Inherit|Confirm
set current attr annotation "Start"
set current attr Confirm.Text "Do you really want to..."
set current attr SetDig.Attribute "rt-Dv1.ActualValue##Boolean"
set current attr DigLowColor.Attribute "rt-Dv2.ActualValue##Boolean"
set current attr DigLowColor.Color YellowGreenMedium4
set current attr Access System|Operator1
```

## Graf attribut

Innan man sparar bör man sätta upp gränser i x- och y-led för grafen och diverse andra attribut. Detta görs med kommandot 'set graphattributes'.

## Exempel

```
set graph x0 -3
set graph y0 -3
set graph x1 40
set graph y1 37
set graph AnimationScanTime 0.2
set graph BackgroundImage "corrado.gif"
```

## Subgraf attribut

Om grafen ska sparas som subgraf använda man också 'set graphattributes', men sätter först attributet 'subgraf' och därefter de attribut som hör till subgrafen.

## Exempel

```
set graph subgraph 1
set graphattr Action ToggleDig
set graphattr DynType1 DigLowColor
set graphattr Color1 YellowGreenMedium4
set graphattr NoConObstacle 1
```

## 20.2 Kommandon

Nedan följer beskrivning på kommandon i Ge

add	add polyline
create	create rectangle create arc create line create polyline create rectangle create text create subgraph
exit	
filter navigator	
group	
move	move currentobject move selectedobject
new	
open	
quit	
replace	
rotate	rotate currentobject rotate selectedobject
save	
scale	scale currentobject scale selectedobject
search object	

select

select currentobject  
select clear

set

set verify  
set noverify  
set fill  
set nofill  
set border  
set noborder  
set shadow  
set noshadow  
set grid  
set nogrid  
set linewidth  
set gridsize  
set textsize  
set textfont  
set bold  
set nobold  
set backgroundcolor  
set fillcolor  
set bordercolor  
set textcolor  
set currentobject fillcolor  
set currentobject colortone  
set currentobject colorlightness  
set currentobject colorintensity  
set currentobject colorshift  
set currentobject gradient  
set currentobject attributes  
set currentobject annotation  
set graphattributes

show

show version

## 20.2.1 add



## 20.2.2 add polyline

Adderar en punkt till senast skapade polyline.

En polyline skapas med 'create polyline' som också skapar den första del-linjen. Övriga del-linjer skapas med 'add polyline'.

### Syntax

```
ge> add polyline /x1= /y1=
```

/x1                      x-koordinat för den adderade del-linjen.

/y1                      y-koordinat för den adderade del-linjen.

## 20.2.3 create

Skapa grafiska element i bilden.

### 20.2.4 create rectangle

Skapar en rektangel.

#### Syntax

```
ge> create rectangel /x1= /y1= /width= /height=
```

/x1	x-koordinat för övre vänstra hörnet.
/y1	y-koordinat för övre vänstra hörnet.
/width	Rektangelns bredd.
/height	Rektangelns höjd.

### 20.2.5 create arc

Skapa en cikel eller oval, eller ett segment av en cirkel eller oval.

#### Syntax

```
ge> create arc /x1= /y1= /x2= /y2= /angle1= /angle2=
```

/x1	x-koordinat för övre vänstra hörnet för den rektangel som omsluter ovalen.
/y1	y-koordinat för övre vänstra hörnet för den rektangel som omsluter ovalen.
/x2	x-koordinat för nedre högra hörnet för den rektangel som omsluter ovalen.
/y2	y-koordinat för nedre högra hörnet för den rektangel som omsluter ovalen.
/angle1	Vinkel som beskriver var ett cirkel-segment startar. Anges i grader. Default-värde 0.
/angle2	Vinkel i grader som anger storleken på ett cirkel-segment. Default-värde 360 grader.

### 20.2.6 create line

Skapa en rät linje mellan två punkter.

#### Syntax

```
ge> create line /x1= /y1= /x2= /y2=
```

/x1	x-koordinat för den första änd-punkten.
/y1	y-koordinat för den första änd-punkten.
/x2	x-koordinat för den andra änd-punkten.
/y2	y-koordinat för den andra ändpunkten.

### 20.2.7 create polyline

Skapa en polylinje, dvs en linje som består av flera rätta del-linjer.

'create polyline' skapar den första del-linjen. Övriga del-linjer skapas med 'add polyline'. Punkten (x1, y1) är startpun

## Syntax

```
ge> create polyline /x1= /y1= /x2= /y2=
ge> add polyline /x1= /y1=
ge> add polyline /x1= /y1=
ge> ...
```

/x1	x-koordinat för start-punkten.
/y1	y-koordinat för start-punkten.
/x2	x-koordinat för första del-linjens änd-punkt.
/y2	y-koordinat för första del-linjens änd-punkt.

### 20.2.8 create text

Skapa en text.

## Syntax

```
ge> create text /x1= /y1= /text=
```

/x1	x-koordinat.
/y1	y-koordinat.
/text	Text omgärdad av dubbel-fnuttar.

### 20.2.9 create subgraph

Skapar ett subgraf-objekt.

## Syntax

```
ge> create subgraph /x1= /y1= /subgraph= [/x2= /y2=]
```

/x1	x-koordinat för övre vänstra hörnet.
/y1	y-koordinat för övre vänstra hörnet.
/subgraph	Namnet på subgrafen. Om subgrafet tillhör ProviewR's bassystem har subgrafen alltid prefixet 'pwr_' och skrivs med gemener, dvs subgrafen 'Valve' har namnet pwr_valve.
/x2	Om punkten (x2, y2) anges skalas subgrafen så att den innesluts i rektangeln med hörnpunkterna (x1,y1) och (x2, y2).
/y2	

### 20.2.10 exit

Avsluta ge och spara aktuell graf först.

## Syntax

```
ge> exit ['filename']
```

### 20.2.11 filter navigator

Filtrera objekt i objektsnavigatorn.

Filtrering kan göras på objektsnamn eller subgraf med wildcard.

## Syntax

```
ge> filter navigator /type= /pattern=  
ge> filter navigator /reset
```

/reset	Återställer filtreringen och visar alla objekt.
/type	'name' eller 'class'. Name filtrerar på objektsnamn och class på subgraf.
/pattern	Filter mönster. Kan innehålla wildcard (*).

### **Exempel**

```
ge> filter navigator /type=class /pattern=*button*
```

## 20.2.12 group

Skapa en grupp av utvalda objekt.

## 20.2.13 move

### 20.2.14 move currentobject

Flyttar senast skapade objekt.

#### Syntax

```
ge> move currentobject /x= /y=  
ge> move currentobject /absx= /absy=
```

/x	Förflyttning i x-led relativt tidigare position.
/y	Förflyttning i y-led relativt tidigare position.
/absx	Förflyttning i x-led till absolut position.
/absy	Förflyttning i y-led till absolut position.

### 20.2.15 move selectedobject

Flyttar utvalt objekt.

#### Syntax

```
ge> move selectedobject /x= /y=  
ge> move selectedobject /absx= /absy=
```

/x	Förflyttning i x-led relativt tidigare position.
/y	Förflyttning i y-led relativt tidigare position.
/absx	Förflyttning i x-led till absolut position.
/absy	Förflyttning i y-led till absolut position.

## 20.2.16 new

Rensar arbetsarean.

### 20.2.17 open

Öppna en graf.

#### Syntax

```
ge> open ['filename']
```

### 20.2.18 quit

Avsluta utan att spara.

#### Syntax

```
ge> quit
```

### 20.2.19 replace attribute

Utbyte av sträng i Dynamic och Action för utvalda object.

#### Syntax

```
ge> replace attribute /from= /to= [/strict]
```

/from	Sträng som ska bytas ut.
/to	Ersättnings sträng.
/strict	Sökning efter 'from' sträng skiljer mellan stora och små bokstäver.

### 20.2.20 rotate

### 20.2.21 rotate currentobject

Roterar senast skapade objekt runt objektets mittpunkt.

#### Syntax

```
ge> rotate currentobject /angle=
```

/angle                      Vinkel i grader som objektet ska roteras.

### 20.2.22 rotate selectedobject

Roterar utvalt objekt runt objektets mittpunkt.

#### Syntax

```
ge> rotate selectedobject /angle=
```

/angle                      Vinkel i grader som objektet ska roteras.

### 20.2.23 **save**

Sparar en graf.

#### **Syntax**

```
ge> save ['filename']
```

### 20.2.24 **scale**

### 20.2.25 **scale currentobject**

Skalar om senast skapade objekt.

#### **Syntax**

```
ge> scale currentobject /scalex= /scaley= [/x= /y=]
```

/scalex	Skalfaktor i x-led
/scaley	Skalfaktor i y-led.
/x	x-koordinat för skalningens referenspunkt.
/y	y-koordinat för skalningens referenspunkt.

### 20.2.26 **scale selectedobject**

Skalar om utvalt objekt.

#### **Syntax**

```
ge> scale selectedobject /scalex= /scaley= [/x= /y=]
```

/scalex	Skalfaktor i x-led
/scaley	Skalfaktor i y-led.
/x	x-koordinat för skalningens referenspunkt.
/y	y-koordinat för skalningens referenspunkt.

### 20.2.27 **search object**

Sök efter ett objekt.  
När sökningen är gjord väljs det hittade objektet ut.

#### **Syntax**

```
ge> search object /name=
```

/name	Objektsnamn.
-------	--------------



**20.2.28 select**

## 20.2.29 **select currentobject**

Lägger in senast skapade objekt i listan av utvalda objekt.

## 20.2.30 **select clear**

Tömmer listan på utvalda objekt.

**20.2.31 set**

## 20.2.32 set verify

Sätter verifierings-mod, dvs att exekverade script-rader skrivs ut i terminal-fönster.

### 20.2.33 set noverify

Avslutar verifierings-mod.

### 20.2.34 set fill

Sätter 'fill' i editorn. Skapade objekt kommer i fortsättningen att ha fill-attributet satt.

### 20.2.35 set nofill

Återställer 'fill' i editorn. Skapade objekt kommer i fortsättningen att ha fill-attributet nollställt.

### 20.2.36 set border

Sätter 'border' i editorn. Skapade objekt kommer i fortsättningen att ha border-attributet satt.

### 20.2.37 set noborder

Återställer 'border' i editorn. Skapade objekt kommer i fortsättningen att ha border-attributet nollställt.

### 20.2.38 set grid

Sätter 'grid' i editorn. Skapade objekt läggs på närmaste grid-punkt.

### 20.2.39 set nogrid

Återställer grid i editorn.

### 20.2.40 set linewidth

Sätter linjebredd i editorn. Skapade objekt kommer i fortsättningen att skapas med angiven linjebredd. Linjebredden kan ha ett värde mellan 1 och 8.

#### Syntax

```
ge> set linewidth 'linewidth'
```

### 20.2.41 set gridsize

Sätter gridstorleken i editorn.

#### Syntax

```
ge> set gridsize 'gridsize'
```

### 20.2.42 set textsize

Sätter 'textsize' i editorn. Alla textobjekt kommer i fortsättning att skapas med denna textstorlek. Textstorleken kan vara 8, 10, 12, 14, 18 eller 24.

### **Syntax**

```
ge> set textsize 'textsize'
```

#### **20.2.43 set textfont**

Sätter typsnitt i editorn. Alla textobjekt kommer i fortsättning att skapas med detta typsnitt. Typsnittet kan vara 'Helvetica', 'Times', 'New Century Schoolbook', 'Curier' eller 'LucidaSans'.

### **Syntax**

```
ge> set textfont 'font'
```

#### **20.2.44 set bold**

Sätter 'bold' i editorn. Textobjekt kommer i fortsättningen att skapas med fet-stil

#### **20.2.45 set nobold**

Återställer 'bold'.

#### **20.2.46 set backgroundcolor**

Sätter bakgrundfärgen.

### **Syntax**

```
ge> set backgroundcolor 'color'
```

#### **20.2.47 set fillcolor**

Sätter fyllnadsfärg i färg-paletten. Objekt kommer i fortsättningen att skapas med denna fyllnadsfärg.

### **Syntax**

```
ge> set fillcolor 'color'
```

#### **20.2.48 set bordercolor**

Sätter kantfärgen i färg-paletten. Objekt kommer i fortsättningen att skapas med denna kantfärg,

### **Syntax**

```
ge> set bordercolor 'color'
```

#### **20.2.49 set textcolor**

Sätter textfärgen i färg-paletten. text-objekt kommer i fortsättningen att skapas med denna textfärg,

### **Syntax**

```
ge> set textcolor 'color'
```

### **20.2.50 set currentobject fillcolor**

Sätter fyllnadsfärg på senast skapade subgraf-objekt.

#### **Syntax**

```
ge> set currentobject fillcolor 'color'
```

### **20.2.51 set currentobject colortone**

Sätter färgton på senast skapade subgraf-objekt.

#### **Syntax**

```
ge> set currentobject colortone 'tone'
```

### **20.2.52 set currentobject colorlightness**

Sätter ljushet på senast skapade subgraf-objekt. Ljusheten är ett heltal. Positivt värde ger ljusare färger, negativt mörkare färger.

#### **Syntax**

```
ge> set currentobject colorlightness 'lightness'
```

### **20.2.53 set currentobject colorintensity**

Sätter färgintensitet på senast skapade subgraf-objekt. Intensiteten är ett heltal. Positivt värde ger intensivare färger, negativt dassigare färger.

#### **Syntax**

```
ge> set currentobject colorintensity 'intensity'
```

### **20.2.54 set currentobject colorshift**

Skiftar färgen på senaste skapade subgraf. Färgtonerna i subgrafen roteras på färgcirkeln, men färgerna behåller sin inbördes färgkontrast. Colorshift är ett heltal som anger hur många steg som färgen ska shiftas. Positivt värde roterar i riktning gult -> guld -> orange -> röd -> magenta -> blå -> turkos -> grön -> gul. Negativt i motsatt riktning.

#### **Syntax**

```
ge> set currentobject colorshift 'shift'
```

### **20.2.55 set currentobject gradient**

Sätter färg-gradient på senast skapade object.  
Gradienten kan vara No,  
HorizontalUp, HorizontalDown, HorizontalTube1, HorizontalTube2,

VerticalLeft, VerticalRight, VerticalTube1, VerticalTube2,  
DiagonalUpperLeft, DiagonalLowerLeft, DiagonalUpperRight, DiagonalLowerRight,  
DiagonalUpTube, DiagonalDownTube,  
Globe, RadialCenter,  
RadialUpperLeft, RadialLowerLeft, RadialUpperRight or RadialLowerRight.

### Syntax

```
ge> set currentobject gradient 'gradient'
```

## 20.2.56 set currentobject attributes

Sätter attribut för senast skapade subgraf.

### Syntax

```
ge> set currentobject attributes 'attrname' 'value'
```

Attributens namn beror på vilken typ av dynamik som är angiven för subgrafen. Typen av dynamik måste sättas först, därefter kan övriga attribut för denna dynamiktyp sättas.

I Ge Användarhandledning finns beskrivet vilka attribut som finns för olika typer av dynamik.

Attribut av en uppräknings-typ, dvs man kan välja en av flera alternativ, sätt värdet till namnet på uppräkningsvärdet, t ex

```
set currentobject attr Cycle Slow
```

Attribut av typ bitmask, där man kan välja flera alternativ, anges namnen för alternativen åtskilda av '|', t ex

```
set currentobject attr Action ToggleDig|Confirm|Tooltip  
set currentobject attr DigColor.Instances 2|3|4
```

### Exempel

```
set currentobject attr Cycle Fast  
set currentobject attr DynType1 DigColor|DigWarning|DigError  
set currentobject attr Action Tooltip|OpenGraph  
set currentobject attr DigColor.Instances 2|3  
set currentobject attr DigColor3.Attribute H1-Dv3##Boolean  
set currentobject attr DigColor3.Color MagentaHigh3  
set currentobject attr DigColor2.Attribute H1-Dv2##Boolean  
set currentobject attr DigColor2.Color OrangeHigh4  
set currentobject attr DigColor.Attribute H1-Dv1##Boolean  
set currentobject attr DigColor.Color BlueHigh3  
set currentobject attr DigError.Attribute H1-Dv4##Boolean  
set currentobject attr DigWarning.Attribute H1-Dv5##Boolean  
set currentobject attr OpenGraph.GraphObject Nodes-MyNode-Op-Overview  
set currentobject attr ToolTip.Text "Temperature switch indicator"
```

## 20.2.57 set currentobject annotation

Lägger in en sträng i annotation-fältet på senast skapade subgraf. Texten läggs in i annotation nr 1.



## Syntax

```
ge> set currentobject annotation 'text'
```

### 20.2.58 set graphattributes

Sätter attribute för aktuell graf (eller subgraf).

#### Graf attribut

Namn	Typ
subgraph	Boolean
x0	Float
y0	Float
x1	Float
y1	Float
ScanTime	Float
AnimationScanTime	Float
JavaWidth	Float
IsJavaApplet	Boolean
IsJavaFrame	Boolean
BackgroundImage	String
Backgroundliled	Boolean
DoubleBuffered	Boolean
MB3Action	Enumeration: No, Close, PopupMenu or Both
Translate	Boolean
BitmapFonts	Boolean
HotIndication	Enumeration: No, LineWidth, DarkColor or LightColor
TooltipTextsize	Integer
AppMotion	Enumeration: Scroll, Slider or Both

#### Subgraf attribut

Namn	Typ
DynType1	Mask: Inherit, Tone, DigLowColor, DigColor, AnalogColor, StatusColor, DigWarning, DigError, DigFlash, FillLevel, Invisible, DigBorder, DigText, Valule, Rotate, Move, AnalogShift, DigShift, Animation, Bar, Trend, FastCurve, XY_Curve, SliderBackground, Video, Table, HostObject, DigSound, DigCommand
DynType2	Mask: DigTextColor, TimeoutColor
Action	Mask: Inherit, PopupMenu, SetDig, ResetDig, ToggleDig, StoDig, SetValue, Command, CommandDoubleClick, Help, OpenGraph, CloseGraph, OpenURL, Confirm, IncrAnalog, RadioButton, ValueInput, ToolTip, InputFocus, PulldownMenu, OptionMenu, MethodPulldownMenu, Slider
Color1	A color or color tone
Color2	A color or color tone
AnimSequence	Enumeration: Cyclic, Dig or ForwBack
NoConObstacle	Boolean
Slider	Boolean
NextSubgraph	String
AnimationCount	Int
JavaName	String
Cycle	Enumeration: Fast or Slow
x0	Float
y0	Float
x1	Float

y1	Float
InputFocusMark	Enumeration: Relief or No.
RecursiveDynamic	Boolean

## Syntax

```
ge> set graphattributes 'name' 'value'
```

## Exempel graf

```
set graphattributes x0 0
set graphattributes y0 0
set graphattributes x1 15
set graphattributes y1 20
set graphattributes DoubleBuffered 1
set graphattributes MB3Action PopupMenu
set graphattributes TooltipTextsize 14
```

## Exempel subgraf

```
set graphattributes Subgraph 1
set graphattributes DynType1 DigLowColor|DigWarning|DigError
set graphattributes Action PopupMenu|OpenGraph
set graphattributes Color1 GrayLow9
set graphattributes Cycle Fast
```

## 20.2.59 show

## 20.2.60 show version

Visa Ge-version.

# 21 Script

Ge-script är ett sätt att programmera ge-kommandon. Scripthanteraren ger dessutom möjlighet till att göra beräkningar, utföra villkors-satser, loop-satser, deklarerar variabler och funktioner.

Ett script startas med '@' följt av scriptfils namnet och eventuella argument.

Ett ge-script bör ha filtypen .ge\_com.

## Exempel

```
ge> @my_script
```

I pwr\_cmd Användarhandledning finns en beskrivning på scriptspråket och beskrivning på inbyggda funktioner. Här följer en lista på de inbyggda funktioner som även kan användas i Ge-script.

## Ge funktioner

### Funktion

BuildGraph  
ClearAll  
CreateArc  
CreateAxis  
CreateAxisArc  
CreateBar  
CreateBarArc  
CreateDsTrend  
CreateDsTrendCurve  
CreateFastCurve  
CreateImage  
CreateLine  
CreateObject  
CreatePie  
CreatePolyLine  
CreateRectangle  
CreateRecRounded  
CreateSevHist  
CreateText  
CreateTrend  
CreateXYCurve  
DashInsertObject  
DeleteObject  
GetCurrentObject  
GetFirstObject  
GetGraphName  
GetNextObject  
GetObjectAttribute  
GetObjectBorder  
GetObjectBorderColor  
GetObjectClass  
GetObjectDynType  
GetObjectFill

### Beskrivning

Bygg aktuell graph .  
Ta bort alla objekt i grafen.  
Skapa ett Arc objekt.  
Skapa ett Axis objekt.  
Skapa ett AxisArc objekt.  
Skapa ett Bar objekt.  
Skapa ett BarArc objekt.  
Skapa ett DsTrend objekt.  
Skapa ett DsTrendCurve objekt.  
Skapa ett FastCurve objekt.  
Skapa ett Image objekt.  
Skapa ett Line objekt.  
Skapa ett subgraph objekt.  
Skapa ett Pie objekt.  
Skapa ett PolyLine objekt.  
Skapa ett Rectangle objekt.  
Skapa ett rounded recangle objekt.  
Skapa ett SevHist objekt.  
Skapa ett Text objekt.  
Skapa ett Trend objekt.  
Skapa ett XYCurve objekt.  
Addera ett object till en DashCell.  
Ta bort ett objekt.  
Returnerar identiteten för senast skapade objekt.  
Hämta första objekt.  
Hämta name på nuvarande graf.  
Hämta nästa objekt.  
Hämta ett attributvärde för ett objekt.  
Hämta kant för ett objekt.  
Hämta kantfärg för ett objekt.  
Hämta klass för ett objekt.  
Hämta dynamik- och aktions-typ för ett objekt.  
Hämta fyllnad för ett objekt.

GetObjectFillColor  
GetObjectGradient  
GetObjectName  
GetObjectShadow  
GetObjectText  
GetObjectTextColor  
GetObjectTransparency  
GetObjectType  
GetRgbColor  
GetTextExtent  
GroupGetFirstObject  
GroupGetNextObject  
GroupSelected  
MeasureObject  
MoveAbsObject  
MoveObject  
OpenGraph  
PolyLineAdd  
Reload  
RotateSelected  
SaveGraph  
ScaleObject  
SelectAdd  
SelectClear  
SetBackgroundColor  
SetColorTheme  
SetCurrentObject  
SetDraw  
SetExtern  
SetExternAll  
SetIntern  
SetInternAll  
SetGraphName  
SetObjectAttribute  
SetObjectBackgroundColor  
SetObjectBorder  
SetObjectBorderColor  
SetObjectClass  
SetObjectFill  
SetObjectFillColor  
SetObjectGradient  
SetObjectShadow  
SetObjectTextColor  
SetObjectTransparency  
SetRgbColor  
SetSelectTextBold  
SetSelectTextFont  
SetSelectTextSize

Hämta fyllnadsfärg för ett objekt.  
Hämta gradient för ett objekt.  
Hämta namn på ett objekt.  
Hämta skuggning på ett objekt.  
Hämta text för ett objekt.  
Hämta textfärg för ett objekt.  
Hämta transparens för ett objekt.  
Hämta typ för ett objekt.  
Hämta rgb-värden för en färg.  
Beräkna storleken på en text.  
Hämta första objektet i en grupp.  
Hämta nästa objekt i en grupp.  
Gruppera utvalda objekt.  
Hämta ett objekts utbredning.  
Flytta ett objekt till en position.  
Flytta ett objekt ett visst avstånd.  
Öppna en graph.  
Addera ett segment till en PolyLine.  
Läs in innevarande graph från fil.  
Roterar utvalda objekt.  
Spara innevarande graph.  
Skala ett objekt.  
Addera objekt till utvalslistan.  
Töm listan av utvalda objekt.  
Sätt bakgrundsfaerg i aktuell graph.  
Sätt faergtema.  
Sätt CurrentObject.  
Sätt ritning till eller från.  
Sätt en subgraf extern.  
Sätt alla subgrafer extern.  
Sätt en subgraf intern.  
Sätt alla subgrafer intern.  
Sätt namn på aktuell graph.  
Sätt värde på ett attribut för ett objekt.  
Sätt bakgrundsfaerg på ett objekt.  
Sätt kant på ett objekt.  
Sätt kantfaerg på ett objekt.  
Ändar subgraf för ett objekt.  
Sätt fyllnad på ett objekt.  
Sätt fyllnadsfaerg på ett objekt.  
Sätt gradient på ett objekt.  
Sätt skuggning på ett objekt.  
Sätt textfaerg på ett objekt.  
Sätt transparens på ett objekt.  
Sätt rgb-värden för en färg.  
Sätt fet text på utvalda objekt.  
Sätt typsnitt på utvalda objekt.  
Sätt textstorlek på utvalda objekt.

## In och utmatning

### Funktion

ask  
say  
printf  
scanf

### Beskrivning

Skriver ut en fråga och läser in ett svar  
Skriver ut en sträng  
Formaterad utskrift  
Formaterad inläsning

## Filhantering

<b>Funktion</b>	<b>Beskrivning</b>
fclose	Stäng en fil
felement	Hämta ett element ur den med fgets senaste lästa raden.
fgets	Läsning av en rad från fil
fopen	Öppna en fil
fprintf	Formaterad skrivning på fil
fscanf	Formaterad läsning från fil
translate_filename	Byt ur miljövariabler i ett filnamn.

## Hantering av strängar

<b>Funktion</b>	<b>Beskrivning</b>
edit	Rensa bort space och tabbar i början och i slutet av en sträng, samt ta bort multipla space och tabbar i strängen
element	Hämta ett element i en sträng
extract	Hämta ett antal tecken i en sträng
sprintf	Formaterat skrivning i en sträng-variabel.
strchr	Leta efter första förekomsten av ett tecken i en sträng
strlen	Längden av en sträng
strrchr	Leta efter sista förekomsten av ett tecken i en sträng
strstr	Leta efter första förekomsten av en teckensekvens i en sträng
tolower	Konvertera till gemener
toupper	Konvertera till versaler

## Databas funktioner

<b>Funktion</b>	<b>Beskrivning</b>
CutObjectName	Hämta de sista segmenten i ett objektsnamn
GetAttribute	Hämta ett attribut
GetChild	Hämta första barnet till ett objekt
GetNextSibling	Hämta nästa syskon till ett objekt
GetNextVolume	Hämta nästa volym
GetParent	Hämta förälder till ett objekt
GetObjectClass	Hämta klassen till ett objekt
GetRootList	Hämta första objekt i rot-listan
GetVolumeClass	Hämta klassen för en volym

## System funktioner

<b>Funktion</b>	<b>Beskrivning</b>
exit	Avsluta exekveringen av ett skript
time	Hämta systemtiden
system	Exekvera ett shell kommando
verify	Sätt verify på eller av

## 21.1 Fördefinierade variabler

### Objekttyper

eObjectType\_Rect  
eObjectType\_Line

eObjectType\_Arc  
eObjectType\_ConPoint  
eObjectType\_Annot  
eObjectType\_PolyLine  
eObjectType\_SubGraph  
eObjectType\_Text  
eObjectType\_Bar  
eObjectType\_Trend  
eObjectType\_Slider  
eObjectType\_Image  
eObjectType\_Group  
eObjectType\_Axis  
eObjectType\_RectRounded  
eObjectType\_ConGlue  
eObjectType\_Menu  
eObjectType\_Window  
eObjectType\_Table  
eObjectType\_Folder  
eObjectType\_XYCurve  
eObjectType\_AxisArc  
eObjectType\_Pie  
eObjectType\_BarChart  
eObjectType\_Toolbar

## **Dynamik och actionstyper**

### **DynType1**

mDynType1\_No  
mDynType1\_Inherit  
mDynType1\_Tone  
mDynType1\_DigLowColor  
mDynType1\_DigColor  
mDynType1\_DigError  
mDynType1\_DigWarning  
mDynType1\_DigFlash  
mDynType1\_Invisible  
mDynType1\_DigBorder  
mDynType1\_DigText  
mDynType1\_Value  
mDynType1\_AnalogColor  
mDynType1\_Rotate  
mDynType1\_Move  
mDynType1\_AnalogShift  
mDynType1\_DigShift  
mDynType1\_Animation  
mDynType1\_Bar  
mDynType1\_Trend  
mDynType1\_SliderBackground  
mDynType1\_Video  
mDynType1\_FillLevel  
mDynType1\_FastCurve  
mDynType1\_AnalogText  
mDynType1\_Table  
mDynType1\_StatusColor  
mDynType1\_HostObject  
mDynType1\_DigSound  
mDynType1\_XY\_Curve  
mDynType1\_DigCommand

mDynType1\_Pie  
mDynType1\_BarChart

### **DynType2**

mDynType2\_No  
mDynType2\_Axis  
mDynType2\_DigTextColor  
mDynType2\_TimeoutColor  
mDynType2\_DigFourShift  
mDynType2\_ScrollingText  
mDynType2\_ColorThemeLightness  
mDynType2\_DigBackgroundColor

### **ActionType1**

mActionType1\_No  
mActionType1\_Inherit  
mActionType1\_PopupMenu  
mActionType1\_SetDig  
mActionType1\_ResetDig  
mActionType1\_ToggleDig  
mActionType1\_StoDig  
mActionType1\_Command  
mActionType1\_CommandDoubleClick  
mActionType1\_Confirm  
mActionType1\_IncrAnalog  
mActionType1\_RadioButton  
mActionType1\_Slider  
mActionType1\_ValueInput  
mActionType1\_TipText  
mActionType1\_Help  
mActionType1\_OpenGraph  
mActionType1\_OpenURL  
mActionType1\_InputFocus  
mActionType1\_CloseGraph  
mActionType1\_PulldownMenu  
mActionType1\_OptionMenu  
mActionType1\_SetValue  
mActionType1\_MethodToolbar  
mActionType1\_MethodPulldownMenu  
mActionType1\_Script

### **Fonter**

eFont\_Helvetica  
eFont\_Times  
eFont\_NewCenturySchoolbook  
eFont\_Courier  
eFont\_LucidaSans

### **Riktning**

eDirection\_Center  
eDirection\_Right  
eDirection\_Left  
eDirection\_Up  
eDirection\_Down

## Colors

### Standardpalett

eDrawType\_Color1 = 0

eDrawType\_Color2 = 1

eDrawType\_Color3 = 2

...

eDrawType\_Color300 = 299

### Custom palett

eDrawType\_CustomColor1 = 310

eDrawType\_CustomColor2 = 314

eDrawType\_CustomColor3 = 318

...

eDrawType\_CustomColor90 = 666

## Access

mAccess\_RtRead

mAccess\_RtWrite

mAccess\_System

mAccess\_Maintenance

mAccess\_Process

mAccess\_Instrument

mAccess\_Operator1

mAccess\_Operator2

mAccess\_Operator3

mAccess\_Operator4

mAccess\_Operator5

mAccess\_Operator6

mAccess\_Operator7

mAccess\_Operator8

mAccess\_Operator9

mAccess\_Operator10

mAccess\_RtEventsAck

mAccess\_RtPlc

mAccess\_RtNavigator

mAccess\_DevRead

mAccess\_DevPlc

mAccess\_DevConfig

mAccess\_DevClass

mAccess\_RtEventsBlock

mAccess\_Administrator

mAccess\_SevRead

mAccess\_SevAdmin

mAccess\_AllRt

mAccess\_RtDefault

mAccess\_AllOperators

mAccess\_AllSev

mAccess\_AllPwr

mAccess\_Default



## 21.2 BuildGraph()

```
void BuildGraph()
```

### **Beskrivning**

Bygg aktuell graph.

### **Exempel**

```
BuildGraph( ) ;
```

## 21.3 ClearAll()

```
void ClearAll()
```

### **Beskrivning**

Ta bort alla object i innevarande graph.

### **Exempel**

```
ClearAll( );
```

## 21.4 CreateArc()

```
int CreateArc(float x1, float y1, float x2, float y2, int angle1, int angle2)
```

### Beskrivning

Skapa ett Arc objekt.

Färg, fyllnad, kant och andra egenskaper hämtas från aktuella inställningar i editorn.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet.
float	y2	Y koordinat för nedre högra hörnet.
int	angle1	Startvinkel i grader. 0 är kl 3.
int	angle2	Segmentets utbredning i grader medsols från startvinkeln.

Returnerar objektets identitet.

### Exempel

```
int id;
```

```
id = CreateArc(2.0, 2.0, 3.0, 3.0, 0, 180);
```

## 21.5 CreateAxis()

int CreateAxis(float x1, float y1 [,float x2, float y2, int colortheme, int dynamic, int direction])

### Beskrivning

Skapa ett axis objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.
int	dynamic	Dynamisk axis (1) eller static (0). Optional.
int	direction	Riktning upp, ner, vänster eller höger. Anges med symbolerna eDirection_Up, eDirection_Down, eDirection_Left och eDirection_Right. Optional.

Returnerar objektets identitet.

### Exempel

```
int id;

id = CreateAxis(1.0, 1.0, 10.0, 2.0, 0, 0, eDirection_Down);
SetObjectAttribute(id, "MaxValue", 100.0);
SetObjectAttribute(id, "MinValue", -100.0);
```

## 21.6 CreateAxisArc()

```
int CreateAxisArc(float x1, float y1 [,float x2, float y2, int colortheme, int dynamic])
```

### Beskrivning

Skapa ett axis arc objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.
int	dynamic	Dynamisk axis (1) eller static (0). Optional.

Returnerar objektets identitet.

### Exempel

```
int id;  
  
id = CreateAxisArc(1.0, 1.0, 5.0, 4.0, 0, 0);  
SetObjectAttribute(id, "MaxValue", 100.0);  
SetObjectAttribute(id, "MinValue", -100.0);  
SetObjectAttribute(id, "Angle1", -45);  
SetObjectAttribute(id, "Angle2", 270);
```

## 21.7 CreateBar()

```
int CreateBar(float x1, float y1 [,float x2, float y2, int colortheme, int direction])
```

### Beskrivning

Skapa ett bar objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.
int	direction	Riktning upp, ner, vänster eller höger. Anges med symbolerna eDirection_Up, eDirection_Down, eDirection_Left och eDirection_Right. Optional.

Returnerar id för det skapade objektet.

### Exempel

```
int id;

id = CreateBar(2.0, 2.0, 2.5, 5.0, 1, eDirection_Up);
SetObjectAttribute(id, "Bar.Attribute", "H1-Av1.ActualValue##Float32");
SetObjectAttribute(id, "Bar.MaxValue", 100.0);
SetObjectAttribute(id, "Bar.MinValue", -100.0);
```

## 21.8 CreateBarArc()

int CreateBar(float x1, float y1 [,float x2, float y2, int colortheme])

### Beskrivning

Skapa ett BarArc objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.

Returnerar id för det skapade objektet.

### Exempel

```
int id;

id = CreateBarArc(2.0, 2.0, 2.5, 5.0, 1);
SetObjectAttribute(id, "Bar.Attribute", "H1-Av1.ActualValue##Float32");
SetObjectAttribute(id, "BarArc.MaxValue", 100.0);
SetObjectAttribute(id, "BarArc.MinValue", -100.0);
SetObjectAttribute(id, "BarArc.Angle1", -45);
SetObjectAttribute(id, "BarArc.Angle2", 270);
```

## 21.9 CreateDsTrend()

```
int CreateDsTrend(float x1, float y1 [,float x2, float y2, int colortheme])
```

### Beskrivning

Skapa ett DsTrend objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.

Returnerar id för det skapade objektet.

### Exempel

```
int id;  
  
id = CreateDsTrend(2.0, 2.0, 7.0, 5.0, 1);  
SetObjectAttribute(id, "DsTrend.Object1", "H1-Trend1");  
SetObjectAttribute(id, "DsTrend.MaxValue1", 100.0);  
SetObjectAttribute(id, "DsTrend.MinValue1", -100.0);  
SetObjectAttribute(id, "DsTrend.ScanTime", 0.5);
```



## 21.10 CreateDsTrendCurve()

int CreateDsTrendCurve(float x1, float y1 [,float x2, float y2, int colortheme])

### Beskrivning

Skapa ett DsTrendCurve objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.

Returnerar id för det skapade objektet.

### Exempel

```
int id;

id = CreateDsTrendCurve(2.0, 2.0, 7.0, 5.0, 1);
SetObjectAttribute(id, "DsTrendCurve.Object", "H1-TrendCurve");
SetObjectAttribute(id, "DsTrendCurve.MaxValue1", 100.0);
SetObjectAttribute(id, "DsTrendCurve.MinValue1", -100.0);
SetObjectAttribute(id, "DsTrendCurve.ScanTime", 0.5);
```

## 21.11 CreateFastCurve()

int CreateFastCurve(float x1, float y1 [,float x2, float y2, int colortheme])

### Beskrivning

Skapa ett FastCurve objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.

Returnerar id för det skapade objektet.

### Exempel

```
int id;

id = CreateFastCurve(2.0, 2.0, 7.0, 5.0, 1);
SetObjectAttribute(id, "FastCurve.Object", "H1-FastCurve");
SetObjectAttribute(id, "FastCurve.MaxValue1", 100.0);
SetObjectAttribute(id, "FastCurve.MinValue1", -100.0);
```

# 21.12 CreateImage()

```
int CreateImage(string image, float x1, float y1 [,float x2, float y2])
```

## Beskrivning

Skapa ett image objekt.

## Argument

string	image	Image filnamn. Filen ska ligga på \$pwrp_exe eller \$pwr_exe.
float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.

Returnerar id för det skapade objektet.

## Exempel

```
int id;  
  
id = CreateImage("pwr_logofully.png", 2.0, 2.0);
```

## 21.13 CreateLine()

```
int CreateLine(float x1, float y1, float x2, float y2)
```

### Beskrivning

Skapa en rät linje.

Färg, bredd och linjetyp hämtas från de aktuella inställningarna i editorn.

### Argument

float	x1	X koordinat för startpunkten.
float	y1	Y koordinat för startpunkten.
float	x2	X koordinat för slutpunkten.
float	y2	Y koordinat för slutpunkten.

Returnerar id för det skapade objektet.

### Exempel

```
int id;  
  
id = CreateLine(2.0, 2.0, 3.0, 3.0);
```

## 21.14 CreateObject()

```
int CreateObject(string subgraph, float x1, float y1 [,float x2, float y2])
```

### Beskrivning

Skapa ett subgraph objekt.

### Argument

string	subgraph	Subgraph name.
float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.

Returnerar id för det skapade objektet.

### Exempel

```
int id;  
  
id = CreateObject("pwr_valve", 2.0, 2.0);
```

## 21.15 CreatePie()

int CreatePie(float x1, float y1 [,float x2, float y2, int colortheme])

### Beskrivning

Skapa ett Pie objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.

Returnerar identiteten på det skapade objektet.

### Exempel

```
int id;

id = CreatePie(2.0, 2.0, 2.5, 5.0, 1);
SetObjectAttribute(id, "Pie.Sectors", 3);
SetObjectAttribute(id, "Pie.FixRange", 1);
SetObjectAttribute(id, "Pie.Attribute1", "H1-Av1.ActualValue##Float32");
SetObjectAttribute(id, "Pie.Attribute2", "H1-Av2.ActualValue##Float32");
SetObjectAttribute(id, "Pie.Attribute3", "H1-Av3.ActualValue##Float32");
SetObjectAttribute(id, "Pie.MinValue", 0.0);
SetObjectAttribute(id, "Pie.MaxValue", 300.0);
```

## 21.16 CreatePolyLine()

```
int CreatePolyLine(float x1, float y1, float x2, float y2)
```

### Beskrivning

Skapa första segmentet av en polyline. Efterföljande segment adderas med PolyLineAdd().

Färg, bredd och linjetyp hämtas från de aktuella inställningarna i editorn.

### Argument

float	x1	X koordinat för startpunkten.
float	y1	Y koordinat för startpunkten.
float	x2	X koordinat för slutpunkten.
float	y2	Y koordinat för slutpunkten.

Returnerar id för det skapade objektet.

### Exempel

```
int id;  
  
id = CreatePolyLine(2.0, 2.0, 3.0, 3.0);  
PolyLineAdd(id, 4.0, 2.0);  
PolyLineAdd(id, 5.0, 3.0);
```

## 21.17 CreateRectangle()

int CreateRectangle(float x, float y, float width, float height)

### Beskrivning

Skapa en rektangel.

Färg, fyllnad, kant och andra egenskaper hämtas från de aktuella inställningarna i editorn.

### Argument

float	x	X koordinat för övre vänstra hörnet.
float	y	Y koordinat för övre vänstra hörnet.
float	width	Rektangelns bredd.
float	height	Rektangelns höjd.

Returnerar id för det skapade objektet.

### Exempel

```
int id;
```

```
id = CreateRectangle(2.0, 2.0, 5.0, 3.0);
```



## 21.18 CreateRectRounded()

int CreateRectRounded(float x, float y, float width, float height)

### Beskrivning

Skapa en rektangel med rundade hörn.

Färg, fyllnad, kant och andra egenskaper hämtas från de aktuella inställningarna i editorn.

### Argument

float	x	X koordinat för övre vänstra hörnet.
float	y	Y koordinat för övre vänstra hörnet.
float	width	Rektangelns bredd.
float	height	Rektangelns höjd.

Returnerar id för det skapade objektet.

### Exempel

```
int id;
```

```
id = CreateRectRounded(2.0, 2.0, 5.0, 3.0);
```

## 21.19 CreateSevHist()

int CreateSevHist(float x1, float y1 [,float x2, float y2, int colortheme])

### Beskrivning

Skapa ett SevHist objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.

Returnerar id för det skapade objektet.

### Exempel

```
int id;  
  
id = CreateSevHist(2.0, 2.0, 7.0, 5.0, 1);  
SetObjectAttribute(id, "SevHist.Object1", "H1-Av1-Hist");  
SetObjectAttribute(id, "SevHist.MaxValue1", 100.0);  
SetObjectAttribute(id, "SevHist.MinValue2", -100.0);  
SetObjectAttribute(id, "SevHist.TimeRange", 120.0);
```

## 21.20 CreateText()

```
int CreateText(string text, float x, float y)
```

### Beskrivning

Skapa ett text objekt. Font, storlek, färg och andra egenskaper hämtas från de aktuella inställningarna i editorn.

### Argument

string	text	Text.
float	x	X koordinat.
float	y	Y koordinat.

Returnerar id för det skapade objektet.

### Exempel

```
int id;  
  
id = CreateText("Start", 2.0, 2.0);
```

## 21.21 CreateTrend()

int CreateTrend(float x1, float y1 [,float x2, float y2, int colortheme])

### Beskrivning

Skapa ett Trend objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.

Returnerar id för det skapade objektet.

### Exempel

```
int id;

id = CreateTrend(2.0, 2.0, 7.0, 5.0, 1);
SetObjectAttribute(id, "Trend.Attribute1", "H1-Av1.ActualValue##Float32");
SetObjectAttribute(id, "Trend.MaxValue1", 100.0);
SetObjectAttribute(id, "Trend.MinValue1", -100.0);
SetObjectAttribute(id, "Trend.ScanTime", 0.5);
```

## 21.22 CreateXYCurve()

int CreateXYCurve(float x1, float y1 [,float x2, float y2, int colortheme])

### Beskrivning

Skapa ett XYCurve objekt.

### Argument

float	x1	X koordinat för övre vänstra hörnet.
float	y1	Y koordinat för övre vänstra hörnet.
float	x2	X koordinat för nedre högra hörnet om objektet ska skalas. Optional.
float	y2	Y koordinat för nedre högra hörnet om objektet ska skalas. Optional.
int	colortheme	1 om färgtema färger ska användas, annars 0. Optional.

Returnerar id för det skapade objektet.

### Exempel

```
int id;  
  
id = CreateXYCurve(2.0, 2.0, 7.0, 5.0, 1);  
SetObjectAttribute(id, "XY_Curve.XAttr", "H1-X1.Value##Float32#100");  
SetObjectAttribute(id, "XY_Curve.YAttr", "H1-Y1.Value##Float32#100");  
SetObjectAttribute(id, "XY_Curve.XMaxValue", 100.0);  
SetObjectAttribute(id, "XY_Curve.XMinValue", 0.0);  
SetObjectAttribute(id, "XY_Curve.YMaxValue", 100.0);  
SetObjectAttribute(id, "XY_Curve.YMinValue", -100.0);  
SetObjectAttribute(id, "XY_Curve.UpdateAttr", "H1-Update.ActualValue##Boolean");
```

## 21.23 DashInsertObject()

```
void DashInsertObject(int dashoid, int oid)
```

### Beskrivning

Addera ett objekt till en dashcell.

### Argument

int	dashoid	Objektidentitet för DashCell-objektet.
int	oid	Objektidentitet för objektet.

### Exempel

```
int doid;  
int oid;
```

```
DashInsertObject(doid, oid);
```

## 21.24 DeleteObject()

```
void DeleteObject( int oid)
```

### Beskrivning

Ta bort ett objekt.

### Argument

int	oid	Objektsidentitet.
-----	-----	-------------------

### Exempel

```
int oid;
```

```
DeleteObject( oid);
```

## 21.25 GetCurrentObject()

```
int GetCurrentObject()
```

### Beskrivning

Returnerar identiteten för det senast skapade objektet.

### Exempel

```
int id;
```

```
id = GetCurrentObject();
```



## 21.26 GetFirstObject()

int GetFirstObject()

### Beskrivning

Hämtar identiteten för det första objektet i en graf. Övriga objekt kan hämtas med GetNextObject().

Returnerar en integer med objektsidentiteten.

### Exempel

```
int oid;

oid = GetFirstObject();
while ( oid != 0)
    ...
    oid = GetNextObject( oid)
endwhile
```

## 21.27 GetGraphName()

```
string GetGraphName()
```

### Beskrivning

Hämta namn på nuvarande graf.

Returnerar en sträng med namnet.

### Exempel

```
string name;
```

```
name = GetGraphName();
```

# 21.28 GetNextObject()

```
int GetNextObject( int oid)
```

## Beskrivning

Hämtar identiteten för nästa objekt i en graf. Det första objektet kan hämtas med GetFirstObject() och övriga objekt med GetNextObject()

## Argument

int	oid	Objektidentitet på objekt vars nästa objekt ska hämtas.
-----	-----	---

Returnerar en integer med objektsidentiteten.

## Exempel

```
int oid;

oid = GetFirstObject();
while ( oid != 0)
    ...
    oid = GetNextObject( oid)
endwhile
```

## 21.29 GetObjectAttribute()

int GetObjectAttribute( int oid, string attribute, (arbitrary type) value)

### Beskrivning

Hämtar värde på ett attribut för ett objekt. Attributet namnges på samma sätt som i attributeditorn för objektet.

### Argument

int	oid	Objektets identitet.
string	attribut	Attributnamn.
(arbitrary type)	value	Hämtat värde på attributet.

Returnerar status för operationen.

### Exempel

```
int sts;
int oid;
int color;

sts = GetObjectAttribute( oid, "DigColor.Color", color);
if ( !(sts & 1))
    printf( "Couldn't get color\n");
endif
```

## 21.30 GetObjectBorder()

```
int GetObjectBorder( int oid)
```

### Beskrivning

Hämtar kant för ett objekt.

Returnerar värdet på kant, 0 eller 1.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

### Exempel

```
int oid;  
int border;  
  
border = GetObjectBorder( oid);
```

## 21.31 GetObjectBorderColor()

```
int GetObjectBorderColor( int oid)
```

### Beskrivning

Hämtar kantfärgen för ett objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar värdet på kantfärgen.

### Exempel

```
int oid;  
int bcolor;  
  
bcolor = GetObjectBorderColor( oid);  
if ( bcolor == eDrawType_Color33)  
    SetObjectBorderColor( oid, eDrawType_CustomColor24);  
endif
```

## 21.32 GetObjectClass()

```
string GetObjectClass( int oid)
```

### Beskrivning

Hämtar namnet på subgrafet för ett objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar namnet.

### Exempel

```
int oid;  
string class;  
  
class = GetObjectClass( oid);  
if ( class == "pwr_roundind")  
    ...  
endif
```

## 21.33 GetObjectDynType()

int GetObjectDynType( int oid, int dyntype1, int dyntype2, int actiontype1, int actiontype2)

### Beskrivning

Hämtar dynamiktyp och aktionstyp för objektet.

### Argument

int	oid	Objektets identitet.
int	dyntype1	Hämtat värde på DynType1.
int	dyntype2	Hämtat värde på DynType2.
int	actiontype1	Hämtat värde på ActionType1.
int	actiontype2	Hämtat värde på ActionType2.

Returnerar status för operationen.

Se Fördefinierade variables för värden på dyntype and actiontype.

### Exempel

```
int sts;
int oid;
int dyntype1;
int dyntype2;
int actiontype1;
int actiontype2;

sts = GetObjectDynType( oid, dyntype1, dyntype2, actiontype1, actiontype2);
if ( dyntype1 & mDynType1_Value)
    ...
endif
```



## 21.34 GetObjectFill()

```
int GetObjectFill( int oid)
```

### Beskrivning

Hämtar fyllnad för ett objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar värdet på fyllnad, 0 eller 1.

### Exempel

```
int oid;  
int fill;  
  
fill = GetObjectFill( oid);
```

## 21.35 GetObjectFillColor()

```
int GetObjectFillColor( int oid)
```

### Beskrivning

Hämtar fyllnadsfärgen för ett objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar värdet på fyllnadsfärgen.

### Exempel

```
int oid;  
int fcolor;  
  
fcolor = GetObjectFillColor( oid);  
if ( fcolor == eDrawType_Color67)  
    SetObjectFillColor( oid, eDrawType_CustomColor23);  
endif
```

## 21.36 GetObjectGradient()

```
int GetObjectGradient( int oid)
```

### Beskrivning

Hämtar gradienten för ett objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar värdet på gradienten, dvs typen av gradient. Värdet 0 innebär ingen gradient.

### Exempel

```
int oid;  
int grad;  
  
grad = GetObjectGradient( oid);
```

## 21.37 GetObjectName()

```
string GetObjectName( int oid)
```

### Beskrivning

Hämtar namnet på ett objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar namnet.

### Exempel

```
int oid;  
string oname;  
  
oname = GetObjectName( oid);
```

## 21.38 GetObjectShadow()

```
int GetObjectShadow( int oid)
```

### Beskrivning

Hämtar skuggning för ett objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar en integer med skuggningen.

### Exempel

```
int oid;  
int shadow;  
  
shadow = GetObjectShadow( oid);
```

## 21.39 GetObjectText()

```
string GetObjectText( int oid)
```

### Beskrivning

Hämtar text för ett text-objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar texten.

### Exempel

```
int oid;  
string text;  
  
text = GetObjectText( oid);
```

## 21.40 GetObjectTextColor()

```
int GetObjectTextColor( int oid)
```

### Beskrivning

Hämtar textfärgen för ett objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar värdet på textfärgen.

### Exempel

```
int oid;
int tcolor;

tcolor = GetObjectTextColor( oid);
if ( tcolor == eDrawType_Color49)
    SetObjectTextColor( oid, eDrawType_CustomColor5);
endif
```

## 21.41 GetObjectTransparency()

```
float GetObjectTransparency( int oid)
```

### Beskrivning

Hämta transparens för ett objekt.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar transparensen. Ett flyttalsvärde mellan 0.0 och 1.0.

### Exempel

```
float transparency;  
  
transparency = GetObjectTransparency( oid);
```



## 21.42 GetObjectType()

```
int GetObjectType( int oid)
```

### Beskrivning

Hämtar ett objects typ.

### Argument

int	oid	Objektets identitet.
-----	-----	----------------------

Returnerar objektstypen.

Se Fördefinierade variabler för värden på objekttyper

### Exempel

```
int oid;
int type;

type = GetObjectType( oid);
if ( type == eObjectType_SubGraph)
    SetObjectFillColor( oid, eDrawType_CustomColor22);
endif
```

## 21.43 void GetRgbColor()

void GetRgbColor( int color, float red, float green, float blue)

### Beskrivning

Hämtar rgb-värden för en färg. Färgen anges med färgindex eller med symbol, se nedan. rgb-värdena är flyttal mellan 0 och 1.

### Argument

int	color	Färg.
float	red	Returnerar värde på röd.
float	green	Returnerar värde på grön.
float	blue	Returnerar värde på blå.

Färger i standardpaletten finns definierade med eDrawType\_Color1 - eDrawType\_Color300, och färger i custom color paletten med eDrawType\_CustomColor1 - eDrawType\_CustomColor90.

Motsvarande färgindex kan också användas, för standardpaletten 0 - 299, och för custom color paletten 310 - 669. För custom color ligger grundfärgen på var fjärde index, dvs 310, 314, 318, ... 666.

### Exempel

```
int i;
float r;
float g;
float b;

printf( "Idx  Red   Green Blue\n");
for ( i = eDrawType_CustomColor1; i <= eDrawType_CustomColor90; i += 4)
    GetRgbColor( i, r, g, b);

    printf( "%3d  %5.3f %5.3f %5.3f\n", i, r, g, b);
endfor
```

## 21.44 GetTextExtent()

```
int GetTextExtent( string text, int textsize, int font, int bold, float widht [, float height,  
float descent])
```

### Beskrivning

Beräknar storleken på en text.

Returnerar text-bredden i width argumentet, och om argumentet height och descent skickats med, även höjd och descent i dessa argument.

### Argument

string	text	Text vars storlek ska beräknas.
int	textsize	Textstorleksindex (0-5)).
int	font	Text font.
int	bold	Fet (1) eller normal (0) text.
float	width	Textens bredd returneras i denna variabel.
float	height	Höjden på texten returneras i denna variabel.
float	descent	Textens descent returneras i denna variabel.

### Exempel

```
float width;
```

```
GetTextExtent("Detta är en text", 3, eFont_LucidaSans, 1, width);
```

## 21.45 GroupGetFirstObject()

```
int GroupGetFirstObject( int group)
```

### Beskrivning

Hämtar identiteten för det första objektet i en grupp. Övriga objekt kan hämtas med GroupGetNextObject().

### Argument

int	group	Identitet på gruppen.
-----	-------	-----------------------

Returnerar en integer med objektsidentiteten.

### Exempel

```
int goid;
int oid;

# Get all objects
goid = GetFirstObject();
while ( goid != 0)
    type = GetObjectType( goid);
    if ( type == eObjectType_Group)
        # This is a group, get all objects in the group
        oid = GroupGetFirstObject( goid);
        while ( oid != 0)
            ...
            oid = GroupGetNextObject( goid, oid)
        endwhile
    endwhile
```

## 21.46 GroupGetNextObject()

int GroupGetNextObject( int group, int oid)

### Beskrivning

Hämtar identiteten för nästa objekt i en grupp. Det första objektet kan hämtas med GroupGetFirstObject() och övriga objekt med GroupGetNextObject()

### Argument

int	oid	Objektidentitet på objekt vars nästa objekt ska hämtas.
-----	-----	---

Returnerar en integer med objektsidentiteten.

## 21.47 GroupSelected()

int GroupSelected()

### Beskrivning

Skapa en grupp med de utvalda objekten.

Returnar en integer med gruppens identiteten.

### Exempel

```
int id1;  
int id2;  
int gid;  
  
SelectAdd(id1);  
SelectAdd(id2);  
gid = GroupSelected();
```

# 21.48 MeasureObject()

```
void MeasureObject( int oid, float ll_x, float ll_y, float ur_x, float ur_y)
```

## Beskrivning

Hämta utbredningen av ett objekt.

Returnerar x och y koordinat för under vänstra och övre högre hörnpunkterna.

## Argument

int	oid	Objektsidentitet
float	ll_x	Returnerad x koordinat för under vänstra.
float	ll_y	Returnerad y koordinat för under vänstra.
float	ur_x	Returnerad x koordinat för övre högra.
float	ur_y	Returnerad y koordinat för övre högra.

## Exempel

```
int oid;  
float ll_x;  
float ll_y;  
float ur_x;  
float ur_y;
```

```
MeasureObject( oid, ll_x, ll_y, ur_x, ur_y);
```

## 21.49 MoveAbsObject()

```
void MoveAbsObject( int oid, float x, float y)
```

### Beskrivning

Flytta ett objekt till koordinaterna x och y.

### Argument

int	oid	Objektsidentitet.
float	x	x koordinat.
float	y	y koordinat.

### Exempel

```
MoveAbsObject( oid, 0.5, 0.5);
```



## 21.50 MoveObject()

```
void MoveObject( int oid, float dx, float dy)
```

### Beskrivning

Flytta ett objekt ett visst avstånd.

### Argument

int	oid	Objektsidentitet.
float	dx	Avstånd att flytta i x-led.
float	dy	Avstånd att flytta i y-led.

### Exempel

```
MoveObject( oid, 0.5, 0.5 );
```

## 21.51 OpenGraph()

```
void OpenGraph(string name)
```

### Beskrivning

Öppna en graph.

### Argument

string	name	Namn på grafen.
--------	------	-----------------

### Exempel

```
OpenGraph( "overview" );
```

## 21.52 PolyLineAdd()

```
int PolyLineAdd(int id, float x, float y)
```

### Beskrivning

Addera ett segment till ett PolyLine objekt.

### Argument

float	x	X koordinat för slutpunkten.
float	y	Y koordinat för slutpunkten.

### Exempel

```
int id;  
  
id = CreatePolyLine(2.0, 2.0, 3.0, 3.0);  
PolyLineAdd(id, 4.0, 2.0);  
PolyLineAdd(id, 5.0, 3.0);
```

## 21.53 Reload()

void Reload()

### Beskrivning

Läser nuvarande graf från fil.

### Exempel

```
# Update subgraph pwr_roundind
SetExtern("pwr_roundind");
save
Reload();
SetIntern("pwr_roundind");
save
```

## 21.54 RotateSelected()

```
void RotateSelected(float angle, int rotationpoint)
```

### Beskrivning

Rotera utvalda objekt runt rotationspunkten.  
Rotationspunkten kan vara

- 0 nedre vänstra hörnet
- 1 nedre högra hörnet
- 2 övre högra hörnet
- 3 övre vänstra hörnet
- 4 mittpunkten (default-värde)

### Argument

float	angle	Vinkel i grader för rotationen.
int	rotationpoint	Rotationspunkt. Default-värde center.

### Exempel

```
int id;  
  
SelectAdd(id);  
RotateSelected(90.0);
```

## 21.55 SaveGraph()

```
void SaveGraph([string name])
```

### Beskrivning

Spara aktuell graph.

### Argument

string	name	Namn på grafen. Optional.
--------	------	---------------------------

### Exempel

```
SaveGraph( ) ;
```

## 21.56 ScaleObject()

```
void ScaleObject( int oid, float scalex, float scaley)
```

### Beskrivning

Skala ett objekt.

### Argument

int	oid	Objektsidentitet.
float	scalex	Skalfaktor i x-led.
float	scaley	Skalfaktor i y-led.

### Exempel

```
ScaleObject( oid, 1.5, 1.5);
```

## 21.57 SelectAdd()

```
int SelectAdd( int objectid)
```

### Beskrivning

Lägger in ett objekt i listan av utvalda objekt.

### Argument

int	objectid	Objektsidentitet.
-----	----------	-------------------

### Exempel

```
int id;  
  
id = GetCurrentObject();  
SelectAdd( id);
```



## 21.58 SelectClear()

int SelectClear()

### Description

Töm listan på utvalda objekt.

## 21.59 SetBackgroundColor()

```
void SetBackgroundColor(int color)
```

### Beskrivning

Sätt bakgrundsfärg för aktuell graph.

### Argument

int	color	Bakgrundsfärg.
-----	-------	----------------

### Exempel

```
SetBackgroundColor(eDrawType_CustomColor1);
```

# 21.60 SetColorTheme()

```
void SetColorTheme(int colortheme)
```

## Beskrivning

Sätt färgtema för aktuell graph.

I runtime måste även ColorTheme i GraphAttributes sättas till \$default. I det här fallet ska SetColorTheme() anropas utan argument.

## Argument

int	colortheme	Index för färgtema.
-----	------------	---------------------

## Exempel

```
SetGraphAttribute("ColorTheme", "$default");  
SetColorTheme(2);
```

## 21.61 SetCurrentObject()

```
void SetCurrentObject( int oid)
```

### Beskrivning

Sätter CurrentObject.

### Argument

int	oid	Objektsidentitet.
-----	-----	-------------------

### Exempel

```
SetCurrentObject( oid);
```

# 21.62 SetDraw()

void SetDraw( int on)

## Beskrivning

Sätter ritning på skärmen till (1) eller från (0).

## Argument

int                                      on                                      1 innebär att ritning sätts till, och 0 att ritning sätts från.

## Exempel

```
SetDraw(0);

for ( i = 0; i < 10000; i++)
  x2 = x + 0.55;
  y2 = y + 0.55;
  create object/sub=pwr_indsquare/x1='x'/y1='y'/x2='x2'/y2='y2'
  set current attr DigLowColor.Attribute H1-DArray.Val['i']##Boolean
  x += 0.65;
  if ( i - (i / 100) * 100 == 99)
    y += 0.65;
    x = 0;
  endif
endfor

SetDraw(1);
```

## 21.63 SetExtern()

```
int SetExtern( string name)
```

### Beskrivning

Sätter en subgraf extern.

### Argument

string	name	Namn på subgrafen.
--------	------	--------------------

### Exempel

```
SetExtern( "MySubgraph" );
```

## 21.64 SetExternAll()

```
void SetExternAll()
```

### Beskrivning

Sätter alla subgrafer extern.

### Exempel

```
SetExternAll();
```

## 21.65 SetIntern()

```
int SetIntern( string name)
```

### Beskrivning

Sätter en subgraf intern.

### Argument

string	name	Namn på subgrafen.
--------	------	--------------------

### Exempel

```
SetIntern( "MySubgraph" );
```



## 21.66 SetInternAll()

```
void SetInternAll()
```

### Beskrivning

Sätter alla subgrafer intern.

### Exempel

```
SetInternAll();
```

## 21.67 SetGraphName()

string SetGraphName()

### Beskrivning

Sätt namn på innevarande graph.

### Argument

string	name	Namn på grafen.
--------	------	-----------------

### Exempel

```
string name = "overview";  
  
SetGraphName(name);
```

# 21.68 SetObjectAttribute()

int SetObjectAttribute( int oid, string attribute, (arbitrary type) value)

## Beskrivning

Sätter värde på ett attribut i ett objekt. Attributet namnges på samma sätt som i attributeditorn för objektet.

Citationstecken i strängar kan sättas med \", och ny rad i attribut med flera rader med \n.

## Argument

int	oid	Objektets identitet.
string	attribut	Attributnamn.
(arbitrary type)	Värde på attributet.	

Returnerar status för operationen.

## Exempel 1

```
int sts;
int oid;

sts = SetObjectAttribute( oid, "DigColor.Color", eDrawType_Color244);
```

## Exempel 2

Script attribut med flera rader och strängar med citationstecket.

```
SetObjectAttribute(id, "Script.Script", "\
int a;\n\
a = GetAttribute(\"H1-Dv1.ActualValue\");\n\
if (a)\n\
    SetSubwindow(\"$current\", \"W1\", \"@plant1\", \"\", 0);\n\
else\n\
    SetSubwindow(\"$current\", \"W1\", \"@plant2\", \"\", 0);\n\
endif");
```

## 21.69 SetObjectBackgroundColor()

```
void SetObjectBackgroundColor( int oid, int color)
```

### Beskrivning

Sätter bakgrundsfärg på ett objekt.

### Argument

int	oid	Objektets identitet.
int	color	Bakgrundsfärg.

### Exempel

```
SetObjectBackgroundColor( oid, eDrawType_CustomColor1);
```

## 21.70 SetObjectBorder()

```
void SetObjectBorder( int oid, int border)
```

### Beskrivning

Sätter kant på ett objekt.

### Argument

int	oid	Objektets identitet.
int	border	1 ritar ut kantlinjen, 0 tar bort kantlinjen.

### Exempel

```
SetObjectBorder( oid, 1);
```

## 21.71 SetObjectBorderColor()

```
void SetObjectBorderColor( int oid, int color)
```

### Beskrivning

Sätter kantfärg på ett objekt.

### Argument

int	oid	Objektets identitet.
int	color	Kantfärg.

### Exempel

```
SetObjectBorderColor( oid, eDrawType_CustomColor1);
```

## 21.72 SetObjectClass()

```
void SetObjectClass( int oid, string subgraph)
```

### Beskrivning

Byt ut subgrafen för ett objekt.

### Argument

int	oid	Objektsidentitet.
string	subgraph	Namn på subgrafen.

### Exempel

```
int oid;  
string class;  
  
class = GetObjectClass( oid)  
if ( class == "pwr_indsquare")  
    SetObjectClass( oid, "pwr_indround");  
endif
```

## 21.73 SetObjectFill()

```
void SetObjectFill( int oid, int fill)
```

### Beskrivning

Sätter fyllnad på ett objekt.

### Argument

int	oid	Objektets identitet.
int	fill	Objektets fyllnad, 1 eller 0.

### Exempel

```
SetObjectFill( oid, 1);
```



## 21.74 SetObjectFillColor()

```
void SetObjectFillColor( int oid, int color)
```

### Beskrivning

Sätter fyllnadsfärg på ett objekt.

### Argument

int	oid	Objektets identitet.
int	color	Fyllnadsfärg.

### Exempel

```
SetObjectFillColor( oid, eDrawType_CustomColor1);
```

## 21.75 SetObjectGradient()

```
void SetObjectGradient( int oid, int gradient)
```

### Beskrivning

Sätter gradient på ett objekt.

### Argument

int	oid	Object identity.
int	gradient	Gradient. 0 betyder ingen gradient.

### Exempel

```
SetObjectGradient( oid, 0);
```

## 21.76 SetObjectShadow()

```
void SetObjectShadow( int oid, int shadow)
```

### Beskrivning

Sätter skuggning på ett objekt.

### Argument

int	oid	Objektets identitet.
int	shadow	Objektets skuggning, 1 eller 0.

### Exempel

```
SetObjectShadow( oid, 1);
```

## 21.77 SetObjectTextColor()

```
void SetObjectTextColor( int oid, int color)
```

### Beskrivning

Sätter textfärg på ett objekt.

### Argument

int	oid	Objektets identitet.
int	color	Textfärg.

### Exempel

```
SetObjectTextColor( oid, eDrawType_CustomColor5);
```

## 21.78 SetObjectTransparency()

```
void SetObjectTransparency( int oid, float transparency)
```

### Beskrivning

Sätter grad av transparens på ett objekt.

### Argument

int	oid	Objektets identitet.
float	transparency	Grad av transparens. Ett värde mellan 0.0 och 1.0.

### Exempel

```
SetObjectTransparency( oid, 0.5);
```

## 21.79 void SetRgbColor()

void SetRgbColor( int color, float red, float green, float blue)

### Beskrivning

Sätter rgb-värden på en färg i custom color paletten. Färgen anges med färgindex eller med symbol, se nedan. rgb-värdena är flyttal mellan 0 och 1.

### Argument

int	color	Färg.
float	red	Värde på röd.
float	green	Värde på grön.
float	blue	Värde på blå.

Färger i standardpaletten finns definierade med eDrawType\_Color1 - eDrawType\_Color300, och färger i custom color paletten med eDrawType\_CustomColor1 - eDrawType\_CustomColor90.

Motsvarande färgindex kan också användas, för standardpaletten 0 - 299, och för custom color paletten 310 - 669. För custom color ligger grundfärgen på var fjärde index, dvs 310, 314, 318, ... 666.

<

### b>Exempel

```
float red = 0.731;  
float green = 0.224;  
float blue = 0.328;
```

```
SetRgbColor( eDrawType_CustomColor1, red, green, blue);
```

## 21.80 void SetSelectTextBold()

void SetSelectTextBold(int font)

### Beskrivning

Sätt fet text på utvalda objekt.

### Argument

int	bold	1 för fet text 0 för normal text.
-----	------	-----------------------------------

### Exempel

```
int id;  
  
SelectAdd(id);  
SetSelectTextBold(1);
```

## 21.81 void SetSelectTextFont()

void SetSelectTextFont(int font)

### Beskrivning

Sätt typsnitt på utvalda objekt.

### Argument

int	font	Typsnitt.
-----	------	-----------

### Exempel

```
int id;  
  
SelectAdd(id);  
SetSelectTextFont(eFont_LucidaSans);
```



## 21.82 void SetSelectTextSize()

void SetSelectTextSize(int size)

### Beskrivning

Sätt text-storleken på utvalda objekt.

### Argument

int	size	Storlek, värde i området 0-5.
-----	------	-------------------------------

### Exempel

```
int id;  
  
SelectAdd(id);  
SetSelectTextSize(2);
```

# 21.83 Exempel

## Exempel 1

```
!  
! Rita några enkla objekt  
!  
main()  
    string name;  
  
    verify(1);  
  
    ! Rita en rektangel  
    set fill  
    set shadow  
    set linewidth 1  
    set bordercolor Black  
    set fillcolor BlueHigh8  
    create rect /x1=10 /y1=1 /width=20 /height=12  
    set current attr shadow_width 2  
    set current attr gradient Globe  
  
    ! Rita en cirkel  
    set fill  
    set shadow  
    set fillcolor MagentaHigh4  
    set linewidth 2  
    create arc /x1=1/y1=1/x2=6/y2=6  
    set current attr shadow_width 12  
    set current attr gradient DiagonalLowerLeft  
  
    ! Rita en polyline  
    set fill  
    set shadow  
    set fillcolor YellowGreenHigh5  
    set linewidth 1  
    create polyline /x1=5 /y1=5 /x2=10 /y2=10  
    add polyline /x1=10 /y1=20  
    add polyline /x1=5 /y1=20  
    add polyline /x1=5 /y1=5  
    set current attr gradient DiagonalLowerLeft  
  
    ! Skriv en text  
    set bold  
    set textsize 24  
    set textfont LucidaSans  
    set textcolor RedHigh8  
    create text/text="Example" /x1=3 /y1=5
```

```

! Skapa en subgraf
set fillcolor YellowGreenMedium3
create object/sub=pwr_valve/x1=1/y1=1/x2=3/y2=2
set current attr DigLowColor.Attribute "Rt-Dv1.ActualValue##Boolean"
set current attr DigLowColor.Color GrayHigh9

! Skapa en tryckknapp
create object /sub=pwr_buttonsetcenter /x1=5 /y1=1
set current annotation "Start"
set current attr SetDig.Attribute "rt-dv1.actualvalue##Boolean"
set current attr Access System|Operator1

! Sätt graph attributes
set graphattr x0 -3
set graphattr y0 -3
set graphattr x1 40
set graphattr y1 25

save
endmain

```

## Exempel 2

```

!
! Det här exemplet letar upp alla dv-objekt i en hierarki,
! skriver ut namnet och visar värdet i en indikator.
! En ram ritas runt objekten.
!
main()

string name;
string segname;
string attr;
float x;
float y;
float x_ind;
float y_ind;
float x1;
float x2;
float y1;
float y2;
float width;
float height;
float t_width;
string class;

verify(1);

x_ind = 2;
x = 4;
y_ind = 1.5;

```

```

y = 2;
name = GetChild( p1);

if ( p1 == "" )
    printf("usage : test3 'parent'\n");
    exit();
endif

set bold
set textsize 12
while ( name != "" )
    class = GetObjectClass( name);
    if ( class == "Dv" )
        create object/sub=pwr_indsquare/x1='x_ind'/y1='y_ind'
        attr = name + ".ActualValue##Boolean";
        set current attr DigLowColor.Attribute "'attr'"
        segname = CutObjectName( name, 1);
        create text/text="'segname'"/x1='x'/y1='y'
        GetTextExtent( segname, 12, 1, t_width);
        if ( t_width > width )
            width = t_width;
        endif
        y += 1;
        y_ind += 1;
    endif
    name = GetNextSibling( name);
endwhile

x1 = x_ind - 1;
x2 = x + width + 1;
y1 = 0;
y1 = y;
width = x2 - x1;
height = y2 - y1;

set fillcolor GrayLow3
set shadow
set noborder
set nofill
create rectangle/x1='x1'/y1='y1'/width='width'/height='height'
set current attr shadow_width 2
set current attr relief Down

x1--;
x2++;
y1--;
y2++;
set background GrayLow3
set graph x0 'x1'
set graph y0 'y1'
set graph x1 'x2'
set graph y1 'x2'
endmain

```