

# GE script

## Användarhandledning

Revision: 02 01 07 Claes Sjöfors

SSAB Oxelösund

Allmänt .....	4
Skapa objekt .....	5
Rektangel .....	5
Cirkel .....	5
Linje .....	5
Polylinje .....	5
Text .....	6
Subgraf objekt .....	6
Graf attribut .....	7
Subgraf attribut .....	7
Färger .....	8
Färgton .....	8
Kommandon .....	9
add polyline .....	10
create .....	11
create rectangle .....	11
create arc .....	11
create line .....	12
create polyline .....	12
create text .....	13
create subgraph .....	13
group .....	15
move .....	16
move currentobject .....	16
/x .....	16
new .....	17
select .....	18
select currentobject .....	18
select clear .....	18
set .....	19
set verify .....	19
set noverify .....	19
set fill .....	19
set nofill .....	19
set border .....	19
set noborder .....	19
grid .....	19
nogrid .....	19
linewidth .....	19
gridsize .....	20
set textsize .....	20
set bold .....	20
set nobold .....	20
set backgroundcolor .....	20
set fillcolor .....	20
set bordercolor .....	20
set currentobject fillcolor .....	21
set currentobject colortone .....	21
set currentobjekt colorlightness .....	21
set currentobject colorintensity .....	21
set currentobject colorshift .....	21
set currentobject attributes .....	22
Type .....	22
color .....	23

colortone .....	23
access .....	23
set currentobject attr1 .....	24
set currentobject attr2 .....	24
set currentobject annotation .....	24
set graphattributes .....	25
rotate .....	26
rotate currentobject .....	26
scale .....	27
scale currentobject .....	27
save .....	28
exit .....	29
quit .....	30
open .....	31
show .....	32
show version .....	32
Script .....	33
In och utmatning .....	33
Filhantering .....	33
Hantering av strängar .....	33
Databas funktioner .....	34
System funktioner .....	34
Ge funktioner .....	34
GetTextExtent() .....	35
SelectAdd() .....	36
GetCurrentObject() .....	37
Exempel .....	40
Exempel 1 .....	40
Exempel 2 .....	41

# Allmänt

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 0
set fillcolor 19
create rect /x1=10 /y1=1 /width=4 /height=2
```

## Cirkel

### Exempel på en halvcirkel

```
set nofill
set linewidth 2
set bordercolor 0
create arc /x1=1 /y1=1 /x2=3 /y2=2 /angle1=0 /angle2
=180
```

## Linje

### Exempel på en linje

```
set linewidth 1
set bordercolor 0
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 0
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 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 45
set current attr attribute "Rt-Dv1.ActualValue##Boolean"
set current attr lowcolor 3
```

### Exempel på trycknapp

```
create object /sub=pwr_buttonset /x1=5 /y1=1
! Change type to SetDigConfirmWithTone
set current attr type 1014
set current attr text "Start"
set current attr ConfirmText "Do you really want to..."
set current attr Attribute "rt-Dv1.ActualValue##Boolean"
set current attr ToneAttr "rt-Dv2.ActualValue##Boolean"
set current attr LowTone 8
set current attr Access 65534
```

# 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
! Type ToggleDigWithTone
set graph TraceAttrType 1012
set graph TraceColor 3
set graph NoConObstacle 1
```

# Färger

Färger anges med index i färgpaletten (0 - 99). 0 ger svart, 10-19 grå nyanser, 20-29 gula, 30-39 gulorange, 40-49 rödorange, 50-59 röda, 60-69 magenta, 70-79 blåa, 80-89 turkosa och 90-99 gröna.

## Färgton

Även färgton anges med index. Se tabell nedan.

För de typer av dynamik ändrar på färgtonen kan man oftast sätta en fyllnads-färg istället för färgton. Man anger då index för fyllnadsfärgen istället (ett värde mellan 11-99).

Index	Ton
0	Ingen ton
1	Grå
2	Gul
3	Guld
4	Orange
5	Röd
6	Magenta
7	Blå
8	Turkos
9	Grön
10	Mörkgrå



# Kommandon

Nedan följer beskrivning på kommandon i Ge

# 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.

# create

Skapa grafiska element i bilden.

## 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 bredd.

## create arc

Skapa en cikel eller oval, eller ett segment av en cirkel eller oval.

### Syntax

```
ge> create arc /x1= /y1= /x2= /y2= /angel1= /angel2=
```

### **/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.

**/angel1**

Vinkel som beskriver var ett cirkel-segment startar. Anges i grader. Default-värde 0.

**/angel2**

Vinkel i grader som anger storleken på ett cirkel-segment. Default-värde 360 grader.

## 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.

## create polyline

Skapa en polyline, dvs en linje som består av flera räta del-linjer.

‘create polyline’ skapar den första del-linjen. Övriga del-linjer skapas med ‘add polyline’.

Punkten (x1, y1) är startpunkten för polylinjen. Nästa del-linje hakas på punkten (x2, y2).

### 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.

## create text

Skapa en text.

### Syntax

```
ge> create text /x1= /y1= /text=
```

**/x1**

x-koordinat.

**/y1**

y-koordinat.

**/text**

Text omgärdad av dubbel-fnuttar.

## 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 proview's bassystem har subgrafen alltid prefixet 'pwr\_' och skrivs med gemener, dvs subgrafen 'Valve' har namnet pwr\_valve.

**/x2**

Om punkten  $(x_2, y_2)$  anges skalas subgrafen så att den innesluts i rektangelns med hörnpunkterna  $(x_1, y_1)$  och  $(x_2, y_2)$ .

**/y2**

# group

Skapa en grupp av utvalda objekt.

# move

## move currentobject

Flyttar senast skapade objekt.

**/x**

Förflyttning i x-led relativt tidigare position.

**/y**

Förflyttning i y-led relativt tidigare position.



# new

Rensar arbetsarean.

# **select**

## **select currentobject**

Lägger in senast skapade objekt i listan av utvalda objekt.

## **select clear**

Tömmer listan på utvalda objekt.

# set

## set verify

Sätter verifierings-mod, dvs att exekverade script-rader skrivs ut i terminal-fönster.

## set noverify

Avslutar verifierings-mod.

## set fill

Sätter 'fill' i editorn. Skapade objekt kommer i fortsättningen att ha fill-attributet satt.

## set nofill

Återställer 'fill' i editorn. Skapade objekt kommer i fortsättningen att ha fill-attributet nollställt.

## set border

Sätter 'border' i editorn. Skapade objekt kommer i fortsättningen att ha border-attributet satt.

## set noborder

Återställer 'border' i editorn. Skapade objekt kommer i fortsättningen att ha border-attributet nollställt.

## grid

Sätter 'grid' i editorn. Skapade objekt läggs på närmaste grid-punkt.

## nogrid

Återställer grid i editorn.

## 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'
```

# gridsize

Sätter gridstorleken i editorn.

## Syntax

```
ge> set gridsize 'gridsize'
```

# 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'
```

# set bold

Sätter 'bold' i editorn. Textobjekt kommer i fortsättningen att skapas med fet-stil

# set nobold

Återställer 'bold'.

# set backgroundcolor

Sätter bakgrundfärgen. Bakgrundsfärgen är ett index i färgtabellen (0-99).

## Syntax

```
ge> set backgroundcolor 'color'
```

# set fillcolor

Sätter fyllnadsfärg i färg-paletten. Objekt kommer i fortsättningen att skapas med denna fyllnadsfärg. Fyllnadsfärgen anges med index i färgtabellen (0-99).

## Syntax

```
ge> set fillcolor 'color'
```

# set bordercolor

Sätter kantfärgen i färg-paletten. Obekt kommer i fortsättningen att skapas med denna kantfärg, Kantfärgen anges med index i färgtabellen (0-99).

### Syntax

```
ge> set bordercolor 'color'
```

## set currentobject fillcolor

Sätter fyllnadsfärg på senast skapade subgraf-objekt. Fyllnadsfärgen anges som index i färgtabellen (0-99).

### Syntax

```
ge> set currentobject fillcolor 'color'
```

## set currentobject colortone

Sätter färgton på senast skapade subgraf-objekt.

### Syntax

```
ge> set currentobject colortone 'index'
```

## 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'
```

## set currentobject colorintensity

Sätter färgintensitet på senast skapade subgraf-objekt. Intensiteten är ett heltal. Positivt värde ger intensivare färger, negativt dämpare färger.

### Syntax

```
ge> set currentobject colorintensity 'intensity'
```

## 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'
```

# 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 subgraf. 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.

## Type

Type anger vilken typ av dynamik som objektet har. Denna anges i form av ett heltal.

## Syntax

```
ge> set currentobject attributes type 'value'
```

Value	Typ av dynamik
0	Inherit
1	Dig
2	DigWithError
3	DigTone
4	DigToneWithError
5	Annot
6	DigWithText
7	Bar
8	Trend
9	DigBorder
10	AnnotWithTone
11	DigTwo
12	DigToneTwo
13	Invisible
14	Rotate
15	AnalogShift
16	Animation
17	DigAnimation

18	AnimationForwBack
19	DigShift
20	Move
1000	SetDig
1001	ResetDig
1002	ToggleDig
1003	Slider
1004	AnnotInput
1005	Command
1006	CommandConfirm
1007	SetDigConfirm
1008	ResetDigConfirm
1009	ToggleDigConfirm
1010	SetDigWithTone
1011	ResetDigWithto
1012	ToggleDigWithTone
1013	AnnotInputWithTone
1014	SetDigConfirmWithTone
1015	ResetDigConfirmWithTone
1016	ToggleDigConfirmWithTone
1017	DigWithCommand
1018	DigWithErrorAndCommand
1019	DigToneWithCommand
1020	DigToneWithErrorAndCommand
1021	StoDigWithTone
1022	DigTwoWithCommand
1023	DigToneTwoWithCommand
1024	IncrAnalog

## color

Färger benämns vanligvis 'LowColor' eller 'LowColor2'. För dess anges index i färgpaletten.

## colortone

Färgton kallas normalt för 'LowTone' och anges som ett heltal.

## access

Access är en bitmask där varje bit ger en användar-kategori behörighet att påverka objektet.

Värde	Användar kategori
1 << 0	RtRead
1 << 1	RtWrite
1 << 2	System
1 << 3	Maintenance

1 << 4	Process
1 << 5	Instrument
1 << 6	Operator1
1 << 7	Operator2
1 << 8	Operator3
1 << 9	Operator4
1 << 10	Operator5
1 << 11	Operator6
1 << 12	Operator7
1 << 13	Operator8
1 << 14	Operator9
1 << 15	Operator10

### Syntax

```
ge> set currentobject access 'value'
```

## set currentobject attr1

Sätter attributet, dvs kopplingen till rtdb, i senast skapade subgraf.  
Motsvarar funktionen vid Ctrl+Dubbelklick MB1.

### Syntax

```
ge> set currentobject attr1 'attribute'
```

## set currentobject attr2

Sätter attribut nr 2 i senast skapade subgraf.  
Motsvarar funktionen vid Ctrl+Shift+Dubbelklick MB1.

### Syntax

```
ge> set currentobject attr2 'attribute'
```

## 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'
```



# 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	String
BackgroundImage	String

Subgraf attribut

Namn	Typ
TraceAttrType	TraceType
TraceColor	ColorIndex eller ColorTone
TraceColor2	ColorIndex eller ColorTone
NoConObstacle	Int
Slider	Int
NextSubgraph	Int
AnimationCount	Int

## Syntax

```
ge> set currentobject graphattributes 'name' 'value'
```

# rotate

## rotate currentobject

Roterar senast skapade objekt runt objektets mittpunkt.

### Syntax

```
ge> rotate currentobject /angel=
```

### /angel

Vinkel i grader som objektet ska roteras.

# scale

## 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.

# save

Sparar en graf.

## Syntax

```
ge> save ['filename']
```

# exit

Avsluta ge och spara aktuell graf först.

## Syntax

```
ge> exit ['filename']
```

# quit

Avsluta utan att spara.

## Syntax

```
ge> quit
```

# open

Öppna en graf.

## Syntax

```
ge> open ['filename']
```

Om något av objekten redan är kopplat görs disconnect före kopplingen.

# show

## show version

Visa Ge-version.



# Script

Ge-script är ett sätt att programmera ge-kommandon. Scripthanteraren ger dessutom möjlighet till att göra beräkningar, utföra villkors-satser, loop-satser, deklarerara 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.

## In och utmatning

Funktion	Beskrivning
ask	Skriver ut en fråga och läser in ett svar
say	Skriver ut en sträng
printf	Formaterad utskrift
scanf	Formaterad inläsning

## Filhantering

Funktion	Beskrivning
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

## Hantering av strängar

Funktion	Beskrivning
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
toupper	Konvertera till versaler

## Databas funktioner

Funktion	Beskrivning
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

Funktion	Beskrivning
exit	Avsluta exekveringen av ett skript
time	Hämta systemtiden
system	Exekvera ett DCL-kommando
verify	Sätt verify på eller av

## Ge funktioner

Funktion	Beskrivning
GetCurrentObject	Returnerar identiteten för senast skapade objekt.
GetTextExtent	Beräkna storleken på en text.
SelectAdd	Addera objekt till utvalslistan.
SetExtern	Sätt en subgraf extern.
SetIntern	Sätt en subgraf intern.

# GetTextExtent()

**int GetTextExtent( string text, int textsize, int bold, float width [, 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	Textstorlek (8, 10, 12, 14, 18 eller 24).
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", 12, 1, width);
```

# 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);
```

# GetCurrentObject()

**int GetCurrentObject()**

## **Beskrivning**

Returnerar identiteten på senast skapade objekt..

## **Exempel**

```
int id;
```

```
id = GetCurrentObject();
```

# SetExtern()

**int SetExtern( string name)**

## **Beskrivning**

Sätter en subgraf extern.

## **Argument**

string	name	Namn på subgrafen.
--------	------	--------------------

## **Exempel**

```
SetExtern( "MySubgraph" );
```

# SetIntern()

**int SetIntern( string name)**

## **Beskrivning**

Sätter en subgraf intern.

## **Argument**

string	name	Namn på subgrafen.
--------	------	--------------------

## **Exempel**

```
SetIntern( "MySubgraph" );
```

# Exempel

## Exempel 1

```
!  
! Rita några enkla objekt  
!  
main()  
    string name;  
  
    verify(1);  
  
    ! Rita en rektangel  
    set fill  
    set linewidth 1  
    set bordercolor 0  
    set fillcolor 19  
    create rect /x1=10 /y1=1 /width=2 /height=2  
  
    ! Rita en halvcirkel  
    set nofill  
    set linewidth 2  
    create arc /x1=1/y1=1/x2=3/y2=2/angel1=0 /angel2=180  
  
    ! Rita en polyline  
    set nofill  
    set linewidth 1  
    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  
  
    ! Skriv en text  
    set bold  
    set textsize 14  
    create text/text="Example" /x1=3 /y1=5  
  
    ! Skapa en subgraf  
    create object/sub=pwr_valve/x1=1/y1=1/x2=3/y2=2  
    set current fillcolor 45  
    set current attr attribute "Rt-  
Dv1.ActualValue##Boolean"  
    set current attr lowcolor 0  
  
    ! Skapa en trycknapp  
    create object /sub=pwr_buttonset /x1=5 /y1=1  
    set current attr text "Start"
```



```

    set current attr attribute "rt-
dv1.actualvalue##Boolean"
    set current attributes access 65000

    ! Sätt graph attributes
    set graph x0 -3
    set graph y0 -3
    set graph x1 40
    set graph 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 t_width;
    string class;

    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 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;
cre
obj/sub=pwr_framethin/x1='x1'/y1='y1'/x2='x2'/y2='y2'

x1--;
x2++;
y1--;
y2++;
set background 11
set graph x0 'x1'
set graph y0 'y1'
set graph x1 'x2'
set graph y1 'x2'
endmain

```