

<http://www.goto10.org/-/pdgop.html>

PureData "Graph on parent" simple howtos

by chun lee

version 080206 - second version, new section added for pd 0.39

version 280205 - first version, made for pd pre 0.39

INTRO

Pd externals and abstractions conventionally have the appearance of (transparent) rectangular box that holds the name of the object and its creation arguments in the middle(fig.1). however, the "Graph on parent" feature allows abstractions to have customized appearance in which typically consists of Pd GUI objects such as sliders, toggle switch, number box and so on(fig.2). in other words, the "Graph on parent" feature literally turns an abstraction into a graph when created and hides all the cables while displaying the gui elements only.

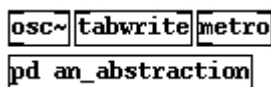


fig.1 the conventional look of Pd objects and abstractions



fig.2 Graph and Parent enables customized look. note that the inlets and outlets are still at where they should be.

The Graph on parent therefore provides a convenient way to interact/control abstractions, especially ones that are complex or full of features. this documentation aims explain the procedures involved in using the Graph On Parent feature.

for pd 0.39

in pd 0.39, GOP is much much easier than previous versions. once you have made your dream patch, all you need to do is right click on the top-level patch and select 'Properties' and you will get this window



fig.3 the Properties window

now, enable the 'graph on parent' option by click on it. the two X and Y 'size' fields will determin how big your gop will be. in this instance, its a 100x100 pixel square. the X and Y 'margin' fields determines the positioning of the 'red box' on your patch. in this example. the red magic box will appear at the most top-left corner of your patch. 'red box'??!



fig.4 shows the magical red box

once you clicked on the 'Apply' or 'OK' button on the Properties window, a red bounding box will appear on your patch. this red box represents the boundaries of the GOP. all you need to do now is put the GUI elements inside of this red box, arrange them and save the patch.

note that the positioning of the red box on your GOP patch does not effect the final look of the GOP. for example, the following patch will produce the same GOP as the patch above.



fig.5 red box is now in different position, but will give the same look at the end

the final look of this patch as GOP...



fig. 6 the final look...

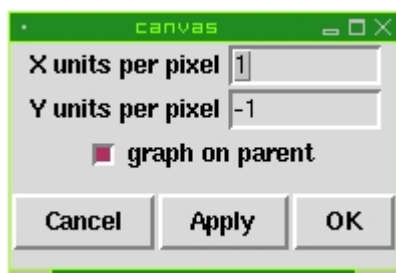
note that in previous versions of Pd, the comment object would not appear in GOP. however, in pd-0.39, comments will appear in the GOP as showed in the example patch. messages still remain hidden.

regarding to the decoration of GOP, its the same as before, so have a read towards the end of the pre 0.39 part of this doc.

for pd pre 0.39

HOW?

All that is needed for your abstraction to have a "face" when instantiated is to enable the Graph on Parent at the top-level patch window of your abstraction. you can enable it by right click (in edit mode) on any empty space and select "Properties" and check the "Graph on Parent" box (fig.3). now save the abstraction and next time you call it, it will have a face:) Please note that the top-level patch window refers to the top most window of your abstraction patch, which contains everything else of your abstraction.




```

#X floatatom 332 21 5 0 0 0 - - -;
#X floatatom 30 10 5 0 0 0 - - -;
#X obj 30 44 tgl 15 0 empty empty empty 0 -6 0 8 -262144 -1 -1 0 1;
#X obj 167 153 hsl 50 15 0 127 0 0 empty empty empty -2 -6 0 8 -262144 -1 -1 0 1;
#N canvas 0 0 450 300 dummy 0;
#X obj 65 90 inlet;
#X obj 109 89 inlet;
#X obj 156 87 inlet;
#X obj 200 86 inlet;
#X restore 122 104 pd dummy;
#X obj 107 1 inlet;
#X msg 99 27 messages will be hidden;
#X connect 0 0 5 0;
#X connect 1 0 5 3;
#X connect 2 0 5 2;
#X connect 3 0 5 1;
#X connect 4 0 5 0;
#X connect 6 0 5 2;
#X connect 7 0 5 2;
#X coords 0 0 1 1 200 140 1;

```

the only parts that concern with our problem is the first line **"#N canvas 519 63 362 203 10;"** and the very last line **"#X coords 0 0 1 1 200 140 1;"**. in the first line, the string of 5 numbers after "canvas" are X, Y coordinates of the top-left corner of the patch window, the width and height of the window and the font size. in the last line the string of 5 numbers after the "coords 0 0" are X,Y units per pixel, width and height of the Graph on Parent object box/canvas and Graph on Parent on/off flag. Please note that the "#X coords" line does not exist by default, it will only appear when you checked the Graph on parent box in the properties window and then save.

what's happening here is that the size of the abstraction's top window is 362x203 and when its instantiated, the canvas that is used to display the abstraction is 200x140(default). and that's why everything seems a bit squashed:)

so, all we need to do to correct this is to set the abstraction's display canvas to the same size of the abstraction's top-level window size. you can do this by either editing the text file and save it, or right click on the abstraction's display canvas and set the screen width/height in its properties window. please note that if you use the latter method, don't forget to open the abstraction after setting the width/height and SAVE!

now, the abstraction looks like this:



Fig.10 now GUI elements have the same distance and proportion as to the original abstraction's parent patch. well, almost, because of the scroll bars:)

also, one trick that you can do is to simply put a canvas as the background of all the gui element you want to include in the abstraction and resize the patch window to fit the canvas. after this, you can get the needed dimensions of the abstraction from the properties of the canvas.

OK, that's basically it (or AFAIK about Graph on Parent feature)!

but, but

SOME MORE!

just a few more things:)

-if you want to make the Graph on Parent abstraction compact, try to arrange all the GUI elements together at the top left corner and re-size the parent window to the size that you want the abstraction's display canvas to be and save. and then modify the display canvas size accordingly as we mentioned before.

-you might also want to give a bit margin for the height so that the abstraction's name and creation arguments can be displayed properly.

-add some color, perhaps:) Issues concerning adding colors to your patch can be found here ([pd-color](#)).

the look of the same abstraction but cleaned up with my favorite color:)

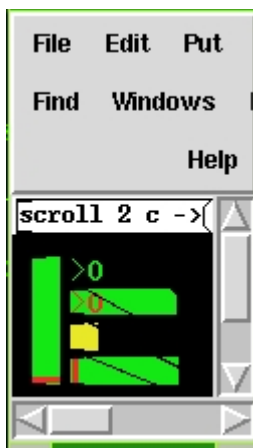


Fig.11 re-size the parent window

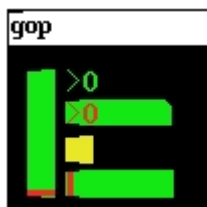


Fig.12 finished!!!!

the text file:

```
#N canvas 519 63 100 100 10;  
#X obj 0 19 cnv 15 100 80 empty empty empty 20 12 0 14 -1 -66577 0;  
#X obj 9 31 vsl 15 60 0 127 0 0 empty empty empty 0 -8 0 8 -24198 -258699 -1 0 1;
```

```
#X obj 28 62 tgl 15 0 empty empty empty 0 -6 0 8 -241291 -1 -1 0 1;
#X obj 31 79 hsl 50 15 0 127 0 0 empty empty empty -2 -6 0 8 -24198 -258699 -1 0 1;
#N canvas 0 0 450 300 dummy 0;
#X obj 65 90 inlet;
#X obj 109 89 inlet;
#X obj 156 87 inlet;
#X obj 200 86 inlet;
#X restore 120 122 pd dummy;
#X obj 105 19 inlet;
#X msg 97 45 messages will be hidden;
#X obj 28 29 nbx 5 14 -1e+37 1e+37 0 0 empty empty empty 0 -6 0 10 -1 -24198 -1 0
256;
#X obj 28 45 nbx 5 14 -1e+37 1e+37 0 0 empty empty empty 0 -6 0 10 -24198 -258699
-1 0 256;
#X obj 155 73 outlet;
#X msg 2 1 scroll 2 c ->;
#X connect 1 0 4 0;
#X connect 2 0 4 1;
#X connect 3 0 4 0;
#X connect 5 0 4 2;
#X connect 6 0 4 2;
#X connect 7 0 4 3;
#X connect 8 0 4 2;
#X coords 0 0 1 1 100 100 1;
```