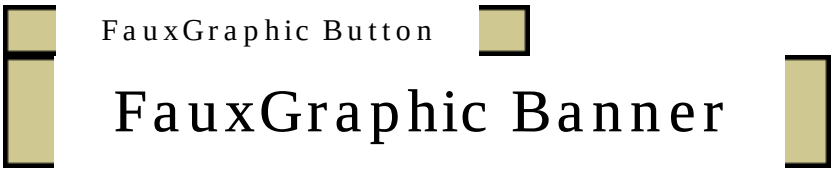
FauxGraphic 5.0

Getting Started

FauxGraphic 5.0 is a powerful component that allows to create pseudo graphics wherever you need them!

Banners, buttons, Stacked pages with custom title graphics - the FauxGraphic can do that and more.

What does a FauxGraphic look like? Anything you like!



A FauxGraphic is based on a table 1 row x 3 columns.

FauxGraphics can be any length or height - it is up to you!

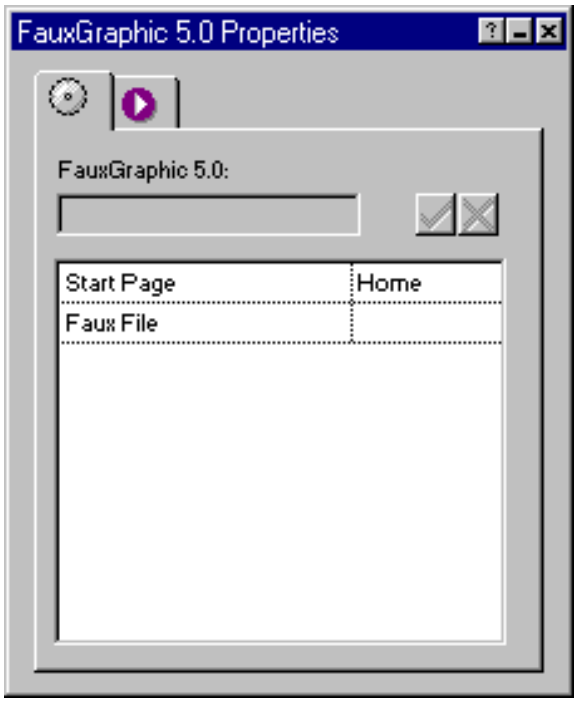
The FauxGraphic ships with 3 styles. Each style has a range of color options. You can also create your own Faux styles, and edit the Faux template file, giving you full control of the component.

You can download additional FauxStyles at the Download Center!

To get started with FauxGraphic:

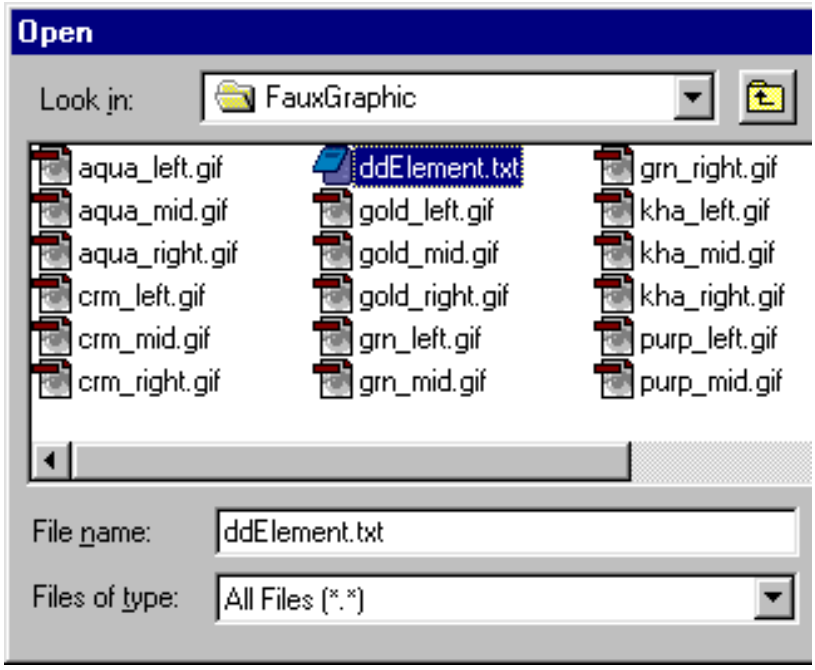
- Run the Installation program, specifying the location of Fusion 4.0/5.0's component directory
- Launch Fusion
- If the coolmaps.com component toolbar is not visible choose View > Toolbars > Component Tools > coolmaps.com components
- Select the FauxGraphic component and drag out an area on the page

The FauxGraphic properties now appears in Fusion's properties palette. The Palette is almost empty.

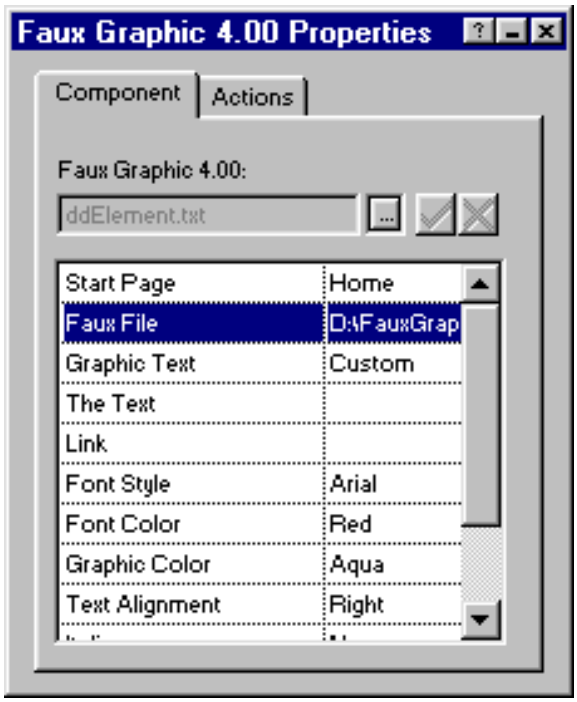


We need to specify a template file for the component to draw on.

- Select the Faux File option in the Palette.
- Browse to the FauxStyles directory.
  default location:
  C:/NetObjects Fusion 5.0(4.0)/Components/coolmaps/Fauxstyles/
- Select one of the three styles and locate the text file within the chosen style's directory.



Once the template is selected, the FauxGraphic Properties will be available:



FauxGraphic Options

The Fauxgraphic template represents one *style*. Within that one *style* you can select from many options, including color, font, links etc.

You can create your own styles and templates, making the FauxGraphic a very versatile component!

Start
Indicates which page the component was first placed on. (If you should ever choose to remove the FauxGraphic, the component should always be deleted from the page it was placed on initially.)

Faux File
Browse to the Faux template file you wish to use.

Graphic Text (Button, Banner, Title, Page, Custom)
Define the text that will appear in your FauxGraphic. When 'Custom' is chosen an additional field appears for the text.

Link
Define the location the FauxGraphic will be hyperlinked to.

Font Style (Arial, Verdana, Times)
Select the font face for the text that will appear in the FauxGraphic.

Font Color
Select a color for the text.

Graphic Color
Select the color of the graphic style

Text Alignment (left, right. center)

Define the alignment of the text within the FauxGraphic

Italics (yes,no)

Font Size
Select a font size from 8-36pt. 8 is recommended.

Bold (off,on)

FauxGraphic Styles

Fauxgraphic ships with three styles:

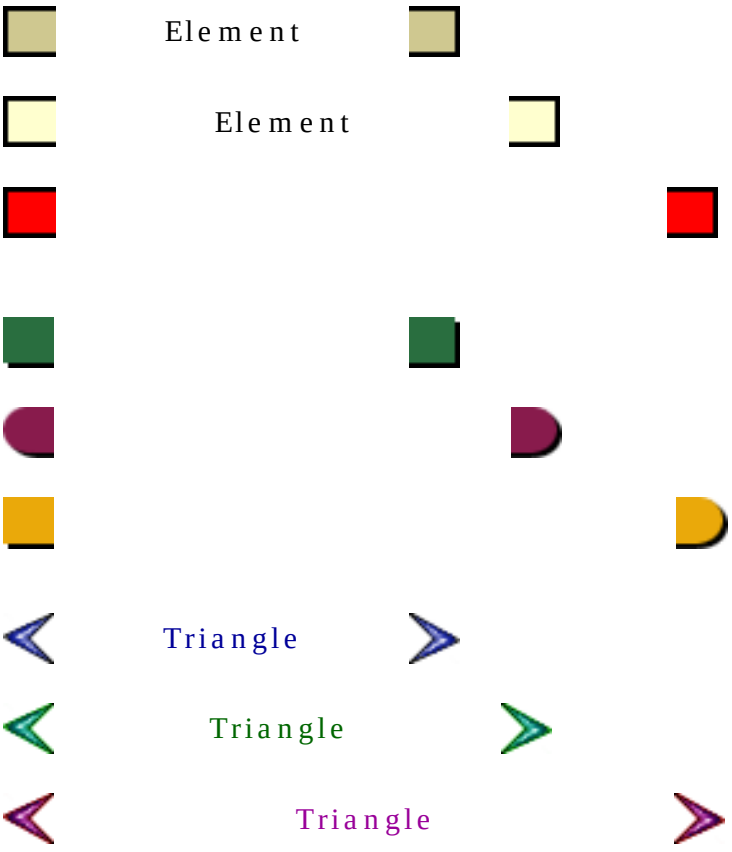
- Element
- Shadow
- Triangles

Once the text file that defines the selected style has been loaded into the Properties Palette in Fusion, the options for that style will be available.

The real power of the FauxGraphics is that the user can create and configure their own styles! Please refer to the Style Tutorial.

You can also download additional FauxStyles at the Download Center!

Below are samples from each style:



Making your own Faux styles

Making your own Faux Styles is fun and easy. The best way to make your own Style is to base it on an existing template. This tutorial is based on the Triangle style (located in C:\NetObjects Fusion 4.0 \Components\coolmaps\FauxStyles).

To help get started you can download additional FauxStyles at the Download Center!

Open the Triangle dir and you will notice that there are seven files in GIF format, and two text files. The gif files are named with the following convention: *style_color_position*

Step 1: Creating the Graphics

- Open up your image editor.
- Create three graphics, all the same size, for the left bullet, the middle section, and the right bullet.

The graphics do not need to be a particular size, but we have been using 20pxls * 20pxls images when creating these graphic sets for our examples. The sizes of the graphics are set by the text file that controls Faux Styles and can be changed to suit.

- Save your files with the aforementioned logical style VAR1_VAR2_VARside.gif (e.g. t_blu_right.gif). This will make your styles easier to manage later on

Step 2: Editing the Text Files

Open the file faux_triangles.txt. Click icon to view 

The first part of this file is a variable declaration for style, and the variable name, as you can see, is called Graphic Style.

There is only one sub-style for this example, named Triangle. This particular declaration will create a drop-down menu within the Faux Component's Object Properties menu that will have one selectable variable called Triangle, which is equal to t.

The variable names can be set to your preferences, the left side will show up in NOF, and the right side is used in the code (be sure this matches the first letter in your filenames for their respective sub-styles!).

When Triangle is selected from the list, the variable for Var1, which we declared as Graphic Style, is now set to t. This corresponds to the first variable in the files we named earlier. Note that this variable, Graphic Style, is referenced later at the end of the document.

The next variable is Graphic Color. This variable, like the first, shows up in the object properties box with selections: Blue, Red, or, Green. Selecting Blue sets the variable Graphic Color to blu, which is our filename variable for the blue triangles and backgrounds. The second part of this variable is just the respective color in HTML format, for those browsers that do no support multiple layers of graphic files

After configuring the variables, we move to the end of the text. Here we see a declaration that ends the variables that will be displayed by NOF in the Object Properties box of the FauxGraphic component.

Next, we see the width and height settings for the two end graphics, the settings are easily adjusted here to whatever size you want to make those graphics. The rest of the code, save three lines, is the HTML coding that puts everything together in the end. You don't need to modify this, but realize that there are two different versions of this code, one for the regular files and one for "_c" files. .

The last three lines of the text determine which filenames to copy over for the sub-style, and color. This is where the variables that we declared earlier come into play.

The pound symbols are the marker for a variable name, and those variables will be replaced by the style and color selected by the user from within NOF. Therefore, if a user selects the Triangle style in Blue, the declaration #Graphic Style#_#Graphic Color#_left.gif is replaced with t_blu_left.gif. There are three Save declarations here, left graphic, center graphic, and right graphic, listed respectively.

Note that in this example we do not use a variable with the center graphic, bg.gif. You could make a matching center file for each set of ends. To do this you would simply name the center graphics, keeping the variables in mind as we mentioned before for the ends, (e.g. blu_bg.gif), and then add the variable declaration to the second save statement (e.g. Save2=#Graphic Color#_bg.gif).

You can add or subtract variables for use in this manner as you see fit to organize your graphics for your style set.