

blinkdagger

an Engineering and MATLAB blog

- [Home](#)
- [Listchecker](#)
- [MATLAB](#)
- [Contact](#)
- [About](#)

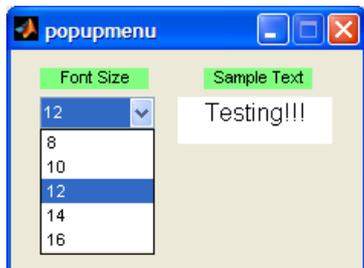
[Matlab GUI Tutorial - Pop-up Menu](#)

29 Oct 2007 [Quan Quach](#) [137 comments](#) 20,912 views

Introduction



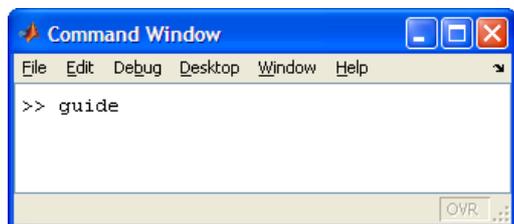
In this Matlab GUI tutorial, you will learn how to create and use the *Pop-up Menu* component. Pop-up menus are used as a control for choosing between a set of options. When the user clicks on the Pop-up menu, the menu expands, revealing a set of choices that the user can pick. A common use for Pop-up menus is a font size selector (shown below).



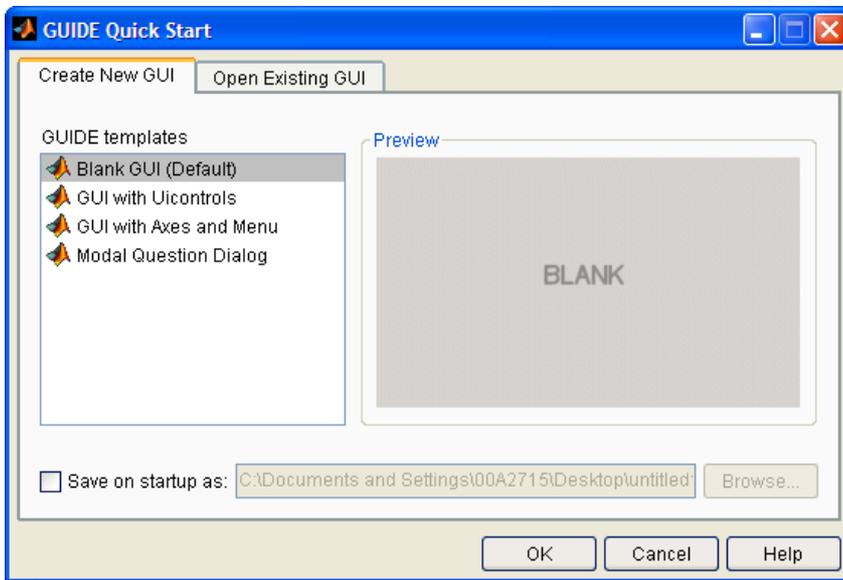
This tutorial is written for those with little or no experience creating a Matlab GUI (Graphical User Interface). If you're new to creating GUIs in Matlab, you should [visit this tutorial first](#). Basic knowledge of Matlab is not required, but recommended. Matlab version 2007a is used in writing this tutorial. Both earlier versions and new versions should be compatible as well (as long as it isn't too outdated). Let's get started!

Create the Visual Aspect of the GUI

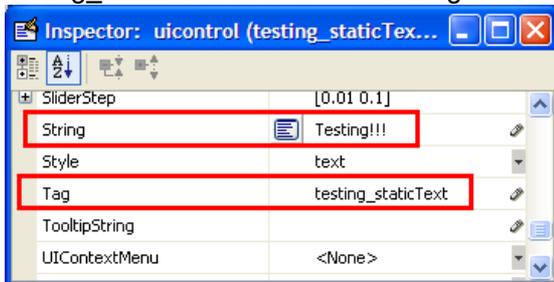
1. First, open up Matlab. Go to the command window and type in `guide`.



2. You should see the following screen appear. Choose the first option Blank GUI (Default).

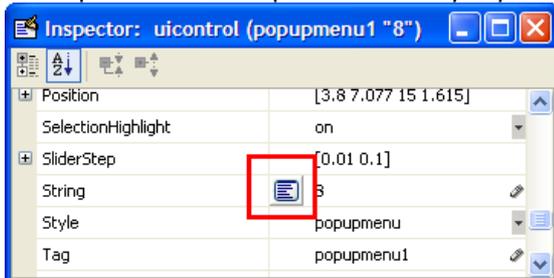


- Click on  and add a *Static Text* component to the GUI figure. Next, click on  and add a *Pop-up Menu* component onto the GUI figure.
- Double click the *Static Text* component to bring up the Property Inspector. Change the *String* property to Testing!!!, and change the *Tag* property to testing_staticText as shown in the figure below:

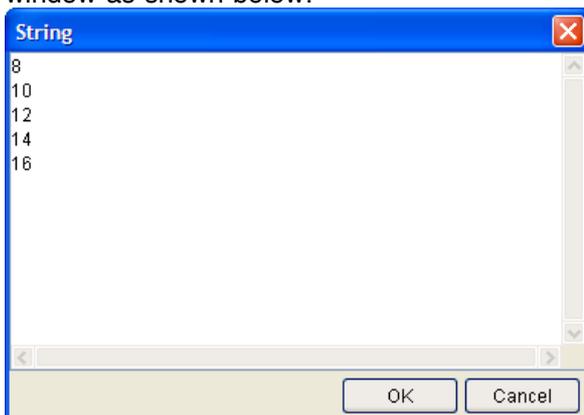


You can also modify the *BackgroundColor* property if you desire.

- Next, let's modify the properties of the *Pop-up Menu* component. First, click on the icon on the *String* property line as shown below. This allows you to edit the description for each option in the *Pop-Up Menu*.

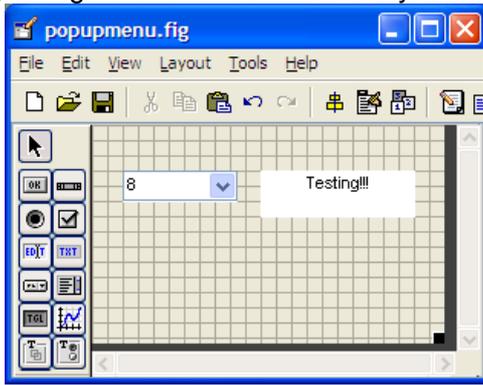


After clicking on the icon, you should now see the following window. Fill in the window as shown below:



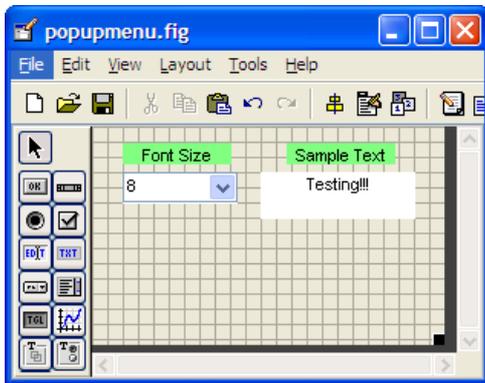
In addition, I set the *Tag* property to `popupmenu1`, which is the default name. You might want to make sure that its named properly before you move on.

6. Here's what your figure should look like after you add the components and



modify them.

7. At this point, you also might want to add some *Static Text* components to add some description tags to the GUI. You can modify their text by double clicking on the component and changing the *String* property. It's not required, but I highly recommend it.

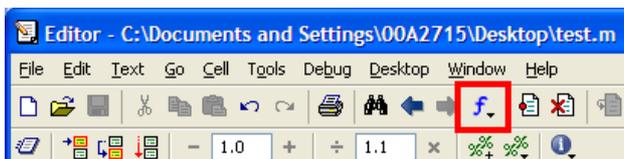


8. Save your GUI wherever you please with your desired filename.

Writing the Code for the GUI

Matlab automatically generates an `.m` file to go along with the figure that you just put together. The `.m` file is where we attach the appropriate code to the callback of each component. For the purposes of this tutorial, we are primarily concerned only with the callback functions. You don't have to worry about any of the other function types.

1. Open up the `.m` file that was automatically generated when you saved your GUI. In the Matlab editor, click on the  icon, which will bring up a list of the functions within the `.m` file. Select `popupmenu1_Callback`.



Add the following code to the function:

```
%gets the selected option
switch get(handles.popupmenu1,'Value')
case 1
    set(handles.testing_staticText,'FontSize',8);
case 2
    set(handles.testing_staticText,'FontSize',10);
case 3
    set(handles.testing_staticText,'FontSize',12);
```

```

case 4
    set(handles.testing_staticText,'FontSize',14);
case 5
    set(handles.testing_staticText,'FontSize',16);
otherwise
end

```

2. Lets quickly go over the code now. The following line of code gets the option that the user selected. Remember that in the visual layout of the GUI, we designated five different font sizes for the *Pop-up Menu* component, giving us five different options for the *Pop-up Menu*. So for example, if the user selected a font size of 8 (which was the first option in the *Pop-up Menu*), then the following line of code would return a value of 1. If the user selected a font size of 10, then the value returned would be 2, and so on.

```
get(handles.popupmenu1,'Value')
```

Depending on the option selected, the font of the *Static Text* component will be adjusted using the following line of code for each case statement:

```

%where ## is the appropriate fontsize value
set(handles.testing_staticText,'FontSize',##);

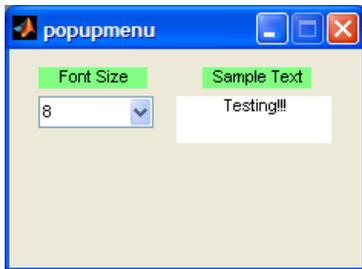
```

3. Save your m-file!

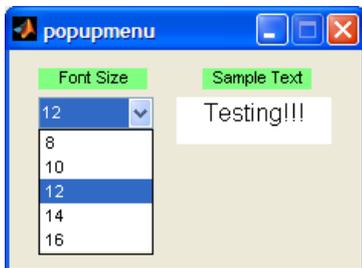
Run and Test the GUI

Now that we've completed both the visual and code aspects of the GUI, its time to run the GUI to make sure it works.

1. From the m-file editor, you can click on the  icon to save and run the GUI. Alternatively, from the GUIDE editor, you can click on the  to launch the GUI. The following GUI should appear once you click the icon:



2. Go ahead and try selecting different font sizes. If everything was done correctly, you should see the font size of the sample text change accordingly.



3. And that's it. Those are the basics of using a *Pop-up Menu* component. You can explore the other options that the slider has to offer through the Property Inspector.

This is the end of the tutorial.

Source files can be downloaded [here](#).

