

[blinkdagger](#)

an Engineering and MATLAB blog

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

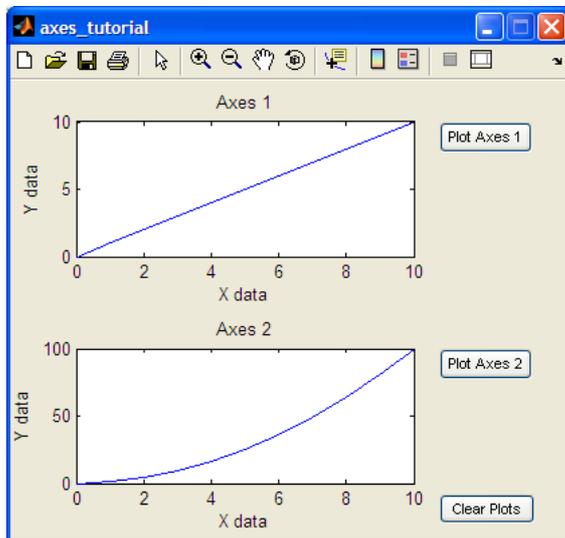
[MATLAB GUI Tutorial - Plotting Data to Axes](#)

31 Oct 2007 [Quan Quach](#) [176 comments](#) 25,632 views

Introduction



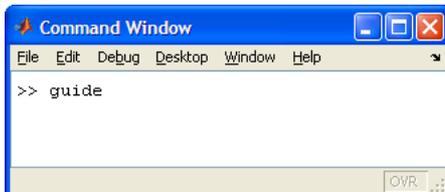
In this Matlab GUI tutorial, you will learn how to create and use the *Axes* component. The *Axes* component allows you to display graphics, such as graphs and images on your GUI. In this tutorial, we will create two axes on the GUI and plot some simple data onto it. In addition, we will include a reset button to clear the axes and we will also add the standard toolbar to allow the user to zoom, pan, and query the plot.



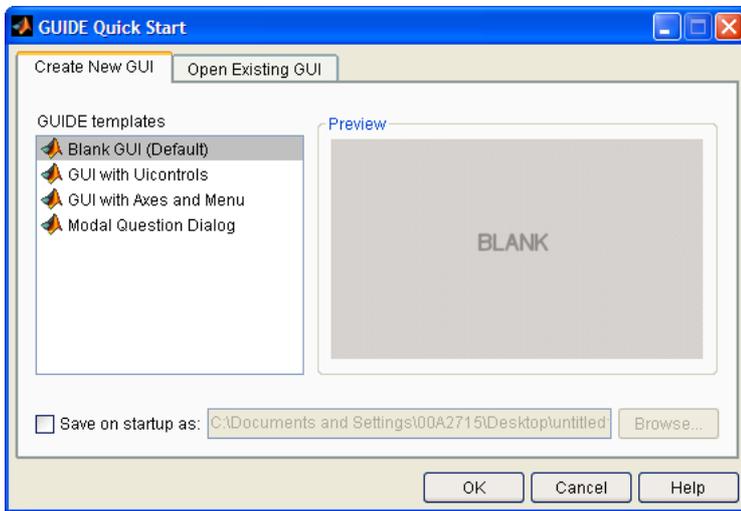
This tutorial is written for those with little or no experience creating a Matlab GUI (Graphical User Interface). If you're new to creating GUIs in Matlab, you should [visit this tutorial first](#). Basic knowledge of Matlab is recommended. Matlab version 2007a is used in writing this tutorial. Both earlier versions and new versions should be compatible as well (as long as it isn't too outdated).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

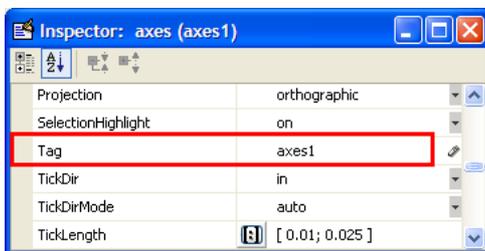
1. First, open up Matlab. Go to the command window and type in `guide`.



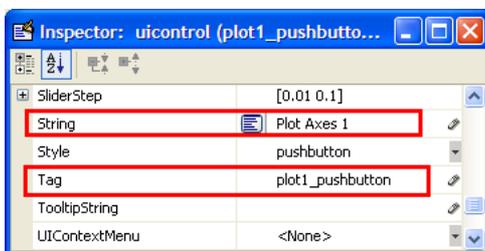
2. You should see the following screen appear. Choose the first option Blank GUI (Default).



- Click on  and add two *Axes* components to the GUI figure. Next, click on  and add three *Pushbutton* components onto the GUI figure.
- Double click the *Axes* component to bring up the Property Inspector. Change the *Tag* property to *axes1*, which should already be the default name. Additionally, make sure the other *Axes* component's *Tag* property is named *axes2*.



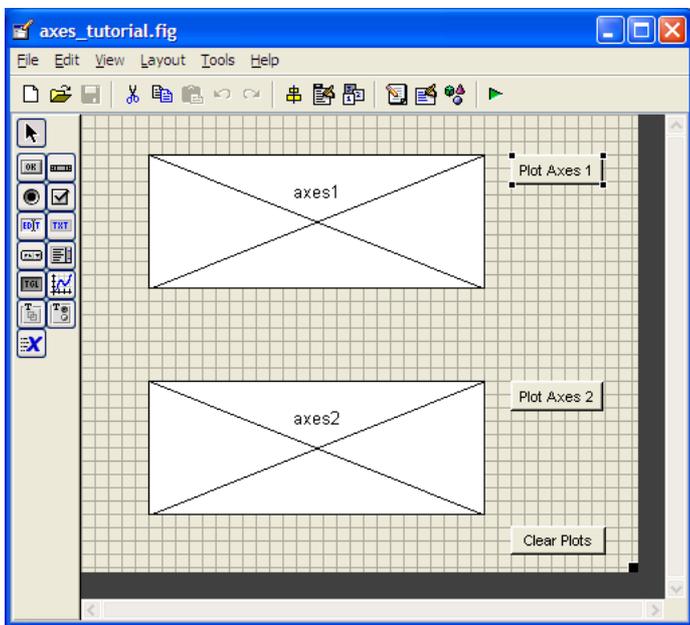
- Next, let's modify the properties of the *Pushbutton* components. Double click on one of the *Pushbutton* components. Change the *String* property to Plot Axes 1, and the *Tag* property to *plotAxes1_pushbutton*, as shown below.



Similarly, double click on the next pushbutton and change the *String* property to Plot Axes 2 and change the *Tag* property to *plotAxes2_pushbutton*.

Finally, double click on the final pushbutton and change the *String* property to Clear Axes and change the *Tag* property to *clearAxes_pushbutton*.

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

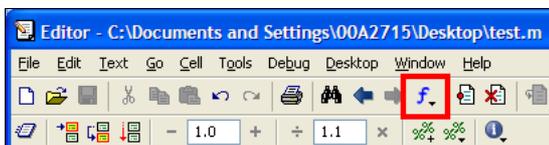


7. Save your GUI wherever you please with your desired filename.

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 `plot1_pushbutton_Callback`.



Add the following code to the function:

```
%selects axes1 as the current axes, so that
%Matlab knows where to plot the data
axes(handles.axes1)

%creates a vector from 0 to 10, [0 1 2 3 . . . 10]
x = 0:10;
%creates a vector from 0 to 10, [0 1 2 3 . . . 10]
y = 0:10;

%plots the x and y data
plot(x,y);
%adds a title, x-axis description, and y-axis description
title('Axes 1');
xlabel('X data');
ylabel('Y data');
guidata(hObject, handles); %updates the handles
```

2. Similarly, we want to put the following code into the `plot2_pushbutton_Callback`:

```
%selects axes2 as the current axes, so that
%Matlab knows where to plot the data
axes(handles.axes2)

%creates a vector from 0 to 10, [0 1 2 3 . . . 10]
x = 0:10;
%creates a vector [0 1 4 9 . . . 100]
y = x.^2
```

```

%plots the x and y data
plot(x,y);
%adds a title, x-axis description, and y-axis description
title('Axes 2');
xlabel('X data');
ylabel('Y data');
guidata(hObject, handles); %updates the handles

```

3. Next, we need to add some code to the *clearPlots_pushbutton_Callback*:

```

%these two lines of code clears both axes
cla(handles.axes1,'reset')
cla(handles.axes2,'reset')
guidata(hObject, handles); %updates the handles

```

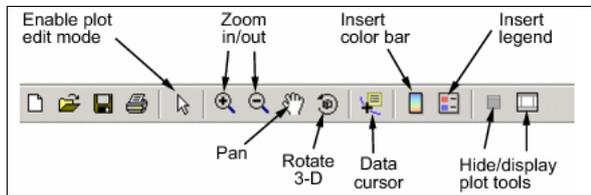
4. And finally, we need to add the following line of code to *axes_tutorial_OpeningFcn*:

```
set(hObject,'toolbar','figure');
```

This line of code should be placed right before:

```
guidata(hObject, handles);
```

This line of code effectively adds the standard toolbar to the GUI, allowing the user to zoom, pan, query the plot, and more. The standard toolbar and a brief description of the icons are shown below:

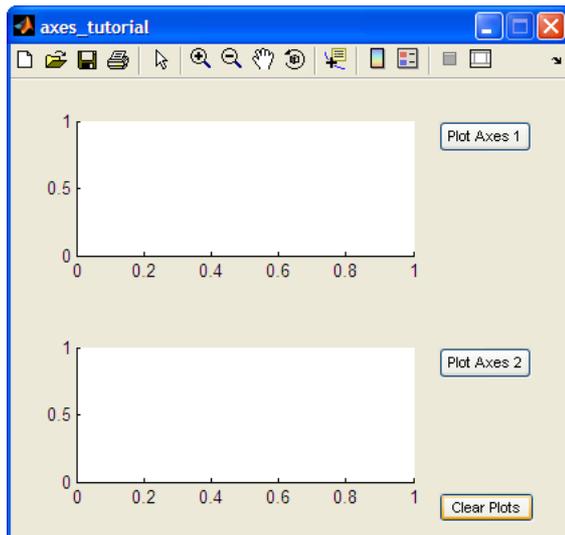


5. Save your m-file!

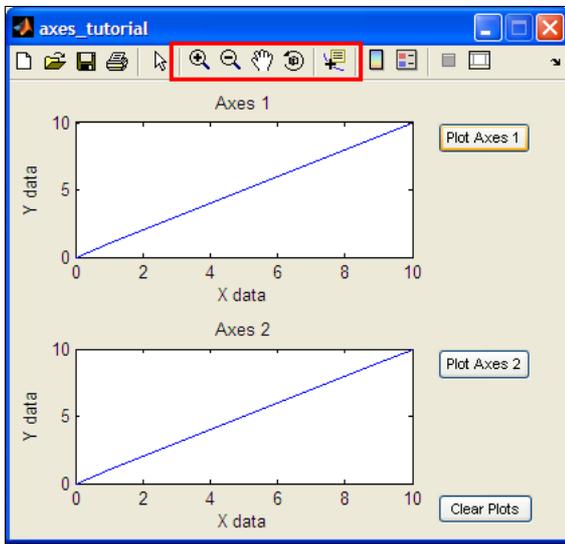
Run and Test the GUI

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 pressing all of the buttons to make sure they work. If everything was done correctly, you should see the following plots. Also, you can use the icons that are within the red box to test out the other functions.



3. And that's it. Those are the basics of using the Axes component. You can explore the other options that the axes has to offer through the Property Inspector.

This is the end of the tutorial.

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



176 Responses to "MATLAB GUI Tutorial - Plotting Data to Axes"

1. on 12 Jan 2008 at 12:48 pm [1](#) Vaibhav Bedia

When i close MATLAB and start guide again my axes loses its TAG and hence becomes invisible. How do i solve this?

2. on 20 Jan 2008 at 4:48 pm [2](#) Alex

nice one! U need to write some more stuff for beginners 😊

3. on 13 Feb 2008 at 3:59 pm [3](#) Saikat

Thank you so much I love your style and find your instructions very helpful. Going through the programs you have written for us, the beginners, I feel more confident of writing my own codes
Thank you once again

4. on 17 Feb 2008 at 11:03 am [4](#) Tanty

What a great tutorial.
Im working on my final project, so this tutorial help me solve my problem writing the code for axes.
Thanks.

5. on 17 Mar 2008 at 9:57 pm [5](#) lovejoy

how to plot a resultant vector in GUI ex. $10N+5N$

6. on 26 Mar 2008 at 4:28 am [6](#) nola

very nice tutorial..however i had a query for you..
when i do a zoom on the image..and click one pushbutton to make an action..the image return to its initial dimensions(i loose the zoom)...could you help me on this subject?

7. on 26 Mar 2008 at 12:37 pm [7](#) Daniel Sutoyo

Hi Nola

good question... although I haven't tried to code it yet, what you need to do is to get the new axes info and store it somewhere.

For example, your axes is called handles.axes1. You will have to use the `get()` and get the min,max axis on handles.axes1 and store in a variable for example myaxis.