

Ex. No. 5 Flow Control – Programming Constructs and User Interaction

A user interface (UI) is a graphical display in one or more windows containing controls, called components that enable a user to perform interactive tasks. The user does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user does not need to understand the details of how the tasks are performed. UI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders—just to name a few. UIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other UIs, and display data as tables or as plots. The following figure illustrates a simple UI that you can easily build yourself.

Graphical user interfaces, or GUIs, are essentially objects that allow users to have input using graphical interfaces such as push buttons, sliders, radio buttons, toggle buttons, pop-up menus, and so forth. GUIs are an example of object oriented programming in which there is a hierarchy. For example, the parent may be a Figure Window and its children would be graphics objects such as push buttons and text boxes. The parent user interface object can be a figure, uipanel, or uibuttongroup. A figure is a Figure Window created by the figure function. A uipanel is a means of grouping together user interface objects (the ui stands for user interface). A uibuttongroup is a means of grouping together buttons (both radio buttons and toggle buttons).

In MATLAB there are two basic methods for creating GUIs: writing the GUI program from scratch, or using the built-in Graphical User Interface Development Environment (GUIDE). GUIDE allows the user to graphically lay out the GUI, and then MATLAB generates the code for it automatically. However, in order to be able to understand and modify this code, it is important to understand the underlying programming concepts. Therefore, this section will concentrate on the programming methodology.

The UI contains these components:

- ✓ An axes component
- ✓ A pop-up menu listing three data sets that correspond to MATLAB functions: peaks, membrane, and sync
- ✓ A static text component to label the pop-up menu
- ✓ Three buttons that provide different kinds of plots: surface, mesh, and contour
- ✓ When you click a push button, the axes component displays the selected data set using the specified type of 3-D plot.

There are different ways to build MATLAB apps:

1. “Use App Designer”
2. “Use GUIDE”
3. “Use MATLAB Functions to Create Apps Programmatically”

Each of these approaches offers a different workflow and a slightly different set of functionality. The best choice for you depends on your project requirements and how you prefer to work.

Use App Designer

App Designer is a rich drag-and-drop environment introduced in R2016a, and it is the recommended environment for building most apps. It includes a fully integrated version of the MATLAB editor. The layout and code views are tightly linked so that changes you make in one view immediately affect the other. A larger set of interactive controls is available, including gauges, lamps, knobs, and switches. Most graphics functionality is supported.

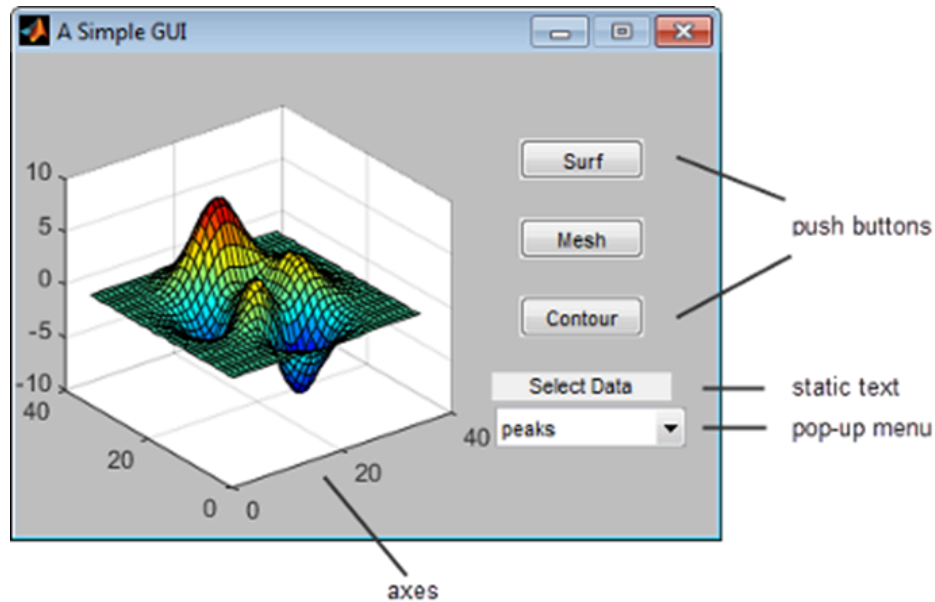
Use GUIDE

GUIDE is a drag-and-drop environment for laying out user interfaces (UIs). You code the interactive behavior of your app separately, in the MATLAB editor. Apps created with GUIDE are compatible with almost all other releases, and they support all the graphics functionality in MATLAB.

Use MATLAB Functions to Create Apps Programmatically

One can also code the layout and behavior of your app entirely using MATLAB functions. In this approach, you create a traditional figure and place interactive components in that figure programmatically. These apps support the same types of graphics and interactive components that GUIDE supports, as well as tabbed panels.

Create A Simple App Using Guide



This example shows how to use GUIDE to create an app that has a simple user interface (UI).

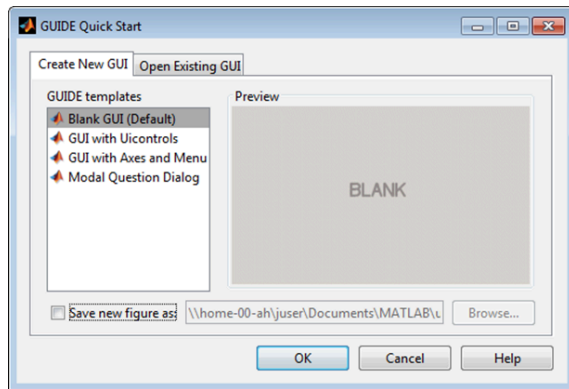
If you only want to view and run the code that created this app, set your current folder to one to which you have write access. Copy the example code and open it in the Editor by issuing the following MATLAB commands:

```
copyfile(fullfile(docroot, 'techdoc', 'creating_guis', ...
'examples', 'simple_gui*. *')), fileattrib('simple_gui*. *', '+w');
guide simple_gui.fig;
edit simple_gui.m
```

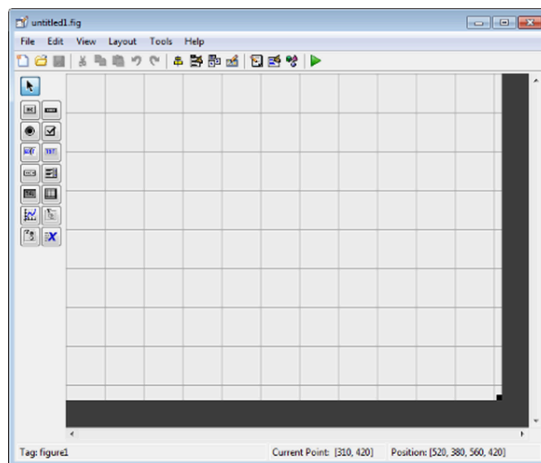
Click the **Run** ► button to run the app.

Open A New UI In The Guide Layout Editor

Start GUIDE by typing `guide` at the MATLAB prompt.



In the GUIDE Quick Start dialog box, select the **Blank GUI (Default)** template, and then click **OK**.

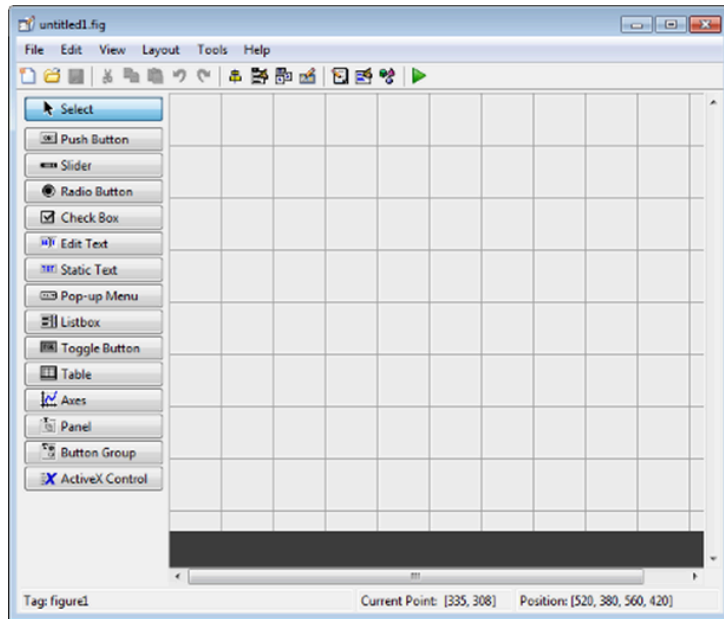


Display the names of the components in the component palette:

Select **File > Preferences > GUIDE**.

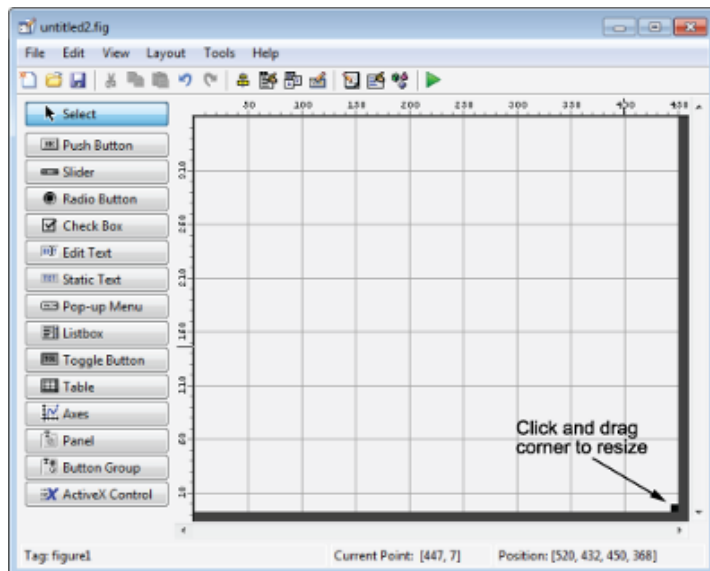
Select **Show names in component palette**.

Click **OK**.



Set the Window Size In Guide

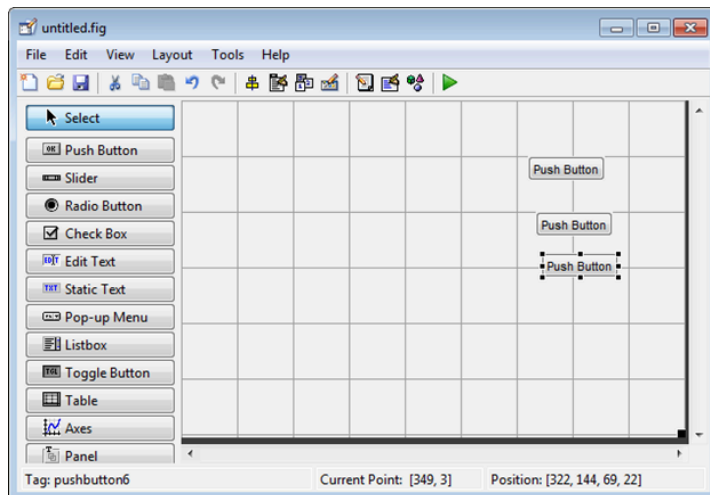
Set the size of the window by resizing the grid area in the Layout Editor. Click the lower-right corner and drag it until the canvas is approximately 3 inches high and 4 inches wide. If necessary, make the canvas larger.



Layout the UI

Add, align, and label the components in the UI.

Add the three push buttons to the UI. Select the push button tool from the component palette at the left side of the Layout Editor and drag it into the layout area. Create three buttons, positioning them approximately as shown in the following figure.



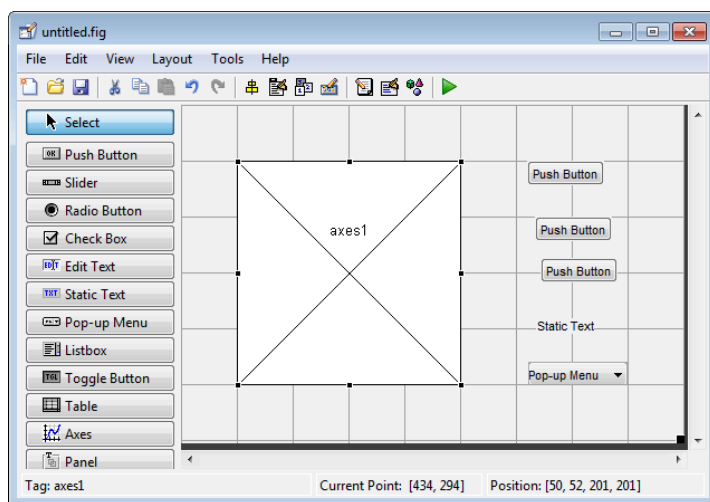
Add the remaining components to the UI.

A static text area

A pop-up menu

An axes

Arrange the components as shown in the following figure. Resize the axes component to approximately 2-by-2 inches.



Align the Components

If several components have the same parent, you can use the Alignment Tool to align them to one another. To align the three push buttons:

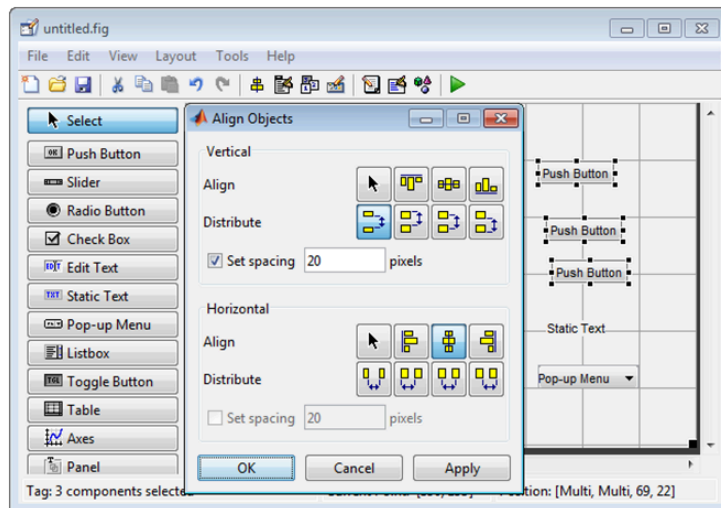
Select all three push buttons by pressing **Ctrl** and clicking them.

Select **Tools > Align Objects**.

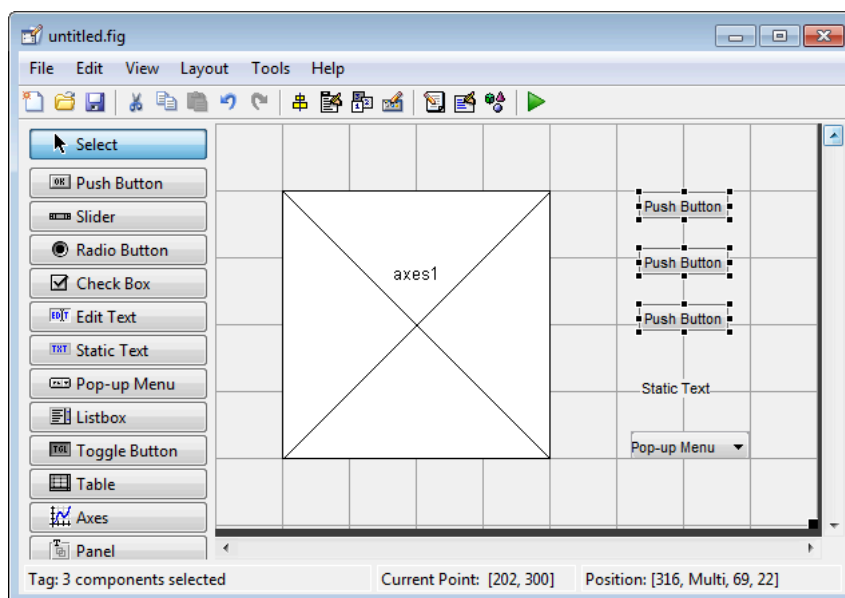
Make these settings in the Alignment Tool:

Left-aligned in the horizontal direction.

20 pixels spacing between push buttons in the vertical direction.



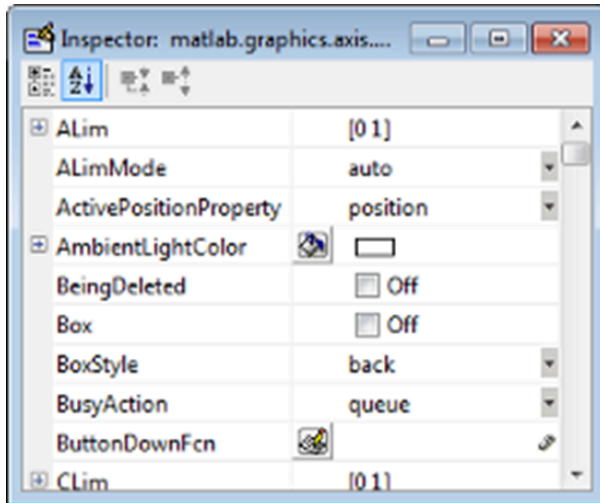
Click **OK**.



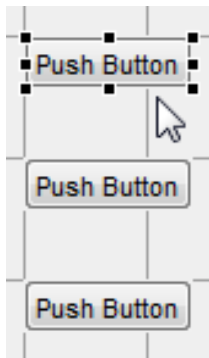
Label the Push Buttons

Each of the three push buttons specifies a plot type: surf, mesh, and contour. This section shows you how to label the buttons with those options.

Select **View > Property Inspector**.



In the layout area, click the top push button.



In the Property Inspector, select the String property, and then replace the existing value with the word Surf.



Press the **Enter** key. The push button label changes to **Surf**.



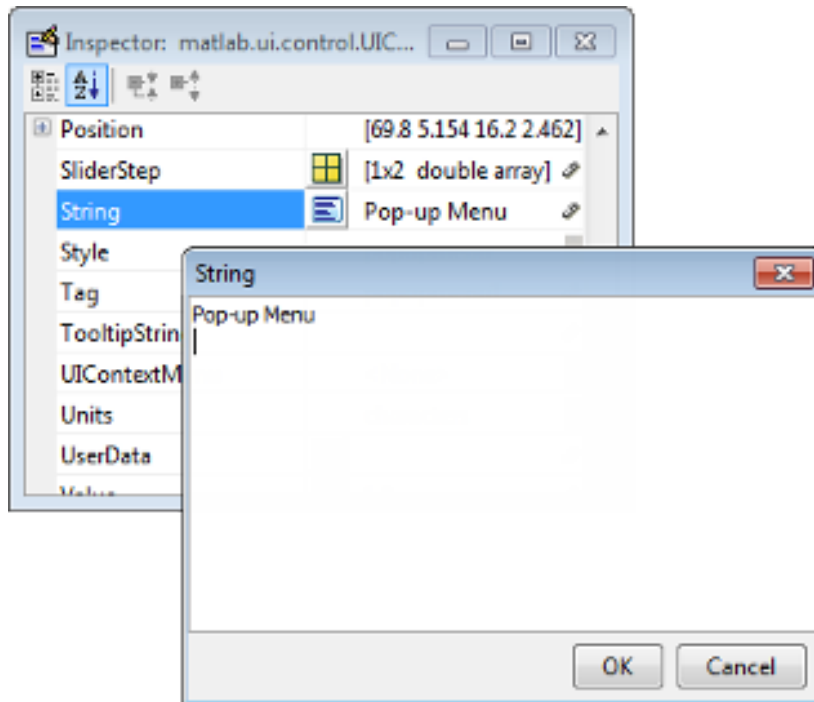
Click each of the remaining push buttons in turn and repeat steps 3 and 4. Label the middle push button **Mesh**, and the bottom button **Contour**.

List Pop-Up Menu Items

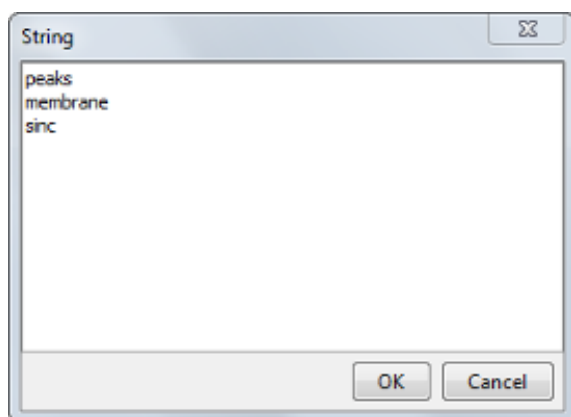
The pop-up menu provides a choice of three data sets: peaks, membrane, and sinc. These data sets correspond to MATLAB functions of the same name. This section shows you how to list those data sets as choices in the pop-menu.

In the layout area, click the pop-up menu.

In the Property Inspector, click the button next to String. The String dialog box displays.

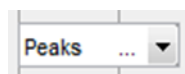


Replace the existing text with the names of the three data sets: peaks, membrane, and sinc. Press **Enter** to move to the next line.



When you finish editing the items, click **OK**.

The first item in your list, peaks, appears in the pop-up menu in the layout area.

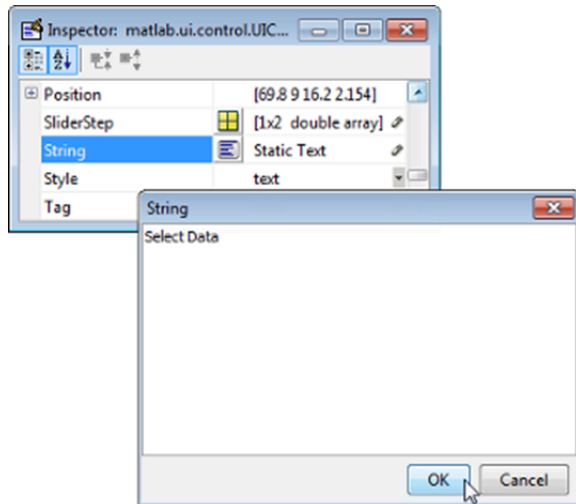


Modify the Static Text

In this UI, the static text serves as a label for the pop-up menu. This section shows you how to change the static text to read Select Data.

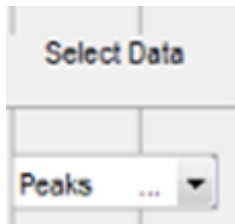
In the layout area, click the static text.

In the Property Inspector, click the button next to String. In the String dialog box that displays, replace the existing text with the phrase Select Data.



Click **OK**.

The phrase Select Data appears in the static text component above the pop-up menu.



Save the Layout

When you save a layout, GUIDE creates two files, a FIG-file and a code file. The FIG-file, with extension .fig, is a binary file that contains a description of the layout. The code file, with extension .m, contains MATLAB functions that control the app's behavior.

Save and run your program by selecting **Tools > Run**.

GUIDE displays a dialog box displaying: “Activating will save changes to your figure file and MATLAB code. Do you wish to continue?”

Click **Yes**.

GUIDE opens a **Save As** dialog box in your current folder and prompts you for a FIG-file name.

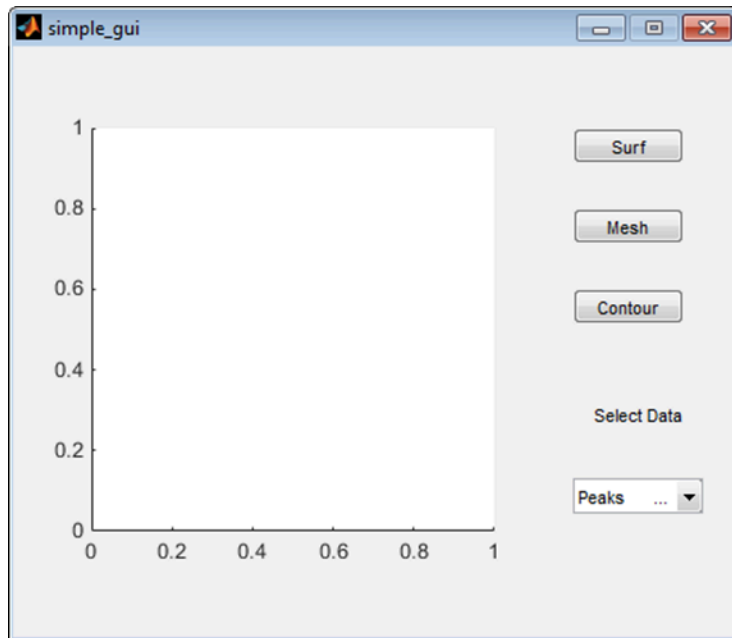
Browse to any folder for which you have write privileges, and then enter the file name `simple_gui` for the FIG-file. GUIDE saves both the FIG-file and the code file using this name.

If the folder in which you save the files is not on the MATLAB path, GUIDE opens a dialog box that allows you to change the current folder.

GUIDE saves the files `simple_gui.fig` and `simple_gui.m`, and then runs the program. It also opens the code file in your default editor.

The app opens in a new window. Notice that the window lacks the standard menu bar and toolbar that MATLAB figure windows display. You can add your own menus and toolbar buttons with GUIDE, but by default a GUIDE app includes none of these components.

When you run `simple_gui`, you can select a data set in the pop-up menu and click the push buttons, but nothing happens. This is because the code file contains no statements to service the pop-up menu and the buttons.



To run an app created with GUIDE without opening GUIDE, execute its code file by typing its name.

`simple_gui.`

You can also use the run command with the code file, for example, `run simple_gui.`