Accessing Workspace Variables from a List Box

On this page…

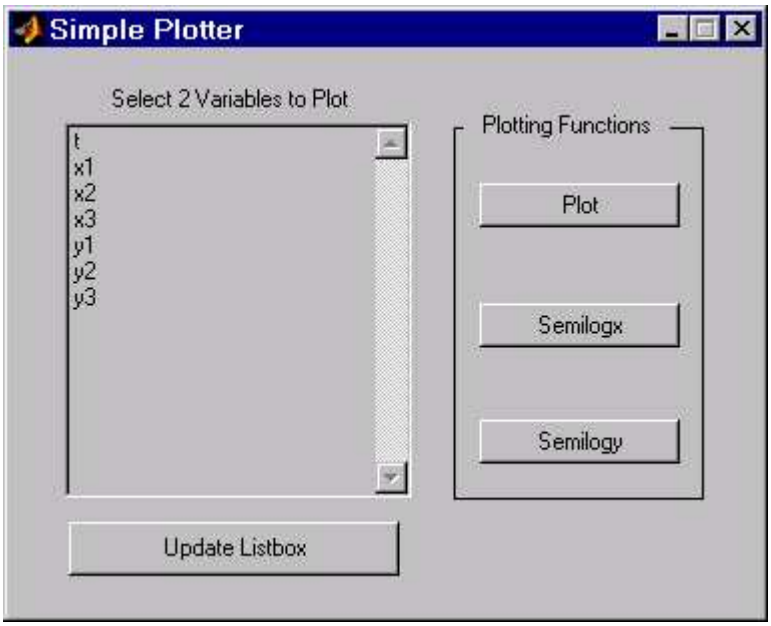
About the Workspace Variable Example

This GUI uses a list box to display names of and plot variables in the base workspace. Initially, no variable names are selected in the list box. The GUI's provides controls to:

- Update the list.
- Select multiple variables in the list box. Exactly two variables must be selected.
- Create linear, `semilogx` and `semilogy` line graphs of selected variables.

The GUI evaluates the plotting commands in the base workspace. It does no validation before plotting. The user is responsible for selecting pairs of variables that can be plotted against one another. The top-most selection is used as the*x*-variable, the lower one as the *y*-variable.

The following figure illustrates the GUI layout.



Back to Top

View and Run the Workspace Variable GUI

If you are reading this document in the MATLAB Help browser, you can access the example FIG-file and code file by clicking the following links. If you are reading this on the Web or in PDF form, go to the corresponding section in the MATLAB Help Browser to use the links.

If you intend to modify the layout or code of this GUI example, first save a copy of its code file and FIG-file to your current folder (you need write access to your current folder to do this). Follow these steps to copy the example files to your current folder and then to open them:

1. Click here to copy the files to your current folder.
2. Type `guide lb` or click here to open the FIG-file in GUIDE.

3. Type `edit lb` or click here to open the code file in the Editor.

You can view the properties of any component by double-clicking it in the Layout Editor to open the Property Inspector for it. You can modify either the figure, the code, or both, and then save the GUI in your current folder using **File** > **Save as** from GUIDE. This saves both files, allowing you to rename them, if you choose.

To just inspect the GUI in GUIDE and run it, follow these steps instead:

1. Click here to add the example files to the MATLAB path (only for the current session).
2. Click here to run the `lb` GUI.
3. Click here to display the GUI in the GUIDE Layout Editor (read only).
4. Click here to display the GUI code file in the MATLAB Editor (read only).

> **Note** Do not save GUI files to the `examples` folder where you found them or you will overwrite the original files. If you want to save GUI files, use **File** > **Save as** from GUIDE, which saves both the GUI FIG-file and the GUI code file.

Back to Top

Reading Workspace Variables

When the GUI initializes, it needs to query the workspace variables and set the list box `String` property to display these variable names. Adding the following subfunction to the GUI code accomplishes this using `evalin` to execute the `who` command in the base workspace. The `who` command returns a cell array of strings, which are used to populate the list box.

```
function update_listbox(handles)
vars = evalin('base','who');
set(handles.listbox1,'String',vars)
```

The function's input argument is the handles structure set up by the GUIDE. This structure contains the handle of the list box, as well as the handles of all other components in the GUI.

The callback for the **Update Listbox** push button also calls `update_listbox`.

Back to Top

Reading the Selections from the List Box

The GUI requires the user to select two variables from the workspace and then choose one of three plot commands to create the graph: `plot`, `semilogx`, or `semilogy`.

No callback for the list box exists in the GUI code file. One is not needed because the plotting actions are initiated by push buttons.

Enabling Multiple Selection

To enable multiple selection in a list box, you must set the `Min` and `Max` properties so that `Max - Min > 1`. You must change the default `Min` and `Max` values of `0` and `1` to meet these conditions. Use the Property Inspector to set these properties on the list box.

How Users Select Multiple Items

List box multiple selection follows the standard for most systems:

- **Ctrl**+click left mouse button — noncontiguous multi-item selection
- **Shift**+click left mouse button — contiguous multi-item selection

Users must use one of these techniques to select the two variables required to create the plot.

Returning Variable Names for the Plotting Functions

The `get_var_names` subfunction returns the two variable names that are selected when the user clicks one of the three plotting buttons. The function:

- Gets the list of all items in the list box from the `String` property.
- Gets the indices of the selected items from the `Value` property.
- Returns two string variables, if there are two items selected. Otherwise `get_var_names` displays an error dialog box stating that the user must select two variables.

Here is the code for `get_var_names`:

```
function [var1,var2] = get_var_names(handles)
list_entries = get(handles.listbox1,'String');
index_selected = get(handles.listbox1,'Value');
if length(index_selected) ~= 2
        errordlg('You must select two variables',...
                         'Incorrect Selection','modal')
else
        var1 = list_entries{index_selected(1)};
        var2 = list_entries{index_selected(2)};
end
```

Callbacks for the Plotting Buttons

The callbacks for the plotting buttons call `get_var_names` to get the names of the variables to plot and then call `evalin` to execute the plot commands in the base workspace.

For example, here is the callback for the `plot` function:

```
function plot_button_Callback(hObject, eventdata, handles)
[x,y] = get_var_names(handles);
evalin('base',['plot(' x ',' y ')'])
```

The command to evaluate is created by concatenating the strings and variables, and looks like this:

```
try
    evalin('base',['semilogx(',x,',',y,')'])
catch ex
    errordlg(...
      ex.getReport('basic'),'Error generating semilogx plot','modal')
end
```

The `try/catch` block handles errors resulting from attempting to graph inappropriate data. When evaluated, the result of the command is:

```
plot(x,y)
```

The other two plotting buttons work in the same way, resulting in `semilogx(x,y)` and `semilogy(x,y)`.

Back to Top

List Box Directory Reader

A GUI to Set Simulink Model Parameters