
Chapter 5 — Graphing Application

In this chapter, we will see how you could build the graphing application that is described in Chapter 11 of *Smalltalk Programming for Windows*. As you will see, with what you have already learned in this manual, constructing this application as a subclass of the `PlotWindow` class in the original book is simple and straight-forward. In fact, the brevity of this chapter is a testament to the transparent power in WindowBuilder Pro.

Getting Started

You should file in the code from the CH11 folder of the disk accompanying WindowBuilder Pro (or the CHAP5 subdirectory in the WindowBuilder Pro installation directory) before beginning your work on this chapter. It might also be a good idea to re-read Chapter 11 of *Smalltalk Programming for Windows* if it isn't fresh in your mind.

In the book, we build this application in three stages. But since the user interface doesn't really change throughout those three steps, we will construct the application in one phase with WindowBuilder Pro.

Constructing the User Interface

There is nothing new or surprising in the user interface for the `PlotWindowWB` class. It is a simple two-pane window with a two-menu menubar. To construct this window manually, follow these steps.

1. Open WindowBuilder Pro to edit a new window.
2. Resize the window you start with so that it is approximately 480 x 360. The window itself need not recognize any events, so there's no need to connect it to any.
3. Select the `TextPane` tool and drag a tall, narrow text pane in the left portion of the window. We found that making it 50 pixels wide and as tall as the window worked well.
4. As you can read in Chapter 11 of *Smalltalk Programming for Windows*, this pane (which we'll call `dataPane`) should respond to several events: `getContents` (by executing the `dataPane` method), `getMenu` (with `dataMenu`, to generate a pane popup menu equivalent to the Data menu on the window menubar), and `save` (with the `redraw` method). Make those connections as you have previously learned to do.

5. Select the GraphPane tool and fill the rest of the window with a GraphPane component.
6. The GraphPane must respond to the `display` event with the `redraw` method, and to the `getMenu` event with the `plotMenu` method. Make those connections.
7. Select the window and create a menubar for it. The menubar contains two menus: Data and Plot. The Data menu has seven active items and two separators. Table 5-1 presents its details.

Table 5-1. Data Menu Items and Associated Methods

Menu Item	Associated Method
&Accept	accept
&Restore	restoreData
----- (Separator)	(none)
File &In	fileIn
File &Out	fileOut
----- (Separator)	(none)
&Copy	(none - supplied by superclass)
Cu&t	(none - supplied by superclass)
&Paste	(none - supplied by superclass)

NOTE

In the First and Second printings of *Smalltalk Programming for Windows*, Chapter 11 omitted the code necessary to implement the editing methods correctly from the application's menus. This problem was corrected in the Third Printing. Check the copy of *Smalltalk Programming for Windows* to see which printing you have. (You can find out on the bottom of the page on the reverse side of the title page.) If you have the Third Printing, just continue. If not, then you'll need to install (file in) the code from the subdirectory called CHAP5 in the WindowBuilder Pro installation directory).

8. Create the plot menu, which is shown in Table 5-2. Notice that this is a three-level hierarchical menu; we use the greater-than sign (“>”) to indicate levels of hierarchy.

NOTE

Remember that the menubar editor treats menu items that are indented under one another as a group. When you create complex menus, this can get a little confusing if you enter all the items and then go back and attempt to demote everything to the right level.

Because of this, we recommend that you click on the “Demote Item” button before you enter an item that is to appear at the next lower level of the hierarchy. Similarly, press **Promote Item** before entering an item that needs to appear higher in the menu hierarchy than the item you just finished entering. This will greatly simplify the process of entering menus.

Table 5-2. Plot Menu Items and Associated Methods

Menu Item	Associated Method
&Erase plot	clearPlot
&Redo plot	newPlot
---- (Separator)	(none)
&Horizontal bar	setHorizontal
&Vertical bar	setVertical
----- (Separator)	(none)
&Options	(none)
> Stretch &Factor	factor
> Bar &Width	barWidth
> &Bar Color	(none)
>> &Black	black
>> &Dark gray	darkGray
>> &Light gray	paleGray
Restore &Defaults	restoreDefaults

9. Save your work, with PlotWindow as the ViewManager subclass to use as the parent of this class.

Reviewing the Code

Apart from the slightly more complex menu entries, there is nothing particularly interesting about the code for PlotWindowWB compared with the code in *Smalltalk Programming for Windows* for PlotWindow or with other `createViews` or `open` methods we’ve seen constructed with WindowBuilder Pro.

Finishing the Application

As is often the case with Smalltalk/V applications of any complexity, we will create an `initWindow` method for PlotWindowWB. Recall from Chapter 2 that we do this to undertake processing that should occur at the time an application starts up and which we might put in the `open` method itself if we weren’t working with WindowBuilder Pro. Since we may use WindowBuilder Pro later to modify the user interface and re-generate the `open` method, we want to avoid losing work. So we create a method

called `initWindow` which will be automatically invoked when the window is opened.

Here is the code for the `initWindow` method in `PlotWindowWB`:

```
initWindow

    dataPane := self paneNamed: 'dataPane'.
    plotPane := self paneNamed: 'plotPane'.

    barPen := plotPane pen.
    barPen setLineWidth: 1.
    self black.
```

Aside from the by-now-familiar process of providing pane names for later ease of use, we initialize the pen for the `GraphPane` in the `initWindow` method.