

## Graphs in Smalltalk – Karthik Narayan

Graphs in Smalltalk can often be frustrating at points; certain items just won't seem to come up, and (more than half the time), you think that it's Smalltalk's fault. If the above describes you, then read on! This guide will help you get set up with graphs in absolutely no time at all.

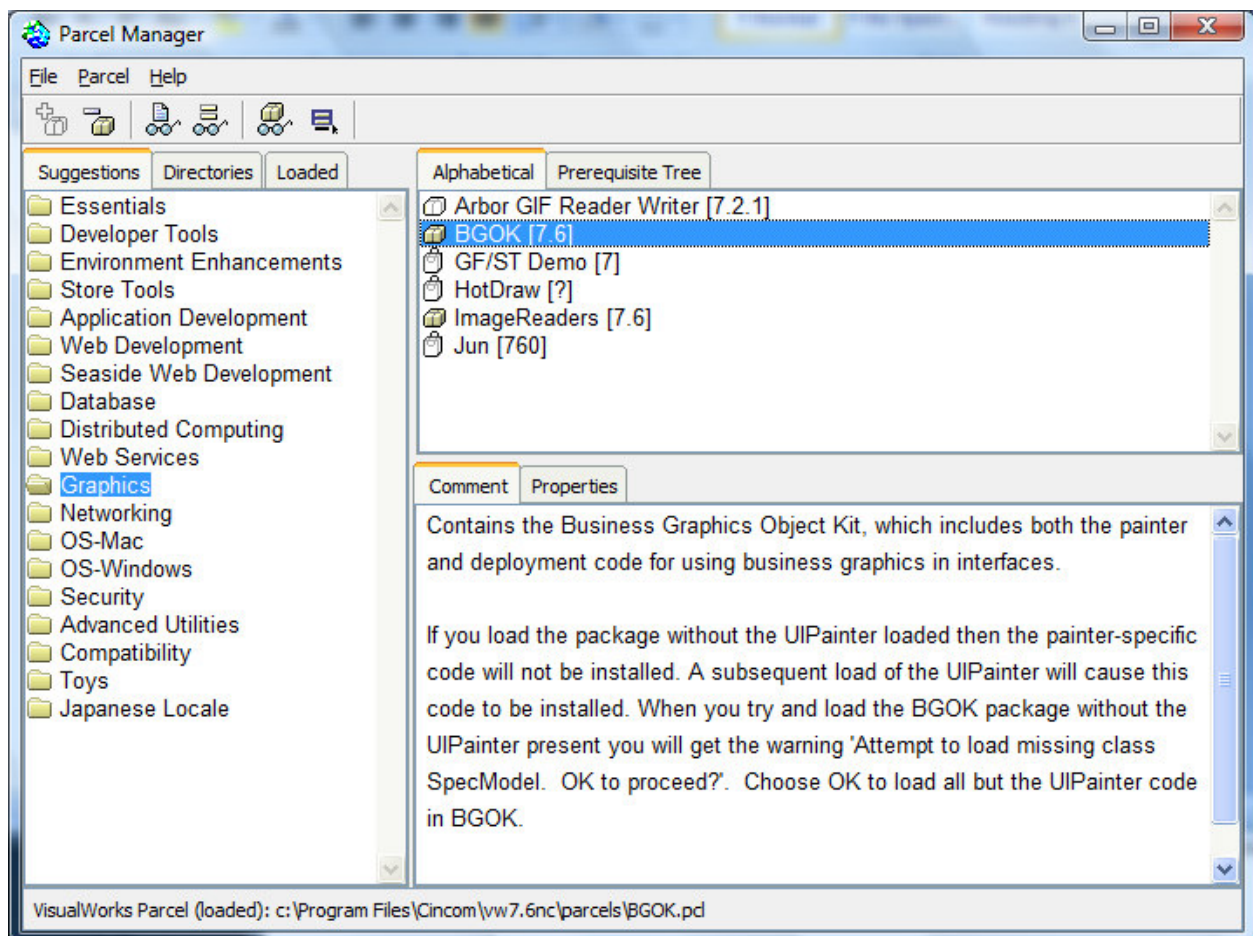
In this case, we will see how to set up a bar graph.

### Step 1: Figure out what type of graph you want to put in.

This is clearly the most important step. If you're trying to give someone percentages, use a pie chart (and not an xy plot).

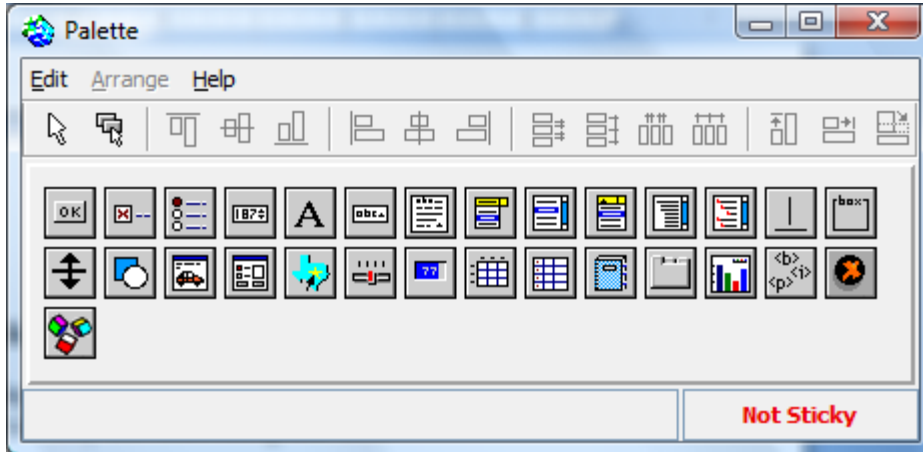
### Step 2: Import the BGOK parcel.

Navigate to the Smalltalk main page and open the Parcel Manager (F3). Click on Graphics, right click on BGOK, and hit Load. You now have all the tools you need to start creating graphs in Smalltalk.

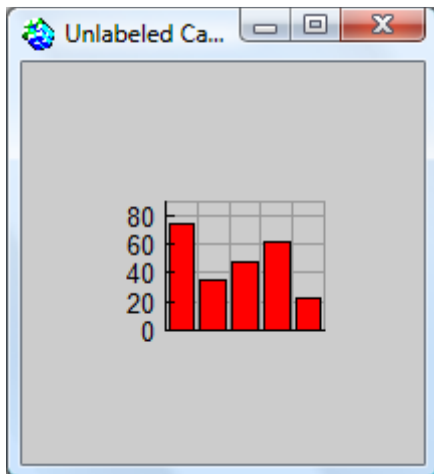


### Step 3: Create a new Application Dialog with a Graph object.

To display a graph, we first need a window to display it in. Create your new canvas and attach it to a class, say GraphAppModel. Ensure that GraphAppModel is in a namespace, GraphData. Now, look at the palette. You may see something like this.



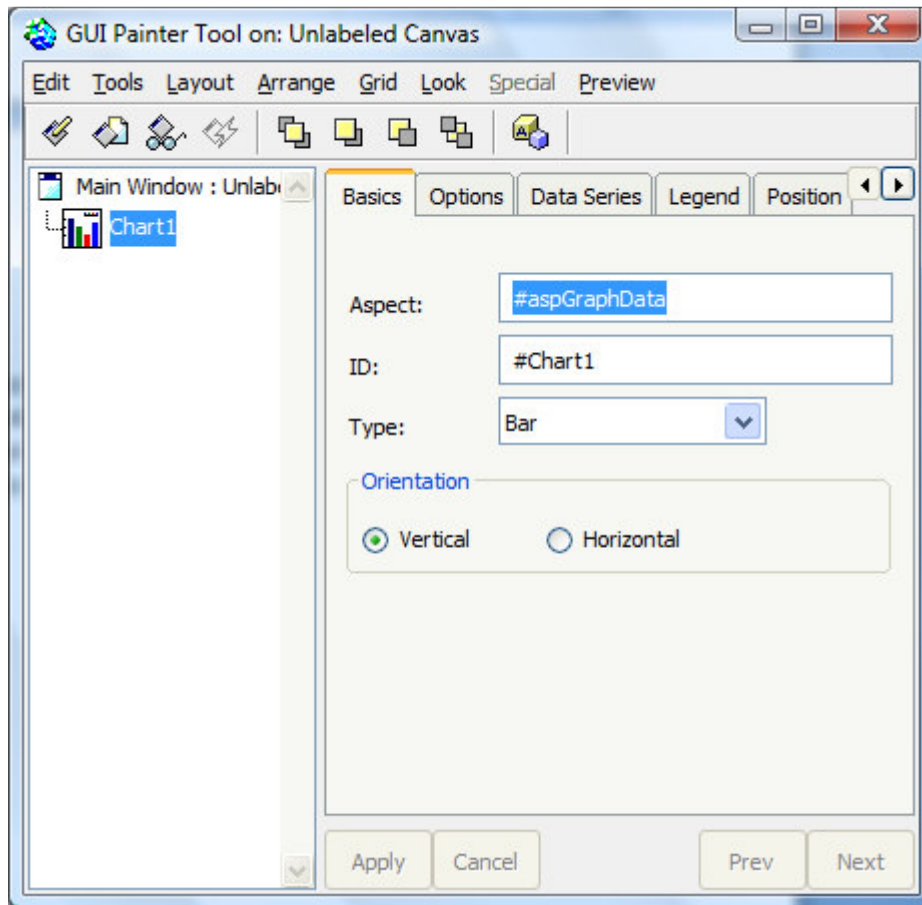
If you don't see all of the above buttons, don't worry. The most important one you're looking for is the one with little bars forming a bar chart (second row, third from the right). Drag and drop and instance of this onto your canvas. You should now have something that looks like this



Great! We now have a graph object on our canvas.

#### Step 4: Modify the graph parameters.

We will first look at how to create a bar graph. In the GUI painter tool, the Type should initially be set to Bar. If not, go ahead and set it to Bar for this exercise. As with any other widget which holds data, graphs also have aspects. For this exercise, we will call our aspect `aspGraphData`. Apply the changes. Your window should now resemble something like this:

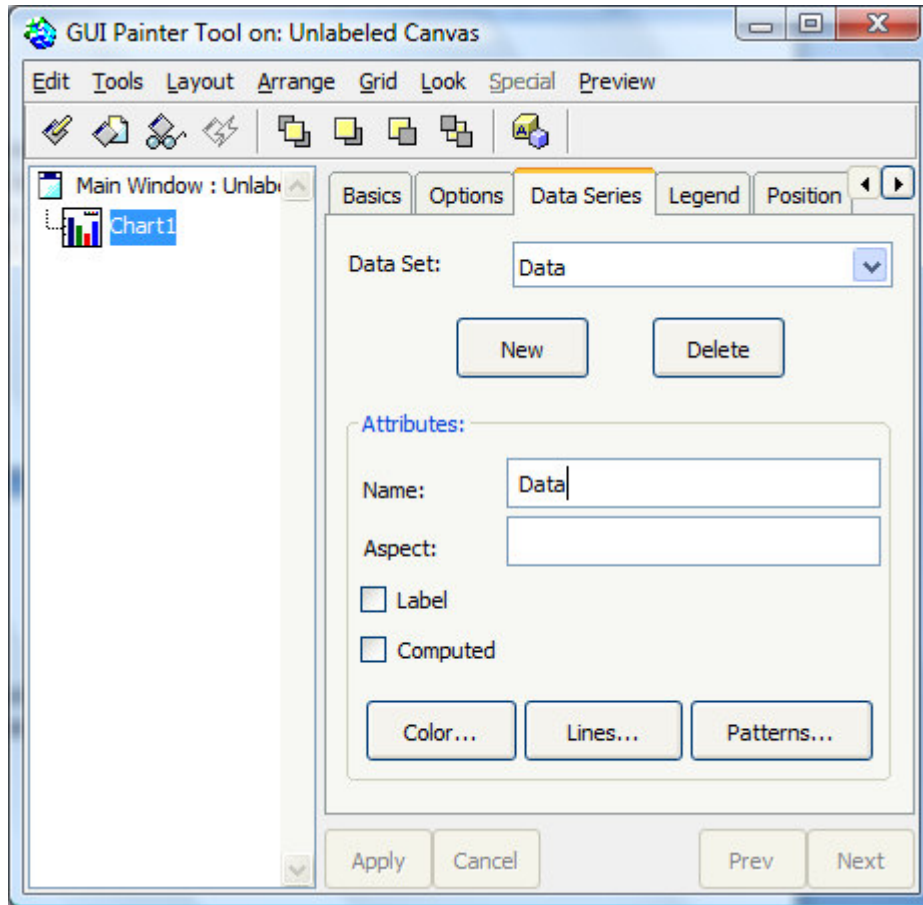


Make sure you define the Chart and install it. Let's say that we want to graph the following data on our Bar chart:

- A: 16
- B: 32
- C: 29
- D: 11
- F: 9

We need two components: labels, and data. Here's how to do this. Click on the Data Series tab and add a new Data Set. Our first data set will be the data itself. Call the Data Set 'Data', delete the aspect, check the label box, and apply the changes. Our second data set will be the labelling system. Call the Data Set 'Labels', delete the aspect, check the label box, and apply the changes. *If you're confused as to why to*

delete the aspects, don't worry. All of the information in the graph will be stored collectively in the `aspGraphData` aspect. Define and install these changes. You should now have something resembling the following:



Perfect! We now move on to the coding.

### Step 5: Format the data appropriately in code.

Now, we have to simply format the graph data appropriately. Smalltalk uses Arrays in order to read the information for graphs. So, this means that `aspGraphData` will contain all the information required, in an Array form. Particularly, it will be an Array of Arrays in this case. Recall that the data we wanted to plot was

A: 16

B: 32

C: 29

D: 11

F: 9

In order to do this, we want to represent the data in an appropriate form so Smalltalk can recognize what has been input. The data representation for a bar graph of the above data is as follows:

```
##(16 'A') #(32 'B') #(29 'C') #(11 'D') #(9 'F'))
```

In other words, `aspGraphData` will store this array. To see this, go into the `initialize` method of `GraphAppModel`, and type in the following code:

```
aspGraphData := ##(16 'A') #(32 'B') #(29 'C') #(11 'D') #(9 'F'))
```

```
^self
```

Now, start up your program, and you will see a nice graph of the above information.

### Extensions

All graphs are generally the same as the above. The only difference is that the `aspect`, `aspGraphData`, changes with each type of graph. The different representations are all given in the GUI Developer Guide for Smalltalk. If you have trouble adding graphs, simply follow the above process and change around the `aspGraphData` value.

### One Final Trick

The above only works for static data. The following code wouldn't work, however:

```
|x|
```

```
x := 3.
```

```
y := 4.
```

```
aspGraphData := ##(x 'A') #(y 'B'))
```

Instead, you can use something similar to this:

```
|entryOne entryTwo|
```

```
aspGraphData := Array new: 2.
```

```
entryOne := Array new: 2.
```

```
    entryOne at: 1 put: x.
```

```
    entryOne at: 2 put: 'A'.
```

```
entryTwo := Array new: 2.
```

```
    entryTwo at: 1 put: y.
```

```
    entryTwo at: 2 put: 'B'.
```

```
aspGraphData at: 1 put: entryOne.
```

```
aspGraphData at: 2 put: entryTwo.
```

Remember that in Smalltalk, array indexes start at 1, and not 0! By using the a combination of the above, you should be able to create virtually any graph you want to, and ace the graph portion of the CS 2340 course!