

- **GraphNormal** puts the user into normal operation mode, and is the equivalent of clicking the top icon in the tool palette.

These operations are provided so a program can allow the user to sketch a region in a graph. The program can then read back what the user did. Unlike the other drawing modes, these wave drawing and edit modes allow user-defined buttons to be active. This is so you can provide a “done” button for the user. The button procedure should call **GraphNormal** to exit the drawing or edit mode.

The **GraphWaveEdit** command operates a little differently depending on whether or not you specify a wave with the command. If you do specify a wave then only that wave can be edited by the user. If you let the user choose a wave then he or she can switch to a new trace by clicking it.

Drawing Programming Strategies

There are three distinct ways you can structure your drawing program:

- **Append:** You can append the contents of one or more layers.
- **Replace Layer:** You can replace the contents of the layers.
- **Replace Group:** You can replace the contents of a named group.

The Replace Layer Method

Use this method when you want to maintain a fairly complex drawing completely under program control. For example you may want to extend Igor by adding a new axis type or a new display method or you may want to create a completely new kind of graph. The Polar Graphs package mentioned above utilizes the replace method.

The key to the replace method is the use of the /K flag with the **SetDrawLayer** command. This “kills” (deletes) the entire contents of the specified layer. After clearing out the layer you must then redraw the entire contents. To do this you will usually have to maintain some sort of data structure to hold all the information required to maintain the drawing.

For example if you are creating an artificial axis package, you will need to maintain user settings similar to those you see in Igor’s modify axis dialog. In many cases setting up a few global variables or waves in a data folder will be sufficient. As an example, see the Drawing Axes procedure file in the WaveMetrics Procedures folder.

The Replace Group Method

With named groups created with the **SetDrawEnv** **gname** keyword, you can use **DrawAction** to delete the group or to set the insertion point for new drawing commands. See the **DrawAction** operation on page V-172 for an example.

The Append Method

Use this method to provide convenience features that automate creation of simple drawings. You add a small drawing, such as a drop line, calibration bar, or shading rectangle, to any existing drawing objects in a given layer when the user runs a procedure or clicks a button. Such drawings are often small and modular — .

Generally, the drawing will be something the user could have done manually and may want to modify. If you need to specify a layer at all it should be a User layer. Often there will be no need to set the drawing layer at all — just use the current layer.

You may, however, need to set the layer for specific circumstances. A shading rectangle is an example of an object that should go in a specific layer, since it must be below the traces of a graph. In this case, if you use the **SetDrawLayer** operation, then you should set the current layer back with “**SetDrawLayer UserTop**”.

If you are using the append method, you should avoid using the Prog layers. This is because they are intended for use where the entire layer is to be replaced under program control.