

### Overview

The following sections describe the various programming techniques available for getting input from and for interacting with a user during the execution of your procedures. These techniques include:

- The simple input dialog
- Control panels
- Cursors
- Marquee menus

The simple input dialog provides a bare bones but functional user interfaces with just a little programming. In situations where more elegance is required, control panels provide a better solution.

### Modal and Modeless User Interface Techniques

Before the rise of the graphical user interface, computer programs worked something like this:

1. The program prompts the user for input.
2. The user enters the input.
3. The program does some processing.
4. Return to step 1.

In this model, the program is in charge and the user must respond with specific input at specific points of program execution. This is called a “modal” user interface because the program has one mode in which it will only accept specific input and another mode in which it will only do processing.

The Macintosh changed all this with the idea of event-driven programming. In this model, the computer waits for an event such as a mouse click or a key press and then acts on that event. The user is in charge and the program responds. This is called a “modeless” user interface because the program will accept any user action at any time.

You can use both techniques in Igor. Your program can put up a modal dialog asking for input and then do its processing or you can use control panels to build a sophisticated modeless event-driven system.

Event-driven programming is quite a bit more work than dialog-driven programming. You have to be able to handle user actions in any order rather than progressing through a predefined sequence of steps. In real life, a combination of these two methods is often used.

### The Simple Input Dialog

The simple input dialog is a way by which a function can get input from the user in a modal fashion. It is very simple to program and is also simple in appearance.

A simple input dialog is presented to the user when a DoPrompt statement is executed in a function. Parameters to DoPrompt specify the title for the dialog and a list of local variables. For each variable, you must include a Prompt statement that provides the text label for the variable.

Generally, the simple input dialog is used in conjunction with routines that run when the user chooses an item from a menu. This is illustrated in the following example which you can type into the procedure window of a new experiment:

```
Menu "Macros"  
    "Calculate Diagonal...", CalcDiagDialog()  
End  
  
Function CalcDiagDialog()  
    Variable x=10,y=20  
    Prompt x, "Enter X component: " // Set prompt for x param  
    Prompt y, "Enter Y component: " // Set prompt for y param  
    DoPrompt "Enter X and Y", x, y
```