

## Chapter I-1 — Introduction to Igor Pro

Originally XOPs were intended to allow adding operations only to Igor. Now XOPs can add much more, including functions, menus, dialogs, and windows, so “XOP” has the meaning “external module that extends Igor”.

To create an XOP, you must be a C or C++ programmer and you need the optional **Igor External Operations Toolkit**. See **Creating Igor Extensions** on page IV-208.

Although *creating* an extension is a job for a programmer, anyone can *use* an extension. The Igor installer automatically installs commonly used extensions in "Igor Pro Folder/Igor Extensions (64-bit)". These extensions are available for immediate use.

Less commonly used extensions are installed in "Igor Pro Folder/More Extensions (64-bit)". Available extensions are described in the “XOP Index” help file (choose Help→Help Windows→XOP Index.ihf). To activate an extension, see **Activating 64-bit Extensions** on page III-512.

## Igor’s User Interface

Igor uses a combination of the familiar graphical user interface and a command-line interface. This approach gives Igor both ease-of-use and programmability.

The job of the user interface is to allow you to apply Igor’s operations and functions to objects that you create. You can do this in several ways:

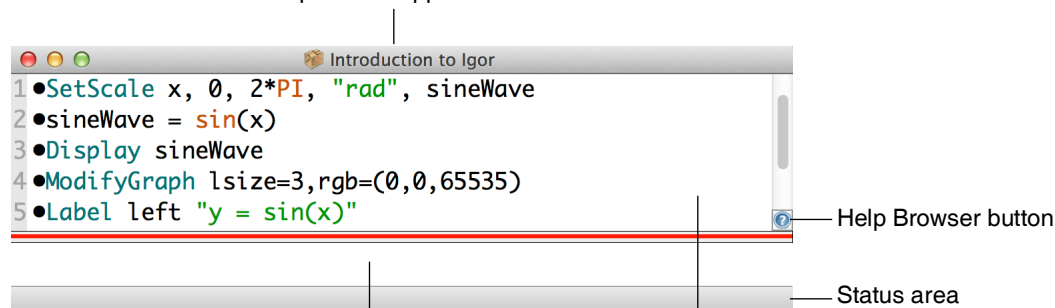
- Via menus and dialogs
- Using the main toolbar
- By typing Igor commands directly into the command line
- By writing Igor procedures

### The Command Window

The command window is Igor’s control center. It appears at the bottom of the screen.

At the bottom of the command window is the command line. Above the red divider is the history area where executed commands are stored for review. Igor also uses the history area to report results of analyses like curve-fitting or waveform statistics.

The name of the current experiment appears as the title of the command window.



You enter commands in the **command line**.

When Igor executes a command it transfers it to the **history area**.