Overview

You can add your own menu items to many Igor menus by writing a menu definition in a procedure window. A simple menu definition looks like this:

```
Menu "Macros"
   "Load Data File/1"
   "Do Analysis/2"
   "Print Report"
End
```

This adds three items to the Macros menu. If you choose Load Data File or press Command-1 (*Macintosh*) or Ctrl+1 (*Windows*), Igor executes the procedure LoadDataFile which, presumably, you have written in a procedure window. The command executed when you select a particular item is derived from the text of the item. This is an *implicit* specification of the item's execution text.

You can also *explicitly* specify the execution text:

```
Menu "Macros"
   "Load Data File/1", Beep; LoadWave/G
   "Do Analysis/2"
   "Print Report"
End
```

Now if you choose Load Data File, Igor will execute "`Beep; LoadWave/G`".

When you choose a user menu item, Igor checks to see if there is execution text for that item. If there is, Igor executes it. If not, Igor makes a procedure name from the menu item string. It does this by removing any characters that are not legal characters in a procedure name. Then it executes the procedure. For example, choosing an item that says

```
"Set Sampling Rate..."
```

executes the SetSamplingRate procedure.

If a procedure window is the top window and if Option (*Macintosh*) or Alt (*Windows*) is pressed when you choose a menu item, Igor tries to find the procedure in the window, rather than executing it.

A menu definition can add submenus as well as regular menu items.

```
Menu "Macros"
   Submenu "Load Data File"
      "Text File"
      "Binary File"
   End

   Submenu "Do Analysis"
      "Method A"
      "Method B"
   End

   "Print Report"
End
```

This adds three items to the Macros menu, two submenus and one regular item. You can nest submenus to any depth.

Menu Definition Syntax

The syntax for a menu definition is:

```
Menu <Menu title string> [,<menu options>]
   [<Menu help strings>]
   <Menu item string> [,<menu item flags>] [,<execution text>]
   [<Item help strings>]
   …
   Submenu <Submenu title string>
      [<Submenu help strings>]
      <Submenu item string> [,<execution text>]
      [<Item help strings>]
      …
   End
End
```

`<Menu title string>` is the title of the menu to which you want to add items. Often this will be Macros but you can also add items to Analysis, Misc and many other built-in Igor menus, including some submenus and the graph marquee and layout marquee menus. If `<Menu title string>` is not the title of a built-in menu then Igor creates a new main menu on the menu bar.

<Menu options> are optional comma-separated keywords that change the behavior of the menu. The allowed keywords are `dynamic`, `hideable`, and `contextualmenu`. For usage, see **Dynamic Menu Items** (see page IV-129), **HideIgorMenus** (see page V-346), and **PopupContextualMenu** (see page V-756) respectively.

`<Menu help strings>` specifies the help for the menu title. This is optional. As of Igor7, menu help is not supported and the menu help specification, if present, is ignored.

`<Menu item string>` is the text to appear for a single menu item, a semicolon-separated string list to define **Multiple Menu Items** (see page IV-131), or **Specialized Menu Item Definitions** (see page IV-132) such as a color, line style, or font menu.

<Menu item flags> are optional flags that modify the behavior of the menu item. The only flag currently supported is /Q, which prevents Igor from storing the executed command in the history area. This is useful for menu commands that are executed over and over through a keyboard shortcut. This feature was introduced in Igor Pro 5. Using it will cause errors in earlier versions of Igor. Menus defined with the `contextualmenu` keyword implicitly set all the menu item flags in the entire menu to /Q; it doesn't matter whether /Q is explicitly set or not, the executed command is not stored in the history area.

`<Execution text>` is an Igor command to execute for the menu item. If omitted, Igor makes a procedure name from the menu item string and executes that procedure. Use "" to prevent command execution (useful only with PopupContextualMenu/N).

`<Item help strings>` specifies the help for the menu item. This is optional. As of Igor7, menu help is not supported and the menu item help specification, if present, is ignored.

The Submenu keyword introduces a submenu with `<Submenu title string>` as its title. The submenu continues until the next End keyword.

`<Submenu item string>` acts just like `<Menu item string>`.