When the a menu event is reported then the following key:value pairs will also be present in `infoStr`:

Key	Value
`MENUNAME`	Name of menu (in English) as used by **SetIgorMenuMode**.
`MENUITEM`	Text of menu item as used by **SetIgorMenuMode**.

The `enablemenu` event does not pass `MENUNAME` or `MENUITEM`.

The `menu` and `enablemenu` messages are not sent when drawing tools are in use in a graph or layout or when waves are being edited in a graph.

Returning a value of 0 for the `enablemenu` message is recommended, though the return value is (currently) ignored.

You can use the **SetIgorMenuMode** operation to alter the enable state of Igor's built-in menus in a way you find appropriate for the window. If you do this, usually you will also handle the menu message and perform your idea of an appropriate action.

Note: Dynamic user-defined menus (see **Dynamic Menu Items** on page IV-129) are built and enabled by using string functions in the menu definitions.

Returning a value of 0 for any menu message allows Igor to perform the normal action. Returning any other value (1 is commonly used) tells Igor to skip performing the normal action.

See the user function description with **IgorMenuHook** on page IV-291 for details on the sequence of menu building, enabling, and handling.

Custom Marker Hook Functions

You can define custom marker shapes for use with graph traces. To do this, you must define a custom marker hook function, activate it by calling SetWindow with the markerHook keyword, and set a trace to use it via the ModifyGraph operation marker keyword.

A custom marker hook function takes one parameter - a WMMarkerHookStruct structure. This structure provides your function with information you need to draw a marker.

The function prototype used with a custom marker hook has the format:

```
Function MyMarkerHook(s)
   STRUCT WMMarkerHookStruct &s
   <code to draw marker>
   ...
   return statusCode       // 0 if nothing done, else 1
End
```