The DoIgorMenu operation will not attempt to select a menu during curve fitting or while a dynamic menu item's function is running. Doubtless there are other times during which using DoIgorMenu would be unwise.

The text of some items in the File menu changes depending on the type of the active window. In these cases you must pass generic text as the *MenuItemStr* parameter. Use “Save Window”, “Save Window As”, “Save Window Copy”, “Adopt Window”, and “Revert Window” instead of “Save Notebook” or “Save Procedure”, etc. Use “Page Setup” instead of “Page Setup For All Graphs”, etc. Use “Print” instead of “Print Graph”, etc.

The Edit→Insert File menu item was previously named Insert Text. For compatibility, you can specify either “Insert File” or “Insert Text” as *MenuItemStr* to invoke this item.

### See Also

**SetIgorMenuMode**, **ShowIgorMenus**, **HideIgorMenus**, **Execute**

The SetIgorMenuModeProc.ipf WaveMetrics procedure file contains SetIgorMenuMode commands for every menu and menu item. You can load it using

```
#include <SetIgorMenuModeProc>
```

# DoPrompt

**DoPrompt** [**/HELP=*helpStr***] ***dialogTitleStr***, ***variable*** [, ***variable***]…

The DoPrompt statement in a function invokes the simple input dialog. A DoPrompt specifies the title for the simple input dialog and which input variables are to be included in the dialog.

### Flags

| | |
|---|---|
| /HELP=*helpStr* | Sets the help topic or help text that appears when the dialog’s Help button is pressed.<br>*helpStr* can be a help topic and subtopic such as is used by `DisplayHelpTopic/K=1 helpStr`, or it can be text (255 characters max) that is displayed in a subdialog just as if `DoAlert 0, helpStr` had been called, or *helpStr* can be `""` to remove the Help button. |

### Parameters

*variable* is the name of a dialog input variable, which can be real or complex numeric local variable or local string variable, defined by a Prompt statement. You can specify as many as 10 variables.

*dialogTitleStr* is a string or string expression containing the text for the title of the simple input dialog.

### Details

Prompt statements are required to define what variables are to be used and the text for any string expression to accompany or describe the input variable in the dialog. When a DoPrompt variable is missing a Prompt statement, you will get a compilation error. Pop-up string data can not be continued across multiple lines as can be done using Prompt in macros. See **Prompt** for further usage details.

Prompt statements for the input variables used by DoPrompt must come before the DoPrompt statement itself, otherwise, they may be used anywhere within the body of a function. The variables are not required to be input parameters for the function (as is the case for Prompt in macros) and they may be declared within the function body. DoPrompt can accept as many as 10 variables.

Functions can use multiple DoPrompt statements, and Prompt statements can be reused or redefined.

When the user clicks the Cancel button, any new input parameter values are not stored in the variables.

DoPrompt sets the variable V_flag as follows:

| | |
|---|---|
| 0: | Continue button clicked. |
| 1: | Cancel button clicked. |

### See Also

**The Simple Input Dialog** on page IV-144, the **Prompt** keyword, and **DisplayHelpTopic**.