

The first and simplest method is to use the **CreateBrowser** operation with the prompt keyword. You can optionally use any of the other keywords supported by CreateBrowser.

The second method is to use the CreateBrowser operation with the /M flag. This creates the modal Data Browser but does not display it. You then call **ModifyBrowser** with the /M flag to configure the modal browser as you like. When you are ready to display the browser, call ModifyBrowser with the /M flag and the showModalBrowser keyword.

Using either method, when the user clicks OK or Cancel or closes the modal browser using the close box, the browser sets the V_Flag and S_BrowserList variables. If the user clicks OK, V_Flag is set to 1 and S_BrowserList is set to contain a semicolon-separated list of the full paths of the selected items. If the user clicks Cancel or the close box, V_Flag is set to 0 and S_BrowserList is set to "".

The Data Browser stores full paths, quoted if necessary, in S_BrowserList. Each full path is followed by a semicolon. You can extract the full paths one-by-one using the **StringFromList** function.

The output variables are local when CreateBrowser is executed from a procedure. There is no reason for you to create the modal Data Browser from the command line, but if you do, the output variables are global and are created in the current data folder at the time the CreateBrowser operation was invoked.

The user can change the current data folder in the modal browser. In most cases this is not desirable. The examples for the **CreateBrowser** operation show how to save and restore the current data folder.

When the user clicks the OK button, the Data Browser executes the commands that you specify using the command1 and command2 keywords to **CreateBrowser** or **ModifyBrowser**. If you omit these keywords, it does not execute any commands.

Managing Data Browser User-Defined Buttons

User-defined buttons allow you to customize your work environment and provide quick access to frequently-used operations. You add a button using ModifyBrowser appendUserButton and delete a button using ModifyBrowser deleteUserButton. The appendUserButton keyword allows you to specify the button name as well as the command to be executed when the button is clicked.

The user button command allows you to invoke operations or functions on the objects that are currently selected in the Data Browser or to invoke some action that is completely unrelated to the selection. For example, the command string "Display %s" will display each of the currently selected waves while the command string "Print IgorInfo(0)" prints some general information in the history window.

When you click a user button, the command is executed once for each selected item if the command string contains a %s. Otherwise the command is executed once regardless of the current selection. If you want to operate once on the whole selection you must not use %s but instead call GetBrowserSelection from your function.

User buttons are drawn in the order that they are appended to the window. If you want to change their position you must delete them and then append them in the desired order.

Buttons are not saved with the experiment or in preferences so they must be added to the Data Browser when Igor starts. To add a set of buttons so that they are available in any experiment, you must write an **IgorStartOrNewHook** hook function. See **User-Defined Hook Functions** on page IV-280 for more information.

Data Folder Notes

You can store notes describing a particular data folder in a string variable in that data folder and view those notes in the Data Browser.

When a single data folder is selected in the Data Browser and the info pane is visible, if the data folder contains a string whose name matches a preset list, the contents of the string are displayed in the info pane. You can edit the list of string names in the Info Pane tab of the Data Browser category of the Miscellaneous Settings dialog. The default list of names is "readme;notes;".

For example, execute the following commands on the command line:

```
NewDataFolder/O TestDF  
String root:TestDF:readme = "This is a data folder note"
```

Now open the Data Browser, make sure the Info Pane checkbox is checked, and select TestDF in the main list. The contents of root:TestDF:readme are displayed in the info pane.

If a data folder contains more than one string matching the list of string names to use as data folder notes, only the first matching string in the list is used.

This feature was added in Igor Pro 9.00. To disable it, set the list of names in the Miscellaneous Settings dialog to an empty string.