Note: Providing for liberal names requires extra effort and testing by Igor programmers (see **Programming with Liberal Names** on page IV-168) so you may occasionally experience problems using liberal names with user-defined procedures.

Commands and Data Folders

Data folders provide a way to keep separate sets of data from interfering with each other. You can examine and create data folders using the Data Browser (Data menu). There is always a root data folder and this is the only data folder that many users will ever need. Advanced users may want to create additional data folders to organize their data.

You can refer to waves and variables either in the current data folder, in a specific data folder or in a data folder whose location is relative to the current data folder:

```
// wave1 is in the current data folder
wave1 = <expression>

// wave1 is in a specific data folder
root:'Background Curves':wave1 = <expression>

// wave1 is in a data folder inside the current data folder
:'Background Curves':wave1 = <expression>
```

(This syntax applies to the command line and macros only, not to user-defined functions in which you must use **Wave References** to read and write waves.)

In the first example, we use an object name by itself (wave1) and Igor looks for the object in the current data folder.

In the second example, we use a full data folder path (root:'Background Curves':) plus an object name. Igor looks for the object in the specified data folder.

In the third example, we use a relative data folder path (:'Background Curves':) plus an object name. Igor looks in the current data folder for a subdata folder named Background Curves and looks for the object within that data folder.

Important: The right-hand side of an assignment statement (described under **Assignment Statements** on page IV-4) is evaluated in the context of the data folder containing the destination object. For example:

```
root:'Background Curves':wave1 = wave2 + var1
```

For this to work, wave2 and var1 must be in the Background Curves data folder.

Examples in the rest of this chapter use object names alone and thus reference data in the current data folder. For more on data folders, see Chapter II-8, **Data Folders**.

Command Tooltips

In procedure windows and the command window, if you hover the mouse cursor over a command, such as an operation or function, Igor displays a tooltip that contains information about the command. If you hover over the name of a built-in structure, such as WMButtonAction, Igor displays the structure definition. You can turn off this feature by choosing Misc→Miscellaneous Settings, selecting the Text Editing section, selecting the Editor Behavior tab, and unchecking the Show Context-sensitive Tooltips checkbox.

Command Completion

Procedure windows and the command line support automatic command completion.

When you type the first few letters of a command, a popup appears and allows you to quickly select a command. Use the arrow keys to move up and down the list of completion options. Press the Enter key or Tab key to insert the highlighted text in the document.