An exception to this is that Igor limits names of XOP files to 31 characters, plus the ".xop" extension. Igor will not recognize an XOP file with a longer name.

Paths in Windows are limited to 259 characters in length. Neither Windows nor Igor can deal with a path that exceeds this limit. For example, if you create a directory with a 250 character name and try to create a file with a 15 character name, neither Windows nor Igor will permit this.

This boils down to the following: Feel free to use long file and directory names, but expect to see errors if you use outrageously long names or if you have directories so deeply nested that paths approach the theoretical limit.

Path Separators

The Macintosh HFS file system uses a colon to separate elements in a file path. For example:

```
hd:Igor Pro Folder:Examples:Sample Graphs:Contour Demo.pxp
```

The Windows file system uses a backslash to separate elements in a file path. For example:

```
C:\Igor Pro Folder\Examples\Sample Graphs:Contour Demo.pxp
```

Some Igor operations (e.g., LoadWave) allow you to enter file paths. Igor accepts Macintosh-style or Windows-style paths regardless of the platform on which you are running.

Note: Igor uses the backslash character as an escape character in literal strings. This can cause problems when using Windows-style paths.

For example, the following command creates a textbox with two lines of text. "\r" is an escape code inserts a carriage return character:

```
Textbox "This is line 1.\rThis is line 2."
```

Because Igor interprets a backslash as an escape character, the following command will not execute properly:

```
LoadWave "C:\Data Files\really good data.ibw"
```

Instead of loading a file named "really good data.ibw", Igor would try to load a file named "Data Files<CR>eally good data.ibw", where <CR> represents the carriage return character. This happens because Igor interprets "\r" in literal strings to mean carriage return.

To solve this problem, you must use "\\" instead of "\" in a file path. Igor will correctly execute the following:

```
LoadWave "C:\\Data Files\\really good data.ibw"
```

This works because Igor interprets "\\" as an escape sequence that means "insert a backslash character here".

Another solution to this problem is to use a Macintosh HFS-style path, even on Windows:

```
LoadWave "C:Data Files:really good data.ibw"
```

Igor converts the Macintosh HFS-style path to a Windows-style path before using it. This avoids the backslash issue.

For a complete list of escape sequences, see **Escape Sequences in Strings** on page IV-14.

If you are writing procedures that need to extract sections of file paths or otherwise manipulate file paths, the **ParseFilePath** function on page V-733 may come in handy.

UNC Paths

"UNC" stands for "Universal Naming Convention". This is a Windows convention for identifying resources on a network. One type of network resource is a shared directory. Consequently, when running under Windows, in order to reference a network directory from an Igor command, you need to use a UNC path.

The format of a UNC path that points to a file in a folder on a shared server volume or directory is:

```
   "\\server\share\directory\filename"
```