Examples

```
String pathIn, pathOut

// Full path
pathIn= "hd:Igor Pro Folder:WaveMetrics Procedures:Waves:Wave Lists.ipf"

// Extract first element.
Print ParseFilePath(0, pathIn, ":", 0, 0)      // Prints "hd"

// Extract second element.
Print ParseFilePath(0, pathIn, ":", 0, 1)      // Prints "Igor Pro Folder"

// Extract last element.
Print ParseFilePath(0, pathIn, ":", 1, 0)      // Prints "Wave Lists.ipf"

// Extract next to last element.
Print ParseFilePath(0, pathIn, ":", 1, 1)      // Prints "Waves"

// Get path to folder containing the file.
// Prints "hd:Igor Pro Folder:WaveMetrics Procedures:Waves:"
Print ParseFilePath(1, pathIn, ":", 1, 0)

// Extract the file name without extension.
Print ParseFilePath(3, pathIn, ":", 0, 0)      // Prints "Wave Lists"

// Extract the extension.
Print ParseFilePath(4, pathIn, ":", 0, 0)      // Prints "ipf"

// Make sure the given path ends with a colon and concatenate file name.
String path = <routine that returns a Macintosh-style path to a folder>
path = ParseFilePath(2, path, ":", 0, 0)
path += "AFile.txt"
```

See Also

Escape Sequences in Strings on page IV-14, **UNC Paths** on page III-451, and **Path Separators** on page III-451 for details. The **RemoveEnding** function.

ParseOperationTemplate

ParseOperationTemplate [***flags***] ***cmdTemplate***

The ParseOperationTemplate operation helps XOP programmers and WaveMetrics programmers write code to implement Igor operations. If you are not an XOP programmer nor a WaveMetrics programmer, it will be of no interest.

ParseOperationTemplate generates starter code for programmers who are creating Igor operations. The starter code is copied to the clipboard, overwriting any previous clipboard contents.

Flags

/C=*c* If *c* is nonzero, ParseOperationTemplate stores code for your ExecuteOperation and RegisterOperation functions in the clipboard.

- *c*=0: Do not generate code
- *c*=1: Generate simplified C code - not recommended
- *c*=2: Generate C code
- *c*=6: Generate C++ code

The only difference between /C=6 and /C=2 is that the ExecuteOperation function is declared as extern "C" instead of static. C++ files that use static work fine although extern "C" is correct.