

SpecialCharacterList

SpecialCharacterList(*notebookNameStr*, *separatorStr*, *mask*, *flags*)

The SpecialCharacterList function returns a string containing a list of names of special characters in a formatted text notebook.

Parameters

If *notebookNameStr* is "", the top visible notebook is used. Otherwise *notebookNameStr* contains either *kwTopWin* for the top notebook window, the name of a notebook window or a host-child specification (an *hcSpec*) such as *Panel0#nb0*. See **Subwindow Syntax** on page III-92 for details on host-child specifications.

separatorStr should contain a single ASCII character, usually semicolon, to separate the names.

mask determines which types of special characters are included. *mask* is a bitwise parameter with values:

- 1: Pictures including graphs, tables and layouts.
- 2: Notebook actions.
- 4: All other special characters such as dates and times.

or a bitwise combination of the above for more than one type. See **Setting Bit Parameters** on page IV-12 for details about bit settings.

flags is a bitwise parameter. Pass 0 to include all special characters or 1 to include only selected special characters. All other bits are reserved and should be passed as zero.

Details

Only formatted text notebooks have special characters. When called for a plain text notebook, SpecialCharacterList always returns "".

Examples

Print a list of all special characters in the top notebook:

```
Print SpecialCharacterList("", ";", -1, 0)
```

Prints a list of notebook action characters in Notebook0:

```
Print SpecialCharacterList("Notebook0", ";", 2, 0)
```

Print a list of selected notebook action characters in Notebook0:

```
Print SpecialCharacterList("Notebook0", ";", 2, 1)
```

See Also

The **Notebook** and **NotebookAction** operations; the **SpecialCharacterInfo** function.

SpecialDirPath

SpecialDirPath(*dirIDStr*, *domain*, *flags*, *createDir*)

The SpecialDirPath function returns a full path to a file system directory specified by *dirIDStr* and *domain*. It provides a programmer with a way to access directories of special interest, such as the preferences directory and the desktop directory.

The path returned always ends with a separator character which may be a colon, backslash, or forward slash depending on the operating system and the *flags* parameter.

SpecialDirPath depends on operating system behavior. The exact path returned depends on the locale, the operating system, the specific installation, the current user, and possibly other factors.

Parameters

dirIDStr is one of the following strings:

- | | |
|---------------|---|
| "Packages" | Place for advanced programmers to put preferences for their procedure packages. |
| "Documents" | The OS-defined place for users to put documents. |
| "Preferences" | The OS-defined place for applications to put preferences. |
| "Desktop" | The desktop. |

SpecialDirPath

"Temporary"	The OS-defined place for applications to put temporary files.
"Igor Application"	The Igor installation folder. This is typically: /Applications/Igor Pro X Folder (Macintosh) or C:\Program Files\WaveMetrics\Igor Pro X Folder (Windows) where "X" is the major version number. Use only with domain = 0 (the current user).
"Igor Executable"	The folder containing the current Igor executable. On Macintosh, this is the path to the executable itself, not to the application bundle. Use only with domain = 0 (the current user). Requires Igor Pro 7.00 or later.
"Igor Preferences"	The folder in which Igor's own preference files are stored.
"Igor Pro User Files"	A guaranteed-writable folder for the user to store their own Igor files, and to activate extensions, help, and procedure files by creating shortcuts or aliases in the appropriate subfolders. Use only with domain = 0 (the current user). This is the folder opened using the Show Igor Pro User Files menu item in the Help menu.

domain permits discriminating between, for example, the preferences folder for all users versus the preferences folder for the current user. It is supported only for certain *dirIDStrs*. It is one of the following:

- 0: The current user (recommended value for most purposes).
- 1: All users (may generate an error or return the same path as 0).
- 2: System (may generate an error or return the same path as 1).

flags a bitwise parameter:

- Bit 0: If set, the returned path is a native path (Macintosh-style on Mac OS 9, Unix-style on Mac OS X, Windows-style on Windows). If cleared, the returned path is a Macintosh-style path regardless of the current platform. In most cases you should set this bit to zero since Igor accepts Macintosh-style paths on all operating systems. You must set this bit to one if you are going to pass the path to an external script.

All other bits are reserved and must be set to zero.

See **Setting Bit Parameters** on page IV-12 for details about bit settings.

createDir is 1 if you want the directory to be created if it does not exist or 0 if you do not want it to be created. This flag will not work if the current user does not have sufficient privileges to create the specified directory. In almost all cases it is not needed, you can't count on it, and you should pass 0.

Details

The *domain* parameter has no effect in most cases. In almost all cases you should pass 0 (current user) for this parameter. For values other than 0, SpecialDirPath might return an error which you must be prepared to handle.

In the event of an error, SpecialDirPath returns a NULL string and sets a runtime error code. You can check for an error like this:

```
String fullPath = SpecialDirPath("Packages", 0, 0, 0)
Variable len = strlen(fullPath)           // strlen(NULL) returns NaN
if (numtype(len) == 2)                   // fullPath is NULL?
    Print "SpecialDirPath returned error."
endif
```

Here is sample output from SpecialDirPath("Packages", 0, 0, 0):

Mac OS X hd:Users:<user>:Library:Preferences:WaveMetrics:Igor Pro X:Packages: