Two Liberty Basic routines to use to retrieve user-specified names for opening or saving files. These replace the built-in filedialog command, which is buggy. Both routines add a default extension if the user does not specify one. In addition, uSaveFileDialog$ checks to see if the file already exists; if it does the user is asked to confirm.

Both routines return the full name (path+file+extension) for the specified file, or in the event of error or cancellation they return blank.

```
function uSaveFileDialog$(
filter$, defaultExt$, initialDir$, initialFile$, windTitle$)

'Opens dialog to find file name for saving file. Returns full path nam
e+file name+extension
    'filter$ specifies the type of file to allow. Sample:
    'filter$ = "Text files" + chr$(0) + "*.txt" + chr$(0) + _
    '        "All files" + chr$(0) + "*.*"
    'filter$ is pairs of strings, or null for no filter

'defaultExt$ is the extension to add if the user does not specify one;
 do not include the period
    'initialDir$ is the directory in which to start
    'windTitle$ is the title of the dialog window

    struct ofn, lStructSize as ulong, hwndOwner as ulong, _
        hInstance as ulong, lpstrFilter$ as
 ptr, lpstrCustomFilter$ as ptr, _
        nMaxCustFilter as ulong, nFilterIndex as ulong
, lpstrFile$ as ptr, _
        nMaxFile as ulong, lpstrFileTitle$ as
 ptr, nMaxFileTitle as ulong, _
        lpstrInitialDir$ as ptr, lpstrTitle$ as ptr, flags as ulong, _
        nFileOffset as word, nFileExtension as
 word, lpstrDefExt$ as ptr, _
        lCustData as ulong, lpfnHook as long, lpTemplateName as long

    ofn.lStructSize.struct = len(ofn.struct)    'len of this struct
    ofn.lpstrFilter$.struct = filter$+ chr$(0) + chr$(0)
'filter for file types
    ofn.nFilterIndex.struct = 1     'Filter initially selected
    ofn.lpstrFile$.struct = initialFile$+chr$(0) + space$(360) +
chr$(0)
'File name used to initialize; receives full selected file path+name
    ofn.nMaxFile.struct = 360
'length of lpstrFile$; buffer may actually be longer
    ofn.lpstrFileTitle$.struct = space$(260) + chr$(0)
```

```
'Receives file name w/o path
    ofn.nMaxFileTitle.struct = 260  'len of lpstrFileTitle$
    ofn.lpstrInitialDir$.struct = initialDir$ + chr$(0)
'initial directory; null for default directory
    ofn.lpstrTitle$.struct = windTitle$ + chr$(0)
'titlebar string for dialog
    ofn.flags.struct =
 _OFN_OVERWRITEPROMPT     'Warn if file exists already
    ofn.lpstrDefExt$.struct = defaultExt$ + chr$(0)
'default extension to add if none specified

    CallDLL #comdlg32, "GetSaveFileNameA", ofn As
 struct, ok As boolean

    'ofn.lpstrFile.struct returns a long integer memory address.
    'Use winstring() to retrieve the string of text

'at that address, as filled by the function. If multiple files were se
lected,
    'this returns only the first
    if ok then uSaveFileDialog$=winstring(ofn.lpstrFile$.struct)
else uSaveFileDialog$=""
end function

function uOpenFileDialog$(
filter$, defaultExt$, initialDir$, initialFile$, windTitle$)

'Opens dialog to find file name for saving a single file. Return full
path name+file name+extension
    'filter$ specifies the type of file to allow. Sample:
    'filter$ = "Text files" + chr$(0) + "*.txt" + chr$(0) + _
    '       "All files" + chr$(0) + "*.*"
    'filter$ is pairs of strings, or null for no filter

'defaultExt$ is the extension to add if the user does not specify one;
 do not include the period
    'initialDir$ is the directory in which to start
    'windTitle$ is the title of the dialog window
    OFN.EXPLORER = hexdec("80000")
    struct ofn, lStructSize as ulong, hwndOwner as ulong, _
        hInstance as ulong, lpstrFilter$ as
 ptr, lpstrCustomFilter$ as ptr, _
        nMaxCustFilter as ulong, nFilterIndex as ulong
, lpstrFile$ as ptr, _
        nMaxFile as ulong, lpstrFileTitle$ as
 ptr, nMaxFileTitle as ulong, _
```

```
        lpstrInitialDir$ as ptr, lpstrTitle$ as ptr, flags as ulong, _
        nFileOffset as word, nFileExtension as
 word, lpstrDefExt$ as ptr, _
        lCustData as ulong, lpfnHook as long, lpTemplateName as long

    ofn.lStructSize.struct = len(ofn.struct)    'len of this struct
    ofn.lpstrFilter$.struct = filter$+ chr$(0) + chr$(0)
'filter for file types
    ofn.nFilterIndex.struct = 1     'Filter initially selected
    ofn.lpstrFile$.struct = initialFile$+chr$(0) + space$(1000) +
chr$(0)
'File name used to initialize; receives full selected file path+name
    ofn.nMaxFile.struct = 1000
'length of lpstrFile$; buffer may actually be longer
    ofn.lpstrFileTitle$.struct = space$(260) + chr$(0)
'Receives file name w/o path
    ofn.nMaxFileTitle.struct = 260  'len of lpstrFileTitle$
    ofn.lpstrInitialDir$.struct = initialDir$ + chr$(0)
'initial directory; null for default directory
    ofn.lpstrTitle$.struct = windTitle$ + chr$(0)
'titlebar string for dialog
    ofn.flags.struct = _OFN_PATHMUSTEXIST or
 _OFN_FILEMUSTEXIST or OFN.EXPLORER
    ofn.lpstrDefExt$.struct = defaultExt$ + chr$(0)
'default extension to add if none specified

    CallDLL #comdlg32, "GetOpenFileNameA", ofn As
 struct, ok As boolean

    'ofn.lpstrFile.struct returns a long integer memory address.
    'Use winstring() to retrieve the string of text

'at that address, as filled by the function. If multiple files were se
lected,
    'this returns only the first
    if ok then uOpenFileDialog$=winstring(ofn.lpstrFile$.struct)
else uOpenFileDialog$=""
end function
```