Safe Directories for File Writing

-
Alyce

For an eBook or printed book on using the API with Liberty BASIC, see:
APIs for Liberty BASIC

UAC and Files

User Account Control in versions of Windows starting with Windows Vista prohibits programs from writing files to many disk locations unless the program is run "as administrator". The reason for this restriction is as follows, quoted from the Wikidpedia article:

- *It aims to improve the security of Microsoft Windows by limiting application software to standard user privileges until an administrator authorizes an increase or elevation. In this way, only applications trusted by the user may receive administrative privileges, and malware should be kept from compromising the operating system.*

Safe Directories

Do not attempt to write files in the folder from which your program is running. The "Program Files" folder is not available for normal file writing operations and attempts to write files there will generate errors.

Instead, write files to the user's "My Documents" folder, temporary folder, or application data folder.

Special Variables

Applications write data to a user's data folder. Application data includes information needed by the program, such as data files and initialization files.

The following quote from the Liberty BASIC helpfile explains how to use the special variables DefaultDir$ (user's data) and StartupDir$ (installation folder)

- Because Windows Vista and Windows 7 enforces certain practices for security purposes Liberty BASIC has been reworked to fit within the expected behavior.
- In previous versions of Liberty BASIC all files would be kept in the Liberty BASIC folder. This

was a very convenient arrangement but newer versions of Windows do not allow an application that is installed in a folder under Program Files to create or write to files in its own folder or a subfolder. Instead applications are expected to keep all their information and user files in the User Data folder. This is the way that Liberty BASIC v4.04 works when installed on Vista or Windows 7. The location of the User Data folder isn't always the same, but an application asks Windows for the location of this folder.

- As a Liberty BASIC programmer you may sometimes need to know the location of the Liberty BASIC application and also the location of **User Data**. To make this easy, Liberty BASIC programs have access to two global variables.
- The variable named DefaultDir$ points to the location where your program will read and write files.
- The variable StartupDir$ points to the location where Liberty BASIC is running from. Unless your copy of Liberty BASIC is not installed in C:\Program Files you will not be able to write to files in that location.
- If you are running on Windows XP or another earlier version of Windows the DefaultDir$ and StartupDir$ variables will point to the same place.

Trailing Backslash in DefaultDir$

The special variable StartupDir$ points to the folder from which a Liberty BASIC program is running. Do not use it to write files. StartupDir$ includes a trailing backslash.

To access the user's data folder, write files to DefaultDir$. The special variable DefaultDir$ does not include a trailing backslash. In your code, check for a trailing backslash and add it if it doesn't exist, as in the following example.

```
print DefaultDir$   'no trailing backslash
print StartupDir$   'has trailing backslash

if right$(DefaultDir$,1)<>"\" then
    DefaultDir$=DefaultDir$+"\"
end if

print DefaultDir$

'write a file:
open DefaultDir$+"mytestfile.txt" for output as #f
print #f, "Test"
close #f
```

My Documents

Windows has long included a folder called "My Documents" which is meant to be the location where users create and save files. You can retrieve this location with the API from shell32.dll called **SHGetSpecialFolderLocation**, using a value of 5 for the desired folder. This is the value assigned to the "My Documents" folder. An additional API is required to retrieve the folder name. It is **SHGetPathFromIDListA**. Both API calls are wrapped in a function:

```
Function GetSpecialfolder$(CSIDL)
'CSIDL.PERSONAL = 5 : My Documents = GetSpecialfolder$(CSIDL.PERSONAL)
    struct IDL,cb As Long, abID As short
    calldll #shell32, "SHGetSpecialFolderLocation",_
        0 as long, CSIDL as long, IDL as struct, ret as long
    if ret=0 then
        Path$ = Space$(512)
        id=IDL.cb.struct
        calldll #shell32, "SHGetPathFromIDListA",id as long,
 Path$ as ptr, ret as long
        GetSpecialfolder$ = Left$(Path$, InStr(Path$, Chr$(0)) - 1)
    else
        GetSpecialfolder$ = "Error"
    end if
    End Function
```

Trailing Backslash

The folder name for "My Documents" does not include a trailing backslash. To assure that the backslash is included in the folder name, check for it and append it if it is not there. Here is a demo that retrieves the location of "My Documents", adds a trailing backslash, and writes a test file there.

```
CSIDL.PERSONAL = 5    'My Documents Folder

myDocuments$ = GetSpecialfolder$(CSIDL.PERSONAL)

print "My Documents folder location is: "
print myDocuments$

if right$(myDocuments$,1)<>"\" then
    myDocuments$=myDocuments$+"\"
end if
```

```
open myDocuments$ + "testxxxx.txt" for output as #f
print #f, "Test"
close #f

end

Function GetSpecialfolder$(CSIDL)
    struct IDL,cb As Long, abID As short
    calldll #shell32, "SHGetSpecialFolderLocation",_
        0 as long, CSIDL as long, IDL as struct, ret as long
    if ret=0 then
        Path$ = Space$(512)
        id=IDL.cb.struct
        calldll #shell32, "SHGetPathFromIDListA",id as long,
 Path$ as ptr, ret as long
        GetSpecialfolder$ = Left$(Path$, InStr(Path$, Chr$(0)) - 1)
    else
        GetSpecialfolder$ = "Error"
    end if
    End Function
```

Temporary Files and Folders

To assure that your program writes temporary files to the Windows folder reserved for temporary files, use GetTempPathA and GetTempFileNameA. These functions are explained here:

GetTempPathA
GetTempFileNameA

```
yourPrefix$ = "xxx"   'prefix you choose
TempFileName$ = GetTempFileName$(yourPrefix$)
print "Temporary File Name is ";TempFileName$
end

function GetTempFileName$(prefix$)
    TempPath$=GetTempPath$()
    TempFile$ = space$(256)+chr$(0)

    calldll #kernel32, "GetTempFileNameA",_
    TempPath$ as ptr,_  'directory for temp file
    prefix$ as ptr,_    'desired prefix for temp filename
    0 as ulong,_        '0=file created,nonzero=you must create file
    TempFile$ as ptr,_
```

```
'string buffer to hold qualified path and filename
    result as ulong     'nonzero=success
    end function

Function GetTempPath$()
    CallDLL #kernel32, "GetTempPathA",_
    0 as long,_
    _NULL as long,_
    length as long

    buf$ = space$(length)

    CallDLL #kernel32, "GetTempPathA",_
    length as long,_
    buf$ as ptr,_
    ret as long

    GetTempPath$ = buf$
End Function
```

Safe Directories for File Writing | UAC and Files | Safe Directories | Special Variables | Trailing Backslash in DefaultDir$ | My Documents | Trailing Backslash | Temporary Files and Folders