

Forum:[Hack Forums](#)

Thread: [Recursively Display Contents In A Directory - Microsoft C++](#)

gurlhackerclub - February 2, 2020, 7:24 am

- Microsoft C++

Should I just use Boost?

Tried using `std::experimental::filesystem`, it seems screwed.

Code using namespace `std`;

`namespace fs = std::experimental::filesystem;`

The following code works, but does not search recursively

Code `#include <windows.h>`

`#include <tchar.h>`

`#include <shlobj.h>`

`#include <string.h>`

`#include <stdio.h>`

`#include <strsafe.h>`

`#pragma comment(lib, "User32.lib")`

`#include <iostream>`

`using namespace std;`

`void DisplayErrorBox(LPTSTR lpszFunction);`

`int _tmain(int argc, TCHAR* argv[])`

`{`

`WIN32_FIND_DATA ffd;`

`LARGE_INTEGER filesize;`

`TCHAR szDir[MAX_PATH];`

`HANDLE hFind = INVALID_HANDLE_VALUE;`

`DWORD dwError = 0;`

`TCHAR path[MAX_PATH];`

`TCHAR subpath[MAX_PATH];`

`// If the directory is not specified as a command-line argument,`

`// print usage.`

`if (SHGetFolderPath(NULL, CSIDL_PROFILE, NULL, 0, path) != S_OK)`

`{`

`cout << "I could not retrieve the user's home directory!\n";`

`}`

`else`

`{`

`}`

`// Prepare string for use with FindFile functions. First, copy the`

`// string to a buffer, then append '*' to the directory name.`

`StringCchCopy(szDir, MAX_PATH, path);`

`StringCchCat(szDir, MAX_PATH, TEXT("\\*"));`

```

// Find the first file in the directory.

hFind = FindFirstFile(szDir, &ffd);

if (INVALID_HANDLE_VALUE == hFind)
{
    _tprintf(TEXT("FindFirstFile"));
    return dwError;
}

// List all the files in the directory with some info about them.

do
{
    if (ffd.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY)
    {

        filesize.LowPart = ffd.nFileSizeLow;
        filesize.HighPart = ffd.nFileSizeHigh;
        _tprintf(TEXT(" %s\\%s  +DIR \n"), path, ffd.cFileName);
    }
    else
    {
        filesize.LowPart = ffd.nFileSizeLow;
        filesize.HighPart = ffd.nFileSizeHigh;
        _tprintf(TEXT(" %s\\%s  \n"), path, ffd.cFileName);
    }
} while (FindNextFile(hFind, &ffd) != 0);

dwError = GetLastError();
if (dwError != ERROR_NO_MORE_FILES)
{
    _tprintf(TEXT("FindFirstFile"));
}

FindClose(hFind);
return dwError;
}

```

m1n0r1tyrep0rt - February 2, 2020, 7:45 am

Yeah that is not going to search recursively, because for recursion "you need to call yourself".

```

#include <string>
#include <functional>
#include <dirent.h>

void listFiles(const std::string &path, std::function<void(const std::string &)> cb) {
    if (auto dir = opendir(path.c_str())) {
        while (auto f = readdir(dir)) {
            if (!f->d_name || f->d_name[0] == '.') continue;
            if (f->d_type == DT_DIR)
                listFiles(path + f->d_name + "/", cb);
        }
    }
}

```

```

        if (f->d_type == DT_REG)
            cb(path + f->d_name);
    }
    closedir(dir);
}
}

```

Call the above function from a main or _tmain with the following

```

listFiles("my_directory/", [](const std::string &path) {
    std::cout << path << std::endl;
});

```

This is c++11 standard.

Hope this helps

gurlhackerclub - February 3, 2020, 8:29 pm

(Feb 2, 2020 08:45 AM)m1n0r1tyrep0rt Wrote: [\[CITE\]](#) Yeah that is not going to search recursively, because for recursion "you need to call yourself".

```

#include <string>
#include <functional>
#include <dirent.h>

```

```

void listFiles(const std::string &path, std::function<void(const std::string &)> cb) {
    if (auto dir = opendir(path.c_str())) {
        while (auto f = readdir(dir)) {
            if (!f->d_name || f->d_name[0] == '.') continue;
            if (f->d_type == DT_DIR)
                listFiles(path + f->d_name + "/", cb);

            if (f->d_type == DT_REG)
                cb(path + f->d_name);
        }
        closedir(dir);
    }
}

```

Call the above function from a main or _tmain with the following

```

listFiles("my_directory/", [](const std::string &path) {
    std::cout << path << std::endl;
});

```

This is c++11 standard.

Hope this helps

Thanks. I will look more into dirent.

AnonMouse - February 6, 2020, 6:35 am

give me a second pal and i will see if i can fix this for you :)

Pretty sure you want FindFirstFileW and FindNextFileW

code (vs2019 toolset v142)

Spoiler ([Click to View](#))

Code #include <stdio.h>

#include <wchar.h>

#include <Windows.h>

```
void recurse_files(wchar_t* directory, void (*callback)(LPWIN32_FIND_DATAW)) {

    wcscat_s(directory, MAX_PATH, L"\\*");
    WIN32_FIND_DATAW findData;
    HANDLE hFile = FindFirstFileW(directory, &findData);

    if (hFile == INVALID_HANDLE_VALUE) {
        printf("Nothing found\n");
        return;
    }

    do {
        if (findData.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY) {
            wchar_t newDir[MAX_PATH] = { 0 };
            wcscpy_s(newDir, MAX_PATH, directory);
            newDir[wcslen(newDir) - 1] = 0;
            wcscat_s(newDir, MAX_PATH, findData.cFileName);
            if (wcscmp(findData.cFileName, L"..") != 0 && wcscmp(findData.cFileName, L"..") != 0) {
                wprintf(L"DIR %s\n", newDir);
                recurse_files(newDir, callback);
            }
        }
        else {
            callback(&findData);
        }
    } while (FindNextFileW(hFile, &findData) != 0);
}

void file_callback(LPWIN32_FIND_DATAW lpFile) {
    wprintf(L"%t - %s\n", lpFile->cFileName);
}

int main()
{
    wchar_t directory[MAX_PATH] = { 0 };
    wcscpy_s(directory, MAX_PATH, L"..");
    recurse_files(directory, file_callback);
    return 0;
}
```

output

Spoiler ([Click to View](#))

Code DIR ..\vs

```

DIR ..\vs\recursiveListDir
DIR ..\vs\recursiveListDir\v16
    - .suo
    - Browse.VC.db
    - Browse.VC.opendb
DIR ..\vs\recursiveListDir\v16\ipch
DIR ..\vs\recursiveListDir\v16\ipch\AutoPCH
DIR ..\vs\recursiveListDir\v16\ipch\AutoPCH\b4127f34282d4880
    - RECURSIVELISTDIR.ipch
DIR ..\Debug
    - recursiveListDir.exe
    - recursiveListDir.ilc
    - recursiveListDir.pdb
DIR ..\recursiveListDir
DIR ..\recursiveListDir\Debug
    - recursiveListDir.log
    - recursiveListDir.obj
DIR ..\recursiveListDir\Debug\recursiveListDir.tlog
    - CL.command.1.tlog
    - CL.read.1.tlog
    - CL.write.1.tlog
    - link.command.1.tlog
    - link.read.1.tlog
    - link.write.1.tlog
    - recursiveListDir.lastbuildstate
    - vc142.idb
    - vc142.pdb
    - recursiveListDir.cpp
    - recursiveListDir.vcxproj
    - recursiveListDir.vcxproj.filters
    - recursiveListDir.vcxproj.user
    - recursiveListDir.sln

```

gurlhackerclub - February 7, 2020, 6:50 pm

(Feb 7, 2020 05:41 PM)f411en Wrote: [\[CITE\]](#) Pretty sure you want FindFirstFileW and FindNextFileW

code (vs2019 toolset v142)

Spoiler [\(Click to View\)](#)

```
Code #include <stdio.h>
```

```
#include <wchar.h>
```

```
#include <Windows.h>
```

```
void recurse_files(wchar_t* directory, void (*callback)(LPWIN32_FIND_DATAW)) {
```

```
    wcscat_s(directory, MAX_PATH, L"\\*");
```

```
    WIN32_FIND_DATAW findData;
```

```
    HANDLE hFile = FindFirstFileW(directory, &findData);
```

```
    if (hFile == INVALID_HANDLE_VALUE) {
```

```
        printf("Nothing found\n");
```

```
        return;
```

```

    }

    do {
        if (findData.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY) {
            wchar_t newDir[MAX_PATH] = { 0 };
            wcscpy_s(newDir, MAX_PATH, directory);
            newDir[wcslen(newDir) - 1] = 0;
            wscat_s(newDir, MAX_PATH, findData.cFileName);
            if (wcscmp(findData.cFileName, L"..") != 0 && wcscmp(findData.cFileName, L"..") != 0) {
                wprintf(L"DIR %s\n", newDir);
                recurse_files(newDir, callback);
            }
        }
        else {
            callback(&findData);
        }
    } while (FindNextFileW(hFile, &findData) != 0);
}

void file_callback(LPWIN32_FIND_DATAW lpFile) {
    wprintf(L"%t - %s\n", lpFile->cFileName);
}

int main()
{
    wchar_t directory[MAX_PATH] = { 0 };
    wcscpy_s(directory, MAX_PATH, L"..");
    recurse_files(directory, file_callback);
    return 0;
}

```

output

Spoiler ([Click to View](#))

Code DIR ..\vs

DIR ..\vs\recursiveListDir

DIR ..\vs\recursiveListDir\v16

- .suo
- Browse.VC.db
- Browse.VC.opendb

DIR ..\vs\recursiveListDir\v16\ipch

DIR ..\vs\recursiveListDir\v16\ipch\AutoPCH

DIR ..\vs\recursiveListDir\v16\ipch\AutoPCH\b4127f34282d4880

- RECURSIVELISTDIR.ipch

DIR ..\Debug

- recursiveListDir.exe
- recursiveListDir.ilc
- recursiveListDir.pdb

DIR ..\recursiveListDir

DIR ..\recursiveListDir\Debug

- recursiveListDir.log
- recursiveListDir.obj

DIR ..\recursiveListDir\Debug\recursiveListDir.tlog

- CL.command.1.tlog

- CL.read.1.tlog
- CL.write.1.tlog
- link.command.1.tlog
- link.read.1.tlog
- link.write.1.tlog
- recursiveListDir.lastbuildstate
- vc142.idb
- vc142.pdb
- recursiveListDir.cpp
- recursiveListDir.vcxproj
- recursiveListDir.vcxproj.filters
- recursiveListDir.vcxproj.user
- recursiveListDir.sln

This worked great, thanks!

gurlhackerclub - April 7, 2020, 5:07 am

Just for the record another way to do this

```
Code #include <boost/filesystem.hpp>
#include <iostream>

using namespace boost::filesystem;

int main()
{
    path p = "C:\\Users\\gurlh\\Desktop";
    recursive_directory_iterator it{ p };
    while (it != recursive_directory_iterator{})
        std::cout << *it++ << '\n';
}
```

gurlhackerclub - May 2, 2020, 3:24 pm

I'm an idiot.

I finally figured out how to use the filesystem library to search recursively. Switch from my default language of C++ 14 to C++17.

So without Boost, setting my Language to C++ 17. That was a headache.

```
Code #include <filesystem>
#include <iostream>
#include <string>
#include <shlobj.h>
#include <KnownFolders.h>

namespace fs = std::filesystem;

int main() {
    PWSTR the_path;
    SHGetKnownFolderPath(FOLDERID_Desktop, KF_FLAG_DEFAULT, NULL, &the_path);
    fs::path current_dir(the_path);
    // fs::path current_dir(".");
    // list all files containing an mp3 extension
    for (auto& file : fs::recursive_directory_iterator(current_dir)) {
        if (file.path().extension() == ".txt")
```

```
std::cout << file.path().string() << file.path().filename().string() << std::endl;  
    }  
}
```