

Forum:[Hack Forums](#)

Thread: [CopyFile](#)

IAmNotHere - November 16, 2014, 12:26 am

Hello,

I have a source where I need to copy a file to a folder, but the CopyFile function I insert in my project almost doubles the filesize after compiling.

The output size grows from 500 kb to 1 MB!

Are there less size consuming alternatives?

Regards,
IANH

HelloMyNameIsWalter - November 16, 2014, 12:27 am

(Nov 16, 2014 01:26 AM)IAmNotHere Wrote: [\[CITE\]](#) Hello,

I have a source where I need to copy a file to a folder, but the CopyFile function I insert in my project almost doubles the filesize after compiling.

The output size grows from 500 kb to 1 MB!

Are there less size consuming alternatives?

Regards,
IANH

Post the source? No one can help you without seeing what you are doing.

0xDEAD10CC - November 16, 2014, 1:46 pm

ever thought about WIN32_LEAN_AND_MEAN?

Masm4ever - November 17, 2014, 10:28 am

CopyFile is a Kernel32 function, i.e. it's already present without any "included" extra libraries. The call itself costs a few bytes, not kBytes. So obviously something is seriously going wrong with your project, but without seeing the source (as IAmNotHere already wrote), there is no way to understand what exactly causes that behaviour.

IAmNotHere - November 17, 2014, 11:11 pm

I replaced :

```
Code bool copyFile(const char SRC[], const char DEST[])
```

```
{
std::ifstream src;
    std::ofstream dest;
    src.open(SRC, std::ios::binary);
    dest.open(DEST, std::ios::binary);
    if (!src.is_open() || !dest.is_open())
        return false;
    dest << src.rdbuf();
    dest.close();
    src.close();
    return true;
```

```
}
```

with:

```
Code BOOL WINAPI CopyFile(  
    __in    LPCTSTR lpExistingFileName,  
    __in    LPCTSTR lpNewFileName,  
    __in    BOOL bFailIfExists  
);
```

and I have the filesize 100 KB smaller. And indeed the code doesn't run as supposed. On some machines the code runs and on some not at all.

IANH

Mouse? - November 18, 2014, 12:46 pm

If you use <<, and your file contains a \0 null byte, what do you think happens when it runs into it? You need to know the size of your file and then copy just that many bytes, so find the size of the file with your ifstream by seekg()ing to the end of the file, tellg()ing your program how big the file is, then ostream::write()ing to the new file the contents of the old file that you ifstream::read() from. If that doesn't make any sense to you, then you better stop talking, and start posting code because you're not going to get anything from us if we don't know what the real problem is, because if you don't know, why the heck would we without seeing it?

IAmNotHere - November 18, 2014, 1:20 pm

Thanks, I have to re-look the whole source first myself.
I will take your advise along coding.

IAmNotHere

_Stoner - November 19, 2014, 4:13 pm

What do your #include header files look like?

HelloMyNameIsWalter - November 19, 2014, 5:45 pm

You are obviously doing something wrong so here you go.

This was slapped together so you may want to add more error checking or fix some syntax errors (if there are any)

```
Code int copyFile(char *src, char *dst)  
{  
    int size = 0;  
    char *buff;  
    FILE *in = NULL;  
    FILE *out = NULL;  
  
    in = fopen(src, "rb");  
    out = fopen(dst, "wb+");  
  
    if(!in || !out)  
        return -1; //Failed  
  
    fseek(in, 0, SEEK_END);  
    size = ftell(in); //Get total bytes  
    fseek(in, 0, SEEK_SET); //Back to beginning
```

```

buff = (char*)malloc(sizeof(char)*size);//Make buffer large enough to hold our data
memset(buff, 0, sizeof(char)*size);//Init buffer to all 0

fread(buff, sizeof(char), size, in);//Read it all into buffer
fwrite(buff, sizeof(char), size, out);//Output buffe

fclose(in); //Close files
fclose(out);
free(buff); //Deallocate buffer

return 1; //Success

}

```

IAmNotHere - November 19, 2014, 8:47 pm

(Nov 19, 2014 05:13 PM)_Stoner Wrote: [\[CITE\]](#) What do your #include header files look like?Code

```

#include <windows.h>
#include "stdafx.h"
#include "structs.h"
#include "strings.h"
#include <fstream>
#include <iostream>
#include <string>
#include <sstream>

```

_Stoner - November 19, 2014, 8:51 pm

That would be why your file size is so huge. ?

iostream is huge, as are some of the other headers you are using. If you want a small file size, you should look into wanting to code natively instead of using standard libraries.

IAmNotHere - November 19, 2014, 9:07 pm

I noticed already. I deleted one by one of the <headers> till my vs 2013 errored to see if the filesize changed.

Regards.