Subject: [SOLVED] WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by navi on Mon, 19 Nov 2012 05:32:58 GMT

View Forum Message <> Reply to Message

Objective: Trying to batch rename files.

Problem: WIN32 API "MoveFile" fails to rename files if the file names are fetched from LineEdit Ctrl.

My guess is it has got something to do with hidden end of line char or CHARSET Unicode Ascii conversion issue. But I did not yet find a way to fix the problem. It is essential that I fetch the file name from the LineEdit Ctrl. Where the LineEdit Ctrl has multiple lines and each line is a file name.

I have coded a sample program to replicate the problem. Attached are the codes and screen-shots.

Thanks,
Navi

Reference:
MoveFile function (Windows) [MSDN]
System Error Codes (0-499) (Windows) [MSDN]

```
#include <CtrlLib/CtrlLib.h>
#include <CtrlCore/CtrlCore.h>

using namespace Upp;

GUI_APP_MAIN
{

 TopWindow w;
 LineEdit e,f;
 Vector<String> v1, v2; // Vectors for Storing File Names
 String s;      // String for temporary use
 bool r=false;  // r to keep the "MoveFile" API Return
 int err_no=0;  // err_no to keep "GetLastError" API Return

 // GUI SETUP CODE
 e.SetRect(10,10,680,100);
 f.SetRect(10,120,680,220);
 w.SetRect(0,0,700,350);

 w<<e<<f;
```

```
// ADDING SOME DUMMY BUT VALID FILE NAMES THAT DO EXIST
v1.Add("E:\\TEMP\\1.txt");
v1.Add("E:\\TEMP\\2.txt");
v1.Add("E:\\TEMP\\3.txt");

// CATENATING VECTOR 'v1' INOT TEMP STRING 's' TO LOAD ALL FILE NAMEs INTO
LINEEDIT
for(int i=0; i<v1.GetCount(); i++){
  s<<v1[i]<<"\n";
}

e.Set(s);


// FETCHING THE FILE NAME BACK FROM THE LINEEDIT AND SPLITTING THEM INTO
SEPERATE STRING
s.Clear();

s=e.Get();

v2=Split(s, "\n", true);

// TEST-1: RENAMEING FILES USING v1 FILE NAME TO v1 FILE NAME (Ya Same Names. for
tests only!)
s.Clear();

s<<"DEBUG INFO: TEST-1:\n";

for(int i=0; i<v1.GetCount(); i++){

 r=MoveFile(v1[i],v1[i]);

 (!r)? err_no=GetLastError() : err_no=0;

 s<<"RENAME "
  <<v1[i]<<" => "<<v1[i]
  <<" [Done="<<r<<"] [Error="
  <<err_no<<"]\n";
}


// TEST-2: RENAMEING FILES USING v1 FILE NAME TO v2 FILE NAME (Ya Same Names. but
from LineEdit Ctrl)
s<<"\nDEBUG INFO: TEST-2:\n";

for(int i=0; i<v1.GetCount(); i++){

 r=MoveFile(v1[i],v2[i]);
```

```
  (!r)? err_no=GetLastError() : err_no=0;

  s<<"RENAME "
   <<v1[i]<<" => "<<v2[i]
   <<" [Done="<<r<<"] [Error="
   <<err_no<<"]\n";
}

  s<<"\n***Error Code 123(0x7B) = ERROR_INVALID_NAME\n";

// PRINTING THE RESULTS
f.Set(s);

// Executing the Window
w.Run();
}
```

File Attachments
1) main.cpp, downloaded 388 times

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by navi on Mon, 19 Nov 2012 05:34:44 GMT
View Forum Message <> Reply to Message

Screen Shot of the Sample App.

File Attachments
1) pic1.PNG, downloaded 1176 times

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by navi on Mon, 19 Nov 2012 05:36:27 GMT
View Forum Message <> Reply to Message

Screen Shot of the Files.

File Attachments
1) pic2.PNG, downloaded 1172 times

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue

Posted by unknown user on Mon, 19 Nov 2012 07:23:31 GMT
View Forum Message <> Reply to Message

Hello Navi,

The problem is not with U++ but how you use WINAPI.
Quote:lpNewFileName [in]
The new name for the file or directory. The new name must not already exist. A new file may be on a different file system or drive. A new directory must be on the same drive.

What are you trying to do is exactly what this API call doesn't allow. It should work if you choose a new name for files.

Best regards,
Andrei

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by navi on Mon, 19 Nov 2012 07:44:27 GMT
View Forum Message <> Reply to Message

Hello Andrei,

Thanks for your reply. but I don't think "MoveFile" API has any issue renaming file with same name or with a different name in that matter. See the Test-1 success even when file is rename with same name. Yet on the Test-2 when the file names are fetched from the LineEdit Ctrl "MoveFile" complains invalid filename regardless if the name is same or different. Once you fetch text from LineEdit those text are not valid filename! My guess it might have hidden backslash "n" at the end or the CHARSET gets changed.

Thanks
Navi

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by koldo on Mon, 19 Nov 2012 07:59:17 GMT
View Forum Message <> Reply to Message

Hello Navi

I am not sure if this can serve you but in U++ there is a FileMove() function. Please check it if there is anything useful for you:

```
bool FileMove(const char *oldname, const char *newname)
{
#if defined(PLATFORM_WIN32)
 if(IsWinNT())
```

```
    ...
 else
   return !!MoveFile(ToSystemCharset(oldname), ToSystemCharset(newname));
#elif defined(PLATFORM_POSIX)
 ...
#else
 ...
#endif//PLATFORM
}
```

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by navi on Mon, 19 Nov 2012 08:54:37 GMT

View Forum Message <> Reply to Message

Hello koldo,

Many thanks for that piece of code. I will much prefer to use the U++ function, as it looks cross platform. however i have tried "FileMove" instead of "MoveFile" API. But the result is still the same error 123. After all FileMove is also calling "MoveFile" API at the end. I feel the problem is out of my deapth. I know there is something strange is going on when the filenames are fetched from LineEdit Ctrl. And I can not think of any wasys to investigate the inside the String. If I prints out the string on LineEdit Ctrl again, they visually looks normal. But my hunch is, its something to do with hidden chars or char codes.

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by koldo on Mon, 19 Nov 2012 09:17:35 GMT

View Forum Message <> Reply to Message

navi wrote on Mon, 19 November 2012 09:54Hello koldo,

Many thanks for that piece of code. I will much prefer to use the U++ function, as it looks cross platform. however i have tried "FileMove" instead of "MoveFile" API. But the result is still the same error 123. After all FileMove is also calling "MoveFile" API at the end. I feel the problem is out of my deapth. I know there is something strange is going on when the filenames are fetched from LineEdit Ctrl. And I can not think of any wasys to investigate the inside the String. If I prints out the string on LineEdit Ctrl again, they visually looks normal. But my hunch is, its something to do with hidden chars or char codes.

Hello Navi

IMHO 99.99% computing problems in U++ have a reason .

Please reduce your example until the simplest possible test case and we would check it.

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by deep on Mon, 19 Nov 2012 10:44:49 GMT

View Forum Message <> Reply to Message

I did some quick check by putting the v2 strings in String Buffer

Results shows there 0x0D at the end of the string. I think this may be the error.

```
 String s3;

 for ( int i = 0; i < v1.GetCount(); i++ ) {
  StringBuffer sb1;
  int len = v2[i].GetLength() ;
  sb1.SetLength ( len );
  sb1 = v2[i];

  for ( int i = 0; i < len; i++ ) {
   s3 << Format ( "%02X", ( byte ) sb1[i] ) << "|" ;
  }

  s3 << "\n\n";

  /*
    r=MoveFile(v1[i],v2[i]);

    (!r)? err_no=GetLastError() : err_no=0;

    s<<"RENAME "
     <<v1[i]<<" => "<<v2[i]
     <<" [Done="<<r<<"] [Error="
     <<err_no<<"]\n";
  */
 }

 s << "\n" << s3 << "\n";



 // PRINTING THE RESULTS
 f.Set ( s );
```

File Attachments
1) movefile.JPG, downloaded 1112 times

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by koldo on Mon, 19 Nov 2012 11:44:31 GMT

View Forum Message <> Reply to Message

Hello Deep

I see that all file names end in a '0D' char that is a '\r'.

v2 Vector has been filled from a LineEdit using:

v2 = Split(s, "\n", true);

This means that every v2 element ends with a '0D' char.

If you would do this no '0D' char would be added:

v2 = Split(s, "\r\n", true);

Have I interpreted it well?

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by deep on Mon, 19 Nov 2012 12:12:24 GMT
View Forum Message <> Reply to Message

Hi Koldo,

I think your solution will work.

It gives filename without any extra chars ("\r\n").

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by koldo on Mon, 19 Nov 2012 12:25:38 GMT
View Forum Message <> Reply to Message

99.99% + 1

Subject: Re: WIN32 API "MoveFile" and LineEdit Ctrl issue
Posted by navi on Mon, 19 Nov 2012 12:50:58 GMT
View Forum Message <> Reply to Message

Hello Deep and Koldo,

First of all let me congratulate and thank you guys for great detective investigation. Deep, great
technique for investigating strings. I also solve it by myself but was a bit late to reply. But I found
the solution and the cause of the problem by chance. Since, unlike Deep I could not figure how to
look inside the string, I assume the string has something invisible and started to write my very own

splitter. char by char. While doing that I stumble upon the issue that it's not that easy to split where '\n' is the delimiter. Upon searching the forum I stumble upon the following post by "great" Mirek: which essentially explains the problem and provides a elegant solution. Applying his Filter() solution I got it working immediately.

  http://www.ultimatepp.org/forum/index.php?t=msg&goto=104 91&

mirek wrote on Tue, 10 July 2007 23:26Well, it is not U++, it is M$... Back in MSDOS days, they have decided that line are to be ended with "\r\n" and this stupid idea is still alive.

Standard C library solves this problem by intruducing "text" and "binary" streams (text streams remove/insert \r on read/write), but IMO that just adds to confusion.

U++ general take on the problem is to ignore all \r characters on input and insert \r before each \n on output - but both is to be handled by the code.

In your case you have many options. One simple is to filter out all \r out before Split:

int NoCr(int c) { return c == '\r' ? 0 : c; }

Split(Filter(s, NoCr), '\n')

As you seem to ignore empty lines, another simple solution (with a bit better performance) is to treat \r like \n (because it will just result in empty line, which is ignored anyway):

int CrOrLf(int c) { return c == '\r' || c == '\n' ? c : 0; }

Split(s, CrOrLf);

Mirek

Screen Shot of the Program applying Mirek's Filter() Solution.

Thank you very much for your help guys. I really appreciate it.

Navi

File Attachments

1) pic3.png, downloaded 1042 times