

Re: RegQueryValueEx retrieved only one letter

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2007-07/msg00785.html>

- *From:* Joseph M. Newcomer <newcomer@xxxxxxxxxxxxx>
 - *Date:* Wed, 11 Jul 2007 00:26:01 -0400
-

See below...

On Wed, 11 Jul 2007 02:16:31 -0000, fifth <fifth8118@xxxxxxxxxx> wrote:

I want to retrieve the string value of a register key which is "C:\program files\test", but every time I only get one letter as "C" returned. My code is as follows, can somebody help me find out what happened?

```
////////////////////////////////////  
HKEY hKEY  
LPCTSTR data_Set=_T("Software\\test\\Settings");  
long ret0=(::RegOpenKeyEx (HKEY_CURRENT_USER, data_Set, 0,  
KEY_READ, &hKEY));  
if(ret0!=ERROR_SUCCESS)  
AfxMessageBox(_T("error: cannot open hKEY!"));
```

And since you can't open the key, why do you continue execution and try to use the nonexistent HKEY value?

```
char *szPath = new char[256];
```

OK, you are declaring this as char *. Note there are several errors here already:

- * You are using char as if it has meaning. It doesn't
- * You are calling 'new' when there is no possible way this makes sense
- * You are expecting telling it to convert an 8-bit character string to Unicode when you actually already have a Unicode string, so it does the obvious thing of treating the sequence 'C' '\0' '\\ '\0' 'P' '\0' etc. as an 8-bit character string, which is just the character 'C' as you got!

Why did you use 'char' at all? The specification of the API is clearly in terms of LPBYTE, so why not declare it as a BYTE array?

```
DWORD type_1=REG_SZ ;
```

Re: RegQueryValueEx retrieved only one letter

```
DWORD cbData_1=80;
```

Why? You have created a 256-character (byte) buffer and then you claim it is only 80 bytes long? Why are you using something weird like this, when the obviously simpler code is

```
BYTE szPath[256];  
DWORD cbData_1 = sizeof(szPath);  
DWORD type_1; // no need to initialize because it is an OUT parameter
```

The documentation clearly states that the type is an [out] parameter.

```
long ret1>::RegQueryValueEx(hKEY,_T("InstallPath"), NULL,&type_1,  
(BYTE *)szPath, &cbData_1);
```

```
long ret1 = ::RegQueryValueEx(hKey, _T("InstallPath"), NULL, &type_1, szPath, &cbData_1);
```

Why declare it as char and then cast it to BYTE*? Why not declare it as BYTE?

```
if(ret1!=ERROR_SUCCESS)  
AfxMessageBox(_T("error, cannot query hKEY!"));
```

Note that if the key failed to open, this call will fail, so you should not have allowed execution to reach this point. Then if it fails, you have an error, but you have not tested for it, and you continue execution as if the call worked!

```
::RegCloseKey(hKEY);
```

If the key didn't open, why are you trying to close it?

```
USES_CONVERSION;  
AfxMessageBox(A2T(ch));
```

And here is the error. Why did you assume the data coming in is an ASCII 8-bit character string? It isn't, it is a Unicode string, and you just insisted that it be treated as an 8-bit character sequence, so the first high-order 0 byte terminates the string and gives you one character.

I notice that you did NOT check the type_1 field to see if you actually read a string, and that is important. If the user has hand-edited this entry to have the wrong type, you are now in serious trouble. You MUST check the return type to make sure the buffer has any

Re: RegQueryValueEx retrieved only one letter

Re: RegQueryValueEx retrieved only one letter

meaning at all.

delete szPath;

This would not be necessary except for the misuse of 'new'. You don't need 'new' and you don't need 'delete'. Do NOT use 'new' when you don't need to do heap allocation, and there is no need for heap allocation here!

You might want to take a look at my Registry library on my MVP Tips site. I haven't written a registry API call in years.

joe

Joseph M. Newcomer [MVP]

email: newcomer@xxxxxxxxxxxxx

Web: <http://www.flounder.com>

MVP Tips: http://www.flounder.com/mvp_tips.htm

.