

Forum:[Hack Forums](#)

Thread: [Registry Query Fails!](#)

stdio - June 6, 2017, 4:51 am

I am trying to read value from outlook keys

Code #include <Windows.h>

#include <string>

using namespace std;

#define MAX_KEY_LENGTH 255

#define MAX_VALUE_NAME 16383

std::string GetLastErrorAsString()

```
{
    //Get the error message, if any.
    DWORD errorMessageID = ::GetLastError();
    if (errorMessageID == 0)
        return std::string(); //No error message has been recorded
```

```
    LPSTR messageBuffer = nullptr;
    size_t size = FormatMessageA(FORMAT_MESSAGE_ALLOCATE_BUFFER |
    FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS,
        NULL, errorMessageID, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
    (LPSTR)&messageBuffer, 0, NULL);
```

```
    std::string message(messageBuffer, size);
```

```
    //Free the buffer.
    LocalFree(messageBuffer);
```

```
    return message;
```

```
}
```

```
void OutlookPasswords() {
```

```
    HKEY hKey = NULL;
```

```
    char *subkey =
```

```
"Software\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6
676";
```

```
    if (RegOpenKeyEx(HKEY_CURRENT_USER, subkey, 0, KEY_READ, &hKey) !=
    ERROR_SUCCESS)
```

```
        return;
```

```
    // Variables for registry work
```

```
    TCHAR achKey[MAX_KEY_LENGTH]; // buffer for subkey name
```

```
    TCHAR achClass[MAX_PATH] = TEXT(""); // buffer for class name
```

```
    DWORD cchClassName = MAX_PATH; // size of class string
```

```
    DWORD cSubKeys = 0; // number of subkeys
```

```
    DWORD cbName, cbMaxSubKey, cchMaxClass, cValues, cchMaxValue, cbMaxValueData,
    cbSecurityDescriptor;
```

```
    FILETIME ftLastWriteTime; // last write time
```

```

DWORD i, retCode;

// Get the class name and the value count
if (RegQueryInfoKey(hKey, achClass, &cchClassName, NULL, &cSubKeys, &cbMaxSubKey,
&cchMaxClass, &cValues, &cchMaxValue, &cbMaxValueData, &cbSecurityDescriptor,
&ftLastWriteTime) != ERROR_SUCCESS)
{
    MessageBox(NULL, "Fail RegQueryInfoKey.\n", "A", MB_OK);
    return;
}

// Enumerate the subkeys, until RegEnumKeyEx fails
if (!cSubKeys)
{
    MessageBox(NULL, "Fail csubkey.\n", "A", MB_OK);
    return;
}
// For each profile entry
for (i = 0; i < cSubKeys; i++)
{
    cbName = MAX_KEY_LENGTH;

    // Get the name of each profile entry
    if (RegEnumKeyEx(hKey, i, achKey, &cbName, NULL, NULL, NULL, &ftLastWriteTime) !=
ERROR_SUCCESS)
    {
        MessageBox(NULL, "Fail enumKey.\n", "A", MB_OK);
        return;
    }
    // Combine the name with the profile path
    string newkey = string(subkey) + "\\\" + achKey;
    char* c_newkey = (char*)newkey.c_str();

    // Get the password registry value from combined path
    char* value = "";
    DWORD dw = 260;

    if (RegQueryValueEx(hKey, c_newkey, 0, 0, (BYTE*)value, &dw) != ERROR_SUCCESS)
    {
        MessageBox(NULL, "Failed RegQueryvalue\n", GetLastErrorAsString().c_str(), MB_OK);
        return;
    }
}

RegCloseKey(hKey);
}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int
nShowCmd)
{

```

```
OutlookPasswords());  
return 0;  
}
```

However it always fail saying "The specific module could not be loaded". I tried the same query using command prompt and it worked. Please help

Annabelle Lee - June 6, 2017, 8:15 am

(Jun 6, 2017 04:51 AM)Homi Wrote: [\[CITE\]](#) I am trying to read value from a specific key stored in variable "c_newkey"

```
Code char* value = "";  
DWORD dw = 260;
```

```
if (RegQueryValueEx(hKey, c_newkey, 0, 0, (BYTE*)value, &dw) != ERROR_SUCCESS)  
{  
    MessageBox(NULL, value, c_newkey, MB_OK);  
    MessageBox(NULL, "Fail RegQueryvalue\n", GetLastErrorAsString().c_str(), MB_OK);  
    //return;
```

}However it always fail saying "The specific module could not be loaded". I tried the same query using command prompt and it worked. Please help

Can you try making a very simple code that does only this? (A complete minimal example that can be compiled?) You'll then be able to see if it's this portion, or some other piece is causing this piece of code to fail.

HFDone - June 6, 2017, 10:46 am

check the example

[https://msdn.microsoft.com/en-us/library/windows/desktop/ms724911\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms724911(v=vs.85).aspx)

also use double slashes for the registry key

0xDEAD10CC - June 6, 2017, 12:08 pm

What are the real values of the first 2 params so we're not having to pull out the crystal ball here and take wild guesses...

You shouldn't be hardcoding buffer sizes. Try providing more of the relevant code.

stdio - June 7, 2017, 5:40 am

(Jun 6, 2017 08:15 AM)Annabelle Lee Wrote: [\[CITE\]](#) Can you try making a very simple code that does only this? (A complete minimal example that can be compiled?) You'll then be able to see if it's this portion, or some other piece is causing this piece of code to fail.

I have updated the code in main thread, please have a look.

thanks

(Jun 6, 2017 12:08 PM)0xDEAD10CC Wrote: [\[CITE\]](#) What are the real values of the first 2 params so we're not having to pull out the crystal ball here and take wild guesses...

You shouldn't be hardcoding buffer sizes. Try providing more of the relevant code.

Thanks for responding mate, I have posted the whole subjected code.

stackoverflowin - June 7, 2017, 5:49 am

What's your project character set in Visual Studio? You implicitly use an ANSI function (FormatMessageA), which makes me think its set to Unicode..

0xDEAD10CC - June 7, 2017, 7:25 am

(Jun 7, 2017 05:49 AM)stackoverflowin Wrote: [\[CITE\]](#) What's your project character set in Visual

Studio? You implicitly use an ANSI function (FormatMessageA), which makes me think its set to Unicode..

This would fail if that was the case:

```
Code MessageBox(NULL, "Fail RegQueryvalue\n", GetLastErrorAsString().c_str(), MB_OK);
```

Because he's not prefixing with L or using _T or TEXT. He's calling the macros but UNICODE isn't defined if this compiles. Anyways I don't have time until later today to look at this but he is mixing the macros with explicit ANSI calls from a quick glance which makes me think he doesn't understand it...

stdio - June 7, 2017, 9:01 am

(Jun 7, 2017 07:25 AM)0xDEAD10CC Wrote: [\[CITE\]](#) This would fail if that was the case:

```
Code MessageBox(NULL, "Fail RegQueryvalue\n", GetLastErrorAsString().c_str(), MB_OK);
```

Because he's not prefixing with L or using _T or TEXT. He's calling the macros but UNICODE isn't defined if this compiles. Anyways I don't have time until later today to look at this but he is mixing the macros with explicit ANSI calls from a quick glance which makes me think he doesn't understand it...

I am not using unicode character set.

0xDEAD10CC - June 7, 2017, 2:53 pm

(Jun 7, 2017 09:01 AM)Homi Wrote: [\[CITE\]](#) I am not using unicode character set.

That's my point. I just mentioned that UNICODE must not be defined if your code compiles which means that you're not using unicode.

I was pointing out however that you're mixing the macros and ANSI functions then for no reason, which doesn't make sense. You're not using the MAX_VALUE_NAME macro constant anywhere in your code. It looks like you took parts of this code (<https://msdn.microsoft.com/en-us/library/windows/desktop/ms724256%28v=vs.85%29.aspx?f=255&MSPPErr=-2147217396>) and mixed it in with your own as it's clear that the TCHAR and TEXT portions you don't understand if you're calling ANSI functions explicitly in other parts of your code.

using namespace std; - but you still specify std namespace in a lot of the code.

I would suggest reading over your code and looking at the documentation for the functions you're using. This isn't C#, you also seemingly need to read about read-only memory:

Hint:

```
Code char *value = "";
```

There's tons wrong with this.

stdio - June 7, 2017, 8:04 pm

(Jun 7, 2017 02:53 PM)0xDEAD10CC Wrote: [\[CITE\]](#) That's my point. I just mentioned that UNICODE must not be defined if your code compiles which means that you're not using unicode.

I was pointing out however that you're mixing the macros and ANSI functions then for no reason, which doesn't make sense. You're not using the MAX_VALUE_NAME macro constant anywhere in your code. It looks like you took parts of this code (<https://msdn.microsoft.com/en-us/library/windows/desktop/ms724256%28v=vs.85%29.aspx?f=255&MSPPErr=-2147217396>) and mixed it in with your own as it's clear that the TCHAR and TEXT portions you don't understand if you're calling ANSI functions explicitly in other parts of your code.

using namespace std; - but you still specify std namespace in a lot of the code.

I would suggest reading over your code and looking at the documentation for the functions you're using. This isn't C#, you also seemingly need to read about read-only memory:

Hint:

Code char *value = "";

There's tons wrong with this.

Fixing the code as per your guidance, really appreciate it mate.

Code #include <Windows.h>

#include <string>

using namespace std;

#define MAX_KEY_LENGTH 255

#define MAX_VALUE_NAME 16383

string GetLastErrorAsString()

{

 //Get the error message, if any.

 DWORD errorMessageID = ::GetLastError();

 if (errorMessageID == 0)

 return "No Error Message ID"; //No error message has been recorded

 char* messageBuffer = nullptr;

 size_t size = FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS,
 NULL, errorMessageID, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
messageBuffer, 0, NULL);

 string message(messageBuffer, size);

 //Free the buffer.

 LocalFree(messageBuffer);

 return message;

}

void OutlookPasswords() {

 HKEY hKey = NULL;

 char *subkey =

"Software\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6
676";

 if (RegOpenKeyEx(HKEY_CURRENT_USER, subkey, 0, KEY_READ, &hKey) !=
ERROR_SUCCESS)

 return;

 // Variables for registry work

 CHAR achKey[MAX_KEY_LENGTH]; // buffer for subkey name

 CHAR achClass[MAX_PATH]; // buffer for class name

 DWORD cchClassName = MAX_PATH; // size of class string

 DWORD cSubKeys = 0; // number of subkeys

 DWORD cbName, cbMaxSubKey, cchMaxClass, cValues, cchMaxValue, cbMaxValueData,
cbSecurityDescriptor;

 FILETIME ftLastWriteTime; // last write time

 DWORD i, retCode;

 // Get the class name and the value count

```

    if (RegQueryInfoKey(hKey, achClass, &cchClassName, NULL, &cSubKeys, &cbMaxSubKey,
&cchMaxClass, &cValues, &cchMaxValue, &cbMaxValueData, &cbSecurityDescriptor,
&ftLastWriteTime) != ERROR_SUCCESS)
    {
        MessageBox(NULL, "Fail RegQueryInfoKey.\n", "A", MB_OK);
        return;
    }

    // Enumerate the subkeys, until RegEnumKeyEx fails
    if (!cSubKeys)
    {
        MessageBox(NULL, "Fail csubkey.\n", "A", MB_OK);
        return;
    }
    // For each profile entry
    for (i = 0; i < cSubKeys; i++)
    {
        cbName = MAX_KEY_LENGTH;

        // Get the name of each profile entry
        if (RegEnumKeyEx(hKey, i, achKey, &cbName, NULL, NULL, NULL, &ftLastWriteTime) !=
ERROR_SUCCESS)
        {
            MessageBox(NULL, "Fail enumKey.\n", "A", MB_OK);
            return;
        }
        // Combine the name with the profile path
        string newkey = string(subkey) + "\\\" + achKey;
        char* c_newkey = (char*)newkey.c_str();

        // Get the password registry value from combined path
        char* value;
        DWORD dw = 260;

        if (RegQueryValueEx(hKey, c_newkey, 0, 0, (BYTE*)value, &dw) != ERROR_SUCCESS)
        {
            MessageBox(NULL, "Failed RegQueryvalue\n", GetLastErrorAsString().c_str(), MB_OK);
            return;
        }
    }

    RegCloseKey(hKey);
}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int
nShowCmd)
{
    OutlookPasswords();
    return 0;
}

```

There still seems to be some mistakes, as I cannot get the desired result. Please check it :)

0xDEAD10CC - June 7, 2017, 9:13 pm

```
Code char* value;
DWORD dw = 260;
if (RegQueryValueEx(hKey, c_newkey, 0, 0, (BYTE*)value, &dw) != ERROR_SUCCESS)
{
    ...
}
```

Let me re-phrase what I meant to say...

Where you have 'value' is meant to be a pre-allocated buffer... And where you have '&dw' you're telling it that you've allocated 260 bytes at the address specified by the pointer 'value' which isn't the case at all... Read the docs. Last time 'value' was a read-only memory address, this time it's unspecified memory address to which you haven't allocated any memory yourself.

stdio - June 7, 2017, 11:18 pm

(Jun 7, 2017 09:13 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Let me re-phrase what I meant to say...

Where you have 'value' is meant to be a pre-allocated buffer... And where you have '&dw' you're telling it that you've allocated 260 bytes at the address specified by the pointer 'value' which isn't the case at all... Read the docs. Last time 'value' was a read-only memory address, this time it's unspecified memory address to which you haven't allocated any memory yourself.

Alright! read some documents about what you said, hope I got it right this time.

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

```
#include <string>
```

```
using namespace std;
```

```
#define MAX_KEY_LENGTH 255
```

```
#define MAX_VALUE_NAME 16383
```

```
string GetLastErrorAsString()
```

```
{
```

```
    //Get the error message, if any.
```

```
    DWORD errorMessageID = ::GetLastError();
```

```
    if (errorMessageID == 0)
```

```
        return "No Error Message ID"; //No error message has been recorded
```

```
    char* messageBuffer = nullptr;
```

```
    size_t size = FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS,
        NULL, errorMessageID, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
messageBuffer, 0, NULL);
```

```
    string message(messageBuffer, size);
```

```
    //Free the buffer.
```

```
    LocalFree(messageBuffer);
```

```
    return message;
```

```
}
```

```
void OutlookPasswords() {
```

```
    HKEY hKey = NULL;
```

```
    char *subkey =
```

```
"Software\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6
```

```

676";
    if (RegOpenKeyEx(HKEY_CURRENT_USER, subkey, 0, KEY_READ, &hKey) !=
ERROR_SUCCESS)
        return;

    // Variables for registry work
    CHAR    achKey[MAX_KEY_LENGTH]; // buffer for subkey name
    CHAR    achClass[MAX_PATH]; // buffer for class name
    DWORD    cchClassName = MAX_PATH; // size of class string
    DWORD    cSubKeys = 0; // number of subkeys
    DWORD    cbName, cbMaxSubKey, cchMaxClass, cValues, cchMaxValue, cbMaxValueData,
cbSecurityDescriptor;
    FILETIME ftLastWriteTime; // last write time

    DWORD i, retCode;

    // Get the class name and the value count
    if (RegQueryInfoKey(hKey, achClass, &cchClassName, NULL, &cSubKeys, &cbMaxSubKey,
&cchMaxClass, &cValues, &cchMaxValue, &cbMaxValueData, &cbSecurityDescriptor,
&ftLastWriteTime) != ERROR_SUCCESS)
    {
        MessageBox(NULL, "Fail RegQueryInfoKey.\n", "A", MB_OK);
        return;
    }

    // Enumerate the subkeys, until RegEnumKeyEx fails
    if (!cSubKeys)
    {
        MessageBox(NULL, "Fail csubkey.\n", "A", MB_OK);
        return;
    }
    // For each profile entry
    for (i = 0; i < cSubKeys; i++)
    {
        cbName = MAX_KEY_LENGTH;

        // Get the name of each profile entry
        if (RegEnumKeyEx(hKey, i, achKey, &cbName, NULL, NULL, NULL, &ftLastWriteTime) !=
ERROR_SUCCESS)
        {
            MessageBox(NULL, "Fail enumKey.\n", "A", MB_OK);
            return;
        }
        // Combine the name with the profile path
        string newkey = string(subkey) + "\\\" + achKey;
        char* c_newkey = (char*)newkey.c_str();

        // Get the password registry value from combined path
        char value[1024];
        DWORD dw = 1024;
        //DWORD dwType;
        if (RegQueryValueEx(hKey, c_newkey, NULL, NULL, (BYTE*)value, &dw) !=

```

```
ERROR_SUCCESS)
```

```
{  
    MessageBox(NULL, "Failed RegQueryvalue\n", GetLastErrorAsString().c_str(), MB_OK);  
    return;  
}
```

```
}
```

```
RegCloseKey(hKey);
```

```
}
```

```
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int  
nShowCmd)
```

```
{  
    OutlookPasswords();  
    return 0;
```

```
}
```

Someone please help me, stuck with this..

HFDone - June 8, 2017, 10:49 am

<http://www.rohitab.com/discuss/topic/39352-outlook-passwords-recover-c>