When you open a securable object, make sure you pass the security mask you actually want (no more, no less)

devblogs.microsoft.com/oldnewthing/20151029-00

October 29, 2015

Raymond Chen

There are two categories of "Access denied" errors. One occurs when you attempt to create the handle, and the other occurs when you attempt to use the handle.

```
HANDLE hEvent = OpenEvent(SYNCHRONIZE, FALSE, TEXT("MyEvent"));
```

If this call fails with *Access denied*, then it means that you don't have access to the object to the level you requested. In the above example, it means that you don't have `SYNCHRONIZE` access to the event.

A common reason for getting an *Access denied* when trying to create a handle is that you asked for too much access. For example, you might write

```
HKEY hkey;
LONG lError = RegOpenKeyEx(
    hkeyRoot, subkeyName, 0, KEY_ALL_ACCESS, &hkey);
if (lError == ERROR_SUCCESS) {
 DWORD dwType;
 DWORD dwData;
 DWORD cbData = sizeof(dwData);
 lError = RegQueryValueEx(hkey, TEXT("ValueName"), nullptr,
                          &dwType, &dwData, &cbData);
 if (lError == ERROR_SUCCESS && dwType == REG_DWORD &&
     cbData == sizeof(dwData)) {
  .. do something with dwData ..
 }
 RegCloseKey(hkey);
}
```

The call to `RegOpenKeyEx` fails with *Access denied*. The proximate reason is that you don't have `KEY_ALL_ACCESS` permission on the registry key, which makes sense because `KEY_ALL_ACCESS` asks for permission to do *everything imaginable* to the registry key, including crazy things like "Change the permissions of the key to deny access to the rightful owner."

But why are you asking for full access to the key if all you're going to do is read from it?

```
HKEY hkey;
LONG lError = RegOpenKeyEx(
    hkeyRoot, subkeyName, 0, KEY_READ, &hkey);
if (lError == ERROR_SUCCESS) {
 DWORD dwType;
 DWORD dwData;
 DWORD cbData = sizeof(dwData);
 lError = RegQueryValueEx(hkey, TEXT("ValueName"), nullptr,
                          &dwType, &dwData, &cbData);
 if (lError == ERROR_SUCCESS && dwType == REG_DWORD &&
     cbData == sizeof(dwData)) {
  .. do something with dwData ..
 }
 RegCloseKey(hkey);
}
```

If you want to go for bonus points, ask for `KEY_QUERY_VALUE` instead of `KEY_READ`, since all you are going to do with the key is read a value.

> When requesting access to an object, it's best to ask for the minimum access required to get the job done.

This is like the old principle of mathematics: After you've proved something, try to weaken the hypothesis as much as possible and strengthen the conclusions as much as possible. In other words, once you've solved a problem, figure out the absolute minimum requirements for your solution to work, and figure out the largest amount of information your solution produces.

On the other hand, if you get an *Access denied* error when trying to use a handle, then the problem is that you didn't open the handle with *enough* access.

```
HKEY hkey;
LONG lError = RegOpenKeyEx(
    hkeyRoot, subkeyName, 0, KEY_READ, &hkey);
if (lError == ERROR_SUCCESS) {
 DWORD dwData = 1;
 lError = RegSetValueEx(hkey, TEXT("ValueName"), nullptr,
             REG_DWORD, (const BYTE*>)&dwData, sizeof(dwData));
 if (lError == ERROR_SUCCESS && dwType == REG_DWORD &&
     cbData == sizeof(dwData)) {
  .. do something with dwData ..
 }
 RegCloseKey(hkey);
}
```

Here, the `RegOpenKeyEx` succeeds, but the `RegSetValueEx` fails. That's because the registry key was opened for `KEY_READ` access, but the `RegSetValueEx` operation requires `KEY_SET_VALUE` access. To fix this, you need to open the key with the access you actually want:

```
HKEY hkey;
LONG lError = RegOpenKeyEx(
    hkeyRoot, subkeyName, 0, KEY_SET_VALUE, &hkey);
if (lError == ERROR_SUCCESS) {
 DWORD dwData = 1;
 lError = RegSetValueEx(hkey, TEXT("ValueName"), nullptr,
             REG_DWORD, (const BYTE*>)&dwData, sizeof(dwData));
 if (lError == ERROR_SUCCESS && dwType == REG_DWORD &&
     cbData == sizeof(dwData)) {
  .. do something with dwData ..
 }
 RegCloseKey(hkey);
}
```

> When requesting access to an object, it's best to ask for the minimum access required to get the job done, but no less.

Armed with this information, you can solve this problem:

> In the main thread, we create an event like this:
>
> ```
> TheEvent = CreateEvent(NULL, TRUE, FALSE, name);
> ```
>
> A worker thread opens the event like this:
>
> ```
> EventHandle = OpenEvent(SYNCHRONIZE, FALSE, name);
> ```
>
> The `OpenEvent` succeeds, but we try to use the handle, we get *Access denied*:
>
> ```
> SetEvent(EventHandle);
> ```
>
> On the other hand, if the worker thread uses the `CreateEvent` function to get the handle, then the `SetEvent` succeeds.
>
> What are we doing wrong?

Raymond Chen

Follow