

# An easy way to determine whether you have a particular file permission

 [devblogs.microsoft.com/oldnewthing/20040604-00](http://devblogs.microsoft.com/oldnewthing/20040604-00)

June 4, 2004



Raymond Chen

Sometimes you might want to determine whether you **can** do something without actually doing it. For example, you might want to know whether you have a particular permission in a directory, say permission to delete files from it.

One way is to retrieve the ACL and then check whether the current user has the desired permission. The [AccessCheck function](#) does most of the heavy lifting there.

Or you can realize, “Hey, wait a second, there is an entire security infrastructure whose job it is to decide who can access which files. Why not use it?”

For example, here’s how you can check whether the user has permission to delete files from a directory:

```
BOOL CanDeleteFilesFromDirectory(LPCTSTR pszPath)
{
    HANDLE h = CreateFile(pszPath, FILE_DELETE_CHILD, FILE_SHARE_READ |
                        FILE_SHARE_WRITE | FILE_SHARE_DELETE, NULL,
                        OPEN_EXISTING, FILE_FLAG_BACKUP_SEMANTICS, NULL);
    if (h != INVALID_HANDLE_VALUE) {
        CloseHandle(h);
    }
    return h != INVALID_HANDLE_VALUE;
}
```

What we did was open the directory (which requires backup semantics) and ask for `FILE_DELETE_CHILD` access. If it succeeded, then we have permission to delete files from it. [Corrected 7:52am]

Note, of course, that this information is purely advisory. You shouldn’t be making security decisions based on this, because the permissions might change between the time you check and the time you try.

[Raymond Chen](#)

**Follow**

