

How do we change permissions on a share as fast as Explorer does it?

 devblogs.microsoft.com/oldnewthing/20151007-00

October 7, 2015



Raymond Chen

A customer wanted to know what they could do to speed up changing permissions on a share.

We have a large share (over 50,000 files in over 5000 folders). If we have four or five users with access to the share, and we add one more user, it takes around two minutes. If we have a hundred users and add one more, it takes around seven minutes. Here's the code we are using to add a user:

```
// Get the current ACL on the directory.
var info = new DirectoryInfo(sharePath);
var security = info.GetAccessControl(AccessControlSections.Access);

// Create a new rule that grants the user read access.
var rule = new FileSystemAccessRule(user,
    FileSystemRights.Read | FileSystemRight.Synchronize,
    InheritanceFlags.ContainerInherit |
    InheritanceFlags.ObjectInherit,
    PropagationFlags.None,
    AccesControlType.Allow);

// Add the rule to the existing ACL.
security.AddAccessRule(rule);
security.SetAccessRuleProtection(true, false);

// Set the modified ACL back on the directory.
info.SetAccessControl(security);
```

What can we do get this code to perform as well as Explorer?

One suggestion was to create a security group that encompasses the people who have access to the share. The job of controlling who has access to the share is now delegated to the security group. When you need to grant Bob access to the share, you merely add Bob to the security group. When you decide that Bob no longer has access, you remove him from the group.

The customer liaison replied, “The customer cannot use this workaround. How can they improve their code so it is as fast as Explorer?”

We were unable to figure out exactly why the customer refused to use security groups. Whenever we asked, the customer merely changed the subject and asked how to get their code to be as fast as Explorer.

I stepped in and pointed out the obvious:

The customer’s code is not doing what they claim they are doing. They claim to be changing share permissions, but the code is actually changing *directory* permissions.

To change share permissions, use `GetNamedSecurityInfo` to read the existing permissions on the share and `SetNamedSecurityInfo` to change the permissions. There doesn’t appear to be a BCL class for doing this, so you may need to pinvoke to the underlying Win32 functions, then use `ObjectSecurity.SetSecurityDescriptorBinaryForm` to convert it into something you can manipulate more easily.

That was the last we heard from the customer, so it’s not clear whether they gave up and decided that we weren’t being helpful, or whether the tap on the shoulder was enough to clue them in to why their comparison was a false one. I don’t know whether they took any of our advice to put the security information in a security group, but I suspect they didn’t.

Raymond Chen

Follow

