

Re: How to programmngly check off "Allow inheritable permissions

Re: How to programmngly check off "Allow inheritable permissions

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2006-08/msg00132.html>

- *From:* tchu <tchu@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sat, 5 Aug 2006 00:06:02 -0700
-

Thanks for your guidance, Dominick'
I am not familiar with .Net 2.0 yet, but maybe now comes the chance to migrate to it~

Timothy

"Dominick Baier" wrote:

I never used this library – but if you are targeting .NET 2.0 you can find all the functionality in System.Security.AccessControl

dominick

Hi, I am using Microsoft.Win32.Security package(C#) to set the NT security of specified folder/file.

My purpose is to protect the content of files under a specified folder from being read by "Users" group members, whereas Users can still list the name of these files, and tranverse through subfolders.

I have tried the code as follows:

```
-----  
-----  
SecurityDescriptor secDesc =  
SecurityDescriptor.GetFileSecurity(strFullPath,  
SECURITY_INFORMATION.DACL_SECURITY_INFORMATION);  
Dacl dacl = secDesc.Dacl;  
Sid sidUsers = new Sid ("Users");  
dacl.RemoveAces(sidUsers);  
FileAccessType FAType = FileAccessType.READ_CONTROL |  
FileAccessType.FILE_READ_ATTRIBUTES |  
FileAccessType.FILE_READ_EA;  
dacl.AddAce (new AceAccessAllowed (sidUsers,  
(AccessType)FAType,AceFlags.CONTAINER_INHERIT_ACE |  
AceFlags.OBJECT_INHERIT_ACE));
```

Re: How to programmngly check off "Allow inheritable permissions

```
DirectoryAccessType DAType =  
DirectoryAccessType.FILE_LIST_DIRECTORY |  
DirectoryAccessType.FILE_TRAVERSE;  
dacl.AddAce (new AceAccessAllowed (sidUsers, (AccessType)DAType,  
AceFlags.CONTAINER_INHERIT_ACE | AceFlags.INHERITED_ACE));  
secDesc.SetDacl(dacl);  
secDesc.SetFileSecurity(strFullPath,  
SECURITY_INFORMATION.DACL_SECURITY_INFORMATION);
```

The code just worked as expected -- I mean just as what I manually set when testing the security setting. However, by running the code, objects under the folder just keep their security setting -- inherit from much upper parent objects...

I think the only difference is that when manually setting the security, I firstly check off the "Allow inheritable permissions from the parent to propagate to this object and all child objects" in the Advanced Security Setting dialog. By doing so, all objects under the folder will change to inherit their security from this folder, expectably. So, is it possible to implement this CheckOff in C# programming?

Best regards,
Timothy Hu