

## Re: Understanding XP file permissions ? (Application Programs not following standards ?)

**Source:**

[http://www.derkeiler.com/Newsgroups/microsoft.public.windowsxp.security\\_admin/2002-12/22482.html](http://www.derkeiler.com/Newsgroups/microsoft.public.windowsxp.security_admin/2002-12/22482.html)

---

**From:** Roger Abell [MVP] ([mvpNOSPAM@asu.edu](mailto:mvpNOSPAM@asu.edu))

**Date:** 12/23/02

From: "Roger Abell [MVP]" <[mvpNOSPAM@asu.edu](mailto:mvpNOSPAM@asu.edu)>

Date: Mon, 23 Dec 2002 02:40:08 -0700

dvk,

That is quite a discourse, and coming from a \*nix background myself, I can relate.

Let me say off the top that I am not going to recommend sites / books that offer a concise, pre-digested view of permissioning in Windows.

Hopefully many others will post to help you with that.

If I were to give one site, it would be [www.reskit.com](http://www.reskit.com) but that again will err on the side of detail and too many trees to let you simply see the forest.

Further comments are inlined . . .

"dvk" <[dvk@fuxorina.net](mailto:dvk@fuxorina.net)> wrote in message  
news:k60d0vgb1ngfgekoiaairqr1t6tcv1nss4@4ax.com...  
> *I've been trying to understand how file permissions in Windows NT/XP  
> work. I want to give certain users on my computer access to certain  
> files and programs. The problem is there really is no accepted  
> standard in how programs store their data files.*

There is a defined standard. The problem is getting the software houses to see a market incentive accept it and so produce Windows logo certifiable software products.

> *So I have to mess  
> with permissions for almost every program I have installed, ,after I  
> have installed it. Seems they change this location in every version of  
> windows also.*

Yes, things have been changing. A big step was taken with the

release of Windows 2000, and a further step with XP, along the road of trying to bring some order to what was before total anarchy. MS cannot dictate, and in fact during the W2k dev cycle we had to push to get MS to be as tough as they ended up being. Do not get me wrong, many within MS wanted to solve the problem, but the reality was that a gentle middle ground had to be taken.

> *In XP, each user has an "Application Data" folder in their user profile folder under "Documents and Settings".*  
>

Yes, and for the purpose of allowing, when this is used in accordance with the logo certification program, the executable to be isolated from the parts needing write access, and also to allow for a per-user experience.

> *Almost no program uses this folder though.*

And they will not until they see it as something hurting sales. I personally am vocal with software houses of what I want to see in their products' behaviors before my dollars flow.

> *I'll use Forte Agent as an example. It wants to store all its data in its own folder under "Program Files". There is no problem with this as long as I am always Administrator. But what if I want to be a limited user and use the Administrator account only for administrative purposes as it is intended to be ? I have to change permissions for the "Agent" folder under "Program Files" so limited users can actually use the program.*  
>

Complain to them. They need to hear this in a way that can be translated into their bottom line since it will cost to adjust their code.

> *This is pretty simple, just add the "Users" group to this folder, give it full permissions, and tell everything under it to inherit these permissions. But this allows anyone to do anything with this program.*

Yep. But you, and every other person, should not have to do this.

Some say the problem with getting applications to run without giving out admin power is MS's fault. In a sense yes, as they did set the strategy of how a program needs to behave, but no in that we corporate users told MS what we need. We come from long use of \*Nix where our executables are locked from change, usually at a high directory if not partition level. MS is just attempting to bring some order into the chaos that resulted from DOS-oriented software companies starting to write products for a system that has a security system.

- > *They can delete the entire program if they wish. Now things get more*
- > *complicated. I want users to be able to use the program to read and*
- > *save newsgroups, but I don't want to let them do anything with the*
- > *program, such as delete needed files. So now I have to find out which*
- > *files get modified with this program under normal use, and set*
- > *specific permissions on specific files. Multiply this by 20 other*
- > *programs I might install that don't follow the standard, and it can*
- > *get to be a hassle.*
- >

The user should have no need to micromanage these things. Vote by buying Windows certified applications. That is the most rapid way to solve this. While it is true that one can protect files to any degree needed even within a folder that allows users to create/modify/delete other files and subfolders, it is just not something that people should have to be doing.

At least now very few third party software products are still using any place of choice in c:\windows as scratch space as they used to do.

- > *If program-specific data files were stored in "Application Data" for*
- > *each user, this problem would be eliminated, and each user would have*
- > *their own copy of their data/settings etc.*
- >

ditto

- > *Yes, even Microsoft doesn't follow their own standards! Log in under a*

MS is a big place. Office did not initially work within W2k due to the changes W2k made in this direction either.

- > *non-administrator user. Open up the calculator program. Change it from*
- > *Standard to Scientific or vice versa. Now close the calc program and*
- > *open it again. Your setting was not saved. This is because*
- > *Calculator's settings are stored under HKEY\_LOCAL\_MACHINE in the*
- > *registry, which normal users can not modify. It should be under*
- > *HKEY\_CURRENT\_USER. Shame on you, Microsoft!*
- >
- > *I also have a question about changing ownership of folders/files. It*
- > *took me about an hour to figure out how to give ownership to somebody*
- > *else. I am used to the Unix way of file permissions, and understand*
- > *them almost fully, so maybe my mind has been corrupted by it's ways.*
- > *File permissions in Unix are much simpler and much better in my*
- > *opinion. Just use the chown program and use the proper arguments.*
- >

Unix permissions are very simple, which make management less complex. However, it also makes the system much less able to expressively model the work environment. Windows, allowing

a user to be in any number of groups, provides a much richer control structure. When used without planning things can get rather messy over time. When used with a design one starts to appreciate what things one can do that are impossible with Unix.

IIRC the original reason why ownership changing was handled this way was to be clearly on the acceptable side of the trusted systems certification process, needed to become a player in the government market. So, originally it was never possible to give ownership, only to give the right to take ownership. The reasoning ran, if an admin can change ownership there are (at least) two results that follow. One, the audit trail when there is an access accomplished by an account outside of its intended permissions is more obscure as it can be done by multiple accounts, one to give another ownership, the new owner to grant access permissions to another, and that other to access. If the account that becomes owner has to take ownership then some of this audit trail is less fragmentable. And two, if the CEO owned some private files, and there is no way to give ownership, then taking ownership in order to be able to grant permissions to access the files may be possible, but the ownership could not be set back as it was to begin with (at least without actually using the CEO account, in which case there would have been access more easily anyway), and each step of ownership change is fully auditable (account X used take ownership permission).

Anyway, there are now ways to grant ownership in script, and there is a resource kit utility that can be used. Too many have apparently found the "grant the right to take ownership" model to be short of benefit for its cost, although I am not sure why. There was wisdom for the original design and I would frankly like to see things strengthened back to that model, but then I have learned to live with the few times that it is actually inconvenient (and they are very few).

> *My first instinct in windows was to click on the Security Tab, click  
> Advanced, then click the Owner tab, and it would give me a list of  
> users to change ownership to. But this was not the case. I had to add  
> this user to the long list of user/groups associated with this file.  
> Then I had to edit this user's "Special Permissions" to allow to take  
> ownership. Then I had to login under this user and "take ownership".  
> This is too much of a hassle IMHO.  
>*

I got carried away in the earlier relating to your comment here . . .

Once you get a little more used to the non–Unix flavors of filesystem (and registry) security in NT/W2k/XP you may start to ease in the ownership area. Leaving things owned by Administrators for the

large part, and where you do want it otherwise using Creator Owner actually does not work badly. I myself had a very hard time at first with the idea that user accounts had areas to which admins had no permissions whatsoever (an idea totally alien to Unix, in fact impossible in it).

> *Maybe I just don't fully understand permissions under XP. So I am asking if there is any resource / book out there that explains this completely? Microsoft's help docs and website help a little bit, but it explains too much specifics and not enough of the general stuff for me. Maybe a good book or website out there has a chapter that explains everything? Personally I think permissions are much too complex in windows.*

They are rich. Their use can be complex, but there are measures that can be used to contain this to a balance where function is obtained in the trade. It takes some time getting used to when coming from an experience \*nix sys admin point of view, but if you first focus on the design implications that fall out from a user being a member in many, many groups you many start to see some of the incentives. Any, hopefully the brief history of nudging anarchy over to a manageable order will help you understand some of this.

--  
Roger Abell  
MS MVP (Security, Windows), MCDBA, MCSE both  
Associate Expert - Windows XP ExpertZone  
<http://www.microsoft.com/windowsxp/expertzone>