Re: How to modify program files in Vista?

Source: http://www.tech−archive.net/Archive/VC/microsoft.public.vc.mfc/2008−08/msg00344.html

- *From*: Daniel James <wastebasket@xxxxxxxxxxxxxxx>
- *Date*: Fri, 08 Aug 2008 11:35:55 +0100

In article news:<65ul94h3dv2qcj83r716q997hue9v7lks2@xxxxxxx>, Joseph M. Newcomer wrote:

> > If there is an application ini file that is common to all users there isn't really any provision to store that file in the prescribed Windows file hierarchy.
>
> ****
> There is a "common AppData" folder (CSIDL_COMMON_APPDATA) that does this.
> ****

There is ... but that directory (C:\Documents and Settings\All Users\Application Data, or something different on Vista) and any directory created within it (by default, at least) is not writable by non−Administrative users.

That's the point: There is no standard place for common WRITABLE files.

An installer can, of course, change the permisions on any file or directory it creates so the problem is far from insoluble. What I'm getting at here is that without any clear guidance from Microsoft everyone and his dog will come up with a different scheme and confusion will ensue.

> > Generally, HKLM is used to set site defaults during installation (which requires admin privileges) and is not changeable by ordinary users. User data is kept in HKCU.

Again: agreed. But if one wants to create a registry entry that can be accessed and updated by all users then one has little option but to create it within HKLM and change the permissions. There isn't even an HKU\AllUsers hive ...

> > Generally, as a concept INI files suck (try storing a set of font

> definitions in one, for example...painful!); I either use the Registry,
> or if I want a file, an XML file.

Also true, but I was addressing the general problem that there isn't a standard, recommended, location for ANY sort of writable data files shared by all users.

INI files have the advantage that most users with access to notepad can understand how to edit them without breaking anything, which is not necessarily the case with XML or other more structured file types. That's also a DISadvantage, of course!

> Individual users should not be allowed to modify global configuration
> data.

I'm not talking only about configuration data in any strict sense. True configuration data should be set up by some kind of administrator and not altered by unprivileged users, of course.

There are plenty of other types of data that it is perfectly proper to allow all users of a system to read and update ... and, yes, sometimes these data are even held in INI files (though I agree that's not necessarily a good choice).

> > 2. Place your INI file in the program directory, as you do at present,
> > but change your installation program so that it grants write access to
> > the INI file (ONLY the INI file) to all users.

[snip]

> > ****
> > THe problem with this is that corporate security may detect a
> > writeable file and automatically reprotect it. Security policies
> > vary, but are often preemptive.
> > ****

I've heard it suggested before that something along those lines might happen. What "corporate security" applications do you know that will do that? I'd be interested to know ...

> > 3. Create an application data folder in the all users profile folder
> > and write your INI file there. You will have to change the permissions
> > on this folder so that it can be written by normal users (but that's
> > less potentially risky than changing permissions in the Program Files
> > directory.

> ****
> It if really is global data, yes, this would be the safest strategy.
> But I doubt that it is truly global state; more likely it is per−user
> state, improperly stored.
> ****

This is, in fact, what I do in one legacy application I maintain. The files stored in the common writable area do include some per−user files that could be stored elsewhere and some data that should really be writable only by administrators ... but most of it is data that genuinely is public and genuinely does have to be writable by all users.

Unfortunately the data are (for historical reasons) all rather mixed together so it's not really possible (without a major rewrite) to separate the bits that could be per−user and the bits that should be read−only from the rest.

Had I the luxury of a complete rewrite I should implement the back−end of the application as a service and run that service under an account that gave it access to all the necessary files. Each user could then update the shared data under the program's control (after authenticating, etc.).

> > 4. Use the registry. Create a suitable key in the HKLM hive for your
> > application and change the permissions on it so that all users can
> > write to it.
>
> ****
> THe same security software that protects Program Files will often
> reprotect HKLM. I've seen this happen once already.
> ****

I can't say I'm entirely surprised ... though it does seem a little draconian, especially as there is no HKU\All Users.

What's the software that does this?

Cheers,
Daniel.

.