# WINDOWS CONFIDENTIAL

## An Administrator Is Not *the* Administrator

Raymond Chen

**I came across** a report from a user who was trying to set the owner of a file to "Administrator." The user was unable to do this even though he was logged on as an administrator. Why won't the system let an administrator change the owner of a file to Administrator? Don't administrators have permission to take ownership of files?

But you see, "Administrator" and "Administrators" are not the same. That plural marker means all the difference. Indeed, the subtlety of that plural marker creates problems for localizers.

My colleague Jesper Holmberg points out that the word in Swedish for "Owner" (which is what Administrator is called) in Windows® XP Home Edition is "Ägare". Unfortunately, that is one of those words that does not take a plural marker. Jesper's workaround was to change the translation of "Owners" to "Ägaren". (You can read more on Jeppe's Weblog.)

So do you know the difference between Administrator and Administrators? Administrator is an account. If a permission or privilege is granted to Administrator, it can be done only by someone logged in with the Administrator account, that is, the account whose name defaults to Administrator (in English).

Administrators, on the other hand, is a group. If you are a member of the Administrators group of a machine, you have been granted administrator privileges on that machine. It is membership in the Administrators group that people refer to when they say things like "I'm an administrator on this machine." The use of an indefinite article ("an") as opposed to a definite article ("the") highlights that the user is just one of many administrators.

Things are more ambiguous when people say something like "I'm running as administrator." This could mean either they are running as the Administrator account or that they are running with an account that is a member of the Administrators group.

Once you understand this difference, it becomes clear why the user I mentioned earlier was unable to reassign ownership of the file. The user was logged on with an account that belongs to the Administrators group—but not with the Administrator account itself. Let's call the user's account "Bob." The SeTakeOwnershipPrivilege privilege is assigned to members of the Administrators group, allowing members of this group to assign ownership to themselves. However, this privilege does not let members assign ownership to somebody else. In this example, Bob could assign ownership of the file to Bob. But he is mistakenly trying to assign ownership to Administrator and since Bob is not the same as Administrator, the operation fails.

The user needs to log on with the Administrator account and take ownership of the files from there. In this case, the Administrator is assigning ownership to himself. (Alternatively, Bob could enable SeRestorePrivilege before setting the owner. This method is somewhat unorthodox, however, since SeRestorePrivilege is intended to be used by backup restore programs.)

When setting security descriptors, it is strongly recommended not to assign a right exclusively to the Administrator account. If you do, anybody who wants to exercise that right would have to log off from their normal account and log back on as the Administrator account.

A better practice is to assign the right to the Administrators group. This allows any member of the Administrators group to exercise the right without you having to give out any passwords.

Using the Security Descriptor Definition Language (SDDL) to build security descriptors translates into avoiding the LA (Local Administrator account) trustee in favor of the BA (Built-in Administrators group).

Of course, it'd be even better to avoid assigning the right to the Administrators group, because that scenario makes it impossible for the right to be delegated to a non-administrator. A better approach is to assign the right to a group, either an existing one (such as Backup Operators) or, ideally, a custom group created specifically for this purpose. This keeps to the Principle of Least Privilege: a user should be given only the privileges necessary to accomplish their task.

If you are developing a new securable object, define access masks for each operation (or class of operations) so that the system administrators can delegate operations to the right people without having to make them administrators. And, of course, remember that *an* administrator is not necessarily *the* Administrator.