OWNER ACCOUNT

PERMISSIONS - UID & GID

IN LEOPARD & SNOW LEOPARD

Owner Account-Permissions-UID & GID in Leopard & Snow Leopard

After my comment "Permissions are so interesting … and confusing at the same time"
I have put a few details down for anyone interested in this subject.
There are numerous articles & books explaining Permissions in Leopard & Snow Leopard.
Most of my article below I have learn't from books I own.
'Take Control Of Permissions in Snow Leopard','Take Control of Permissions in Leopard' &
'Mac OS X Support Essentials v10.6".

WARNING: Take extreme care with using Terminal Utility especially executing Root commands:

Root is the most powerful account on the system. A mis-executed recursive chown command as root can have disastrous consequences.

Always be absolutely certain you know the effects of any command you intend to execute as root.

Every item on your computer is owned by an account and carries a set of permissions.

These permissions control the access that each of three classes—owner, group, and other—has to an item.

Here is a quick explanation of what I mean by owner, group, and other:

Owner: The owner is the user account that owns an item, such as a file, folder, or disk.
Every item is owned by an account. (Traditionally in Unix, this is known as the user class, and Unix commands abbreviate it with a u.)

Group: In addition to being owned by a user account, every item is also owned by a group.
A group is a set of user accounts conceptually clumped together so permissions can apply to its members collectively.
Mac OS X provides a number of default groups, and you can create additional groups.

Other: Everyone else! Other refers to all user accounts on the system other than the owner and members of the group.
You will see this type referred to as "others" (in the Finder's Info window) and "world" (by other tools).

Permissions for an item say whether owner, group, and other have these three permissions,
Read: View the contents of the item.
Write: Change the item.
Execute: Execute the item.

System accounts and groups:

In OS X you have the standard user accounts and groups associated with those accounts, but there are numerous special users and hidden groups that the system uses to control access for services and system files.

Terminal Command: id

To reveal your account's UID and primary GID, type id in a Terminal window (and then press Return).   The output of the id command looks like this:

```
Rons-500GB-HD:~ ronni$ id
uid=501(ronni) gid=20(staff) groups=20(staff),204(_developer),100(_lpoperator),98(_lpadmin),81(_appserveradm),
80(admin),79(_appserverusr),61(localaccounts),12(everyone),401 (com.apple.access_screensharing)
Rons-500GB-HD:~ ronni$
```

Terminal Command: dscl . -list /Users UniqueID

To reveal all accounts's UID

```
Last login: Wed May 19 11:28:07 on console
Rons-500GB-HD:~ ronni$ dscl . -list /Users UniqueID
_amavisd              83
_appowner              87
_appserver            79
_ard                67
_atsserver            97
_calendar             93
_carddav              206
_clamav               82
_coreaudiod            202
_cvmsroot             212
_cvs                72
_cyrus              77
_devdocs              59
_dovecot              214
_dpaudio              215
_eppc               71
_installer            96
_jabber              84
_lda                211
_locationd            205
_lp                26
_mailman              78
_mcxalr              54
_mdnsresponder          65
_mysql               74
_pcastagent            55
_pcastserver           56
_postfix             27
_qtss               76
_sandbox              60
_screensaver           203
_securityagent          92
_serialnumberd          58
_softwareupdate         200
_spotlight            89
_sshd               75
_svn                73
```

(continued on page 3)

(continued from Page 2)

```
_teamsserver           94
_timezone              210
_tokend               91
_trustevaluationagent   208
_unknown               99
_update_sharing        95
_usbmuxd              213
_uucp                4
_windowserver          88
_www                 70
_xgridagent           86
_xgridcontroller       85
daemon                1
Guest                201
nobody                -2
ronni                501
root                0
trouble              504
Rons-500GB-HD:~ ronni$
```

Ownership:

Ownership in Mac OS X, and all flavours of Unix, is based on an account's user identification number (UID) and a group's group identification number (GID), not on the user name and group name you see in the Info window or other folder listings. Those names are for our convenience; the computer cares only about the numeric identifiers.

It's possible to have the same user name for multiple accounts (e.g. one account when you boot from your internal drive, and another account when you boot from your external drive—both named ronni), but end up with a different UID for each.

In this case, when you try to access an item owned by "your other self," you may be unpleasantly surprised to find that you don't own it, even though you're logged in to the account called 'ronni'.

Problems resulting from multiple boot volumes:

If you boot your Macintosh from an external drive instead of from the internal drive more than occasionally, you may run into ownership issues because the UIDs used on one volume may not correspond to the same accounts that are used on the other volume. So, if you created a file when your account's UID was 501, you will not own that file when you boot from the other drive if your account's UID on that other drive is 502.

Two kinds of problems can result from UIDs and multiple bootable volumes:

1. Someone else can reboot your computer from an external drive and take ownership of your files if she controls an account on the external drive whose UID matches yours.

2. You can have an account on each of the two boot volumes, but not have access to your own files at times because your own UIDs do not match.

The best way to avoid UID problems in multi-boot environments is to create accounts on your computers in the same order so each account always has the same UID.

Problems resulting from copying:

One weakness becomes apparent when you have accounts on multiple computers. For example, let's say I have two computers, and on each I have an account called ronni.

If the UIDs of the two ronni accounts are the same, 501 for instance, then items I own on one computer and items I own on the other are owned by the same entity: UID 501. When I copy files between the two computers, regardless of method, it's unlikely I'll run into ownership problems because the numbers match on both sides.

But let's say the UIDs do not match. My UID is 501 on one computer and 502 on the other. Now, the stage is set for ownership weirdness. For example, if I move files from the computer where my UID is 501 to the other in such a way that ownership is preserved, not reset, then the files on the receiving side will be owned by UID 501, and—on the other computer—501 is not ronni!

If the Mac doesn't happen to have an account with UID 501, the ls command will simply show 501 instead of a user name. Worse yet, if there does happen to be an account with UID 501, then that account will own the files. In other words, if the UID 501 account is snowy, snowy is now the owner.

Whether UID 501 corresponds to an account or not, to change the ownership to your account you must use administrative privilege.

The best way to avoid these problems is to create accounts on your computers in the same order so each account always has the same UID. If you need to retroactively change an account's UID, read the following:

Change an account's UID:

Before Mac OS X 10.5 Leopard, you would use an application called NetInfo to change UIDs.
Apple removed NetInfo from Leopard, but in Leopard and 10.6 Snow Leopard you can change a UID using the simpler Accounts preference pane or the more powerful dscl command.
I provide the steps for each option next.

But, first back up and log in properly! Before you change UIDs, make sure you have a backup Administrator account to fix problems that may arise should you make a mistake while changing a UID. It's also wise to perform this procedure while logged in to an account other than the one you're changing.

Be aware that if you change an account's UID, you will have to change the ownership of all files owned by that account.

To change a UID in the Accounts preference pane:

1. Open the Accounts System Preferences pane.

2. Click the lock icon at the lower left. You will be prompted for an administrator's password, which you should enter.

3. Control-click the account you'd like to change, and choose Advanced Options.

4. Change the numeric UID in the User ID field.

5. Click OK.

To change a UID in Leopard or Snow Leopard using dscl:

(Assuming your account short name is ronni.)

1. Launch Terminal.

2. Enter: sudo su -

3. Enter your password when prompted.

4. Enter: dscl localhost
You're now using dscl's interactive mode: dscl replaces your shell prompt with >.

5. Enter: cd /Local/Default/Users
The dscl prompt will now be /Local/Default/Users >.

6. Enter: cat ronni

7. In the output of the cat command—which may be considerable, particularly if the user has a photo—find the line that begins with UniqueID:. It should be very close to the end of the output.
Mentally note the number following the colon.

8. Enter: change ronni UniqueID XXX YYY (where XXX is the old UniqueID and YYY is the new UniqueID)

9. Enter: quit

This exits dscl and returns you to your shell session.

10. Enter: exit

This exits su, so all subsequent commands will be executed from your account, rather than the all-powerful root account.

11. If you're logged in to the account whose UID you've changed, log out of the account, then log back in.

If you inspect the items in that user's home folder, you'll see that the user no longer owns them! Why?
Because UID determines ownership, not user name.
The existing items belong to the same UID they did before you changed the account.

Fortunately, you can use the Unix command chown recursively to change the ownership of the items in the user's home folder, as I explain next.

Recursively change ownership:

Now that you’ve changed the UID on a user account, you’ll want to switch the items in the account to the new UID.

To recursively change the ownership of files in a folder and all subfolders with the Unix chown command, assuming the account in question is ronni:

1. Launch Terminal.

2. Enter: cd /Users

Warning—you are about to execute a command as root!

Root is the most powerful account on the system. A mis-executed recursive chown command as root—as shown in these steps—can have **disastrous consequences.** Always be absolutely certain you know the effects of any command you intend to execute as root.

3. Enter: sudo chown -R ronni ronni/ (When prompted, type your password. If you’ve run sudo recently, you won’t be prompted for your password.)

The first ronni in Step 3 is the user name, and the second ronni is ronni’s home folder (relative to your current working directory: /Users).
Note: This strategy of retroactively re-aligning UIDs becomes increasingly difficult the more users you have.

Research Articles:

‘Tackling file account association and permissions changes in OS X’

‘Take Control of Permissions in Snow Leopard’

‘Take Control of Permissions in Leopard’

‘Resetting home folder permissions

‘Mac OS X Support Essentials v10.6’

Document Collated By Ronni

June 2010