

# Access Control Lists

How to plan, implement, and troubleshoot ACLs

Josh Wisenbaker  
[www.afp548.com](http://www.afp548.com)

# The Problem

- Most other OSes at least offer ACLs for files
- Many other OSes offer service controls with SACLS
- Many other OSes offer directory access controls with DACs

# The Problem

- Mac OS X 10.3- does not do any of this
- Standard POSIX file modes and ownerships
  - Easy to get stuck
- No SACL support from the OS (Yes, yes. The VPN)
- No real DAC support from Apple

# The Solution

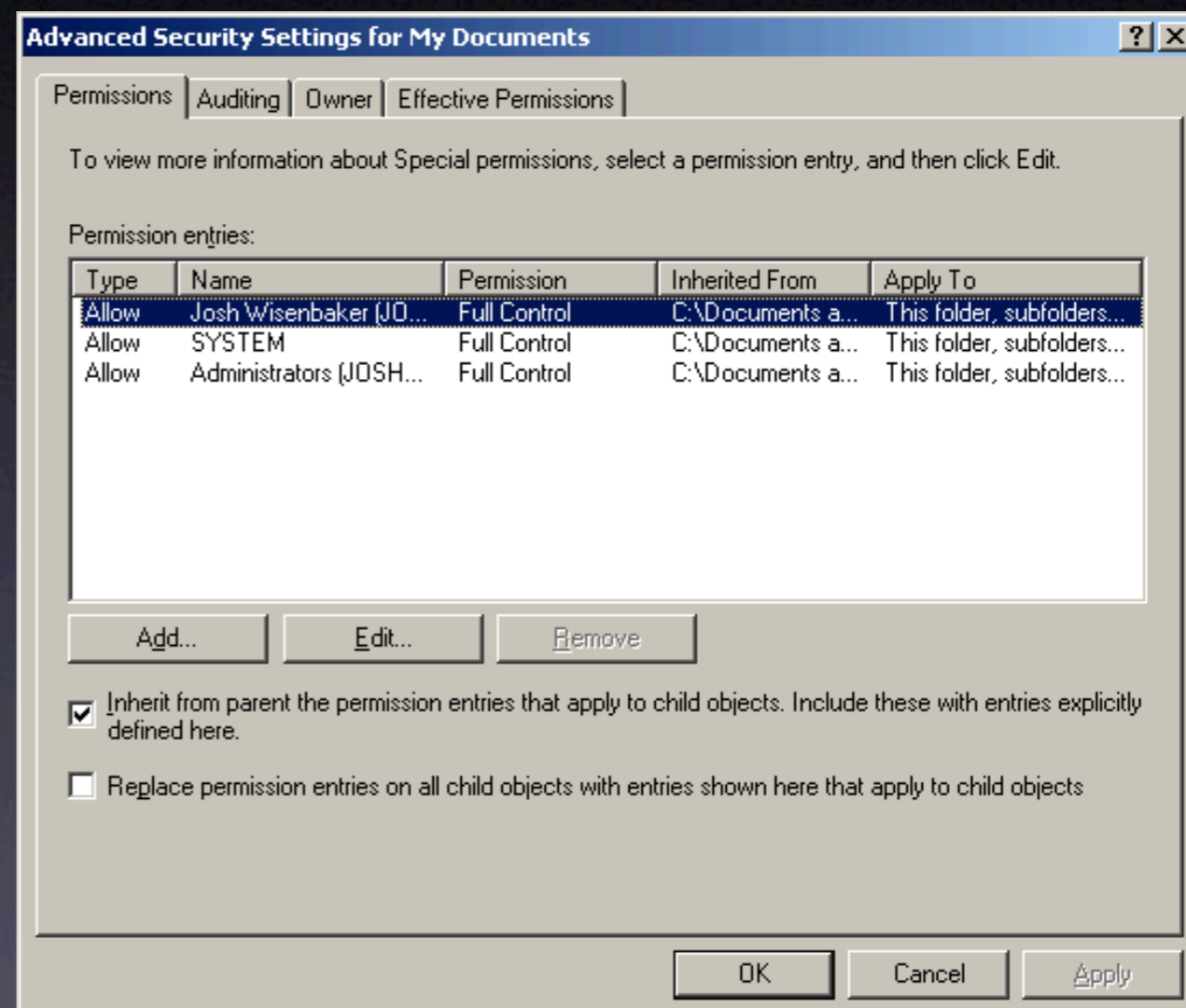
Mac OS X 10.4 Tiger

Apple needed an example  
for file ACLs

They looked to  
Redmond...

# Windows ACLs

- Microsoft has a very nice filesystem ACL model
- Easy presets for common settings






# Windows ACLs

- Microsoft has a very nice filesystem ACL model
- Easy presets for common settings

Access Control List

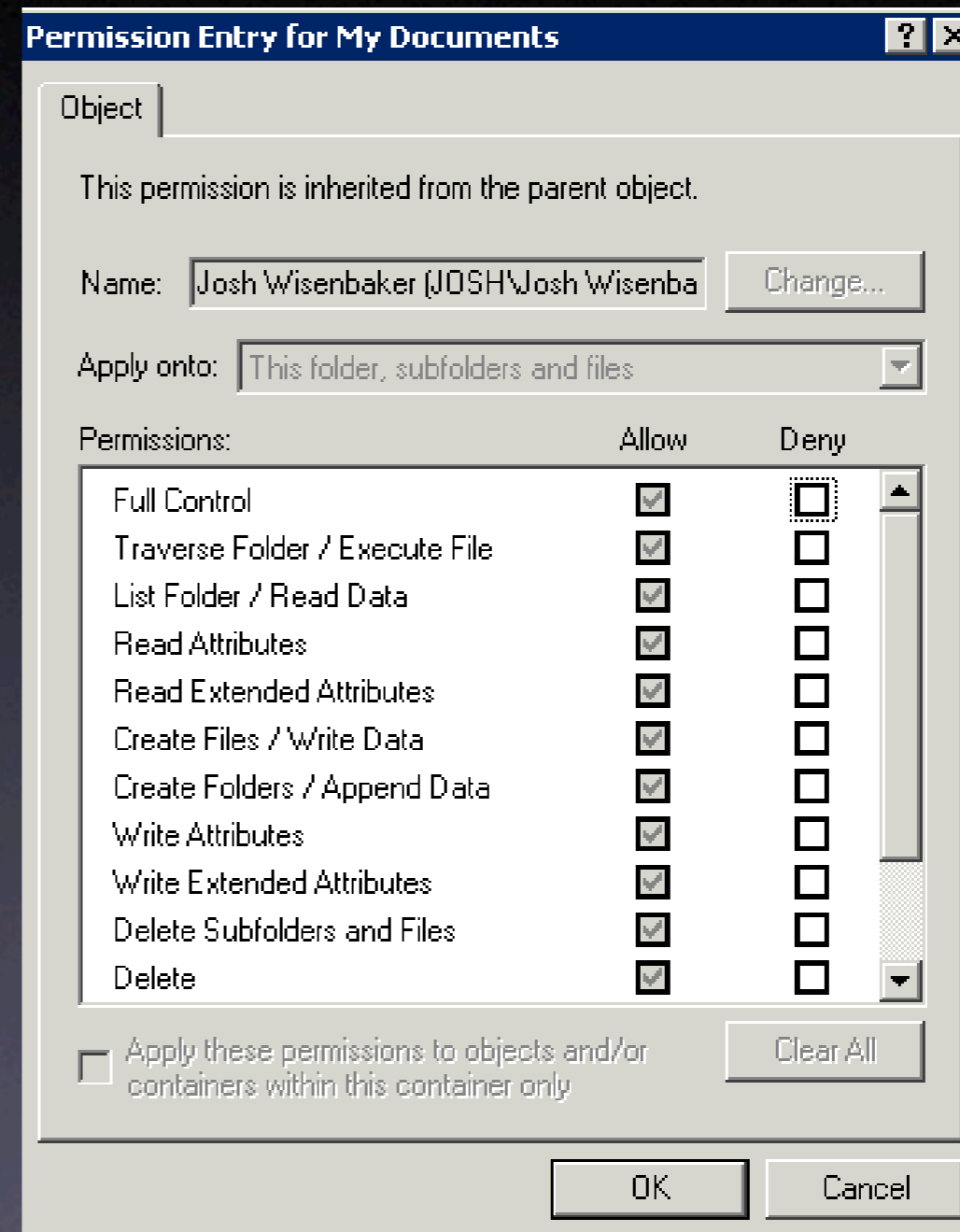
Access control list entries take precedence over the standard permissions listed above.

User or Group	Type	Permission	Inherited	Applies To
 peons	Deny	Full Control	No	This folder and its contents
 Managers	Allow	Full Control	No	This folder and its contents
 World Wide	Allow	Read	No	This folder and its contents



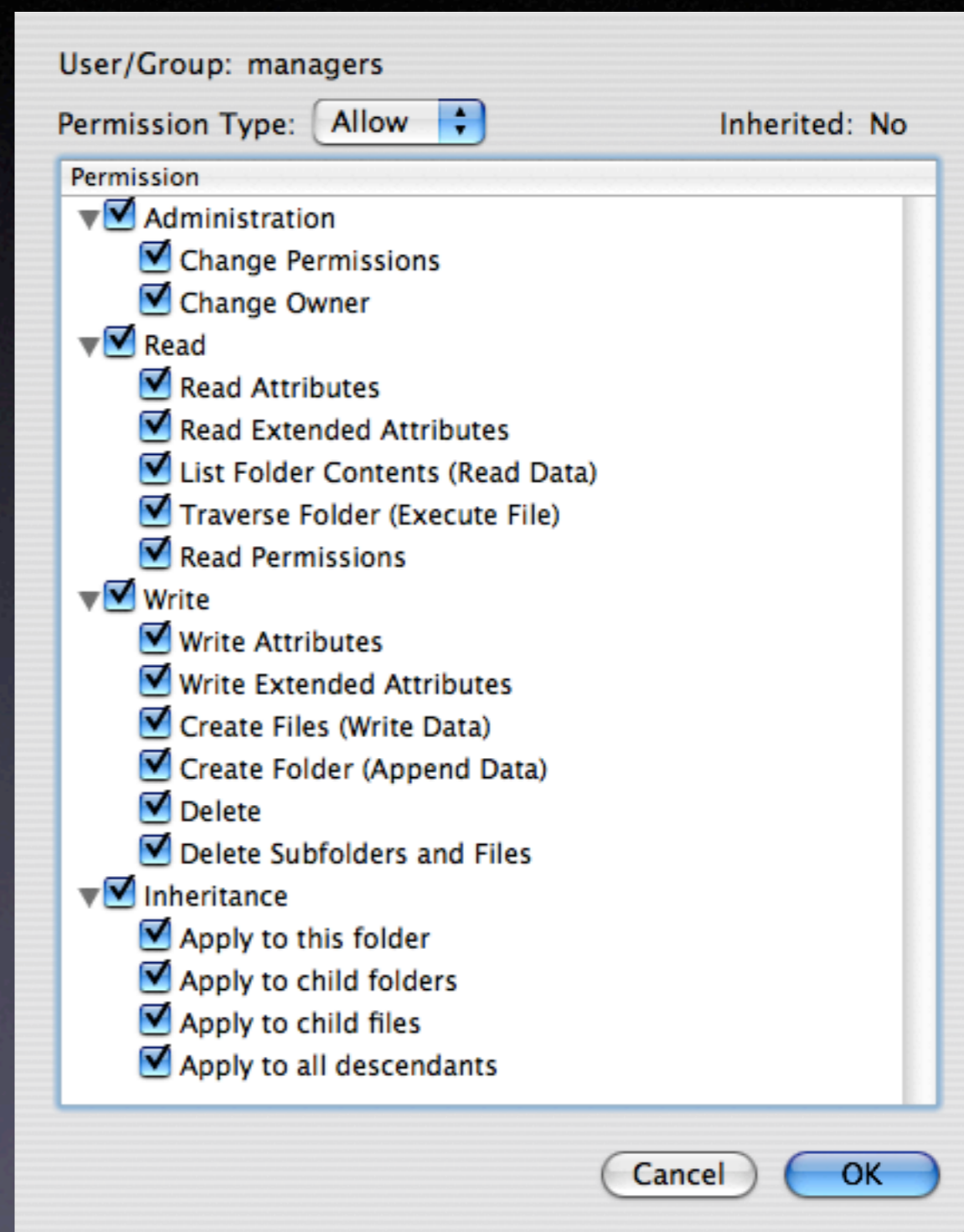
# Windows ACLs

- Advanced editing of individual ACEs



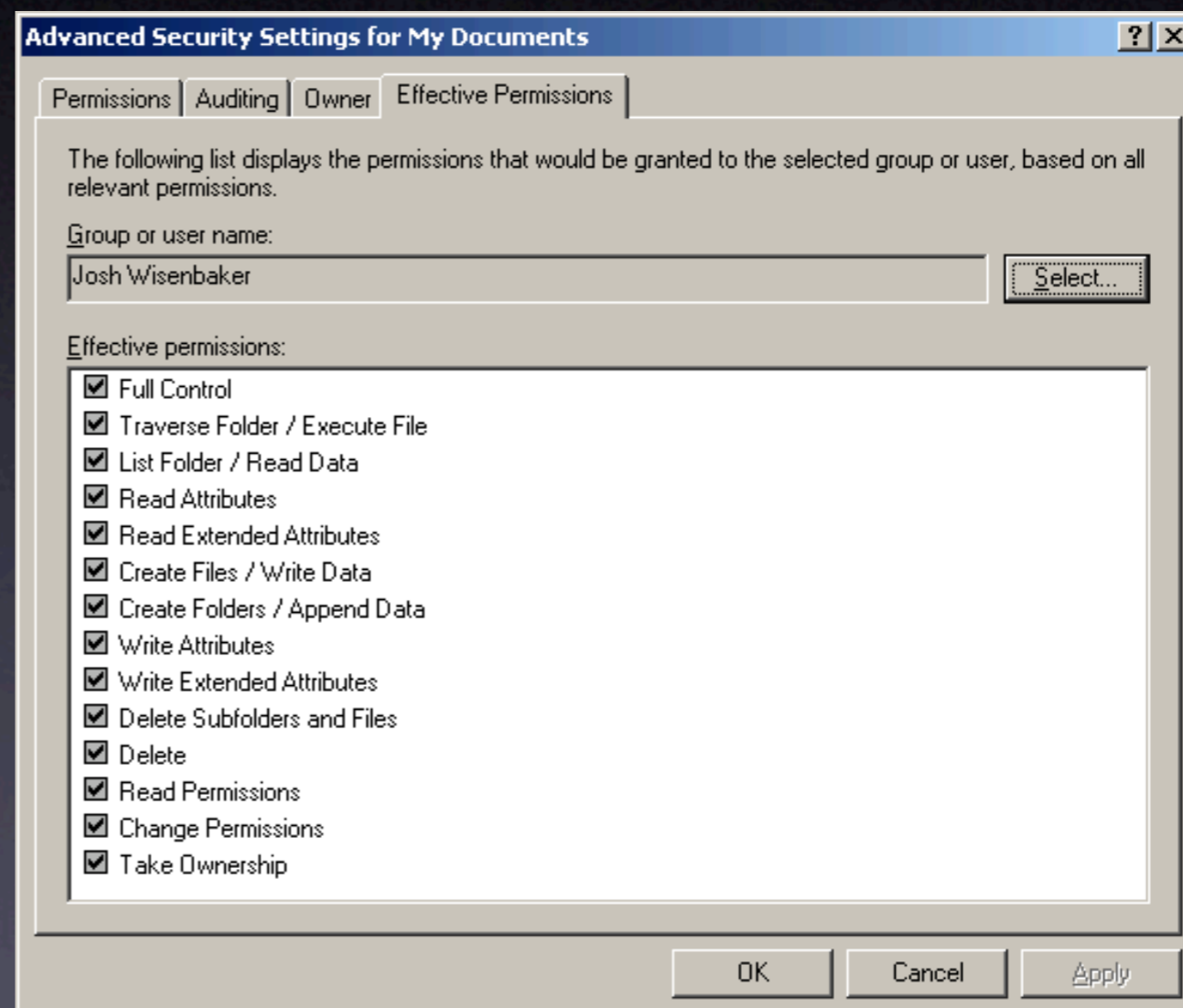
# Windows ACLs

- Advanced editing of individual ACEs



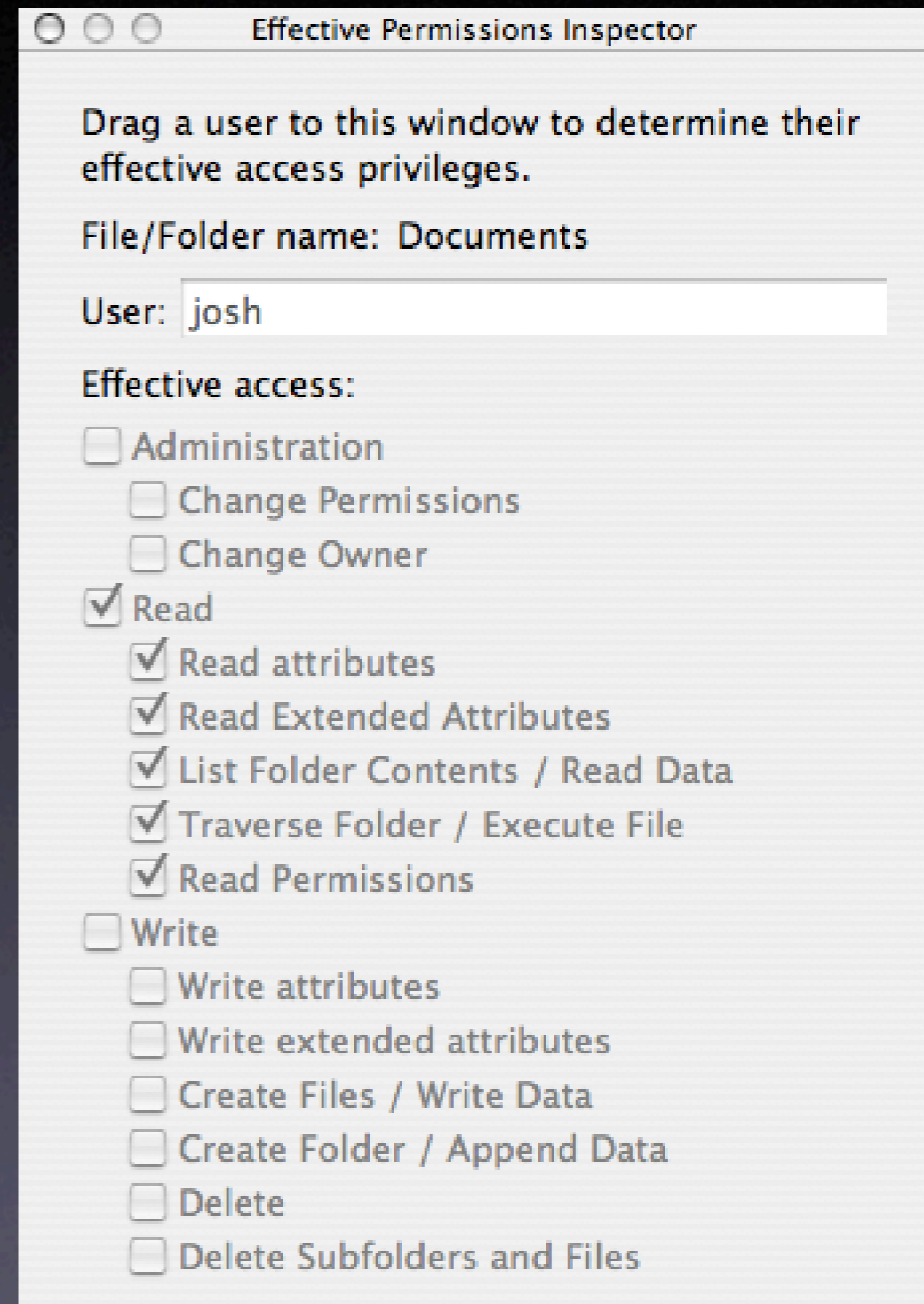
# Windows ACLs

- Effective Permissions tool



# Windows ACLs

- Effective Permissions tool



# Why Windows?

- The Windows file ACLs are very good
- Well known and understood theory
- Mac OS X is now compatible with Windows ACLs

# Basics of ACLs

- ACLs are made of ACEs (Access Control Entries)
- Each ACL has at least one ACE
- Each ACE is tied to the GUID of a user or group

# Basics of ACLs

- ACLs are evaluated from the top down
- POSIX permissions are consulted if no ACE matches
- Each ACL is applied to a directory
- Files pickup ACLs through inheritance

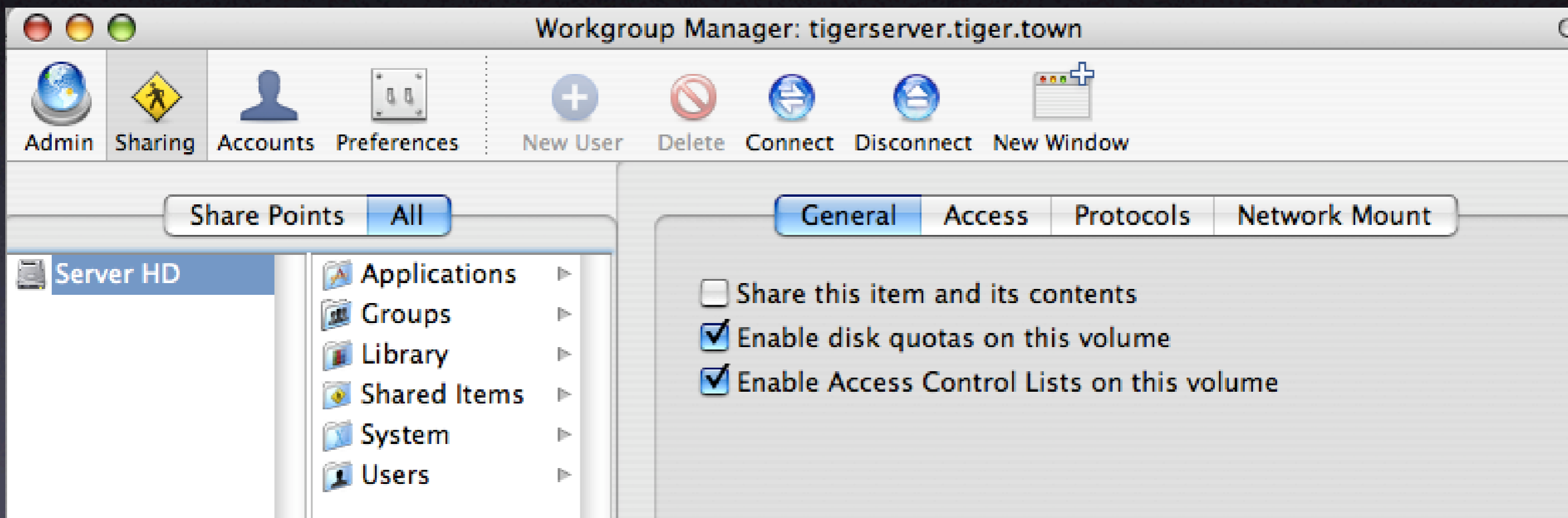
# Basics of ACLs

- ACL data is stored in the filesystem metadata
- So, only HFS+ is supported
- Only AFP and SMB connections support ACLs

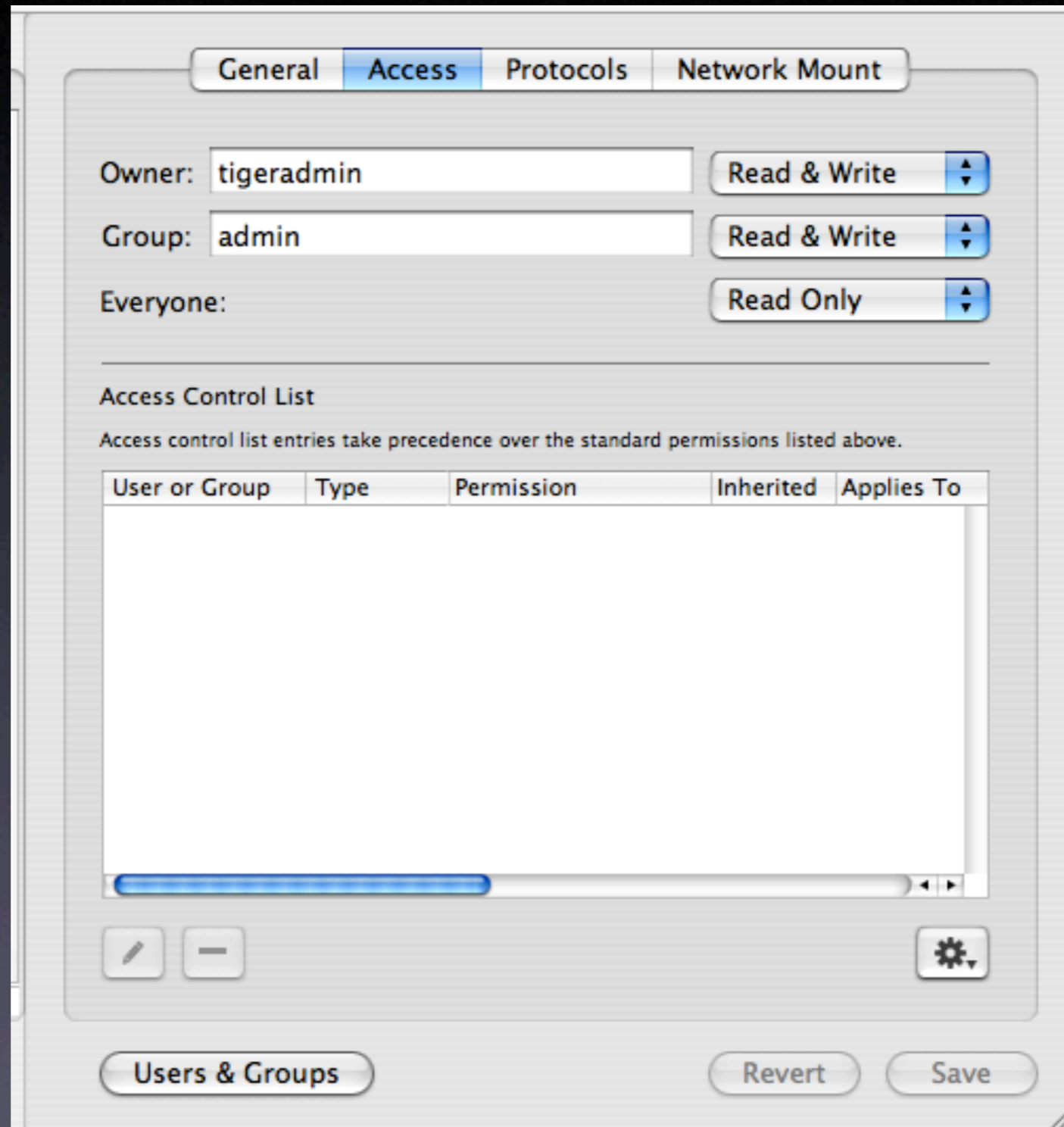


# Mac OS X Server How-To

# First, enable ACLs




# Next, select a share



# Add a user or group

## Access Control List

Access control list entries take precedence over the standard permissions.

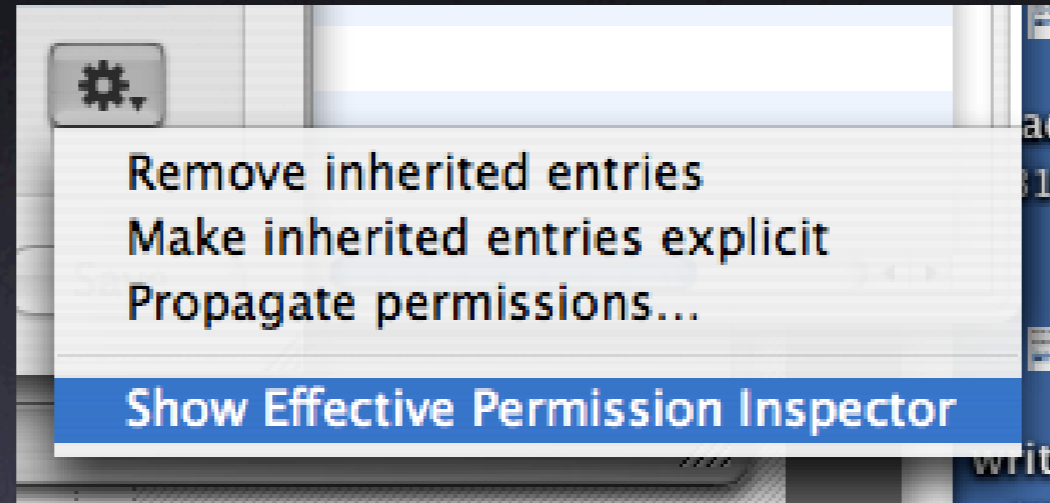
User or Group	Type	Permission	Inher
 managers	Allow	Full Control	No

# Add a user or group

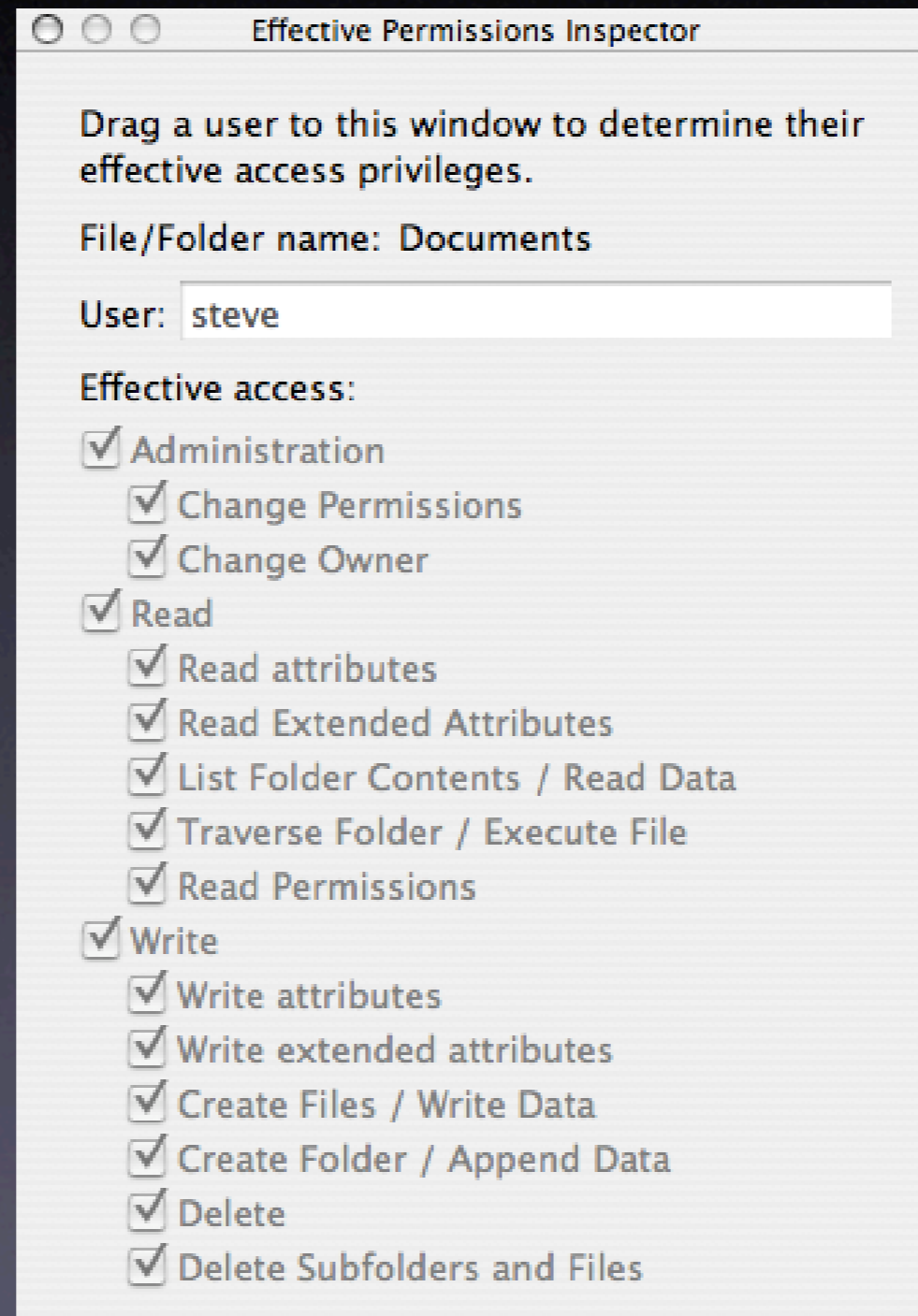
The screenshot shows the 'Access' tab of a file sharing utility. It features three tabs: 'General', 'Access', and 'Network Mount'. The 'Access' tab is active. Below the tabs, there are three rows for permissions: 'Owner: tigeradmin' with 'Read & Write' permissions, 'Group: admin' with 'Read & Write' permissions, and 'Everyone:' with 'Read Only' permissions. Below this is the 'Access Control List' section, which includes a note that 'Access control list entries take precedence over the standard permissions listed above.' and a table with the following data:

Group	Type	Permission	Inherited	Applies To
Managers	Allow	Full Control	No	This folder.

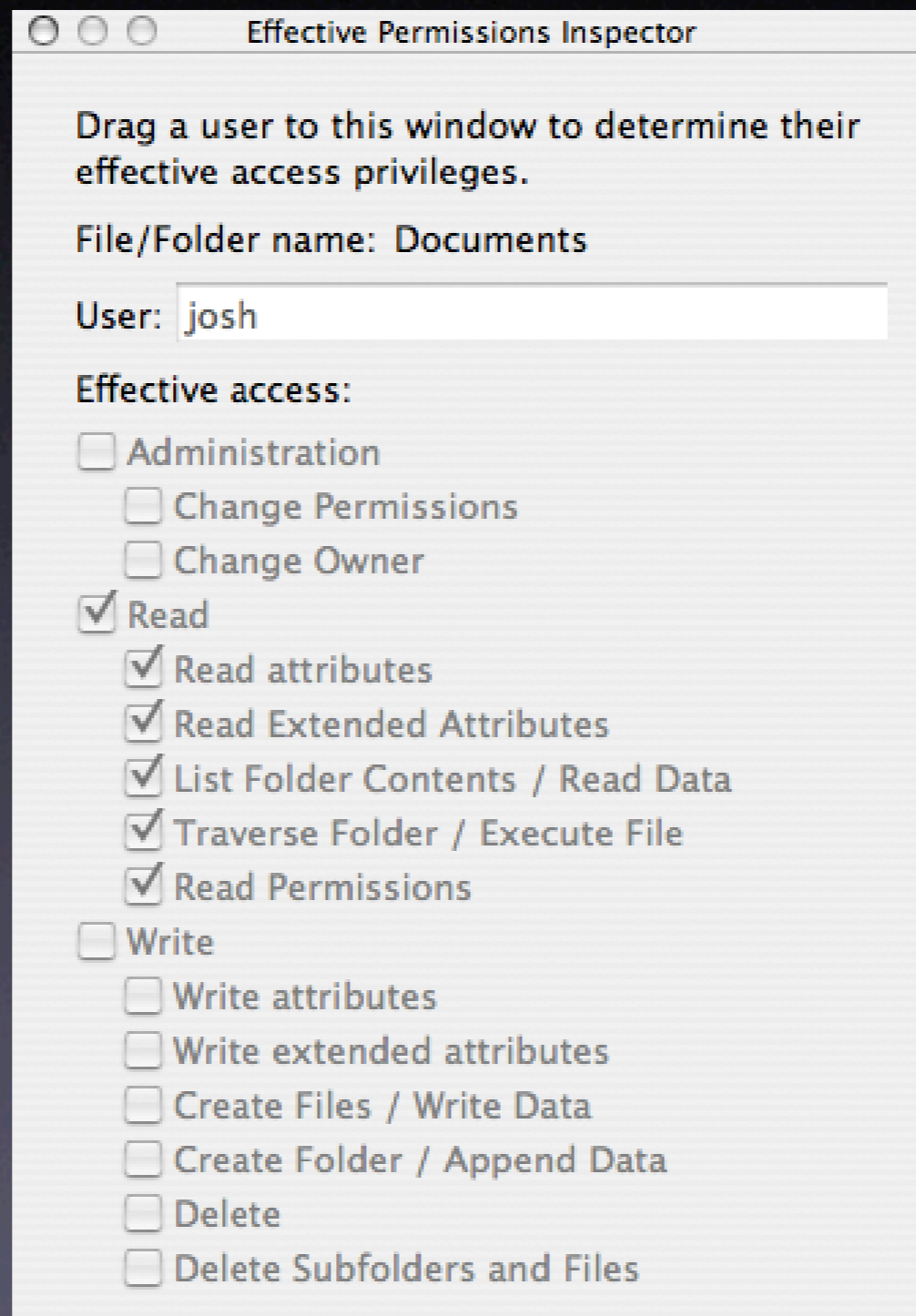
# Check effective permissions



# Check effective permissions



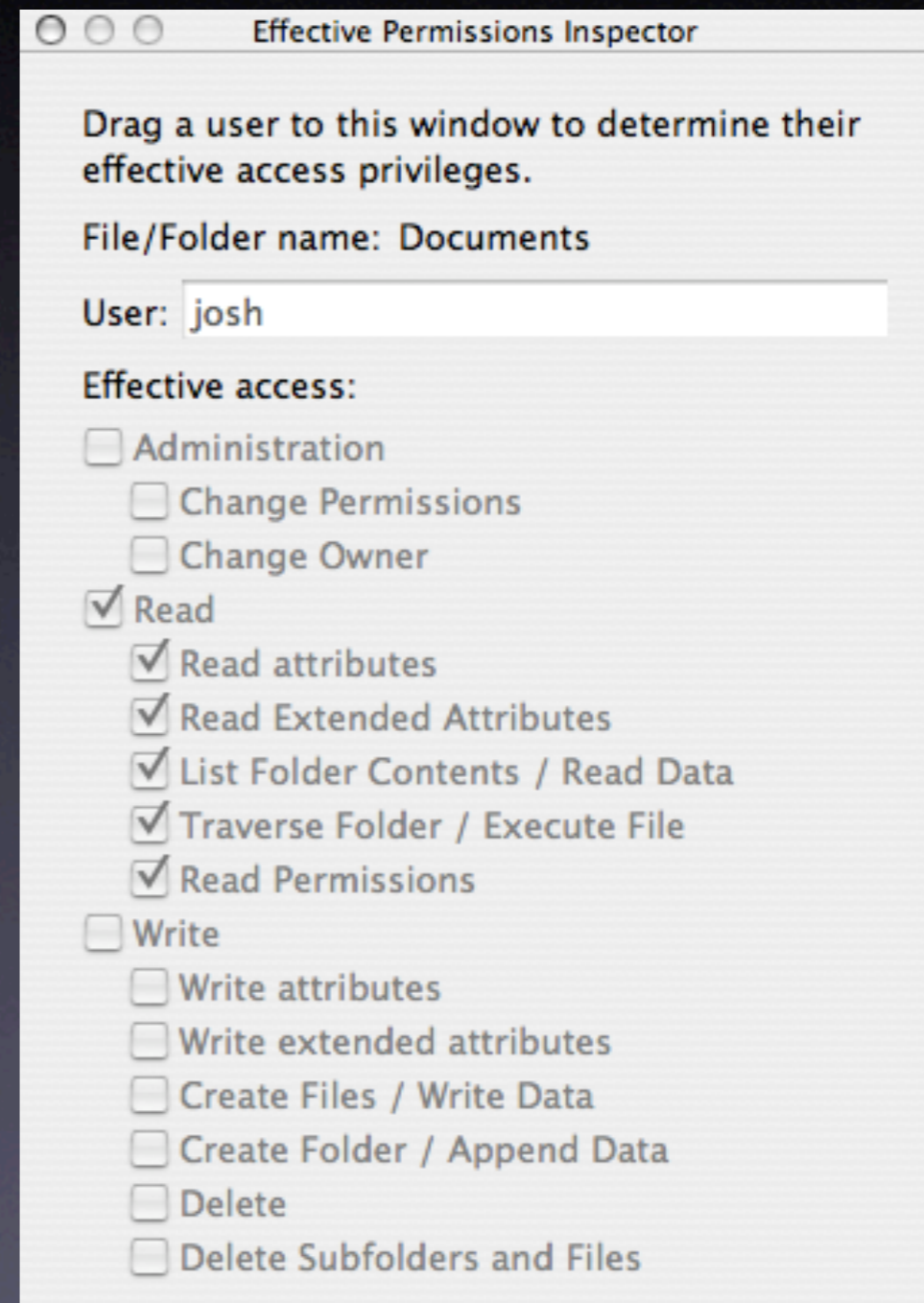
# Check effective permissions





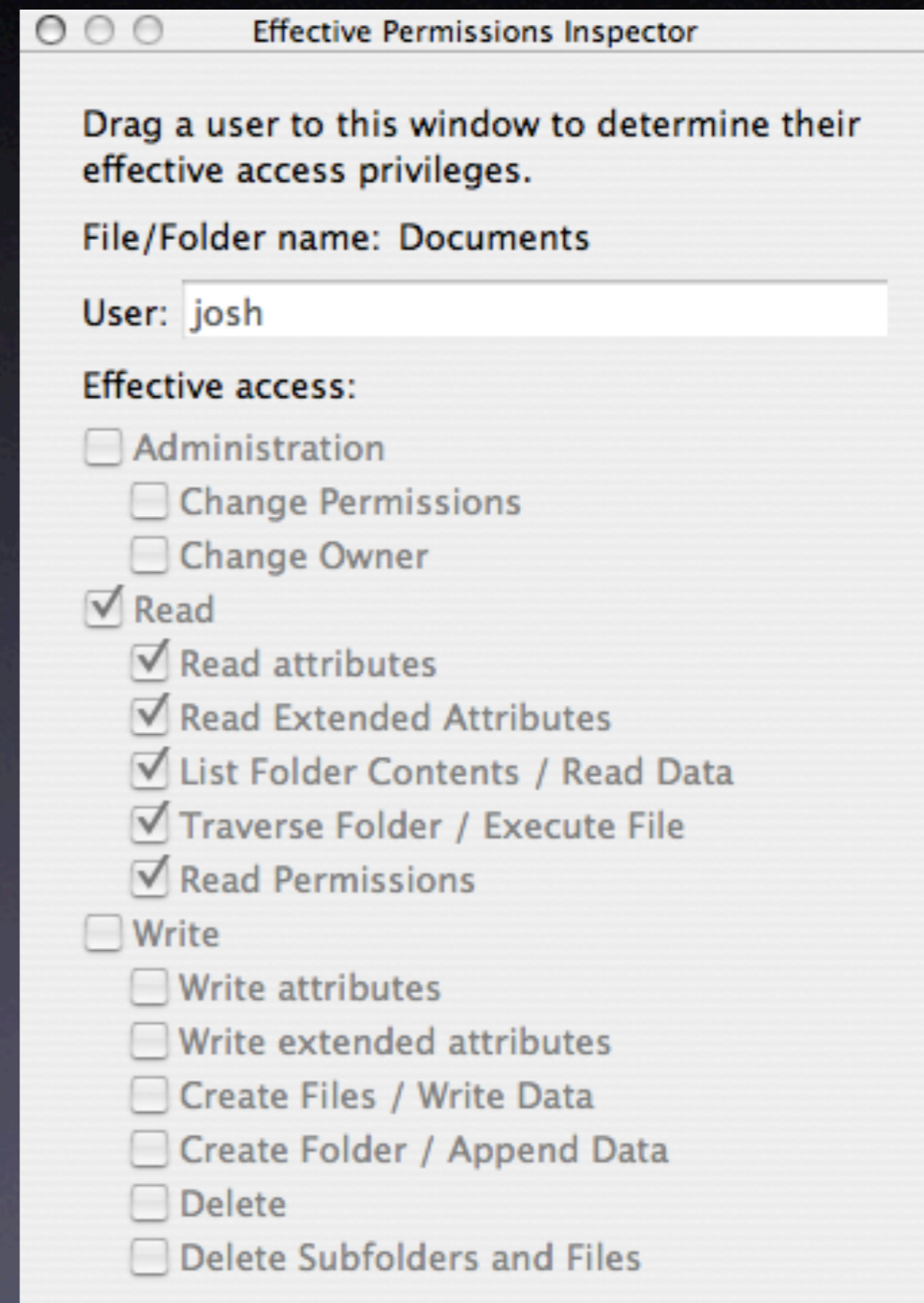
# Check effective permissions

- Josh is not a member of Managers
- Read-only is from POSIX
- Mac OS X evaluates all possibilities and composites them
- If an ACE matches for a user evaluation stops
- Deny ACEs **always** match first!



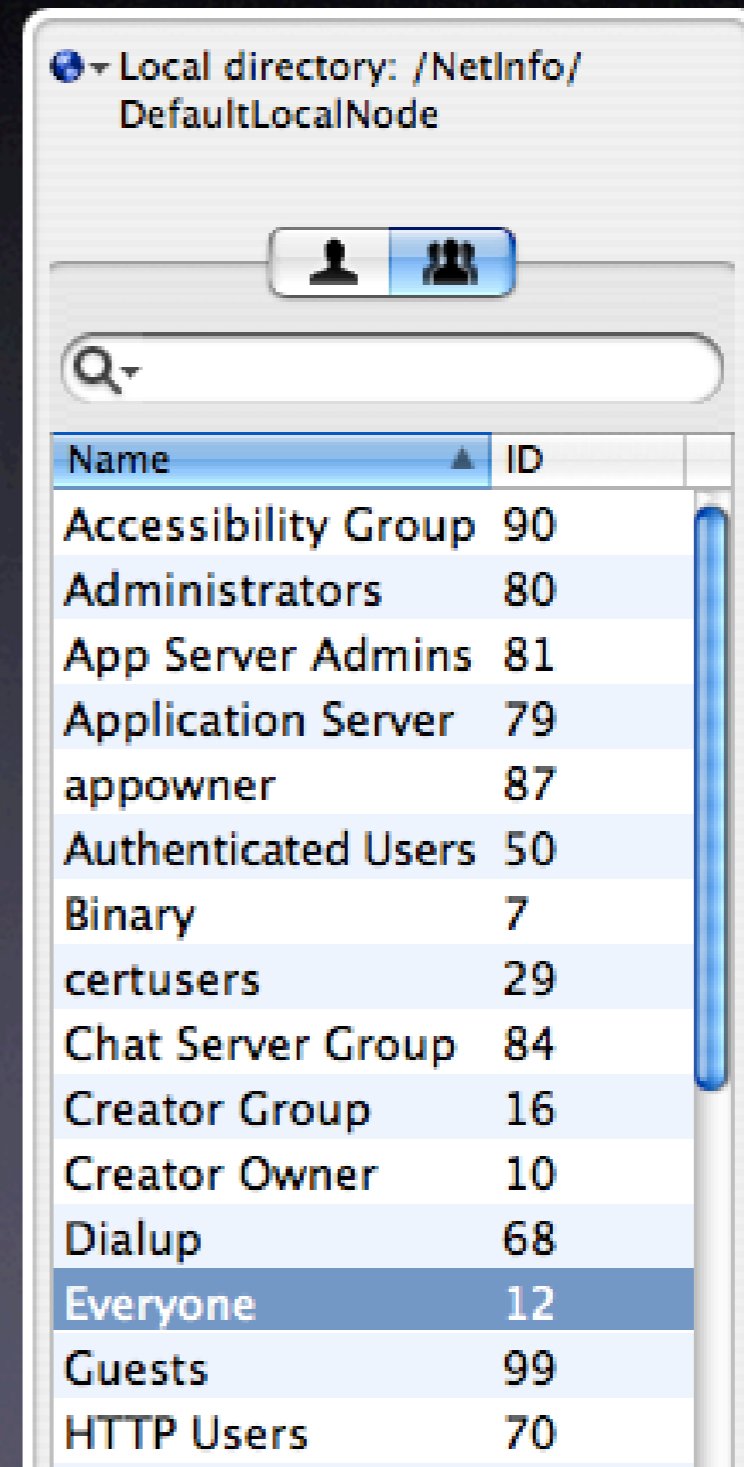
# Check effective permissions

- You could add an ACE for josh to the ACL



# Everyone can play

- You could add an ACE for josh to the ACL
- Simpler in many cases to add the Everyone group



# Everyone can play

Effective Permissions Inspector

Drag a user to this window to determine their effective access privileges.

File/Folder name: Documents

User: josh

Effective access:

- Administration
  - Change Permissions
  - Change Owner
- Read
  - Read attributes
  - Read Extended Attributes
  - List Folder Contents / Read Data
  - Traverse Folder / Execute File
  - Read Permissions
- Write
  - Write attributes
  - Write extended attributes
  - Create Files / Write Data
  - Create Folder / Append Data
  - Delete
  - Delete Subfolders and Files

Owner: tigeradmin Read & Write

Group: admin Read & Write

Everyone: Read Only

Access Control List

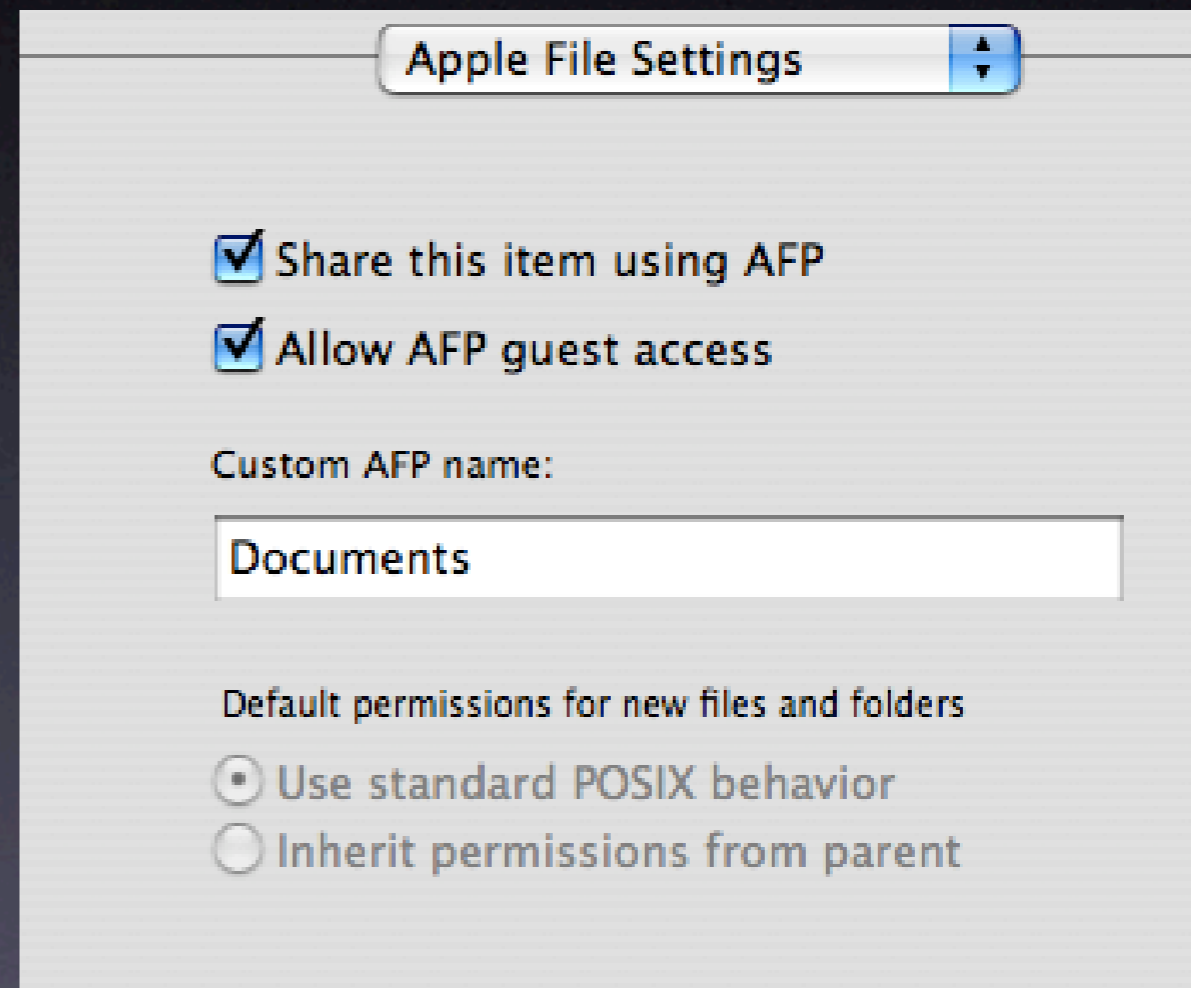
Access control list entries take precedence over the standard permissions listed above.

User or Group	Type	Permission	Inherited	Applies To
Managers	Allow	Full Control	No	This folder
everyone	Allow	Custom	No	This folder

Users & Groups Revert Save

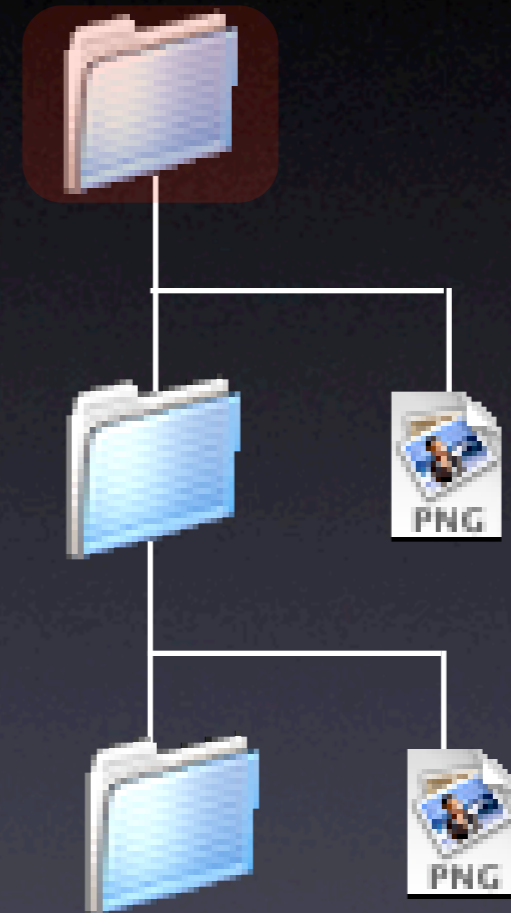
# Everyone can play

- You could add an ACE for josh to the ACL
- Simpler in many cases to add the Everyone group
- Combined with ACL inheritance, this also provides a way around the broken AFP inherit permissions feature



# Four Inheritance Methods

- Apply to this folder



# Four Inheritance Methods

- Apply to this folder
- Apply to child folders



# Four Inheritance Methods

- Apply to this folder
- Apply to child folders
- Apply to child files





# Four Inheritance Methods

- Apply to this folder
- Apply to child folders
- Apply to child files
- Apply to all descendants

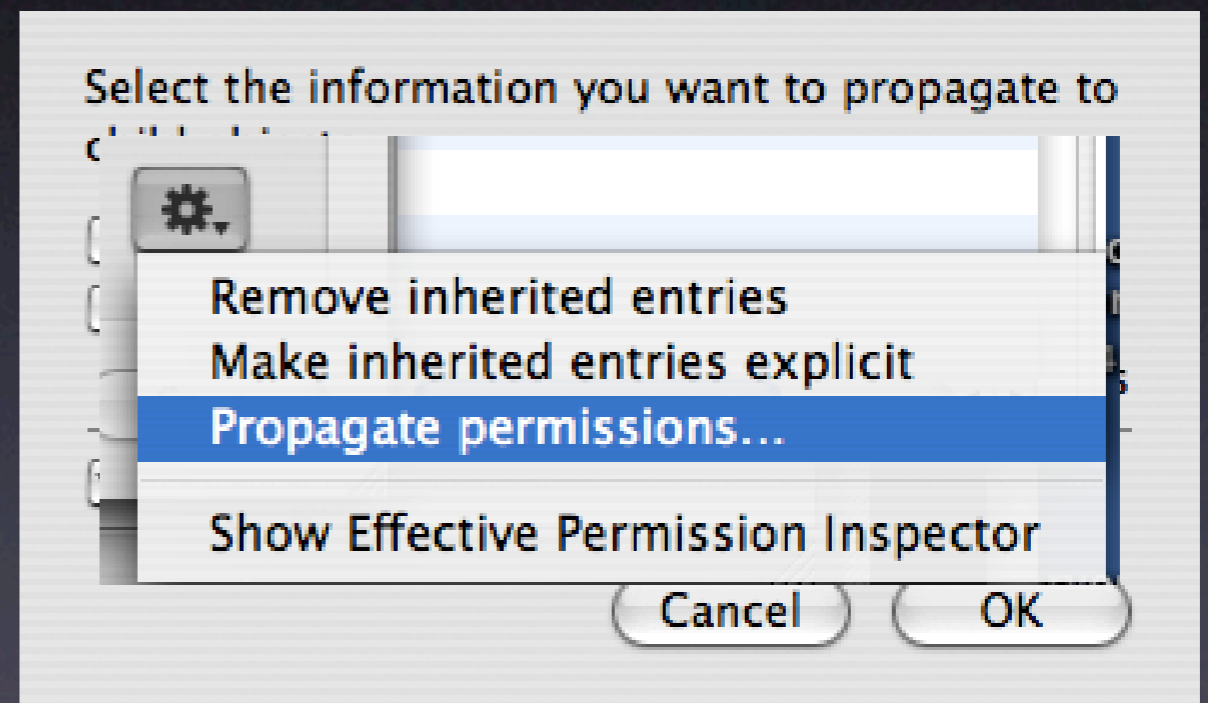


# Four Inheritance Methods

You must combine methods to get your needed results!

# Propagation

- Propagation happens at two specific times
- At file or folder creation time
- When you select Propagate permissions... in Workgroup Manager



# Propagation

The screenshot shows the Windows File Sharing permissions interface. On the left, a tree view shows the share structure: 'Share Points' (selected) and 'All'. Under 'Share Points', there are folders for 'Groups', 'Public', 'Users', and 'Documents'. Under 'All', there are folders for 'Performance' (selected) and 'Projects'. The main pane shows the 'Access' tab for the 'Performance' share. The 'Owner' is 'tigeradmin' with 'Read & Write' permissions. The 'Group' is 'admin' with 'Read Only' permissions. 'Everyone' has 'Read Only' permissions. Below this is the 'Access Control List' section, which is currently empty. A note states: 'Access control list entries take precedence over the standard permissions listed above.'

Share Points All

Groups Performance  
Public Projects  
Users  
Documents

General Access Protocols Network Mount

Owner: tigeradmin Read & Write  
Group: admin Read Only  
Everyone: Read Only

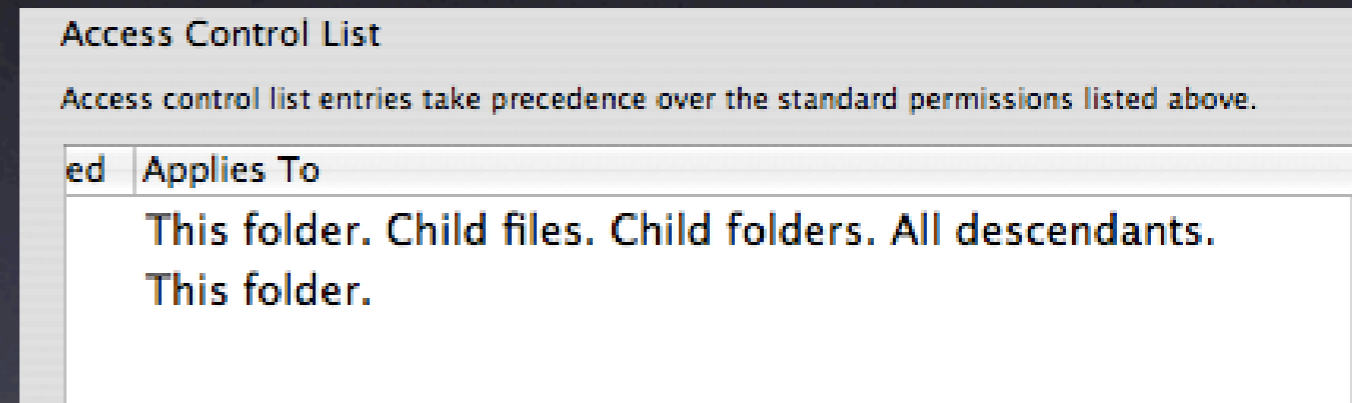
Access Control List

Access control list entries take precedence over the standard permissions listed above.

User or Group	Type	Permission	Inherited	Applies To
---------------	------	------------	-----------	------------

# Inheritance

- Files and folders inherit ACEs from propagation





# Inheritance

- Files and folders inherit ACEs from propagation
- Inherited ACEs are dimmed and noted

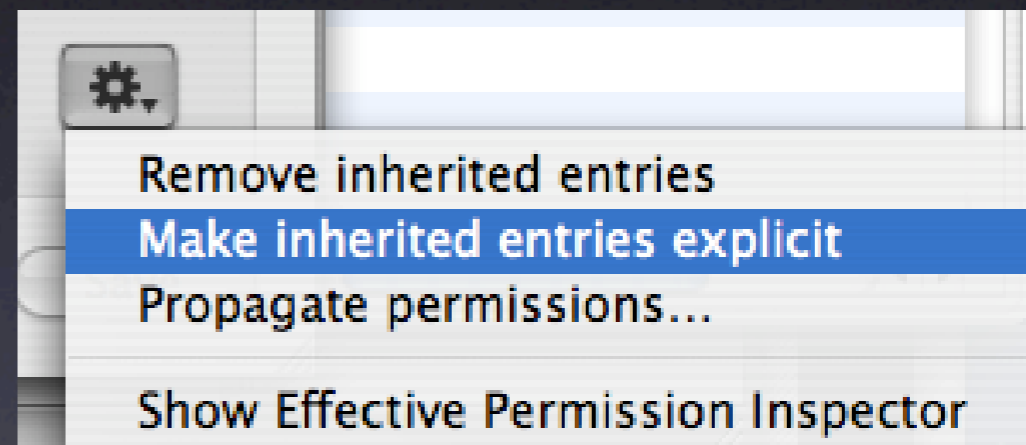
Access Control List

Access control list entries take precedence over the standard permissions listed above.

User or Group	Type	Permission	Inherited	App
 Managers	Allow	Full Control	Yes	This
 Everyone	Allow	Custom	Yes	This

# Inheritance

- Files and folders inherit ACEs from propagation
- Inherited ACEs are dimmed and noted
- You can remove inherited ACEs or make them explicit for editing



# File ACL Tips

- Keep it simple
- Manage at a group level
- Gradually add permissions, consult the EPI often
- Reserve the deny bomb
- Use propagation to your advantage!



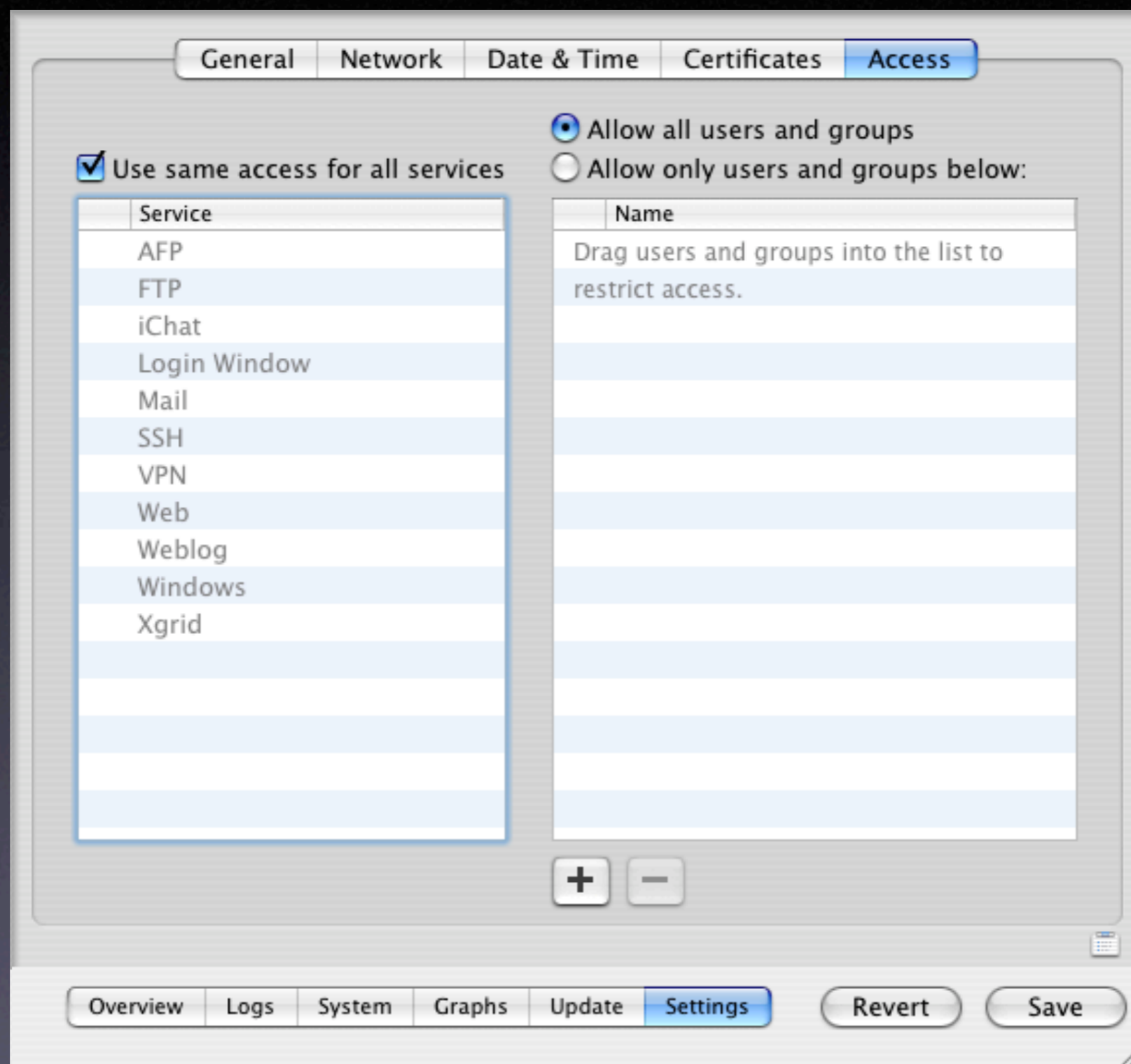
# Service Access Controls

SACLs at last!

# SACLs

- Except for VPN, Mac OS X Server 10.3- had no easy way to restrict services
- 10.4 brings easy SACLs to the party

# SACLs



# Directory Access Control

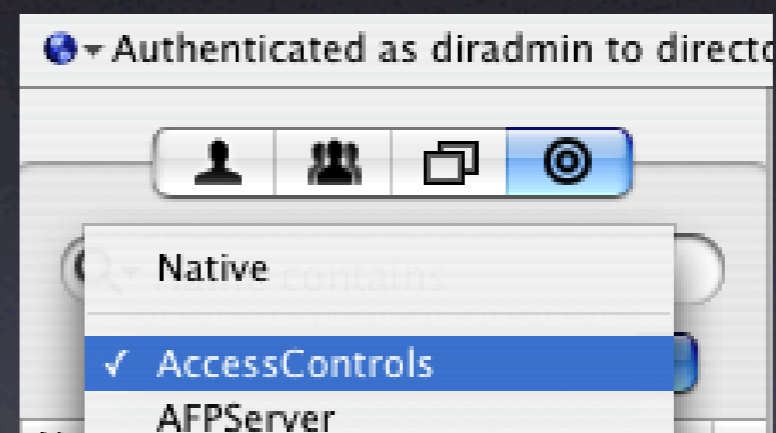
DACs are tricky but powerful

# Directory Access Control

- DACs allow you to restrict access to any part of the LDAP db
- Set using Inspector in WGM
- Uses standard apple-acl records
- Not easy to do without some thought!

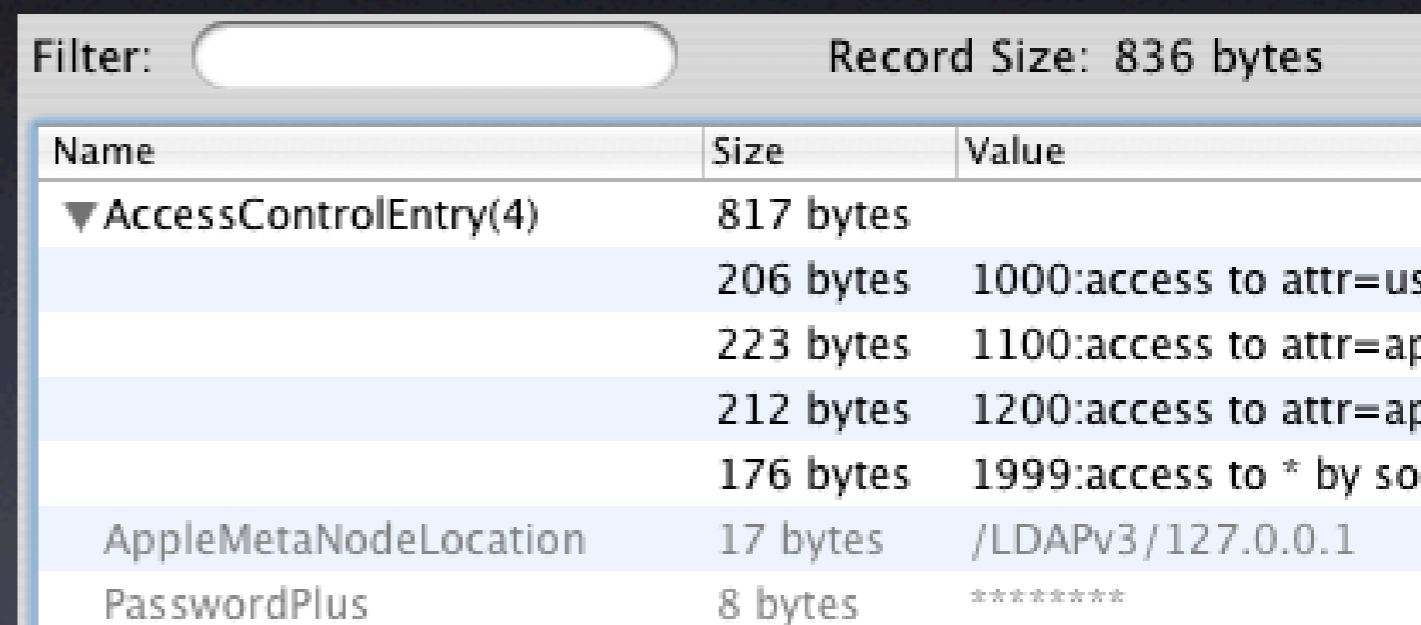
# Directory Access Control

- First, enable the inspector and select the All Records bull's eye
- Next, select AccessControls



# Directory Access Control

- First, enable the inspector and select the All Records bull's eye
- Next, select AccessControls
- Twiddle open the AccessControlEntry attribute



Filter:  Record Size: 836 bytes

Name	Size	Value
▼ AccessControlEntry(4)	817 bytes	
	206 bytes	1000:access to attr=us
	223 bytes	1100:access to attr=ap
	212 bytes	1200:access to attr=ap
	176 bytes	1999:access to * by so
AppleMetaNodeLocation	17 bytes	/LDAPv3/127.0.0.1
PasswordPlus	8 bytes	*****

# Directory Access Control

- First, enable the inspector and select the All Records bull's eye
- Next, select AccessControls
- Twiddle open the AccessControlEntry attribute
- Select a value and click edit or create a new value

Attribute Name: dsAttrTypeStandard:AccessControlEntry

Text:

```
1000:access to attr=userPassword by self write by sockurl="ldapi://%  
2Fldapi" write by group/posixGroup/  
memberUid="cn=admin,cn=groups,dc=dhcp172-21s10n169,dc=09.  
" write by * read|
```

Hexadecimal:

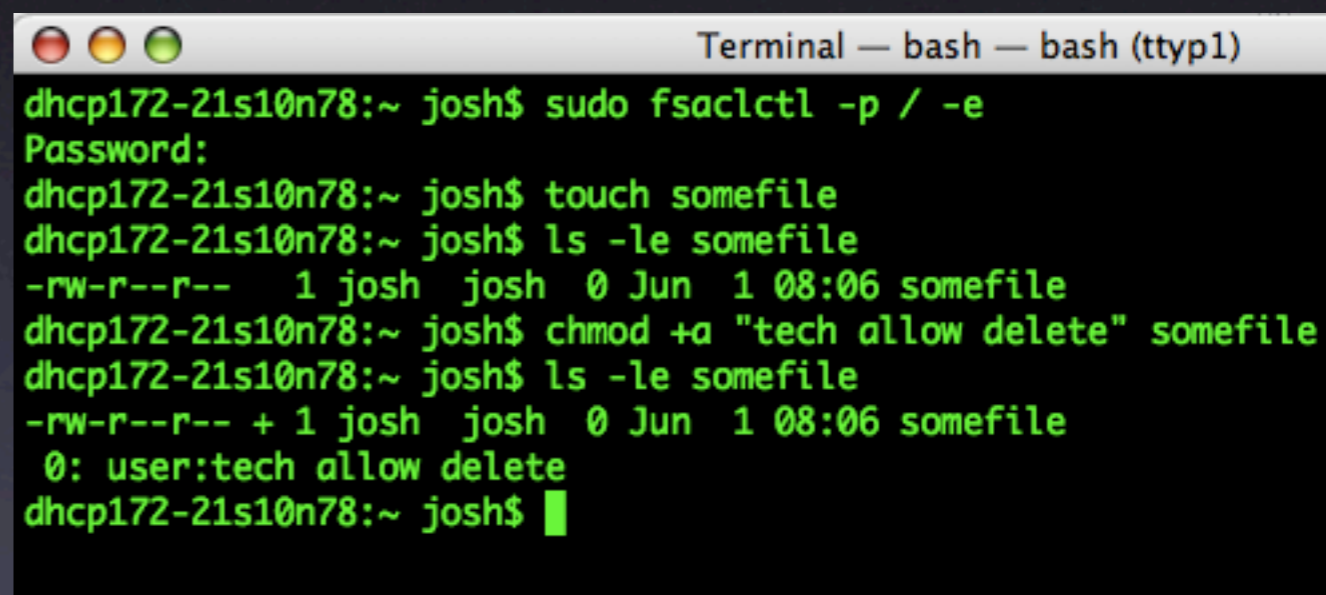
```
31303030 3a616363 65737320 746f2061 7474723d 75736572 5061737  
73656c66 20777269 74652062 7920736f 636b7572 6c3d226c 6461706  
72253246 72756e25 32466c64 61706922 20777269 74652062 7920677  
7847726f 75702f6d 656d6265 72556964 3d22636e 3d61646d 696e2c6  
2c64633d 64686370 3137322d 32317331 306e3136 392c6463 3d30393  
65732c64 633d636f 6d222077 72697465 20627920 2a207265 6164
```

Search in Text



# What about non-servers?

- You can enable ACLs on a volume with the `fsaclctl` command
- Then use `chmod` and `ls` to work with ACLs
- Notice you can apply an ACL directly to a file with `chmod`

A terminal window titled "Terminal — bash — bash (tty1)" showing a series of commands and their outputs. The user 'josh' runs 'sudo fsaclctl -p / -e' to enable ACLs, then 'touch somefile' to create a file. The first 'ls -le somefile' shows permissions '-rw-r--r--'. Then 'chmod +a "tech allow delete" somefile' is run, and the second 'ls -le somefile' shows the updated permissions '-rw-r--r-- + 1 josh josh 0 Jun 1 08:06 somefile' and the ACL entry '0: user:tech allow delete'.

```
Terminal — bash — bash (tty1)
dhcp172-21s10n78:~ josh$ sudo fsaclctl -p / -e
Password:
dhcp172-21s10n78:~ josh$ touch somefile
dhcp172-21s10n78:~ josh$ ls -le somefile
-rw-r--r--  1 josh  josh  0 Jun  1 08:06 somefile
dhcp172-21s10n78:~ josh$ chmod +a "tech allow delete" somefile
dhcp172-21s10n78:~ josh$ ls -le somefile
-rw-r--r-- + 1 josh  josh  0 Jun  1 08:06 somefile
 0: user:tech allow delete
dhcp172-21s10n78:~ josh$ █
```

# Summing up

- Apple now has pervasive ACL support
  - File ACLs allow for granular permissions
  - SACLs allow for easy service restrictions
  - DACs allow for directory control, albeit difficult

Demo