Linux CIFS client support

Steve French
Senior Engineer –
IBM Linux Technology Center
Samba Team

samba opening windows to a wider world

© 2006 IBM Corporation

Outline

- Linux CIFS Client
  - *Why CIFS? A little history*
  - *Why IBM and CIFS and Samba?*
- CIFS Client Components
- CIFS Configuration
- Recent Progress
- CIFS Client Comparison
  - *What about smbfs? Smbclient? NFS?*
- What is missing?
- What is likely to be coming soon?

Note: Opinions expressed are those of the author and not necessarily those of IBM.

© 2006 IBM Corporation

Why CIFS?

- Windows matters in the enterprise! Windows is heavily tied to SMB/CIFS protocol, but what about Linux to Samba using CIFS client …
- CIFS is a very broad, rich protocol
- Existing CIFS servers and clients need fewer changes to achieve functional and performance goals than alternative approaches
- Reasonable performance for certain workloads already, no unnecessary intermediate RPC layer, and straightforward caching model
- Broad support for many platforms including all of most common ones.
- Synergy with large installed base of CIFS clients and servers
- Lots of existing Windows and Samba server knowledge which help make deployment easier than if completely new protocol used

© 2006 IBM Corporation

Need we remind you all …

- CIFS is the defacto standard network filesystem for tens or hundreds of millions of machines (and not just for Windows).
- CIFS clients and servers exist for most or all major platforms
- "*CIFS is a platform-independent file sharing system*" [CIFS: A Common Internet Filesystem, Leach and Perry, 11/96]
- "CIFS is intended to provide an open cross-platform mechanism for client systems to request file and print services from server systems over a network. It is based on the standard Server Message Block (SMB) protocol widely in use by personal computers and workstations running a wide variety of operating systems. " ("A Common Internet File System (CIFS/1.0) Protocol " Leach and Naik 1997)

© 2006 IBM Corporation

30 second History of SMB/CIFS as a standard

- Initial SMB Specification book published by IBM at IBM PC Conference 1984, and also a detailed Netbios “LAN Technical Reference”
- “SMB has been an Open Group (formerly X/Open) standard for PC and Unix interoperability since 1992 (X/Open CAE Specification C209)” [Leach and Perry, ibid], mailslot and other IPC protocols also standardized
- SMBFS for Linux developed
- Leach and Naik, various CIFS IETF Drafts 1997
- CIFS Documentation activity moves to SNIA
- CIFS VFS for Linux included in 2.6 kernel
- SNIA CIFS Technical Reference 3/2002 (include Unix and Mac Extensions)
- 2003: NFSv4 RFC 3530 released (integrates some key ideas from CIFS)
- 2003: “POSIX CIFS Extensions” proposed, and Linux and Samba prototype implementations begun:
  - *2004-2005 POSIX ACLs, POSIX Paths, statfs, very large read/write*
  - *4/2006 POSIX Byte Range Locks in CIFS Version 1.43 (Linux Kernel 2.6.16 includes version 1.42). Additional POSIX CIFS extensions in development*

© 2006 IBM Corporation

Why IBM and CIFS

- IBM (Dr. Barry Feigenbaum) invented original SMB) protocol (1984)
- Over the years IBM has developed more distinct SMB/CIFS server implementations on more platforms than anyone
- We are very active in Samba server development
- SMB/CIFS clients and Samba server are very important to our customers, not just to enable basic enterprise file interoperability but to provide advanced file services
- We want to help move the CIFS protocol forward
- … and help Linux by giving back kernel enhancements in key areas

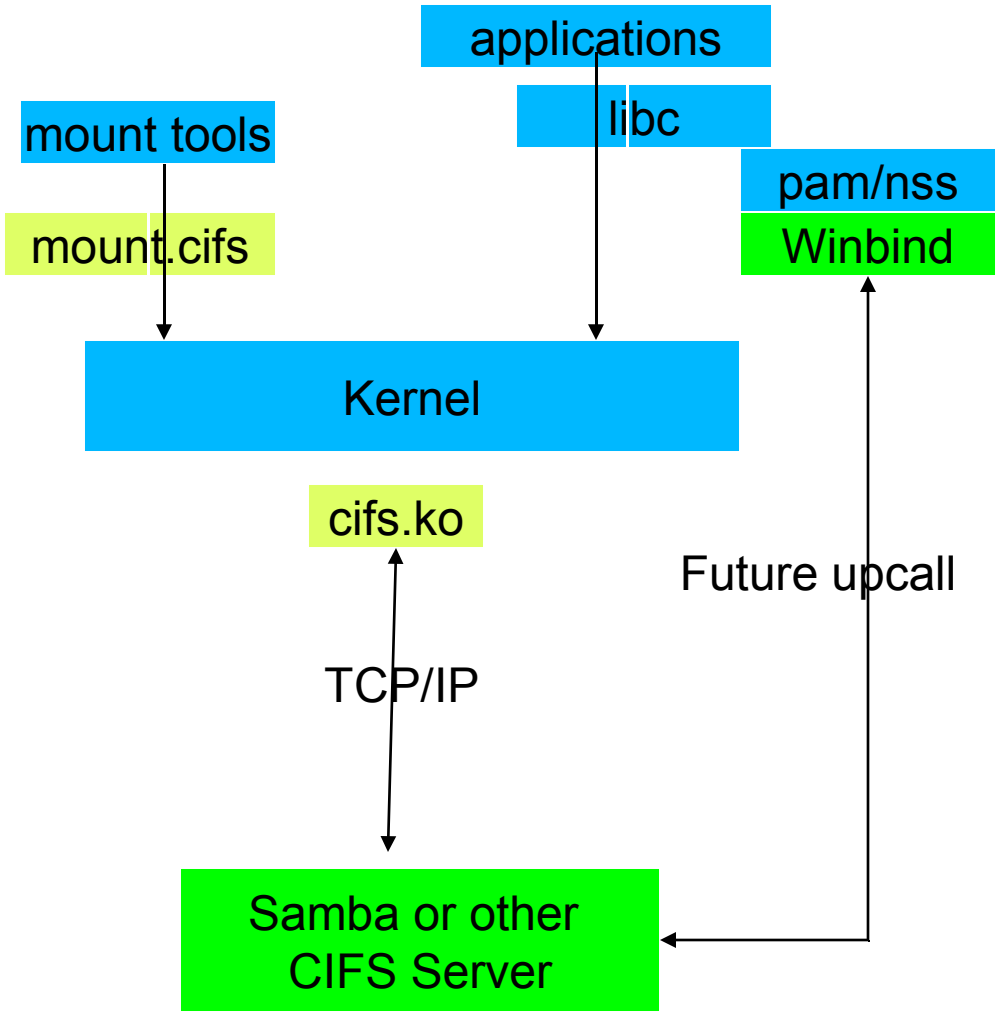
© 2006 IBM Corporation

We worry about two worlds …

- Getting MS Office to run over NET USEs to Samba servers – **perfectly** … but
- We should also worry about getting **perfect (**"better than cluster" fs) semantics (over remote cifs mounts) for OpenOffice, and complex POSIX applications and servers running over CIFS on Linux and certain Unix (not just MacOS).
- And we now worry about these whether the Samba server has local or cluster FS under it

© 2006 IBM Corporation

CIFS and related components



net, smbcacls, smbclient

© 2006 IBM Corporation

CIFS Client Components

- cifs.ko: (kernel code, usually built as module)
- mount.cifs: mount helper that parses a few mount options (uid=name) and converts tcp names to ip addresses before calling kernel mount. Normally called indirectly from mount –t cifs. Kept in Samba client source tree, but not dependent on Samba libraries
- umount.cifs: generally unneeded but can be made setuid for certain configurations
- Various Samba client utilities are useful in conjunction with cifs client
  - *Smbcacls to set Windows ACLs (until CIFS can map POSIX->CIFS ACLs). Not as useful for cifs to Samba servers since both support POSIX ACLs (and setfacl can be used)*
  - *Net command (for misc. remote administration)*
  - *Smbclient to browse servers in domain*
- Pam module for easier cifs automounts also exists

© 2006 IBM Corporation

CIFS Client Configuration

- Parameters can be passed three ways:
  - *Module installation parameters (/sbin/insmod cifs)*
    - MaxBufSize (mostly for faster read and readdir to servers which also are configured with bigger default buffer size)
    - Minimum # of small and large buffers in cifs memory pool
    - cifs_max_pending: Max simultaneous requests can override maxmux
  - *Mount parms (approximately 34 distinct parms understood)*
  - *Runtime changes via /proc/fs/cifs/*
    - Disable, enable or mandate signing, Disable Unix/POSIX Extensions
    - Disable lookup (metadata caching), Disable oplock (caching)
    - Enable MultiuserMount
    - Turn on debugging, Turn on experimental features
- Smb.conf not used by the client (unlike smbfs)

© 2006 IBM Corporation

Security configuration choices

- Optional Disabling acl checks
- MultiuserMounts
- Trusted clients vs. Trusted servers

© 2006 IBM Corporation

Recent Progress

- Focus for last 1 to 2 years has been
  - *Improved stability under stress*
  - *Reduced memory utilization and improved performance (especially in key write behind paths)*
  - *Improving POSIX compliance through*
    - sfu style file mappings approach to Windows
    - Additional POSIX CIFS extensions for Samba
  - *Eliminating remaining "smbfs gaps"*
- And as time permits, extending function

© 2006 IBM Corporation

Recent Progress

- 1.40 through 1.43
  - *POSIX Advisory Locking code improved.*
  - *Improved server interop for non-Samba, non-Windows servers: various bug fixes (and workarounds for server bugs) for issues found during "Connectathon" testing (functional testing against most key modern CIFS servers done)*
  - *Default write size increased to 52K (improved perf)*
  - *Share mode security support*
  - *Reads/Writes over 64K supported (useful for largepage archs)*
- 1.35 through 1.40
  - *Improved large write performance, especially over GigE*
  - *Allow case insensitive mounts*
  - *Improved "mapchars" support and SFU style compensations to Windows srv*
  - *Allow mounts to Windows9x, Windows ME*
  - *Allow local only emulation (no remote) of byte range locking; Allow software suspend*

© 2006 IBM Corporation

SMB/CIFS Client Comparison

(E = excellent, S= Satisfactory, P=Poor, N=none

	Overall Grade	stability	Performance & oplock caching	CIFS ACLs	POSIX ACLs	POSIX compat	Compatibility
Linux CIFS VFS	B-	S	S+	N	Y	S+	Newer servers Win200x, Samba, Windows ME/9x (partial). Best Samba compatibility of all kernel clients.
Samba userspace smbclient	B-	S	P	Y	Y	N/A	Broad compatibility, has krb5
Linux smbfs VFS (obslte)	C-	P	P	N	N	P	No signing support, but has krb5 support
WindowsXP	B+	E	S+	Y	N	S/E*	Broad compatibility for simple ops, but no support for CIFS Unix extensions. Has Krb5 support

© 2006 IBM Corporation

When to Use SMBFS instead of CIFS

- Although we are trying to deprecate smbfs, there are a few cases where cifs can not be used yet (and smbfs or alternatively the user space smbclient is necessary)
  - *Most secure mounts including Krb5 session setup (Samba only)*
  - *NTLMv2 session setup (expected in cifs in mainline by 2.6.18)*
  - *Mounts to old "LANMAN" servers: OS/2 LAN Server, Windows 95, Windows for Workgroups*
- Current userspace smbclient can handle DFS (cifs DFS code is not complete)

© 2006 IBM Corporation

What about NFS?

- NFSv3, dispatching more read requests in parallel, provides faster reads than CIFS in some key cases (especially where network latency is relatively long). NFSv3 skips open/close (sometimes helps perf)
- NFSv3 very stable and provides pretty good POSIX compliance although not as good as NFSv4
- NFS “Not For Security” is no longer strictly the case, but it is easier for many to setup CIFS security features in Samba and/or Windows and CIFS has some additional security features
- NFSv3 provides unsafe caching based on a timer, while CIFS has oplock which helps performance in some key workloads
- NFS write performance also is severely limited by sync window
- NFSv4 is stable enough for some production use in current mainline but performance is often worse (despite addition of safer caching) and supported servers more limited

© 2006 IBM Corporation

Various reasons to use cifs vfs

- Safe caching (oplock support)
- CIFS Packet signing
- Handles high stress scenarios better
- Better performing in most cases
  - *including support for multipage readahead and writebehind*
  - *Better parallelized, can get more requests on wire*
- Better POSIX app conformance (to both Samba and non-Samba servers)
  - *Including POSIX ACLs, hardlinks, symlinks, statfs*
  - *SFU style POSIX file emulation to Windows*
- Implements multiply open files more accurately
- MultiuserMount option
- Byte range locking support
- Forcedirectio (disable client caching)
- Autoreconnection of state after network failure
- More configurable

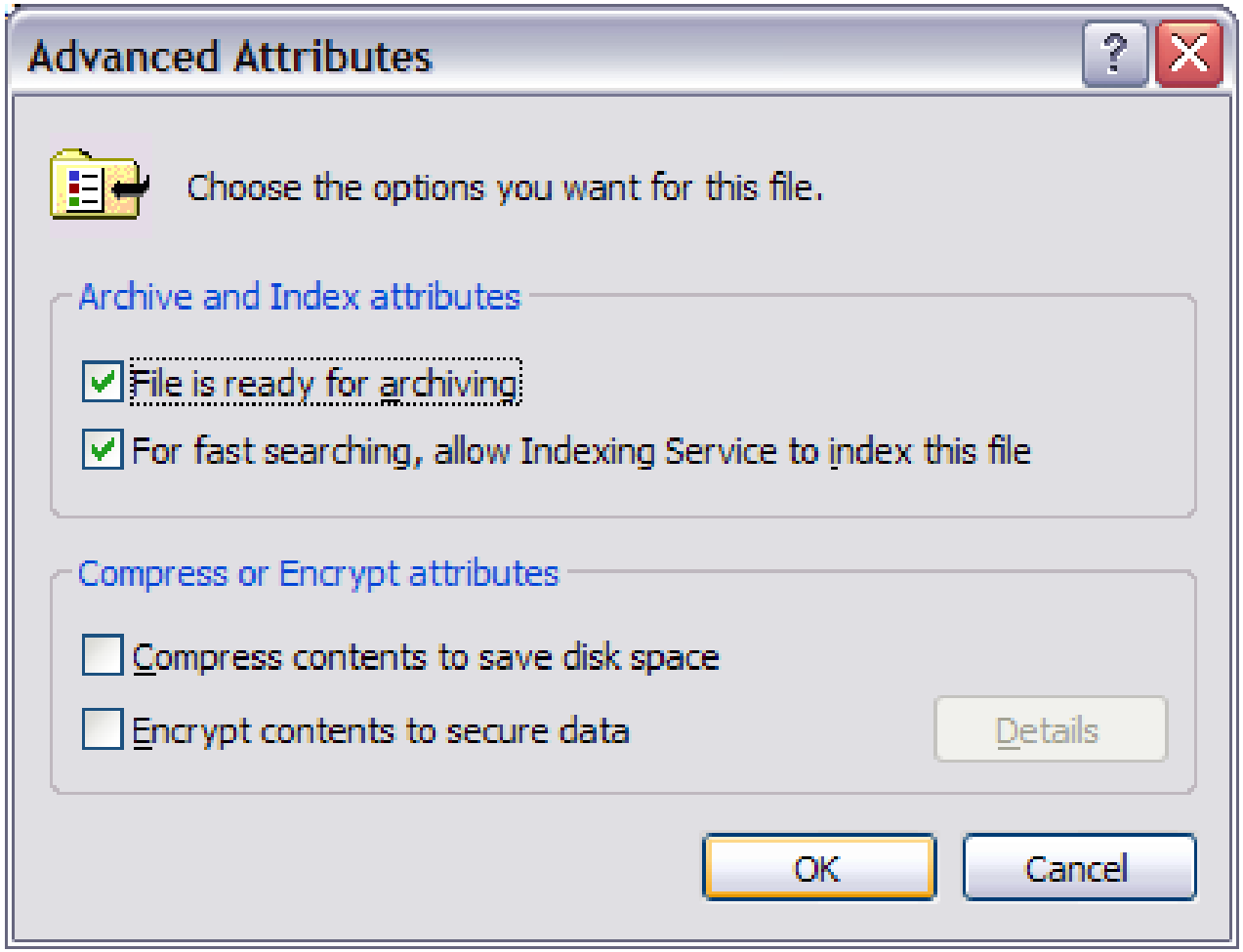
© 2006 IBM Corporation

TODO cifs vfs and general Linux client

- Make it easier to use than Windows in key areas (not just faster, more secure, more transparent, expensible and stable)

© 2006 IBM Corporation

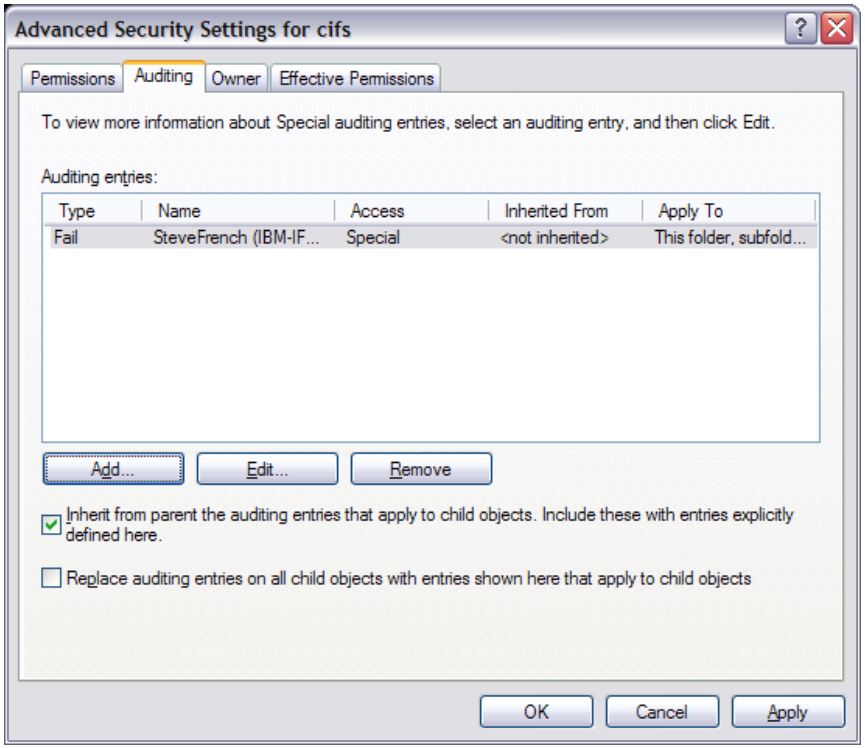
Easy to use Windows Filesystem Encryption/Compression



© 2006 IBM Corporation

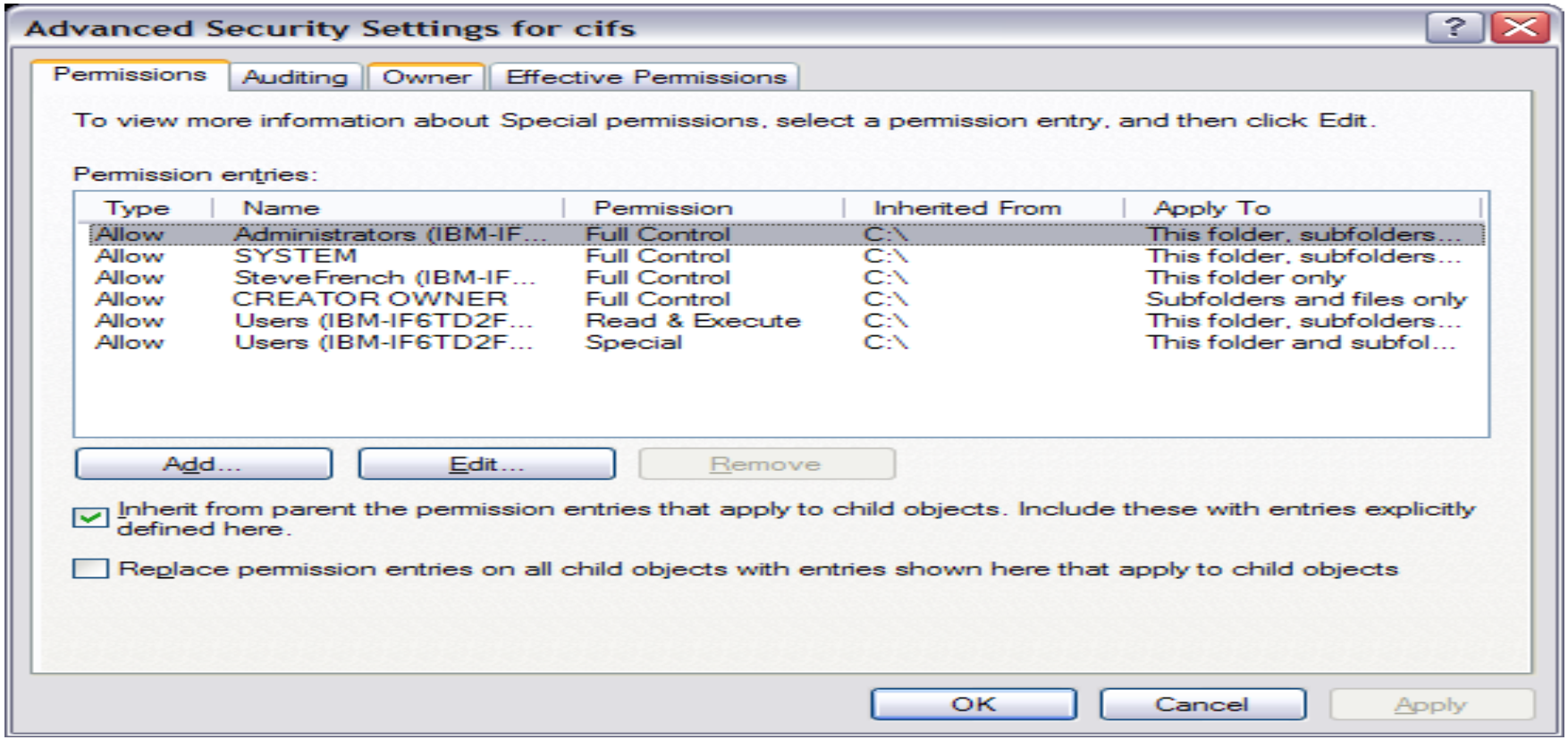
Windows FS Auditing

- Rich Auditing events, nicely integrated with the Access Control tools



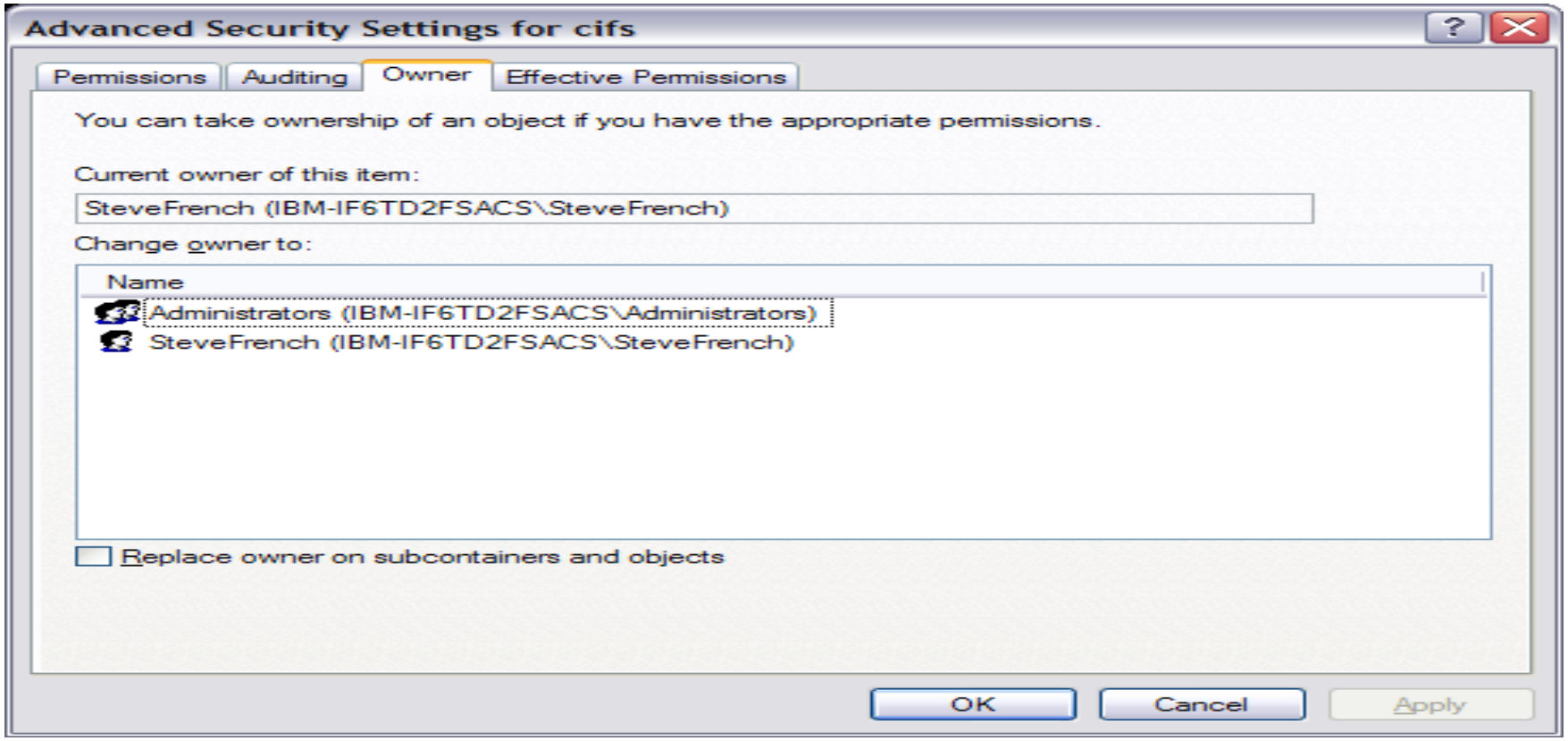
© 2006 IBM Corporation

NTFS permissions, example



© 2006 IBM Corporation

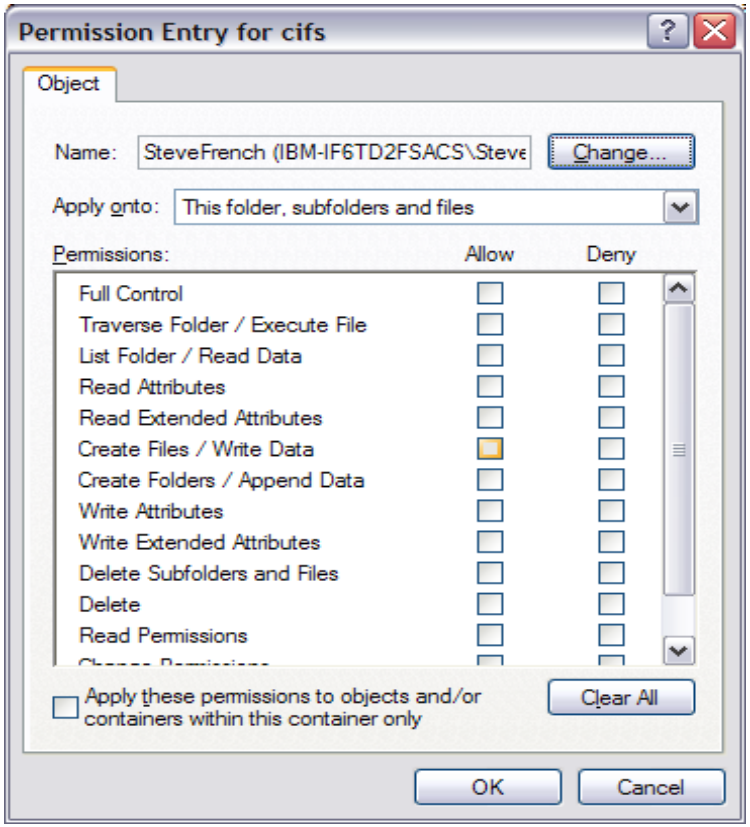
NTFS Owner information



© 2006 IBM Corporation

Rich NTFS Access Control Entries

- Note inheritence options



© 2006 IBM Corporation

CIFS VFS TODOs

- Mapping of chmod/chown/chgrp to CIFS/NTFS ACLs
- Mapping of POSIX ACLs to CIFS/NTFS ACLs
- PAM/Winbind integration: uid mapping and credential caching, refresh
- Kerberos/SPNEGO session setup (userspace Samba helper)
- Secure inkernel NTLMv2 finishup (note no upcall required, reliable reconnect of secure sessions even in low memory, high stress scenarios). Allow insecure lanman mount if configured and forced
- FindNotify finish up, IPv6 finish up
- Support “unlock all” over all mandatory byte range locks on file
- Reduce redundant opens
- SFU symlink finishup
- O_DIRECT
- DOS time conversion routines so can read Windows 9x/ME timestamp
- Additional backlevel server support testing, finish up
- Samba server version 4 support for CIFS Unix and POSIX extensions

© 2006 IBM Corporation

TODO CIFS (the protocol)

- SMB2 brings up interesting possibilities
- And also the POSIX Extensions to the CIFS protocol will continue to be extended
- NFS community is also extending NFSv4 and there is opportunity for work together between the two communities
  - *ACL mapping*
  - *Kerberos security integration*
  - *Global name space, DFS-like features*

© 2006 IBM Corporation

Note planned NFSv4 changes listed by NFS community charts mirror ours in a few places …

- Transport independence (including IPv6, RDMA, perhaps SCTP)
- POSIX ACL to NFSv4 ACL mapping

© 2006 IBM Corporation

What about the NFS community observations on NFS requirements in RFC 2624 – do they apply?

- "Reliable and Available"
- "Scalable Performance"
  - *Throughput and latency and congestion control*
  - *Disconnected client*
- Transport independence
- Data privacy
- [Avoid] Denial of service
- Recovery
- Internet accessibility
- And now pNFS – parallel i/o across servers

© 2006 IBM Corporation

What are some of the key holes in CIFS protocol?

- Various platform specific VFS extensions e.g.
  - *Inotify (in addition to previous handle based fcntl DNOTIFY)*
  - *Linux specific fcntls (e.g. getlease)*
  - *Esoteric mmapping behavior*
  - *Xattr namespace (Security and Trusted EAs)*
  - *lsattr/chattr/chflags*
  - *Generalized client UID to server UID mapping for particular mounts*
  - *New transports (SCTP) and transport QoS*
  - *New SPNEGO security dialects*
  - *POSIX locking (begun on client)*
  - *IPv6 support (begun on client)*

© 2006 IBM Corporation

Other dark secrets of CIFS (the protocol, not the implementation) … problems to address that we don't always want to talk about

- Allow signing of [only] SMB header, (and also fix signing over 4K writes)
- Writes and Reads over 64K are ugly with bcc ignored (need to get to 1MB or larger writes and reads), writes over 128K are little tested (RFC1001 reserved bits may work). SMB2 might help here.
- Allow encryption and compression bits on flag (probably already works to Windows via appropriate FCNTL)
- MID is too small (wraps too fast). SMB2 might help here.
- FS flow control (server fs/block device – return congestion bit?) so writepages and readpages on client can avoid overloading server
- Oplock timeouts too large for LAN – one size fits all not appropriate for varying cifs target networks
- Deferred commit – or avoid deferred commits by two response pattern to SMBWriteX (response = written, 2nd resp = committed to disk)
- Need better, more atomic, NTCreateX for POSIX

© 2006 IBM Corporation

Coming soon …

- First up …
  - *NTLMv2 finish up*
  - *Better SFU support*
  - *Cifs 1.43 build and limited testing across three or four distros*
  - *Note that cifs 1.42 , and thus 2.6.16 kernel, broken when signing turned on! (cifs fixed in refresh e.g. current 2.6.16.11 point release). 1.43 and 2.6.17 are fine (as were previous kernels)*
- Then
  - *Upcall finish up*
  - *Which allows Kerberos/SPNEGO calls to smb libraries*
  - *And hostname resolution needed for DFS*
  - *CIFS ACL mapping (for POSIX ACLs, chmod, chown, chgrp)*
  - *Additional performance improvements*

© 2006 IBM Corporation

Where to go for more information?

- http://linux-cifs.samba.org (note that the download page and a few other links are not updated as regularly as they should be)
- fs/cifs/README contains list of configuration options (more detailed and usually more up to date than mount.cifs.8 man page)
- CIFS FAQ/HOWTO in progress (will be available from cifs download page)
- http://www.samba.org contains links to lots of information including to Chris Hertel's book on the CIFS protocol "Implementing CIFS"

© 2006 IBM Corporation

Thank you for your time …

© 2006 IBM Corporation