CIFS Extensions for UNIX systems v2.0

Introduction

This document is a working draft, detailing proposed extensions to the CIFS specification for the UNIX operating system.

The purpose of these extensions is to allow UNIX based CIFS clients and servers to exchange information used by UNIX systems, but not present in Windows based CIFS servers or clients. Two simple examples are symbolic links and UNIX special files (e.g. UNIX named pipes).

Participants

Anyone is welcome to join this effort, although CIFS client and server implementers are the most appropriate.

Servers

HP CIFS/9000 Server – Hewlett-Packard

Clients

HP CIFS/9000 Client - Hewlett-Packard

Benefits

In addition to enhancements of standard CIFS, there are two benefits of this extension that would not be possible without the extension.

- A UNIX client could have all of its filesystems on another server. SMB is attract for this (vs NFS) due to nice features like oplocks (very nice on slow links) and the security. To do this requires device nodes at the very least.

- Allowing other clients to understand what is really going on with the server's filesystem. It would be very nice if Windows could have a property sheet listing nine permission checkboxes, and for loops caused by symbolic links were avoided when scanning the filesystem.

Principles

These are a set of principles which the extensions must meet.

- Minimal changes
  To make the extensions easier to implement, the number of changes and additions must be minimized.

- Can be implemented on non-UNIX systems
  While being useful for UNIX, the extension should allow one end of the connection to be a non-UNIX system. This is so that other CIFS servers and clients can better integrate with a UNIX CIFS client or server.

- Use current commands
  The changes should only affect current commands. There should be no need for UNIX CIFS clients to use CIFS commands marked as obsolete, nor should there be any changes to obsolete requests.

- Retain existing CIFS semantics
  The existing semantics of CIFS are retained. Perhaps the most notable is that file names are case insensitive, but case should be preserved.

- Use CIFS security model
  The standard CIFS security model is still used. This requires each distinct user to be logged into the server.

- Addition to dialect
  This specification is an addition to the CIFS dialect, currently NT LM 0.12. It is selected by the capability bit in the server's Negprot response.

- Future resilient
  Field sizes should not require updating in the future. Hence they should be bigger than those currently in use.

Specification

Negprot (Response)

- The CAP_UNIX value in the first version of the CIFS Extension for UNIX specification was 0x80000000.
- This value was assigned to CAP_EXTENDED_SECURITY in the CIFS Specification.
- The new CAP_UNIX value will be 0x800000. An official CAP_UNIX value needs to be added to the CIFS Specification

Trans2_Query/Set File/Path Information

New info levels are defined for this set of requests.

SMB_QUERY_FILE_UNIX_BASIC (0x200)

LARGE_INTEGER ENDOFFILE	FILE SIZE
LARGE_INTEGER NumOfBlocks	Number of file system block used to store file
LARGE_INTEGER LastStatusChange	Last time the status of the file was changed. This is in DCE time
LARGE_INTEGER LastAccessTime	Last access time. This is in DCE time.
LARGE_INTEGER LastModificationTime	Last modification time. This is in DCE time.
LARGE_INTEGER Uid	Numeric user id for the owner
LARGE_INTEGER Gid	Numeric group id of owner
ULONG Type	Enumeration specifying the pathname type: • 0 -- File • 1 -- Directory • 2 -- Symbolic link • 3 -- Character device • 4 -- Block device • 5 -- FIFO (named pipe)
LARGE_INTEGER devmajor	Major device number if type is device
LARGE_INTEGER devminor	Minor device number if type is device
LARGE_INTEGER uniqueid	This is a server-assigned unique id for the file. The client will typically map this onto an inode number. The scope of uniqueness is the share.
LARGE_INTEGER permissions	Standard UNIX file permissions
LARGE_INTEGER nlinks	The number of directory entries that map to this entry (number of hard links)

SMB_QUERY_FILE_UNIX_LINK (0x201)

STRING LinkDest	Destination for the symbolic link

SMB_QUERY_FILE_UNIX_HLINK (0x203)

STRING LinkDest	Destination for the HARD link

Trans2_FindFirst/FindNext

SMB_FIND_FILE_UNIX (0x202)

ULONG NextEntryOffset	Offset from this structure to the beginning of the next one
ULONG ResumeKey	Used for continuing search.
SMB_QUERY_FILE_UNIX_BASIC	All fields from SMB_QUERY_FILE_UNIX_BASIC info level
STRING Name	Case-preserved alternative filename

Case

The minimum requirement is that the server preserve case when new names are created, and is case insensitive for other operations. This is normal behavior for the NT LM 0.12 dialect.

If there is a separate specification for case sensitivity, then that can be used in addition to this specification.

Guidelines for implementers

Clients or servers using this extension have no specific reserved filenames (eg CON AUX PRN), and need to take no specific action to protect the other end of the connection from them. If they have any such requirements, they must do them internally. This also applies to reserved characters in filenames (eg : \ |).

Inodes can be generated using the uniqueid field (per share).

Clients should operate in UNICODE if at all possible. A useful bridging step is to implement UTF-8

Symbolic links are created by calling Trans2_setpathinfo with the SMB_QUERY_FILE_UNIX_LINK infolevel data structure provided.

Device nodes are created by calling Trans2_setpathinfo with the SMB_QUERY_FILE_UNIX_BASIC infolevel data structure appropriately filled in for a device node.

Servers should return their timezone as UTC. This will then require no timezone mapping by the client or server. The NetRemoteTimeOfDay IPC should still return the real local time.

Creates with particular permissions can be achieved by doing a create_and_X, and a set_file_info.

Future Features Under consideration

- Quota management using Trans2_Query|Set_FS_INFO.
- Case sensitivity negotiation
- Support for socket file type
- Remote user/group mapping
- Posix ACL
- In IPC, an info level for NetUserInfo that allows retrieving user information (eg the fields in /etc/passwd)

Features From First Revision Not Implemented

Root File System Support

All the necessary functionality for UNIX clients should be present. It should be possible to mount the root file system from a UNIX CIFS server.

Session Setup and X (Request) (Not Implemented in 1st Release)

The account name can be of the form #12345678. The server should interpret this as a user id number. If the server trusts the client machine, or it trusts the user from that machine, it may ignore the password. The mechanism by which this trust is established is not specified here.

The following usernames are recommended for special purposes.

unixkernel
: When the username or number is unavailable, or not relevant to the client redirector. The server may choose to allow this name root equivalence.

bootmonitor
: Used when accessing the CIFS server at boot time.

anonymous
: A user the client system considers a guest.

Errors

The following errors are added to the standard set in ERRDOS:

0x200
: ERRDOS__ErrQuota
: The operation would cause a quota limit to be exceeded.

0x201
: ERRDOS__ErrNotALink
: A link operation was performed on a pathname that was not a link.

History

This specification grew out of discussions at the first CIFS Implementor's Workshop. Some further discussions happened by email, and it was decided to set up a mailing list. The mailing list archives contain the major points of discussion from the email quoted in the first few messages.

SCO created version 1.0 of the specification and presented it at the second CIFS Implementor's Workshop in the Spring of 1997

Hewlett-Packard took over ownership of the specification in the Spring of 2000. Presented the specification at the CIFS 2000 Workshop.

Change Log

10 January 2000
: Changed CAP_UNIX value to 0x800000 since the previous chosen value of 0x8000 was found to be used by Windows 2000. The original CAP_UNIX value was assigned by Microsoft to CAP_EXTENDED_SECURITY in the latest CIFS Protocol Specification.

16 August 1999

Added field NumOfBlocks of type LARGE_INTERGER in the SMB_QUERY_UNIX_BASIC (0x200). This field will contain the number of file system blocks allocated by the file by the target server. This field will be the 2nd field in the data portion to the SMB right after the EndOfFile field.

29 July 1999

Modified spec with following changes:
1) Changed CAP_UNIX value to 0x8000 since the original value of 0x80000000 was assigned by Microsoft to CAP_EXTENDED_SECURITY in the latest CIFS Protocol Specification.
2) Added SMB_QUERY_FILE_UNIX_HLINK (0x203) info level to enable the creation of Hard links.
3) Fixed probably typo error by changing infolevel of SMB_FIND_FILE_UNIX from (0x200) to (0x202).

7 October 1997

Converted more fields to LARG_INTEGER and changed the proposed numbers.

17 January 1997

Added benefits and history sections and items to principles and guidelines sections.

13 January 1997

Added constants for levels, section on case, resume key to SMB_FIND_FILE_UNIX, named pipe type.

29 November 1996

Updated with various comments. All file sizes are 64bit, more implementor guidelines

25 November 1996

Initial web version

19 November 1996

Draft text