

Unix Extensions for SMB2 Protocol Initiative

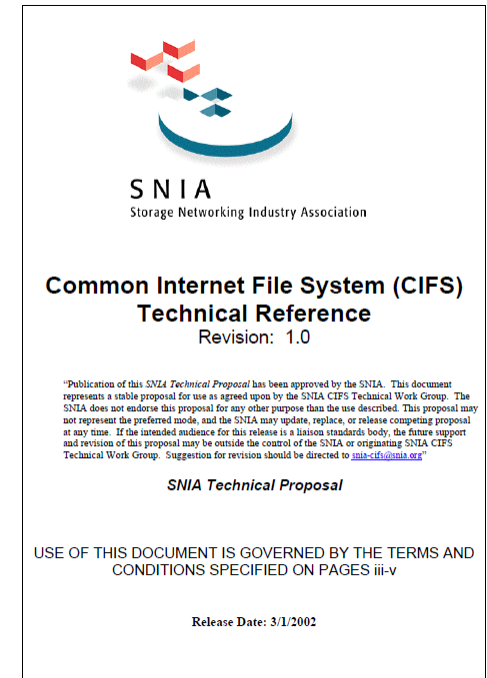
Tom Talpey
Microsoft

- ❑ SMB2 is the file sharing protocol for modern Windows® interoperability
 - ❑ Posix semantics are needed for broad deployment of SMB2 on Unix (Linux, Mac, etc)
- ❑ Posix is supported only by earlier CIFS/SMB extensions
- ❑ Transport encryption is not supported by SMB2
- ❑ Close these gaps!

- ❑ Enable SMB->SMB2 heterogeneous migration
- ❑ Broaden SMB2 interoperability
- ❑ Strengthen SMB2 security

HISTORY

- ❑ SNIA CIFS Technical Working Group
 - ❑ http://www.snia.org/tech_activities/CIFS
- ❑ Unix and Mac extensions as appendices to SNIA CIFS technical specification
 - ❑ Published March 2002
 - ❑ (and prior draft versions)



- ❑ “Carveout” in SMB Information Levels
 - ❑ 0x200-0x2ff (only a few actually used)
- ❑ Single bit in server capabilities
 - ❑ CAP_UNIX negotiation
- ❑ Related set of “Macintosh” extensions
 - ❑ Similar approach

- ❑ “Carveout” approach is problematic
 - ❑ Works only between well-matched peers
 - ❑ Not readily extensible
 - ❑ Led to selective implementation:
 - “12. Appendix D – CIFS UNIX Extension
 - 12.1. Introduction
 - 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.
These extensions may not be implemented by all UNIX systems.”
- ❑ Limited addressing of needed requirements
 - ❑ Overload of existing ops e.g. COM_NT_RENAME
- ❑ Not extensible to SMB2

- ❑ Mixed implementation support
 - ❑ Partial, or not implemented on servers
- ❑ Minimal common feature interoperation
 - Clients unable to count on server support
 - Clients forced to reduce expectations
 - Clients forced to reduce functionality guarantees

CURRENT INITIATIVE

- ❑ A public collaborative effort to create new protocol extensions for the SMB2 protocol, to support Posix/Unix behaviors
- ❑ A lightweight, focused process which will draw on prior CIFS/SMB extensions and industry experience
- ❑ Will produce an open published specification, available for implementation industry-wide
- ❑ Specification(s) to be independent of Microsoft Open Protocols document set
 - ❑ MS-SMB2

New Scope and Approach

- Scope:
 - Support Posix
 - Support transport encryption
 - Provide explicit negotiation
- Approach
 - Explicit reserved Information Levels
 - Explicit reserved ioctls/fsctls, if needed
- Independent of MS-SMB2 protocol spec
 - May make references into other document(s)

- ❑ Protocol changes
 - ❑ New capabilities, operations, etc
 - ❑ Wire-visible only
- ❑ Server best practice
 - ❑ Rules and recommendations to ensure interop
- ❑ Restrictions
 - ❑ Character set, reserved characters, reserved names
- ❑ Improvements, protocol relationships, etc

PROJECT

- ❑ Negotiation of server extension by client (*)
 - ❑ Including negotiation of specific supported items below
- ❑ Encapsulation of Posix/Unix capabilities:
 - ❑ Filename case sensitivity support (*)
 - ❑ Posix attributes tunneled: UID/GID, permissions, sizes, times, etc. (*)
 - ❑ Posix filename character set support
 - ❑ Posix semantics on rename of open files
 - ❑ Posix semantics on unlink of files
 - ❑ Posix semantics on read-only directories
 - ❑ Posix file mode set/get
 - ❑ Posix file locking (fcntl(F_GETFL/F_SETFL), advisory vs. mandatory)
 - ❑ fcntl(F_FULLFSYNC) (Darwin FUA)
- ❑ Transport encryption support

(*) indicates support by prior SMB/Unix extensions

- ❑ Protocol support not explicitly included above
 - ❑ Add to scope only with unanimous agreement
- ❑ Implementation
 - ❑ Producing only a document.

- ❑ Specification
 - ❑ The document itself
- ❑ Discussion
 - ❑ Exchange of ideas and experience to shape the importance of the specification details
 - ❑ What's most important, what's less
 - ❑ Understand approaches to each
 - ❑ Understand comparisons with NFS
- ❑ Consensus
 - ❑ Community willingness to advance and implement the effort

- “Project Coordinator”
 - Tom Talpey, Microsoft
- Contributors
 - recruiting now!
- Project administration
 - Chris Hertel and José Rivera, ubiqx Consulting

Contributor's Agreement

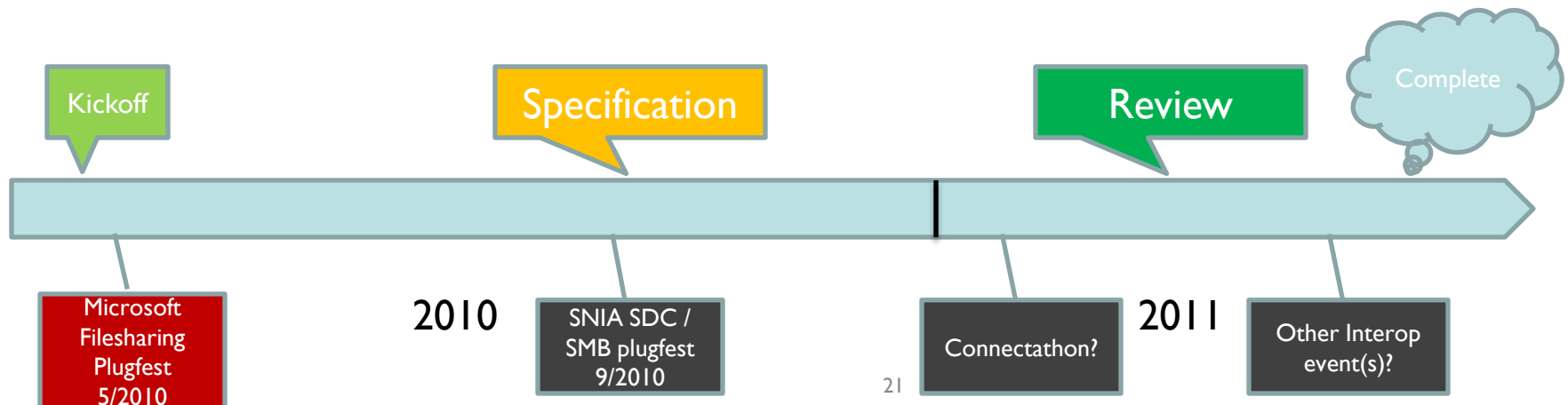
- ❑ Companies and/or individuals will grant certain rights to their contributions to the Project
- ❑ Ensures that use and development of the Specification is available to all
- ❑ Final review process (45 days) at project completion
- ❑ Click-through acceptance

- ❑ Willingness to contribute to specification
 - ❑ Content (ideas, experience)
 - ❑ Writing/editing
 - ❑ Reviewing
- ❑ Intention to implement is not necessary
 - ❑ Though exciting!

- ❑ All discussion to take place via email reflector
- ❑ Face-to-face or conference calls possible
- ❑ Wiki-style issues list, work-in-progress, etc
- ❑ Postings to email and wiki are “Contributions” and may be included in the Specification
- ❑ Discussions will be open to public as read-only
- ❑ Work-in-progress restricted to Contributors

Schedule

- A 6-12 month effort envisioned
- Project kickoff and recruiting – under way
- Project work begins – now
- Specification Development – 2010-2011
- Specification Complete – early-mid 2011



For discussion! These are **not** yet committed as plan.

DETAILS

- Incrementally extend the SMB2 protocol to support access and sharing of server resources (e.g. files) by Posix clients
- Design the extensions to have no impact on data integrity, minimal impact on performance, and maximal transparency to applications
- Allow the extensions to be supported by a wide variety of servers and clients, including non-Posix

- ❑ Posix extensions are for benefit of Posix **clients**
 - ❑ Allow Posix operations to flow over the wire
- ❑ Implementation difficulty arises at SMB2 **servers**
 - ❑ Posix operations may not line up with server and SMB2 operation processing
- ❑ Therefore, work items will generally combine both wire protocol and best practice in their specification

- ❑ Change-notify semantics with Posix extensions
- ❑ Read and write operation semantics with Posix locking in effect per-file
 - ❑ Mandatory lock behavior exposed to Posix client
 - ❑ Advisory lock behavior impact on SMB2 client
- ❑ Leases
 - ❑ Any issues using them from Posix client?

Non-wire protocol issues (2)

- ❑ Rename/delete semantics
 - ❑ A Big One
 - ❑ Exposing behavior on wire an alternative?
 - ❑ i.e. leave it up to the client...
- ❑ Atomicity/replay
 - ❑ Does Posix rename/delete introduce new issues?
 - ❑ Non-handle-based, with different required behaviors
 - ❑ Introduction of new issues ideally to be avoided

- ❑ Prior SMB Extensions support per-server capabilities
 - ❑ CAP_UNIX enabled all of Appendix D
- ❑ SMB2 Extensions possibilities:
 - ❑ Per-server to indicate extended capability
 - ❑ Per-Session to request/provide Posix behavior(s)?
 - ❑ Per-TreeConnect??
 - ❑ Per-Open???

- ❑ Advisory vs Mandatory
 - ❑ Fundamental issue, with consensus on approach required
 - ❑ E.g., advertise server behavior, or require it?
- ❑ Lock stacking/merge/split/overlap
 - ❑ Differing behaviors per-personality
 - ❑ Possibly implementable in server via best practice
- ❑ Conflict handling
 - ❑ If SMB2 or server-local client holds mandatory lock, how to reflect to Posix client, and vice versa?
- ❑ Lock state recovery after transport loss
 - ❑ Durable handle requirement?

- ❑ Potentially large issue requiring other protocols and services/facilities
- ❑ UID/GID and mode bit alternatives
 - ❑ Storage in ACL or extended attribute
 - ❑ Require a mapping facility
 - ❑ RFC2307/LDAP, other “idmapper”, etc
- ❑ ACL interoperability
 - ❑ Define a Posix ACL mapping? (ouch)
 - ❑ Expose server native ACL format?
 - ❑ Possibly large enough to warrant subgroup effort

- ❑ Encapsulate in new suboperations?
 - ❑ E.g. new SMB2_{QUERY,SET}_INFO type(s) and class(es)
 - ❑ New SMB2_CREATE create context(s)
- ❑ Including...
 - ❑ UID/GID/mode bits
 - ❑ Special files
 - ❑ Inode number
 - ❑ Requirements for uniqueid generation and persistence
 - ❑ Statfs/Fsinfo
 - ❑ Per-filesystem, but e.g. blocksizes feed back to file attributes
 - ❑ Other mode bits?
 - ❑ Immutable, append-only, etc
- ❑ Extended attributes (or is existing support enough?)
- ❑ Expose Posix flush requirement on stat(2)?

- ❑ Symbolic links (or is existing support enough?)
- ❑ Case sensitive / case preserving
 - ❑ Expose, negotiate, or other?
 - ❑ Case-sensitive search/open?
- ❑ Character set
 - ❑ Filenames on the wire
 - ❑ Filesystem support requirements
 - ❑ Reserved characters
 - ❑ Remapping support?
- ❑ Errors
 - ❑ Optionally expose errors from Posix server?

- ❑ Transport encryption
 - ❑ Not currently addressed by SMB2
 - ❑ Wide-open for discussion
- ❑ Transport compression
 - ❑ Closely related issue
 - ❑ Interest in this?

Call to Action

- ❑ Visit the website
 - ❑ <http://unixsmb2.org>
- ❑ Read the Contributor's Agreement there
- ❑ Accept it and sign up
- ❑ Contribute!

QUESTIONS?

Tom Talpey: ttalpey@microsoft.com

<http://unixsmb2.org>