A Developer’s View of the Microsoft SMB Protocol Documentation

Tom Talpey

Microsoft

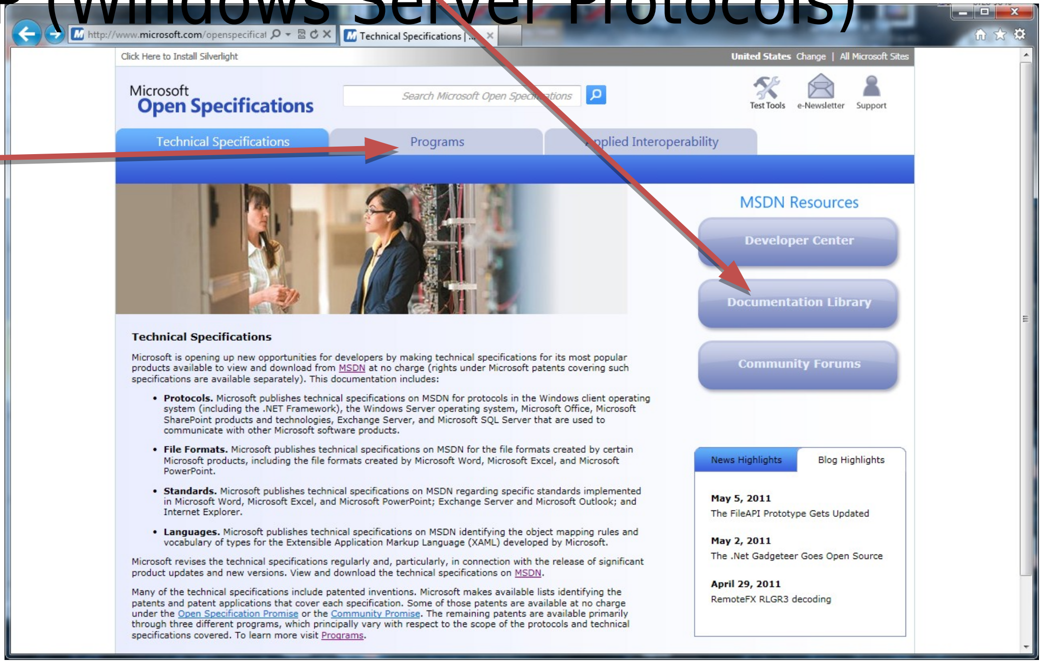
May 10, 2011

Outline

- Filesharing document set overview
- Managing change
- Understanding and use of the documents
- Future

Microsoft Open Specifications

- http://www.microsoft.com/protocols
  - Ø Documentation Library
    - ØWindows Protocols
      - ü MCPP (Windows Communication Protocols)
      - ü WSPP (Windows Server Protocols)
- Also see
  - Ø Programs
    - Ø MIPP
    - Ø MCPP
    - Ø WSPP


http://www.microsoft.com/openspecificat
Technical Specifications | ...
Click Here to Install Silverlight
United States Change | All Microsoft Sites
Microsoft Open Specifications
Search Microsoft Open Specifications
Test Tools
e-Newsletter
Support
Technical Specifications
Programs
Applied Interoperability
MSDN Resources
Developer Center
Documentation Library
Community Forums
News Highlights
Blog Highlights

Technical Specifications

Microsoft is opening up new opportunities for developers by making technical specifications for its most popular products available to view and download from MSDN at no charge (rights under Microsoft patents covering such specifications are available separately). This documentation includes:

- **Protocols.** Microsoft publishes technical specifications on MSDN for protocols in the Windows client operating system (including the .NET Framework), the Windows Server operating system, Microsoft Office, Microsoft SharePoint products and technologies, Exchange Server, and Microsoft SQL Server that are used to communicate with other Microsoft software products.
- **File Formats.** Microsoft publishes technical specifications on MSDN for the file formats created by certain Microsoft products, including the file formats created by Microsoft Word, Microsoft Excel, and Microsoft PowerPoint.
- **Standards.** Microsoft publishes technical specifications on MSDN regarding specific standards implemented in Microsoft Word, Microsoft Excel, and Microsoft PowerPoint; Exchange Server and Microsoft Outlook; and Internet Explorer.
- **Languages.** Microsoft publishes technical specifications on MSDN identifying the object mapping rules and vocabulary of types for the Extensible Application Markup Language (XAML) developed by Microsoft.

Microsoft revises the technical specifications regularly and, particularly, in connection with the release of significant product updates and new versions. View and download the technical specifications on MSDN.

Many of the technical specifications include patented inventions. Microsoft makes available lists identifying the patents and patent applications that cover each specification. Some of those patents are available at no charge under the Open Specification Promise or the Community Promise. The remaining patents are available primarily through three different programs, which principally vary with respect to the scope of the protocols and technical specifications covered. To learn more visit Programs.

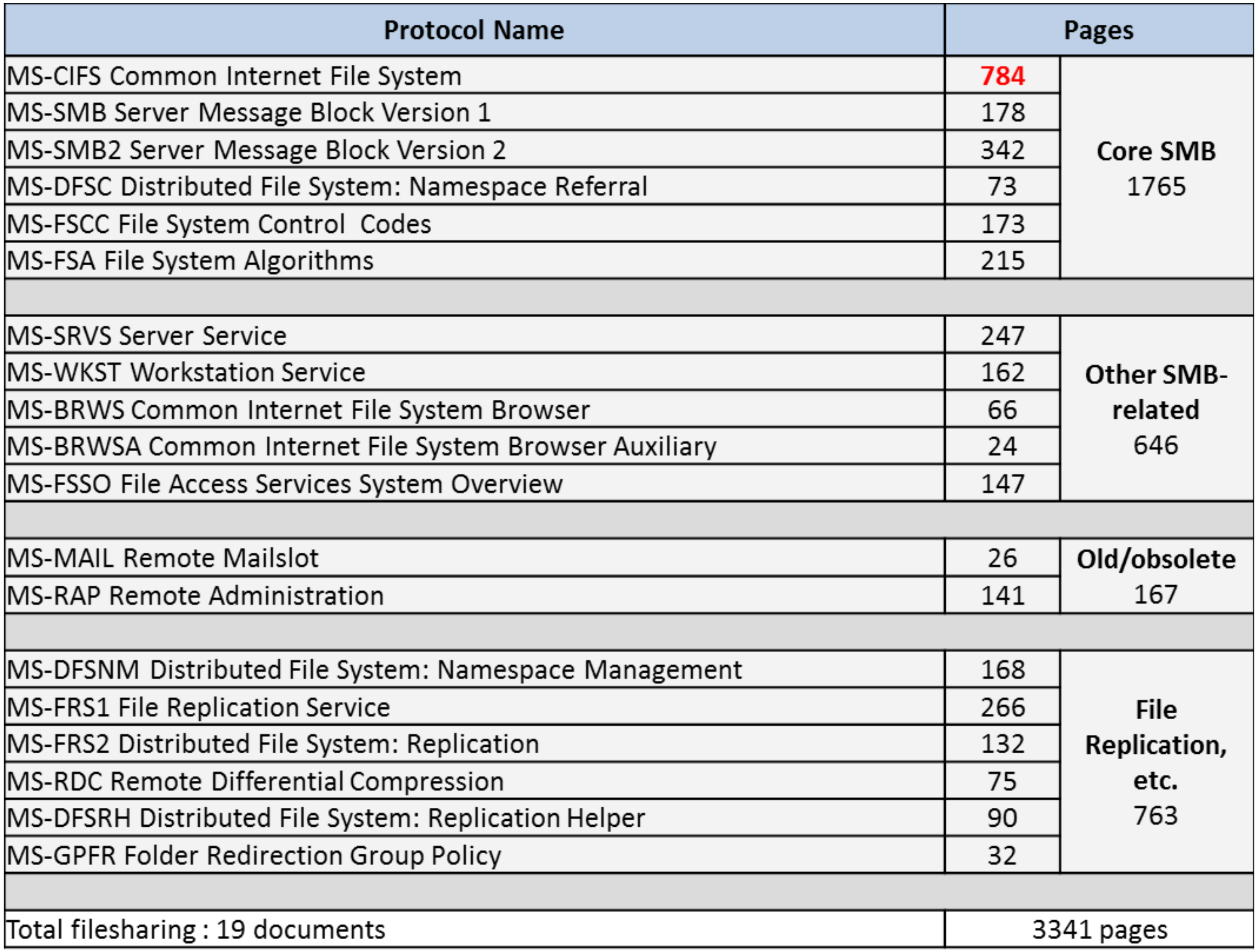
May 5, 2011
The FileAPI Prototype Gets Updated

May 2, 2011
The .Net Gadgeteer Goes Open Source

April 29, 2011
RemoteFX RLGR3 decoding

The SMB Document Family

The SMB family documents



Protocol Name	Pages	
MS-CIFS Common Internet File System	784	**Core SMB** 1765
MS-SMB Server Message Block Version 1	178	
MS-SMB2 Server Message Block Version 2	342	
MS-DFSC Distributed File System: Namespace Referral	73	
MS-FSCC File System Control Codes	173	
MS-FSA File System Algorithms	215	
MS-SRVS Server Service	247	**Other SMB-related** 646
MS-WKST Workstation Service	162	
MS-BRWS Common Internet File System Browser	66	
MS-BRWSA Common Internet File System Browser Auxiliary	24	
MS-FSSO File Access Services System Overview	147	
MS-MAIL Remote Mailslot	26	**Old/obsolete** 167
MS-RAP Remote Administration	141	
MS-DFSNM Distributed File System: Namespace Management	168	**File Replication, etc.** 763
MS-FRS1 File Replication Service	266	
MS-FRS2 Distributed File System: Replication	132	
MS-RDC Remote Differential Compression	75	
MS-DFSRH Distributed File System: Replication Helper	90	
MS-GPFR Folder Redirection Group Policy	32	
Total filesharing : 19 documents	3341 pages	

SMB Document relationships

- MS-CIFS
  - Common Internet File System protocol (historic)
  - Coverage ends at Windows NT 4.0
- MS-SMB
  - Server Message Block protocol (also historic)
  - Extension to MS-CIFS (not standalone)
  - Coverage to present day Windows
- MS-SMB2
  - Server Message Block version 2
  - Coverage from Windows Vista forward

SMB Support Relationships

- MS-DFSC
  - DFS Namespace Referral protocol
    - Resolves DFS paths
- MS-FSCC
  - Filesystem Control codes
    - Types, constants, and FSCTL definitions
- MS-FSA
  - Filesystem Algorithms
    - Underlying filesystem abstraction
- All invoked by all – MS-CIFS, MS-SMB, MS-SMB2

SMB Support Protocols

- MS-SRVS
  - “Server Service”
    - management and common state for SMB protocols (server and client)
- MS-WKST
  - “Workstation Service”
    - About 1/3 of document related to SMB administration and management
- MS-BRWS, MS-BRWSA
  - NetBIOS browser protocol
-

Other SMB-related

- MS-FSSO
  - File Access System Overview
- MS-MAIL, MS-RAP
  - Legacy SMB access and administration
- Replication and DFS management
  - MS-FRSx – replication
  - MS-DFSNM – DFS administration
  - Related but not subject of this talk
  -

Change

- The past year has seen a LOT of change from
  - Intense scrutiny by the Technical Committee
  - Formal protocol and document test process
  - Very active plugfests and licensee feedback
  - Internal review

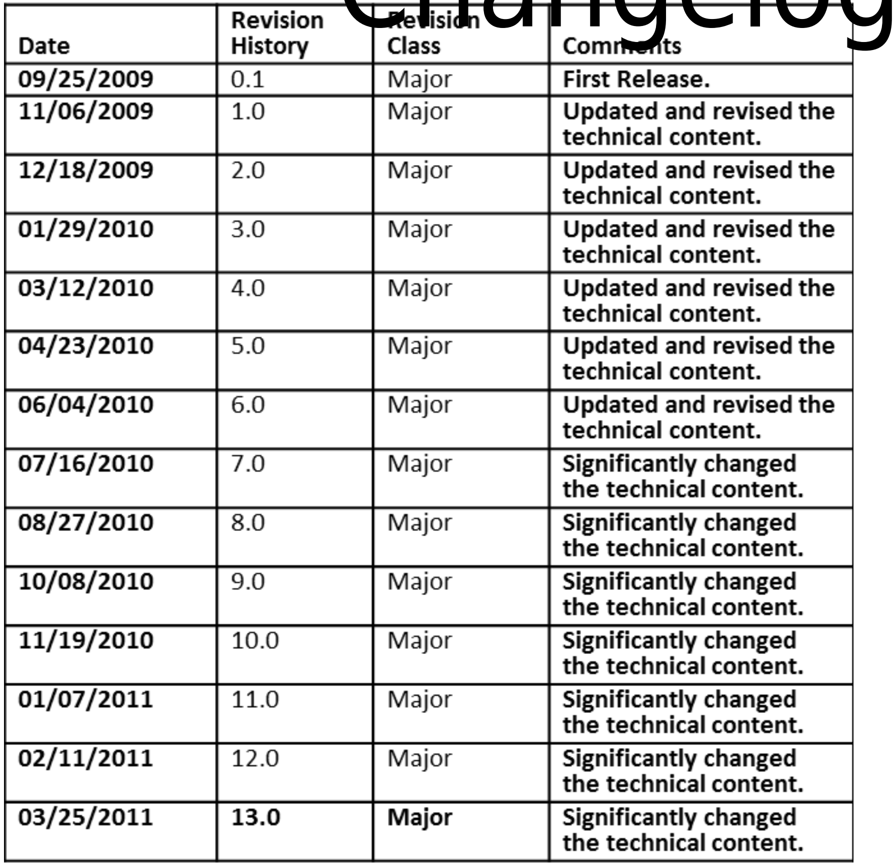
Change in MS-CIFS

- For example, MS-CIFS:
  - Largest document in filesharing
  - Came to exist September 2009 (thanks Chris!)
  - By April 2010 it was 763 pages
- From April 15, 2010 to April 15, 2011
  - Grown to 784 pages
  - Resolutions for 706 issues (1 issue/page!)
  - 2660 total changes
  - 5443 body text revisions
    - 2801 insertions
    - 2510 deletions
    - 132 moves

Versions

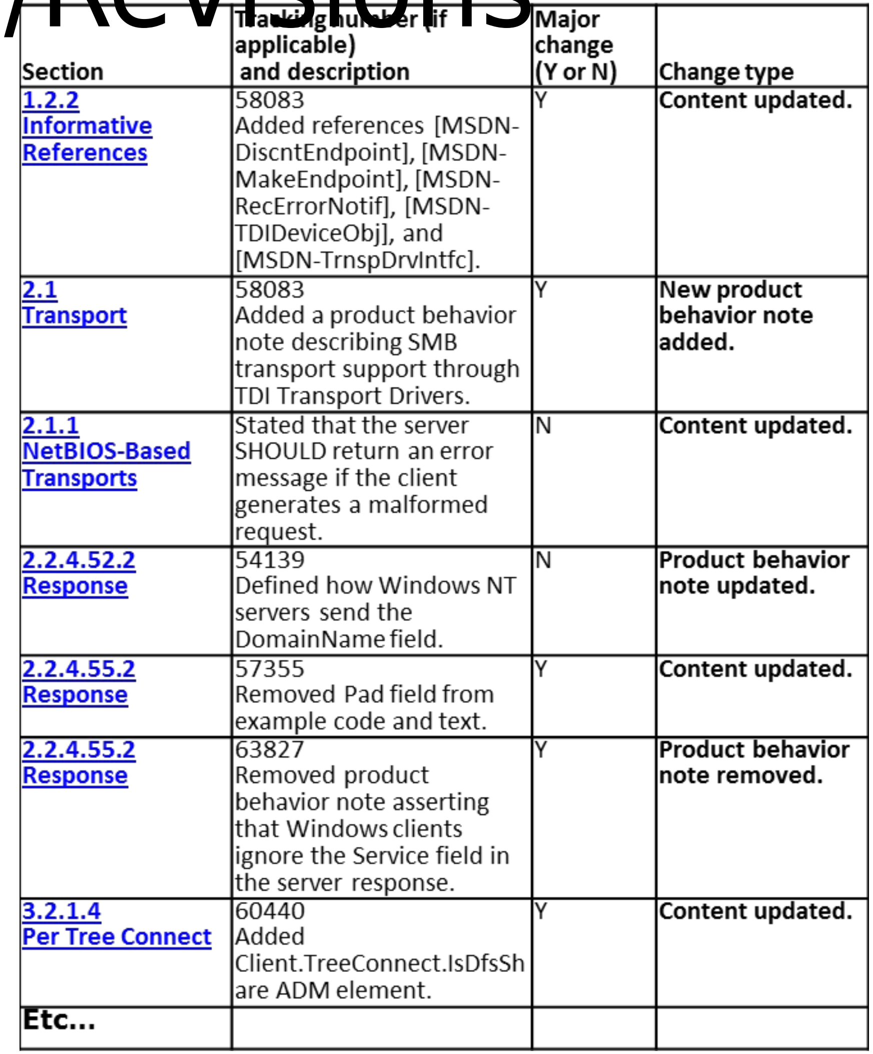
- Living documents – republished regularly
- Documents are stamped with
  - Section 1: Date, revision #
  - Section 7: Changelog
    - Marked with tracking number ("TDI") if available
    - Note: the changelog resets with each release
- Generally speaking, use the latest!

E.g. MS-CIFS Changelog/Revisions



Date	Revision History	Revision Class	Comments
09/25/2009	0.1	Major	**First Release.**
11/06/2009	1.0	Major	**Updated and revised the technical content.**
12/18/2009	2.0	Major	**Updated and revised the technical content.**
01/29/2010	3.0	Major	**Updated and revised the technical content.**
03/12/2010	4.0	Major	**Updated and revised the technical content.**
04/23/2010	5.0	Major	**Updated and revised the technical content.**
06/04/2010	6.0	Major	**Updated and revised the technical content.**
07/16/2010	7.0	Major	**Significantly changed the technical content.**
08/27/2010	8.0	Major	**Significantly changed the technical content.**
10/08/2010	9.0	Major	**Significantly changed the technical content.**
11/19/2010	10.0	Major	**Significantly changed the technical content.**
01/07/2011	11.0	Major	**Significantly changed the technical content.**
02/11/2011	12.0	Major	**Significantly changed the technical content.**
03/25/2011	**13.0**	**Major**	**Significantly changed the technical content.**

Note: Issues are all “major”.
This will change. ☺



Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.2.2 Informative References	58083 Added references [MSDN-DiscntEndpoint], [MSDN-MakeEndpoint], [MSDN-RecErrorNotif], [MSDN-TDIDeviceObj], and [MSDN-TrnspDrvIntfc].	Y	**Content updated.**
2.1 Transport	58083 Added a product behavior note describing SMB transport support through TDI Transport Drivers.	Y	**New product behavior note added.**
2.1.1 NetBIOS-Based Transports	Stated that the server SHOULD return an error message if the client generates a malformed request.	N	**Content updated.**
2.2.4.52.2 Response	54139 Defined how Windows NT servers send the DomainName field.	N	**Product behavior note updated.**
2.2.4.55.2 Response	57355 Removed Pad field from example code and text.	Y	**Content updated.**
2.2.4.55.2 Response	63827 Removed product behavior note asserting that Windows clients ignore the Service field in the server response.	Y	**Product behavior note removed.**
3.2.1.4 Per Tree Connect	60440 Added Client.TreeConnect.IsDfsShare ADM element.	Y	**Content updated.**
Etc…			

Normative Statements

SHOUTING

- Each document states:

> **MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

- Which is true, but note…

These statements are normative to Windows

- The protocol template requires this for compliance.

Shouting implications

- The statements are not necessarily written to constrain other implementations
  - But to match Windows, they’re necessary
- Protocol implementers need to think through the implications
- Also note, no other RFC2119 terms are used
  - RECOMMENDED, OPTIONAL, etc

Shouting, simplified

- MUST:
  - Windows **always does it**
- MUST NOT:
  - Windows **never does it**
- SHOULD:
  - Windows **does it**, except as noted
- MAY, SHOULD NOT:
  - Windows **doesn’t do it**, except as noted

Beware the Shouting Not Made

- When not qualified with a behavior note, the statement “Windows MAY <x>” is invalid, by template
  - It would be a non-testable assertion
  - So, it’s often not made at all
- Occasionally, documents give informative advice with “can” (for example)
- Proceed with care

So, SHOULD you (NOT) do it?

- Generally, yes!
- Occasionally, maybe not
  - If not supported
  - If reason is well-understood
  - If avoiding bug
- Very hard to generalize fully
- You need to think it through
  -

Good sense

- **MS-CIFS 3.2.4.2.5 Connecting to the Share (Tree Connect)**

  In the LAN Manager 1.0 dialect and above, it is a protocol violation to send a tree connect request without completing an SMB_COM_SESSION_SETUP_ANDX (section 2.2.4.53) exchange. When using share level access control, the client MUST perform anonymous authentication (empty username and password) in the Session Setup.

  If a tree connect is already established to the target share in Client.Connection.TreeConnectTable, it SHOULD be reused. If not, the client creates an SMB_COM_TREE_CONNECT_ANDX Request (section 2.2.4.55.1), as specified in section 2.2.4.55. Alternately, the client MAY use the deprecated SMB_COM_TREE_CONNECT Request (section 2.2.4.50.1).

  If **Client.Connection ShareLevelAccessControl** is TRUE and a null session has been established (see section 3.2.4.2.4), the plaintext password or authentication response MUST be passed in the **Password** field of the SMB_COM_TREE_CONNECT_ANDX.Request or SMB_COM_TREE_CONNECT Request. There is only one **Password** field in the tree connect message, so only one response value can be sent. The client MUST determine the authentication type that it uses based upon **Client.Connection.ServerChallengeResponse** and the local configuration (the **Client.PlaintextAuthenticationPolicy**, **Client.LMAuthenticationPolicy**, and **Client.NTLMAuthenticationPolicy** values), as specified in section 3.2.4.2.4.
-
- Implement the MUSTs.
- Strongly consider implementing the SHOULDs.
- Don't implement the MAY, unless you really really mean to.

Nonsense

- **MS-WKST 3.2.4.7 NetrUseAdd (Opnum 8)**

  The **NetrUseAdd** method establishes a connection between the workstation server and an SMB server. Workstation servers SHOULD NOT allow this method to be invoked remotely<72> and SHOULD return ERROR_CALL_NOT_IMPLEMENTED.

  ```
  unsigned long NetrUseAdd(
    [in, string, unique] WKSSVC_IMPERSONATE_HANDLE ServerName,
    [in] unsigned long Level,
    [in, switch_is(Level)] LPUSE_INFO InfoStruct,
    [in, out, unique] unsigned long* ErrorParameter
  );
  ```

  …

  <72> Section 3.2.4.7:Even though Windows servers expose this RPC call to remote callers, it is intended to be called only by processes on the local machine. Windows clients will never issue this RPC call to a remote machine.

- Don’t implement this remotely. It’s documented only because Windows does it. Uselessly. ☺
- Technically, we should have said MUST implement. Shhh…

Examples of Change

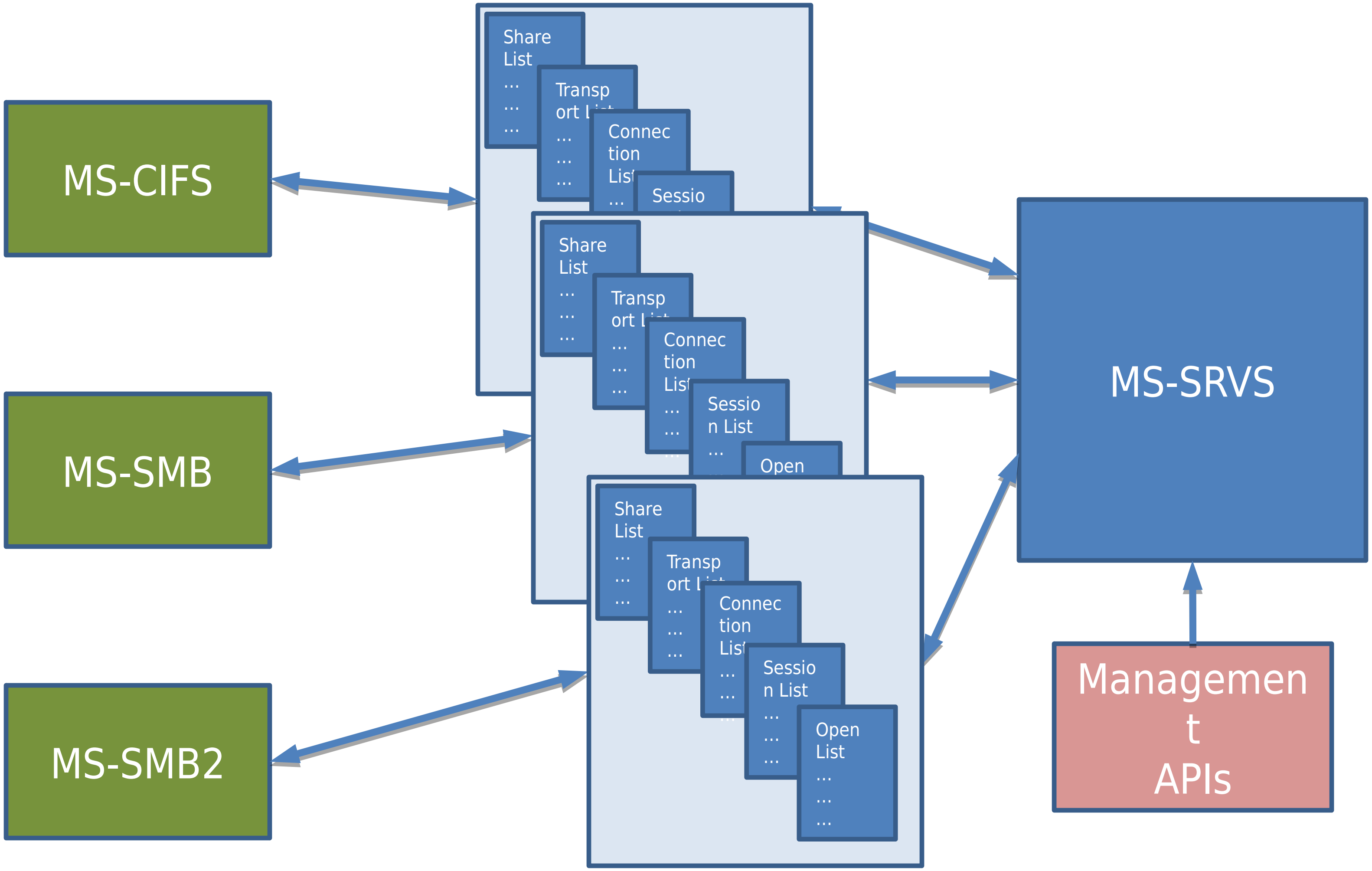
Structural changes

- Examples:
  - Shared state
  - Filesystem actions

“Shared State”

- Formerly, the SMB documents followed a “shared state” abstract model with MS-SRVS
    - Each protocol was responsible for its objects
    - These objects were “shared” with MS-SRVS for management control and statistics
- Highly unmanageable
    - Did not to consistently cover all 3 protocols
    - Violated locality and ownership
    - Documents buggy as heck

Shared State - old



Shared state approach – SMB2 session

- MS-SMB2:

  3.3.1.6 Global Structures

  The server implements the following:

  § **GlobalSessionTable**: A list of all the active sessions established to this server, indexed by the *Session.SessionId*. The server MUST also be able to search the list by security principal, and the list MUST allow for multiple sessions with the same security principal on different connections.

  …

  3.3.5.5.1 Authenticating a New Session

  A session object MUST be allocated for this request. The session MUST be inserted into the **GlobalSessionTable** and a unique **Session.SessionId** assigned to serve as a lookup key in the table.

- MS-SRVS:

  3.1.1.1 Global

  The server MUST implement the following:

  …

  SessionList: A sorted list of Sessions where each element is a tuple of the form <SessionId, Session>. The SessionId is a unique 32-bit value identifying the Session in **SessionList**. This list MUST contain both SMB sessions as described in [MS-SMB] section 3.3.1.9 and SMB2 sessions as described in [MS-SMB2] section 3.3.1.9.

  …

  3.1.4.5 NetrSessionEnum (Opnum 12)

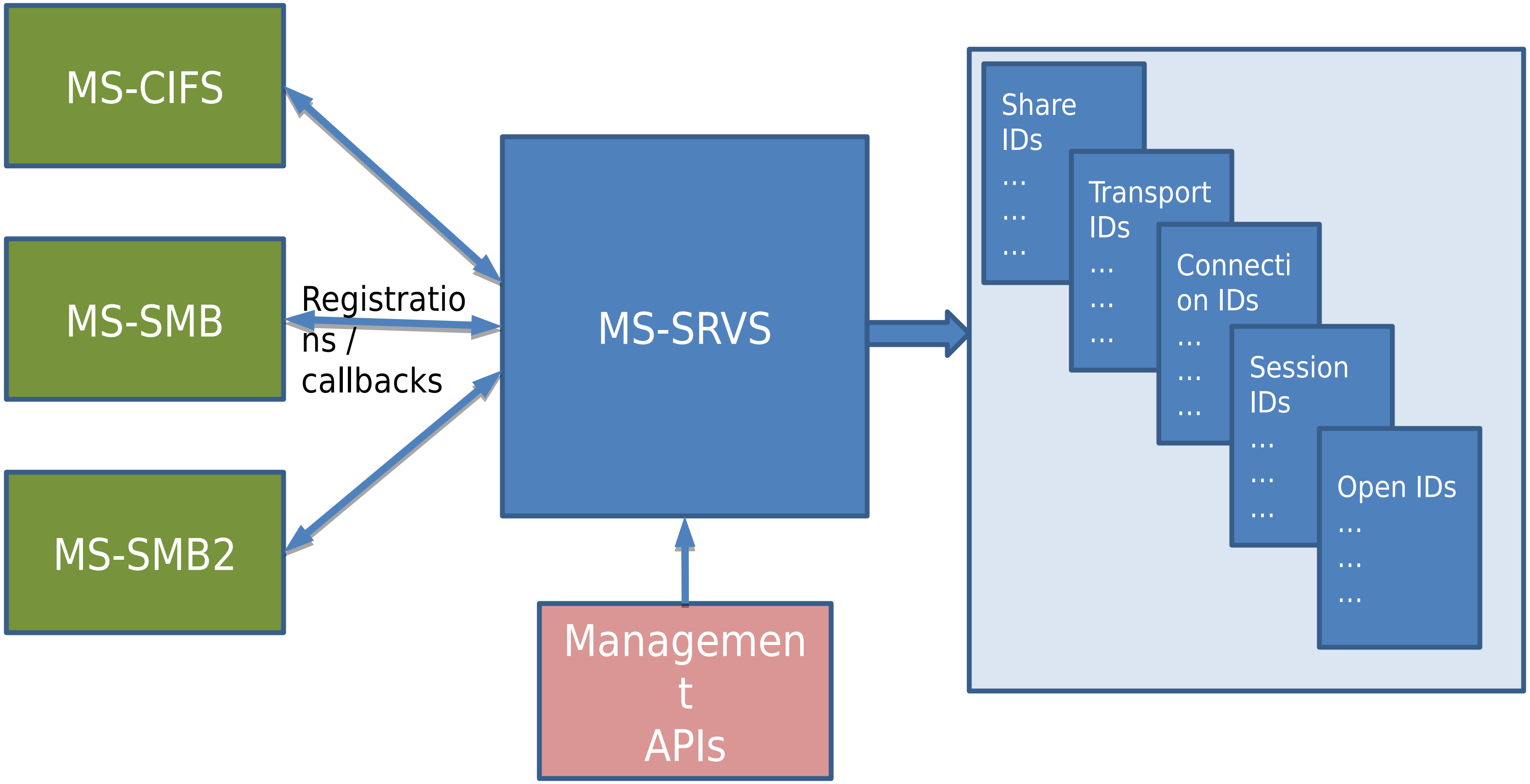
  …

  In response to the **NetrSessionEnum** message, the server MUST return information about sessions that are established on the server contained in **SessionList**, or return an error code.

“Managed State”

- Moved to abstract common registry of objects
- Objects allocated and owned by protocols
  - E.g. session = session_create(); id = register(session);
- Identifiers assigned and registered by exported MS-SRVS interface
  - E.g. id = &object; insert(list, id); return id;
- Management, query, destroy etc. all by id

Managed State - new



Managed state approach – SMB2 session

- MS-SMB2:

  3.3.1.5 Global Structures

  § **GlobalSessionTable**: A list of all the active sessions established to this server, indexed by the **Session.SessionId**.

  …

  3.3.5.5.1 Authenticating a New Session

  A session object MUST be allocated for this request. The session MUST be inserted into the **GlobalSessionTable** and a unique **Session.SessionId** is assigned to serve as a lookup key in the table. The server MUST register the session by invoking the event specified in [MS-SRVS] section 3.1.6.2 and assign the return value to **Session.SessionGlobalId**.

- MS-SRVS:

  3.1.6.2 Server Registers a New Session

  The CIFS or SMB2 server requesting registration of a Session provides no parameters. The server MUST insert a new Session into **SessionList**, and MUST assign *Session.GlobalSessionId* the value that uniquely identifies the entry in the list. This value MUST be returned to the caller.

  3.1.6.3 Server Deregisters a Session

  The CIFS or SMB2 server MUST provide the SessionId of the Session that is being deregistered.

  The SRVS server MUST look up the Session in **SessionList** where Session.GlobalSessionId is equal to the SessionId provided by the caller, and remove it from **SessionList**.

  3.1.4.5 NetrSessionEnum (Opnum 12)

  In response to the **NetrSessionEnum** message, the SRVS server MUST enumerate the **Session** entries in **SessionList** based on the value of the *ResumeHandle* parameter. For each entry, the server MUST query session properties by invoking the underlying server events as specified in [MS-CIFS] section 3.3.4.14 and [MS-SMB2] section 3.3.4.18, providing *Session.GlobalSessionId* as the input parameter.

MS-FSA

- Possibly your most useful document for understanding the Windows filesystem behaviors which underly filesharing services:

  > **1 Introduction**
  >
  > This document defines an abstract model for how an object store can be implemented to support the Common Internet File System (CIFS) Protocol, the Server Message Block (SMB) Protocol and the Server Message Block (SMB) Version 2 Protocol (described in [MS-CIFS], [MS-SMB] and [MS-SMB2], respectively).

-

FSA Relevance to SMB

- Error codes
- Algorithms
  - Oplocks, leases
  - Locking and lock conflicts
  - Change-notify
  - Wildcarding and matching
  - Access checking
  - etc

Use of MS-FSA – SMB2

- **Inline description style:**

  3.3.4.6 Object Store Indicates an Oplock Break

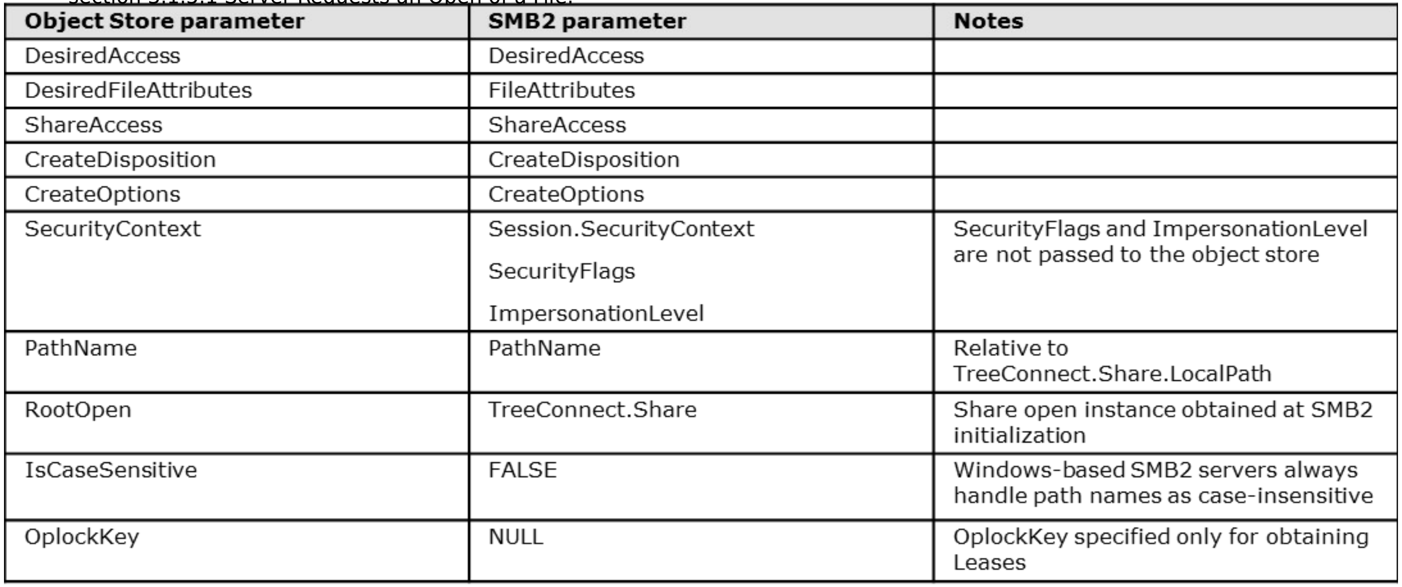
  The underlying object store on the local resource indicates the breaking of an opportunistic lock, specifying the **LocalOpen** and the new oplock level, a status code of the oplock break, and optionally expects the new oplock level in return. The new oplock level MUST be either SMB2_OPLOCK_LEVEL_NONE or SMB2_OPLOCK_LEVEL_II. The conditions under which each oplock level is to be indicated are described in [MS-FSA] section 3.1.5.17.3.

- **Behavior note style:**

  3.3.5.9 Receiving an SMB2 CREATE Request

  The server MUST use the security context of the session in **Session.SecurityContext** to attempt to open the named object in the underlying object store using the parameters specified for **DesiredAccess**, **FileAttributes**, **ShareAccess**, **CreateDisposition**, **CreateOptions**, and the **PathName**. The **PathName** MUST be parsed relative to **TreeConnect.Share.LocalPath**. The server MUST map these flags to match the semantics of its implementation-specific object store [MS-FSA].<186>

  <186> Section 3.3.5.9: Windows performs the following open/create mappings from SMB2 parameters to the object store as described in [MS-FSA] section 3.1.5.1 Server Requests an Open of a File.



Object Store parameter	SMB2 parameter	Notes
DesiredAccess	DesiredAccess	
DesiredFileAttributes	FileAttributes	
ShareAccess	ShareAccess	
CreateDisposition	CreateDisposition	
CreateOptions	CreateOptions	
SecurityContext	Session.SecurityContext SecurityFlags ImpersonationLevel	SecurityFlags and ImpersonationLevel are not passed to the object store
PathName	PathName	Relative to TreeConnect.Share.LocalPath
RootOpen	TreeConnect.Share	Share open instance obtained at SMB2 initialization
IsCaseSensitive	FALSE	Windows-based SMB2 servers always handle path names as case-insensitive
OplockKey	NULL	OplockKey specified only for obtaining Leases

Next steps

- Continued refinements and changed from
  - Peer review
  - Real-world testing and plugfests
  - Formal protocol testing
  - New protocol features
- The process continues…
  - And your feedback/participation is key!

Questions?

Tom Talpey: ttalpey@microsoft.com