Implementing NFS on the Windows Platform

Jeff Biseda

Senior Software Development Engineer
Microsoft Corporation

Agenda

- History of NFS in Windows
- Architecture/Design
- Interop Issues for NFS Server
- Interop Issues for NFS Client
- Identity Mapping
- Management

Windows Server 2003 R2 (2005)

- "Services for Unix" integrated into Windows
- NFS Client/Server, Username Mapping Server
- Gateway for NFS discontinued
- 64-bit support
- New Management Interface

Windows Server 2008

- Active Directory Lookup (ID mapping with LDAP)
- IPv6 Support
- Username Mapping Server discontinued

Windows Server 2008 R2 (2010)

- RPCSEC_GSS support
  - Kerberos v5 Authentication (Krb5)
  - Kerberos v5 Authentication + Integrity (Krb5i)
- Unmapped Unix User Access (U3A)
- Netgroup support
- WMI Provider

NFSv4.1 Client for Windows

- Microsoft funded research project
- NFSv4.1 & pNFS capable standalone Windows client
- Center for Information Technology Integration (CITI), University of Michigan
- Beta version released 9/3/2010
- http://www.citi.umich.edu/projects/nfsv4/windows/
- Microsoft is now actively participating in the NFSv4 Working group

Agenda

- History of NFS in Windows
- **Architecture/Design**
- Interop Issues for NFS Server
- Interop Issues for NFS Client
- Identity Mapping
- Management

General Challenges

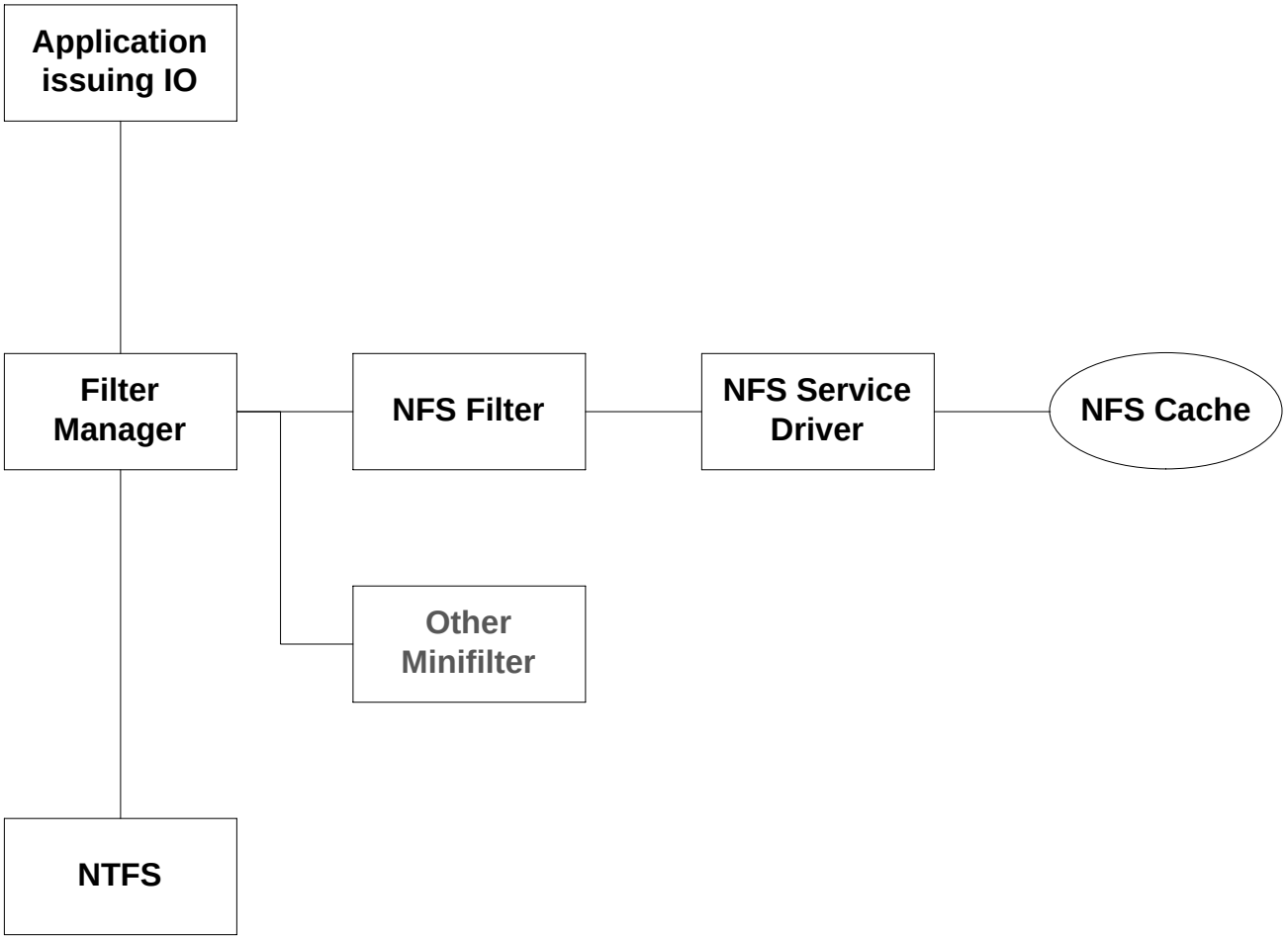
- Stateless versus stateful
- Caching and coordination with other servers or local filesystem access
- Identity mapping

Stateless vs Stateful

- NFSv2/NFSv3 are stateless protocols.
- Local filesystem semantics are stateful and open/close are expensive operations.
- NFS Server caches metadata for files both with and without open filesystem handles.

How do we maintain consistency with parallel access over SMB/local filesystem access?

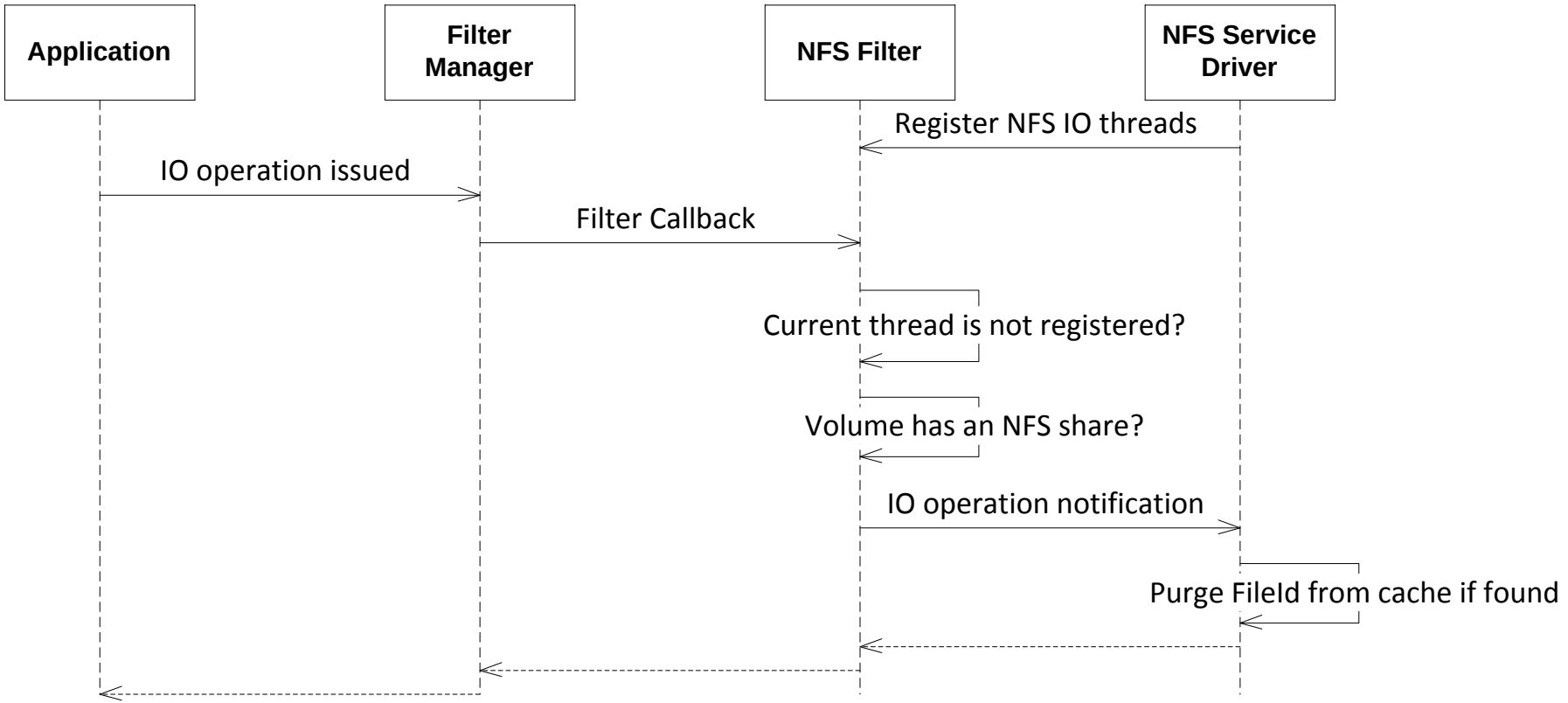
NFS Filter Model



NFS Filter driver

- NFS Filter driver watches for all create, open or modifying operation (write, set attributes, etc.) on volumes where there are NFS shares.
- If there is a collision for a given file that the NFS Server has cached it attempts to flush any outstanding data and close the handle. Lock/share reservations will prevent this.
- The IO operation proceeds

Filter sequence



Persistent handles

- Because NFS handles are persistent the Server encodes the volume ID and file ID in the NFS handle.
- FILE_OPEN_BY_FILE_ID is used to open files.
- Open by File Id precludes directory change notifications.
- A reopen file by name option is provided although expected behavior is not guaranteed (hard links).

Agenda

- History of NFS in Windows
- Architecture/Design
- **Interop Issues for NFS Server**
- Interop Issues for NFS Client
- Identity Mapping
- Management

Server Interop

- Windows OS challenges
- NTFS challenges

NFS Server/Windows OS

Windows is case insensitive by default, most NFS Servers support case sensitivity. Change with registry key:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Kernel

Value Name: ObCaseInsensitive

Data Type: REG_DWORD

Value Data: 0 (Case sensitive)

Server filesystem requirements

Requirements from (FileFsAttributeInformation)

- FILE_UNICODE_ON_DISK
- FILE_PERSISTENT_ACLS
- FILE_SUPPORTS_EXTENDED_ATTRIBUTES
- FILE_SUPPORTS_OPEN_BY_FILE_ID

Also require FileHardLinkInformation for export path security checking.

FS semantics

- Directory rename when child objects are open is not supported
- Rename/delete semantics differ from Unix, rename/delete of a file when other users have the file open does not immediately change the namespace.
- Atomic pre/post attributes not supported. NFS Server provides best effort to obtain attributes.

FS semantics continued

- Directory and file attributes are updated asynchronously. A READDIRPLUS of the parent directory may not have updated attributes of modified children.
- Not possible to commit a range of data, all outstanding data must be flushed.

READDIR cookie collisions

- ZwQueryDirectoryFile() doesn't return cookie values
- NFS Server generates cookie values based on File ID and File name string
- Collisions can occur with many hard links with similar names in the same directory
- NFSv2 up to 32-bit cookies
- NFSv3 up to 64-bit cookies

READDIR continued

- Many apps don't handle using the full range of 32/64-bits well.
- 32-bit applications may be using signed 32-bit integers and only handle 31-bits (highest order bit must be 0)
- Cookie size can be changed to minimize collisions while maximizing application compatibility. By default the NFS Server uses only 31 bits for both NFSv2 and NFSv3.

Cluster support

- NFS Server registers a single endpoint for all interfaces on the machine
- When using UDP reply messages may not be sent on the same interface on which the request message was received
- FreeBSD has “mount_nfs -c” option to support this

Other notes

- MKNOD calls for device types are supported
- An option is provided to enable ACL inheritance for new files for better interoperability with SMB. NFS Server propagates the ACL of the parent directory and adds to it a user SID, group SID, and world SID.
- If the NFS Server is not domain joined there is no way to set the primary group of a Windows account. The server works around this by using the gidNumber attribute.

Agenda

- History of NFS in Windows
- Architecture/Design
- Interop Issues for NFS Server
- **Interop Issues for NFS Client**
- Identity Mapping
- Management

Client Interop Challenges

- Windows share vs Unix export semantics
- Directory change notifications
- Case sensitivity

Windows share vs Unix export

Unix export semantics allow for exports to be multi-component paths (/users/home/a) while Windows WNet* APIs and RDBSS APIs support only single component paths.

The NFS client solves this problem for mapped drive letters by claiming the first component of the path as the share and simulating attributes for intermediate directories.

Share vs export continued

Unix allows the concept of exporting the root directory "/" while Windows share names must be non-zero length. The Windows client supports this by mapping "/" to "\!".

"nfsserver:/" == "\\nfsserver\!"

Directory Change Notifications

NFSv2 and NFSv3 have no mechanism to inform a client of a change to a directory of interest.

The Windows NFS client supports only directory change notifications originating from the client machine itself.

Case-insensitive lookup

Win32 applications expect case insensitive behavior.

NFS client will issue READDIR requests and search for case insensitive matches in the directory when attempting to lookup a file.

This will have performance implications and can be disabled. Case preserving lookup will be used.

UNC access

- Mounts are performed on a per-session basis
- Mapped drive letters are per-session, there is no machine global mount
- Services should use UNC access
- Multiple redirectors are called to try to claim the path
- Provider order can be changed to increase performance for a specific provider

Agenda

- History of NFS in Windows
- Architecture/Design
- Interop Issues for NFS Server
- Interop Issues for NFS Client
- **Identity Mapping**
- Management

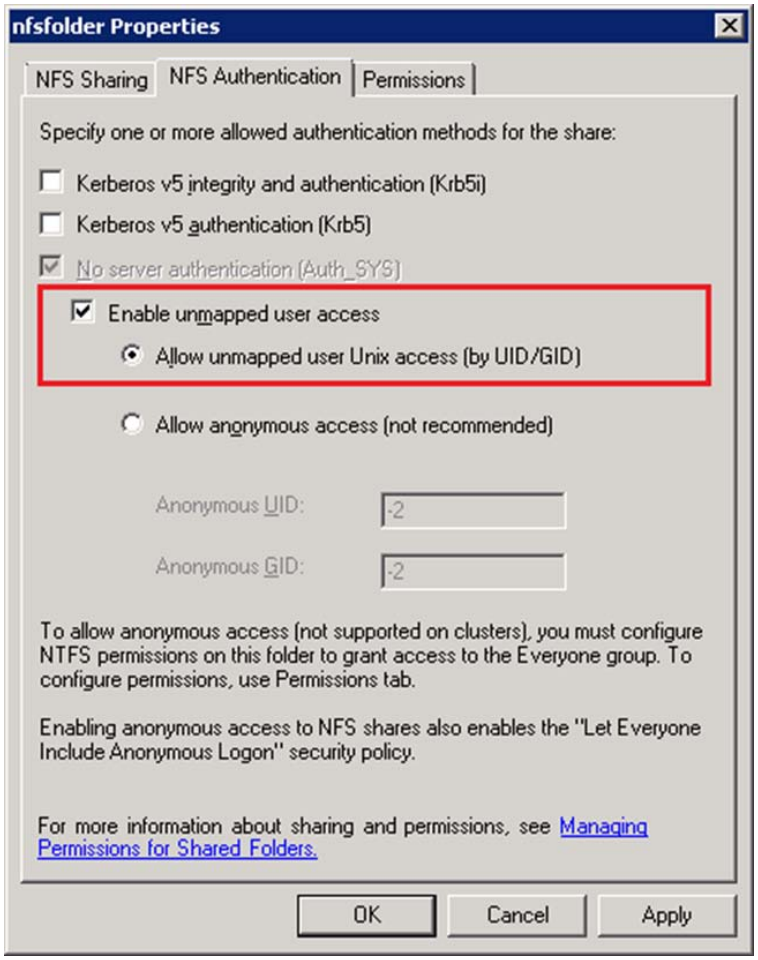
Identity Mapping

- Username Mapping Server (deprecated in Windows 2003 R2)
- Active Directory Lookup
- ADLDS (Active Directory Lightweight Directory Services)
- U3A (Unmapped Unix User Access)
- RPCSEC_GSS Support

AD Lookup

- Uses RFC 2307: An Approach for Using LDAP as a Network Information Service
- uidNumber and gidNumber attributes are set on a per user basis
- gidNumber attribute is set on group objects
- Supplementary GID support is not defined by the RFC

U3A (Unmapping UNIX User Access)



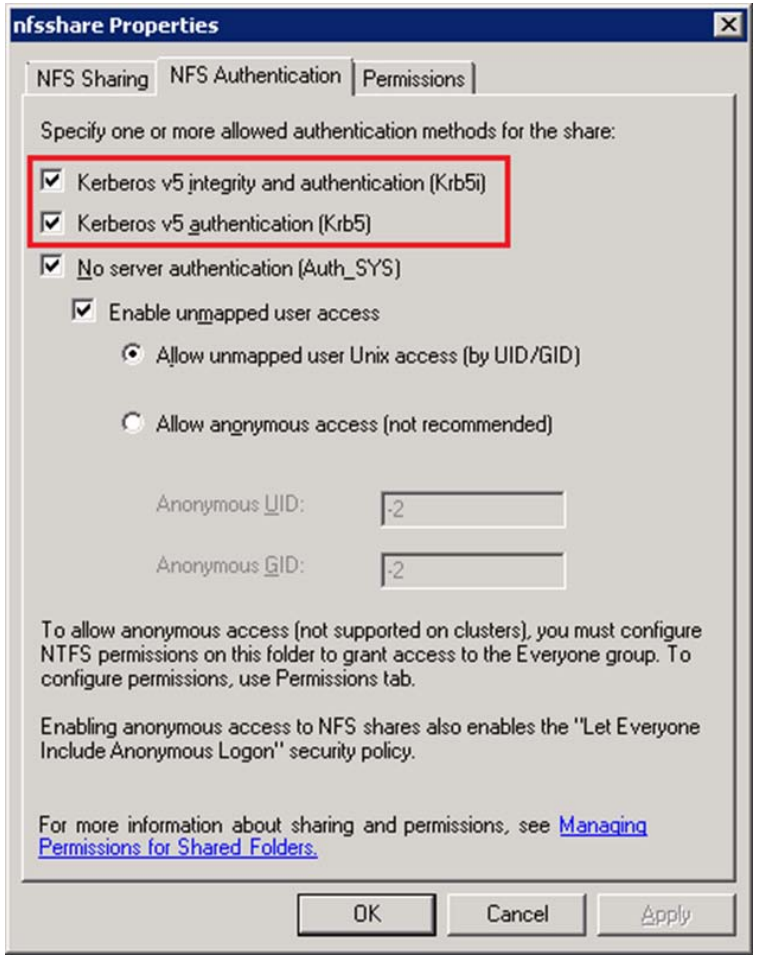
- If this setting is enabled, Server for NFS will generate custom SIDs for UNIX users whose identities are not mapped (both UIDs & GIDs)
- Custom SIDs are placed in owner & group fields (as appropriate) and used in the ACEs for owner & group permissions – for files created by unmapped users.
- Easy to configure – toggle setting while sharing the folder using 'Share and Storage Management' snap-in.

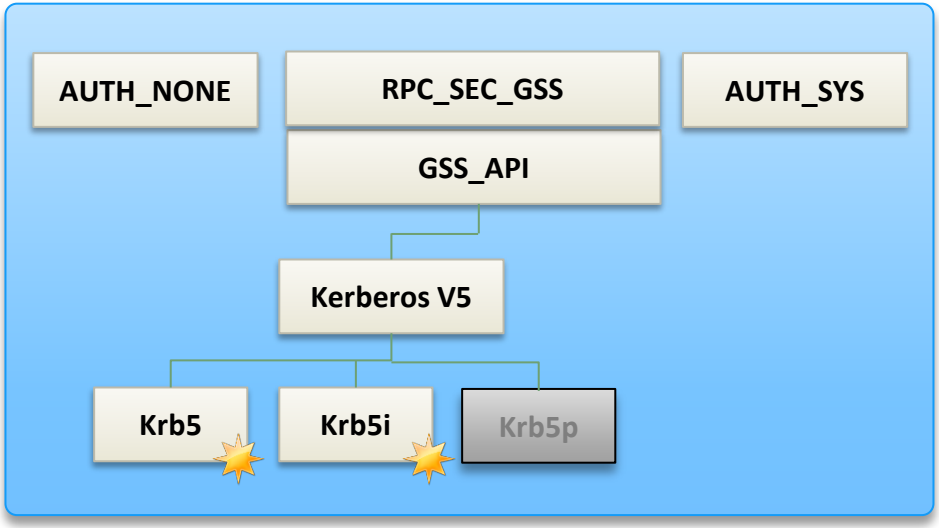
U3A SID Construction

- "<NTSecurityAuthority>-<SECURITY_NFS_ID_BASE_RID>-<NfsSidType>-<NfsSidValue>"
- Owner SID for UID: "S-1-5-88-1-<uid>"
- Group SID for GID: "S-1-5-88-2-<gid>"
- Mode SID: "S-1-5-88-3-<mode>"

http://msdn.microsoft.com/en-us/library/ee380665(PROT.10).aspx

RPCSEC_GSS Support





- RPCSEC_GSS flavors supported:
  - Kerberos v5 Authentication (Krb5)
  - Kerberos v5 Authentication and Integrity (Krb5i)
- Can be configured on a per-share basis.
- AUTH_SYS and AUTH_NONE continue to function as before.

Kerberos Configuration

- Keys must be exported from AD to Unix clients
- Supported encryption types must match between clients, servers, and AD (arcfour-hmac-md5, des-cbc-md5, etc)
- SPN's (Service Principal Names) must be mapped to the proper accounts

Kerberos security still requires ID mapping

When Kerberos security is being used a UID/GID to Windows User/Group mapping is still required.

Although the RPC header specifies only the Kerberos principal being used, NFSv2 and NFSv3 still provide only for UID/GID fields in GETATTR, SETATTR, post-op attributes, etc.

RPCSEC_GSS isn't used for the NLM protocol so locking requests must match IO operations. UID and Kerberos principal must be mapped.

Challenges

- Non-domain joined Windows machines have no "Primary group" for a user's token.
- RFC 2307 (LDAP ID Mapping) has no provision for supplementary GID's
- SE_RESTORE_PRIVILEGE is required to change ownership (chown)

Agenda

- History of NFS in Windows
- Architecture/Design
- Interop Issues for NFS Server
- Interop Issues for NFS Client
- Identity Mapping
- **Management**

NFS WMI Provider

- WMI namespace – 'root\msnfs'
- Available WMI classes:
  - MSNFS_Server
  - MSNFS_Client
  - MSNFS_UserNameMapping
  - MSNFS_ClientGroup
  - MSNFS_NetGroup
  - MSNFS_ClientLock
  - MSNFS_Export
  - MSNFS_ExportFencing
- Enables remote management of NFS server and client.

Questions?

Appendix

NFS product behavior notes
http://msdn.microsoft.com/en-us/library/ee380665(PROT.10).aspx

RFC 2203: RPCSEC_GSS Protocol Specification
http://www.ietf.org/rfc/rfc2203.txt

RFC 2307: An Approach for Using LDAP as a Network Information Service
http://www.ietf.org/rfc/rfc2307.txt

RPCSEC_GSS with Kerberos configuration guide
http://blogs.technet.com/b/filecab/archive/2010/05/13/using-kerberos-security-with-server-for-nfs.aspx

NFS Server ACL construction
http://technet.microsoft.com/en-us/library/bb463216.aspx

Contact: nfsfeed (at) microsoft.com