Network File System (NFS) Overview in Windows Server

By robinharwood

Archived: 2026-04-06 01:22:27 UTC

Network File System (NFS) is a distributed file system protocol available in Windows Server that enables file sharing between Windows and non-Windows systems. It's based on the protocol specification RFC 1094.

In Windows Server, NFS is implemented as part of the File and Storage Services role, providing robust interoperability between Windows-based systems and non-Windows platforms such as Linux and UNIX. This cross-platform compatibility is beneficial for organizations operating mixed IT environments, where seamless integration and data sharing between different operating systems are essential.

Windows Server's NFS implementation includes both **Server for NFS** and **Client for NFS** components. Server for NFS enables Windows Server to act as a file server for Windows and non-Windows clients, allowing them to access shared files and directories using the NFS protocol. Client for NFS allows Windows-based systems to access files stored on NFS servers.

This article explains supported NFS versions, practical scenarios, management tools, and identity mapping and authentication.

NFS is available in all supported versions of Windows Server and Windows client operating systems. The following table summarizes the supported NFS protocol versions and their corresponding Windows operating systems.

Operating system	Server for NFS	Client for NFS
Windows Server (all currently supported versions)	NFSv2, NFSv3, and NFSv4.1	NFSv2 and NFSv3
Windows client (all currently supported versions)	N/A	NFSv2 and NFSv3

Here are some practical scenarios where Network File System (NFS) can be effectively utilized within your organization:

- **Multi-protocol file sharing**: Deploy a Windows Server configured as an NFS file server to provide simultaneous access to shared files and directories using both SMB (Server Message Block) and NFS protocols. This allows seamless collaboration between Windows-based clients and non-Windows clients, such as Linux and UNIX systems, enabling users across different platforms to access and modify shared resources without compatibility issues.
- **Cross-platform file access in mixed environments**: In environments predominantly using non-Windows operating systems, such as Linux or UNIX, a Windows-based NFS file server can provide reliable and efficient file sharing capabilities. This setup allows non-Windows client computers to easily access, store,

and manage data on Windows-hosted NFS shares, simplifying data management and improving interoperability across diverse IT infrastructures.

- **Provision file shares in UNIX-based environments**: Deploy Windows file servers in predominantly UNIX-based environments to provide NFS file shares for UNIX-based clients. Use the Unmapped UNIX User Access (UUUA) option to simplify NFS deployment without requiring UNIX-to-Windows account mapping. UUUA creates custom security identifiers (SIDs) for unmapped users while using standard Windows SIDs for mapped accounts, enabling quick provisioning and efficient management of NFS shares.

- **Simplified application migration**: Facilitate the migration of applications and workloads between different operating systems by using NFS file shares accessible through both SMB and NFS protocols. By storing application data on shared file systems, organizations can smoothly transition applications from one platform to another without extensive downtime or complex data migration processes, significantly reducing the complexity and risk associated with cross-platform migrations.

- **Centralized data management and backup**: Utilize NFS to centralize data storage and simplify backup and recovery processes. By consolidating data from multiple operating systems onto a single Windows-based NFS file server, organizations can streamline data management, enhance data protection strategies, and reduce administrative overhead associated with managing separate storage solutions for different platforms.

- **High-performance computing (HPC) and research environments**: Deploy NFS in HPC clusters or research environments where multiple computing nodes require rapid and concurrent access to shared datasets. Windows Server's NFS implementation provides efficient data access and improved performance, enabling researchers and engineers to collaborate effectively and accelerate computational workloads.

- **Virtualization and container environments**: Use NFS file shares as persistent storage solutions for virtualization platforms and container orchestration systems, such as VMware, Hyper-V, Kubernetes, or Docker. NFS enables virtual machines and containers running on various operating systems to access shared storage resources seamlessly, simplifying storage provisioning, and management in dynamic, multi-platform environments.

Windows Server provides graphical and command line tools and methods for effectively managing and utilizing both Server for NFS and Client for NFS components. Administrators can use:

- **Services for Network File System MMC snap-in**: This snap-in allows administrators to manage NFS shares, configure authentication methods, set permissions, and monitor active connections. It provides a centralized graphical interface for managing both Server for NFS and Client for NFS components.

- **Windows PowerShell cmdlets**: A comprehensive set of PowerShell cmdlets is available for managing NFS shares, configuring identity mapping, and monitoring NFS operations. These cmdlets enable administrators to automate tasks and streamline management processes.

- **Windows command-line tools**: Several command-line utilities are available for managing NFS shares and monitoring NFS operations.

Identity mapping is crucial for ensuring proper access control and permissions management between Windows and non-Windows systems. Windows Server supports multiple identity mapping methods, including Active Directory, Active Directory Lightweight Directory Services (AD LDS), and local flat files. Administrators can configure identity mapping using graphical tools, command-line utilities, or PowerShell cmdlets.

Authentication methods supported by Server for NFS include:

- **Anonymous authentication**: Allows access without explicit user authentication.
- **AUTH_SYS (UNIX-style authentication)**: Uses UID and GID for authentication.
- **Kerberos authentication**: Provides secure authentication using Kerberos v5, including `krb5`, `krb5i` (integrity), and `krb5p` (privacy).

By using these management tools and authentication methods, administrators can effectively deploy, configure, and maintain robust NFS solutions within Windows Server environments.

To perform identity mapping, you need to deploy one of the following:

- A Windows domain controller running Active Directory Domain Services (AD DS) and a User Name Mapping service. The User Name Mapping service is installed as part of Server for NFS.
- A mapping file that contains the identity mapping information. The mapping file is stored on the computer that's running Server for NFS.
- An RFC 2307-compliant LDAP store, such as Active Directory Lightweight Directory Services (AD LDS), that contains the identity mapping information. The LDAP store is stored on the computer that's running Server for NFS.
- A User Name Mapping service that uses a password file and a group file. These files are stored on the computer that's running the User Name Mapping service.

To learn more about identity mapping, see NFS Identity Mapping in Windows Server.

Learn how to Deploy Network File System (NFS).
