



ReadMe for NW3DFS.EXE

Restriction Status: Open

| Name | Size | |
|-----------------------|-----------------|----------------------------|
| license_agreement.txt | 2.1 KB (2252) | → download |
| nw3dfs.exe | 28.1 KB (28798) | → download |
| readme_2916105.html | 9.3 KB (9616) | → download |

| Related Product(s) | Patch Status for Product | Superseded By |
|--------------------|--------------------------|---------------|
| NetWare 3.11 | Active | |
| NetWare 3.12 | Active | |
| NetWare 3.0 | Active | |
| Localizations | | |
| English | | |

Download Manager Links

→ [how to use a download manager](#)

[patchfiles/nw3dfs.exe](#)
[license_agreement.txt](#)

Reminder: By downloading this product, you reaffirm your [agreement](#) to comply with the export laws of the United States and those of other countries.

MD5 Verification Checksums:

MD5 checksum values are used to verify the data integrity of a downloaded file by comparing it to the checksum value of the original file. Linux/UNIX and Mac devices have this capability built-in; for Windows, several free utilities are available on the web to help you determine the checksum value of your downloaded files.

license_agreement.txt 7977d223b2d1eea336a0385e025ce769
nw3dfs.exe f19019b253302a7a51c6fc8eb5c19843
readme_2916105.html

This document (2916105) is provided subject to the [disclaimer](#) at the end of this document.

Associated File

Click the filename to download: [nw3dfs.exe](#) 28798 bytes 07Jan1997 05:01PM

Abstract

Released patch of the NetWare Direct File System.

Installation Instructions

1. You need to have previously installed a NetWare 3.1x patch kit (312PT9.EXE or 311PTF.EXE) which includes the patch manager (PM312.NLM or PATCHMAN.NLM). The patch manager must be loaded in order to add NetWare server operating system patches.
2. Copy the DIRECTFS.NLM file into the the SYSTEM directory on the servers SYS volume (SYS:SYSTEM).
3. Edit the AUTOEXEC.NCF file to include the loading of the DIRECTFS.NLM (Load DIRECTFS.NLM)
4. Down and restart the file server.

Issue

Fix #1

Problem: A problem occurs when a DFS file size is increased and the newly acquired block was recovered from a recently deleted file still being represented in the NetWare disk cache and is marked as dirty (or needs to be written to disk), and the cache block is written to by DFS before the cache subsystem performs its periodic update. The effect is that if the newly expanded DFS file area is written to and then read back, the read data will not match the written data.

Detail: Opening a file in DFS mode evacuates that file from NetWare cache. New disk blocks are acquired as the DFS file size is increased. New disk blocks are acquired from a list of free blocks. The disk blocks on the free block list either never been assigned to any file, or they are recovered from deleted files. Blocks from recently deleted files may still be represented in cache.

Solution: This has been fixed by having DFS verify that newly acquired blocks, obtained while expanding a file, are evacuated from the NetWare disk cache.

Fix #2

Problem: A problem occurs when a DFS file size is increased and written to in a non-sequential mode, causing the formation of a sparse file. The effect is that data previously written to the DFS file might be zeroed out by subsequent writes to adjacent blocks.

Detail: It is possible to write to a file in DFS mode in a manner that causes the file to become a NetWare sparse file (either by creating a new file or extending an existing file). As a security measure, the newly acquired disk blocks are zeroed out before use.

Fixed: This problem has been fixed by changing the method of zeroing out the newly acquired disk blocks.

Technology Background:

NetWare Disk Caching Subsystem Description

NetWare's disk caching subsystem technology decreases data access time for frequently accessed disk data. The technology involves storing a representation of disk in the (RAM) memory of the NetWare file server.

When a client reads and writes files that are stored on a NetWare file server, the data does not interface directly with the server's disk storage hardware, it is accessed in the file server's memory cache (RAM). NetWare periodically updates the disk storage hardware as needed.

NetWare Direct File System

The Direct File System (DFS) allows an application to bypass the NetWare Disk Cache and Transaction Tracking Systems. DFS is necessary for some database applications that provide their own caching and transaction tracking facilities.

Backup applications often access large amounts of data not being accessed by other applications. If these accesses are made through the cache, the cache becomes non-relevant for all other accesses, and general server performance suffers from other users.

Some utilities may also require the ability to specify exactly where files are placed, so that volumes can be de-fragmented and optimal file access can be made.

NetWare does not allow a file to be opened in both the cache and DFS modes simultaneously. DFS file open requests can only be made by an NLM application, and fails if the specified file is already open by any other (NLM or NetWare client) entity. While a particular file is opened in DFS mode, NetWare will deny all other open requests (for both DFS and cache modes) for that file.

Files opened in DFS mode must not have any portion represented in NetWare cache. Therefore, as the file is opened in DFS mode, NetWare searches all cache for blocks that belong to the file. If a cache block belonging to the file being opened is encountered, the cache block content is written to the server's disk storage hardware (if required), and the cache block is disassociated from the file being opened. This process may take considerable time, depending on the size of the file being opened in DFS mode, and the total number of cache blocks in server RAM.

NetWare Sparse Files

The NetWare file system allows the creation of what Novell calls "sparse files." A sparse file is a file which has "holes" or areas which have not been yet been written to. A sparse file is formed by opening a new file writing to it non-sequentially.

An example of a sparse file is one where the file is opened, and data is written to the file starting at offset 1000 and then closing the file. In this state, the file offsets 1 through 999 have not been written to and represents a hole in the data of the file. If an attempt is made to read from offset 1 through 999, NetWare returns zeros as the data.

NetWare only allocates disk blocks to portions of a sparse file that have been written to. NetWare does not allocate disk blocks to a sparse file's holes.

File Contents

Self-Extracting File Name: nw3dfs.exe

| Files Included | Size | Date | Time | Version | Checksum |
|----------------|-------|-----------|---------|---------|----------|
| \ DIRECTFS.NLM | 18192 | 18Oct1996 | 10:20PM | | |
| NW3DFS.TXT | 6664 | 07Jan1997 | 05:01PM | | |

document

Document Title: ReadMe for NW3DFS.EXE

Document ID: 2916105

Creation Date: 31Oct1996

Modified Date: 07Jan1997

Revision: 4

Novell Product Class: NetWare

disclaimer

The Origin of this information may be internal or external to Novell. Novell makes all reasonable efforts to verify this information. However, the information provided in this document is for your information only. Novell makes no explicit or implied claims to the validity of this information. Any trademarks referenced in this document are the property of their respective owners. Consult your product manuals for complete trademark information.