Windows Hardware and Driver Central

Local File Systems for Windows

WinHEC 2004 Version—May 5, 2004

Abstract

This paper provides information about local file systems for the Microsoft® Windows® family of operating systems. It provides guidelines to help system manufacturers and server administrators choose the local file system that is best suited for a particular use and media type. This information applies for the following operating systems:

Microsoft Windows 98
Microsoft Windows 2000
Microsoft Windows XP
Microsoft Windows Server™ 2003
Microsoft Windows Vista™

The current version of this paper is maintained on the Web at:
http://www.microsoft.com/whdc/device/storage/default.mspx

Contents

Disclaimer

1 Introduction

This paper describes the features, benefits, and architectural limits of three local file systems currently supported on the Microsoft® Windows® 98, Microsoft Windows XP, Microsoft Windows 2000, Microsoft Windows Server™ 2003, and Microsoft Windows Vista platforms:

- File Allocation Table (FAT)
- NTFS
- Universal Disk Format (UDF)

There are many important factors that must be considered when selecting a local file system format. This paper is intended to provide system manufacturers and server administrators with the necessary information to choose the local file system that is best suited for a particular usage scenario and media type.

This paper assumes basic knowledge of local file systems. For a description of the terminology used in this paper, see Appendix A—Glossary.

This paper includes the following sections:

- **Local File System Overview:** Provides an overview for each of the three local file systems and describes usage scenarios for each type of local file system.
- **Local File System Limitations:** Provides information about the limits imposed by the on-disk structures and those imposed by the Windows file system driver implementation for each of the file systems.

2 Local File System Overview

Over the past 25 years, as PCs have become more powerful and storage media, such as hard disk drives, have become exponentially larger, the demand for local file systems that store larger file sizes and quantities of files has emerged.

At the same time, there is an increasing demand for removable media, such as Flash and CD/DVD media. These media types are being used on PCs, as well as consumer electronic devices, to provide mobility of data.

As PCs have moved into the enterprise space that serves a variety of purposes, including file share servers, mail servers, and database servers, the need has also arisen for robust local file systems with advanced features, such as recoverability, availability, security, and performance.

These demands have made it necessary to choose the appropriate file system that meets the unique requirements of each of the usage scenarios.

This section provides history and feature overview information about the NTFS, FAT, and UDF file systems and describes the media on which each of these three local file systems is best suited.

2.1 About the FAT File System

The FAT file system originated in the late 1970s and early1980s and was the file system supported by the Microsoft MS-DOS® operating system. It was originally developed as a simple file system suitable for floppy disk drives less than 500 KB in size. Over time it has been enhanced to support larger and larger media. Currently there are three FAT file system types: FAT12, FAT16, and FAT32. The basic difference in these FAT sub types, and the reason for the names, is the size (in bits) of the FAT entries in the actual FAT structure on the disk. The following table shows the number of bits used in the FAT entry for each FAT file system type and what each type is used for in Windows.

Table 2—UDF versions supported by Windows

FAT type	Bits in the FAT entry	Uses
FAT12	12	Current versions of Windows use FAT12 for floppy disk media and removable media.
FAT16	16	Windows supports FAT16 for hard disk volumes and removable media.
FAT32	32	Windows supports FAT32 for hard disk volumes and removable media.

For information about the size limits for FAT16 and FAT32 hard disk volumes, see the “Local File System Limitations” section of this paper.

2.1.1 Media Suitable for FAT

The FAT file system is ideal for removable Flash media that is used in consumer electronic devices, such as digital cameras, media players, Flash drives, and so on. FAT12 is, of course, the only file system that can be used on floppy disks.

Note: The FAT file system is ideal for use with devices that are optimized for quick media removal.

2.1.1.1 What Makes FAT Suitable for These Media

The following list provides reasons why the FAT file system is suitable for devices like memory sticks and floppy disk drives:

- **Consumer electronics device friendly:** FAT is a light-weight file system that is simple and easy to implement.
- **Backward compatibility:** FAT 12 and FAT 16 are recognized by all operating systems from MS-DOS forward. FAT 32 is recognized by all operating systems from Windows 98 forward. This backward compatibility enables the use of memory stick media or floppy disks to transfer files between a consumer electronics device and a PC that is running an out-dated operating system.
- **No unnecessary features:** FAT is not a journaled file system and is not as reliable as NTFS. However, because these features are not a design goal for removable media, the FAT file system is adequate for this purpose. FAT also has no means of specifying security access rights and privileges. This is not a design goal for removable media where the user typically has physical access to the media. For more information about the journaling feature, see the “Local File System Feature Comparison” section later in this paper.

2.1.1.2 Scenarios for using FAT on Hard Drive Volumes

For some usage scenarios, FAT16 or FAT32 may be suitable for a hard disk drive volume. One scenario is when the operating system, for example Windows 98, does not provide support for NTFS. Another scenario is when the user wants to boot the PC by using a floppy disk to access data (typically some system recovery tools) on a hard drive volume.

Because a FAT volume does not provide the benefits of NTFS, careful consideration should be taken to determine whether the disadvantages of using the FAT file system outweigh the need to use FAT on a hard drive volume.

2.1.1.3 Issues to Understand When Choosing FAT

The following list describes the most important issues to understand when choosing the FAT file system for the usage scenarios described in the previous section:

- **Scalability:** The largest volume size supported by FAT32 on-disk structures is 2 terabytes. Windows 2000 and later versions further limit the size of a volume that can be formatted with FAT32 to 32 GB. Because the maximum size of a file on FAT32 is limited to 32 bits, it cannot be used to store files larger than 4 GB.
- **Data transfer limitations:** FAT does not provide support for alternate data streams, other than the default data stream. Copying data from an NTFS or UDF volume, which supports alternate data streams, to a FAT volume can cause data that is stored in the named streams to be lost. This makes FAT unsuitable for transferring data from an NTFS or UDF volume that contains named steams.
- **No incremental-write support:** FAT requires its metadata structures to be overwritten in place during updates to the metadata. Because of this, FAT cannot be used on incremental-write removable media, such as CD-R and DVD-R.
- **No defect management support:** FAT does not provide a mechanism for defect management at the file system level. The FAT file system expects the underlying hardware to manage and remap defective blocks on the media. It does, however, support marking some allocation units as "bad" to prevent their use by the file system. FAT cannot be used on removable rewritable media, such as CD-RW, DVD-RW, and DVD+RW. This is because these media types do not support defect management in the hardware layer.
- **Performance and recoverability:** FAT does not provide clustering of metadata on the disk. The metadata is typically scattered all over the volume. This scattering of metadata causes performance degradation during runtime and also impacts the ability of the operating system to recover corrupted data.

2.2 About the NTFS File System

Microsoft designed the Microsoft Windows NT® File System (NTFS), specifically for the Windows NT operating system, to support and include features required from an enterprise-class file system. When NTFS was created, Microsoft already supported two file systems: FAT and High Performance File System (HPFS). NTFS was designed as a more robust replacement for HPFS.

2.2.1 Media Suitable for NTFS

Most PCs are configured with fixed hard drives. These PCs are used for a variety of purposes, including file share servers, mail servers, database servers, and home or information worker desktops. There are many requirements for a file system that arise from these different usage scenarios, such as recoverability, availability, security, and performance. NTFS meets all of these requirements, which makes NTFS is the ideal file system for use with fixed hard drives.

2.2.1.1 What Makes NTFS Suitable for These Media

The following list provides reasons why the NTFS file system is suitable for hard disk drives:

- **Recoverability:** NTFS is a mature journaling file system that uses database-like logging techniques in order to provide high availability. Journaling file systems commit metadata changes to the file system in transactions. In the event of a power failure or operating system crash, NTFS quickly rolls back the uncommitted transactions to quickly return the file system back to a consistent state—typically within a few seconds.

- **Availability:** Because NTFS can recover from crashes and return the operating system back to a consistent state due to journaling, it reduces the necessity to run CHKDSK as frequently as other file systems. As a result, the downtimes associated with running CHKDSK are eliminated or at least reduced, which increases the availability of the operating system.

In rare occasions, such as an operating system crash that occurs due to defective or misbehaving hardware, it would be necessary to run CHKDSK to repair the file system. In every new version of Windows, improvements have been made (to either the file system or CHKDSK itself) to reduce the amount of time required by CHKDSK to check the volume; thereby increasing the availability of the system.

- **Security:** NTFS supports the Windows NT security model, which provides access control and auditing to file objects. Files and directories can be protected from being accessed by unauthorized users. NTFS enables administrators to allow or deny access to file objects in a granular fashion, ranging from no access, to full access. Windows 2000 and later versions include a feature called the Encrypted File System, which enables users to encrypt files that are stored on NTFS drive volumes.
- **Optimized space utilization:** If the data in the file is small (typically a few hundred bytes), then this data can be completely contained within the Master File Table (MFT) record of the file. Otherwise, this data would have required space allocation outside of the MFT. This provides better space utilization for volumes having a large number of small files.
- **Speed:** The NTFS file system uses a B-tree structure for all folders. This structure minimizes the number of disk accesses that a hard drive must perform to find a file. If the data in the file is small (typically a few hundred bytes), then this data can be completely contained within the MFT record of the file. This makes locating and retrieving a file on NTFS faster than on the FAT file system.
- **Support for large files and disks:** As shown in Table 4, the Windows file system driver limits the size of files on an NTFS volume to approximately 256 terabytes, which is smaller than the NTFS file system format limit of (2^{64} - 1) bytes. Similarly, the maximum volume size supported by current versions of Windows is 2^{32} allocation units, which is smaller than the NTFS file system format limit of 2^{64} allocation units. This shows that the 64-bit design of NTFS is adequate for supporting the file and volume size requirements for future operation systems.
- **Advanced Features:** NTFS boasts numerous features that enterprises require in a file system in order to build and deploy value-added services in their systems. Reparse points, sparse file support, file compression, change journal, and indexing are examples of these advanced features. For more information about these advanced features, see Table 3.

2.3 About the UDF File System

UDF is a file system that is defined by the Optical Storage Technology Association (OSTA). UDF is compliant with ISO-13346/ECMA-167 and is the successor to the CD-ROM file system (CDFS or ISO-9660). At publication, the most current version of the standard is UDF 2.50. UDF is targeted for use with DVD, CD, and magneto-optical (MO). UDF can also be used for exchanging data among computers running different operating systems.

The UDF format supports a number of advanced features including; long and Unicode filenames, access control lists (ACL)s, alternate data streams, 64-bit file sizes, sparse files, and writing to many different types of media. Note that the Microsoft UDF file system driver in current versions of Windows does not support

ACLs or alternate data streams. However, alternate data streams are supported in Windows Vista.

UDF is an evolving specification, and there are several versions of UDF defined by OSTA. The following table shows the UDF versions that are supported by different versions of Windows.

Table 2—UDF versions supported by Windows

Windows version	UDF 1.02	UDF 1.5	UDF 2.00 / 2.01	UDF 2.50
Windows 95	Not supported	Not supported	Not supported	Not supported
Windows 98	Read-only	Not supported	Not supported	Not supported
Windows NT 4.0	Not supported	Not supported	Not supported	Not supported
Windows 2000	Read-only	Read-only	Not supported	Not supported
Windows XP	Read-only	Read-only	Read-only	Not supported
Windows Server 2003	Read-only	Read-only	Read-only	Not supported
Windows Vista	Read/write	Read/write	Read/write	Read/write

Because UDF is not supported by all operating systems, be sure to consider issues needed for backwards compatibility.

2.3.1 Media Suitable for UDF

UDF is designed for and is ideally suited for storing media files on CD and DVD media. Examples of these media types are:

CD Media:
CD-R
CD-RW
CD-MRW

DVD Media:
DVD-R
DVD+R
DVD-RW
DVD+RW
DVD+MRW
DVD-RAM

UDF on CD/DVD media is ideal for storing media files. Because of the storage capacity of DVD media types, DVD media with UDF is replacing tape as the best choice for archiving data. UDF, revision 2.50 in particular, is also suitable for high-capacity next generation optical media, such as optical media that uses blue laser technology.

2.3.1.1 What Makes UDF Suitable for These Media

The following list provides reasons why the UDF file system is suitable for these media types:

- Cross operating system interoperability: Because UDF is an international standard, it provides a way for data interchange among different operating systems.
- Ideal for media files: UDF provides specific support for marking a file as a real-time file. Real-time files are guaranteed to support a minimum data transfer rate when reading or writing audio and video data. This makes UDF an appropriate file system for streaming audio or video media.

UDF allows a 64-bit file size, making it suitable for devices and applications that may need to save huge files, for example, a high quality, long-duration audio and video file. Most consumer electronics devices that play audio and video, such as DVD players, contain a minimum implementation of the UDF file system. This standard makes UDF even more attractive for storing media files on CD and DVD media because the CDs and DVDs that are created by using such a consumer electronics device can then be recognized and played by other consumer electronics devices.

- **Incremental-write media support:** UDF is the only file system described in this paper that can be used on incremental-write media, such as CD-R, DVD-R, and DVD+R.
- **Defect management support:** UDF provides support for defect management at the file system layer. It provides a mechanism for mapping bad blocks to their new locations on the media, even when the underlying hardware provides no defect management. This mechanism makes UDF the only file system described in this paper that can be used on devices that do not support defect management in hardware, such as CD-RW, DVD-RW, and DVD+RW.
- **Scalability:** UDF allows 2^{32} allocation units. With the typical allocation unit size of 2 KB on CD/DVD media, UDF supports a maximum volume size of 8 terabytes. This volume size makes UDF suitable for CD/DVD media whose capacities are constantly growing. UDF allows a 64-bit file size, thus allowing terabyte-sized files to be stored on the media.
- **Performance and recoverability:** As of version 2.50, UDF supports clustering of metadata on non-incremental-write media. This media significantly boosts performance and recoverability as compared to FAT. This media also results in reduced CHKDSK times.

3 Local File System Feature Comparison

The following table shows a comparison of features for the NTFS, FAT, and UDF file systems. The column for FAT represents both the FAT16 and FAT32 file systems.

Table 3—Comparison of features

Feature	Description	FAT	NTFS	UDF
Journaling	File systems that support this feature use transactions to commit metadata changes to the file system. In the event of a power failure or operating system crash, the file system quickly rolls back the uncommitted transactions to quickly return the file system back to a consistent state.	X	√	X
Metadata clustering	The file system attempts to store metadata in contiguous locations on the media. Note that metadata clustering is only supported on UDF 2.5 and later on nonincremental-write media.	X	√	√
Directories are B-trees	A B-Tree index–based directory provides more efficient access to directory entries than a linear directory.	X	√	X
Volume mount points	Volume mount points enable administrators to mount a file system to a directory instead of mounting the file system to a drive letter.	X	√	X
Resident file storage	This feature allows small files (typically a few hundred bytes for each file) to be stored within the file system metadata.	X	√	√

Feature	Description	FAT	NTFS	UDF
File and directory security	The file system can restrict access to files and directories individually for each user or for a group of users.	X	√	X*
Compression	This feature enables users to store files in a compressed format on the media.	X	√	X
Hard links	A hard link is the file system representation of a file by which more than one path references a single file in the same volume.	X	√	√
Directory symbolic links	This feature is similar to hard links but allows one directory name to point to another directory on the same PC.	X	√	X*
Encryption	This feature enables users to store files in an encrypted format on the media.	X	√	X
Change journal	This feature provides a persistent log of changes made to files on a volume.	X	√	X
Quotas	This feature enables an administrator to limit the amount of disk space a particular user can use on a given volume.	X	√	X
Sparse file support	This feature allows for efficient utilization of disk space for sparse files.	X	√	√
Alternate data stream support	This feature allows a file to contain multiple streams of user-defined data.	X	√	X*+
Configurable allocation size	This feature allows the allocation unit size of the file system to be configured.	√	√	X
Unicode naming	The file system supports Unicode file names.	X	√	√
Defect management	The file system supports sparing of bad blocks on the media when the underlying hardware does not provide defect management.	X	X	√
Incremental-write media support	The file system can work with media that needs data to be written incrementally.	X	X	√

*This feature is defined by the UDF 2.01 specification; however, it is not supported by the Microsoft UDF file system driver in current versions of Windows.

+Alternate data streams are supported in Windows Vista.

4 Local File System Limitations

This section describes the design limits of the FAT, NTFS, and UDF local file systems and describes any additional limits imposed by different versions of Windows.

There are two factors that determine the limitations of any local file system: the file system format and the file system driver. The file system format defines the data structures that contain the file system metadata on the media. The file system driver is the software in the operating system that stores and retrieves user data on media by using the file system metadata structures.

The file system driver that uses a particular file system format may not use the format to its full capability. Because of this, the limits imposed by the file system driver for a particular version of the operating system may be more restrictive than those limits imposed by the file system format itself.

4.1 File System Limits Imposed by the File System Format

The following table shows the limits for FAT16, FAT32, NTFS, and UDF that are imposed by the design of the file system format.

Table 4—File system limits imposed by local file system design

Limits	FAT16	FAT32	NTFS	UDF
Maximum file size	2^{32} -1 bytes See Note 1	2^{32} -1 bytes See Note 1	2^{64} -1 bytes See Note 2	2^{64} -1 bytes See Note 2
Maximum allocation units per volume	2^{16}–18 See Note 3	2^{28}–18 See Note 4	2^{64}	2^{32}
Maximum volume size for Windows 95/98	Approx. 2 GB See Note 5	2 terabytes See Note 7	Not supported on Windows 95/98	Not supported on Windows 95 and no write support on Windows 98
Maximum volume size for Windows 2000, Windows XP, Windows Server 2003, and Windows Vista	Approx. 4 GB See Note 6	2 terabytes See Note 7	2^{33} terabytes See Note 8	No write support on Windows 2000, Windows XP, and Windows Server 2003 For Windows Vista: 2 terabytes for hard disk drives 8 terabytes for CD/DVD 16 terabytes for MO media See Note 9

4.1.1.1 Notes for Table 4:

1. Fields related to the file size are 32 bit in size.
2. Fields related to the file size are 64 bit in size.
3. The value of an entry in the file allocation table for each allocation unit in FAT16 can range from 0x0002 to 0xFFEF.
4. The value of an entry in the file allocation table for each allocation unit in FAT32 can range from 0x00000002 to 0x0FFFFFEF.
5. On Windows 95 and Windows 98, the maximum allocation unit size supported is 32 KB.

   The maximum volume size for FAT16 on Windows 95 and Windows 98:

   (Maximum allocation units per volume–Number of reserved FAT entry values)×Maximum allocation unit size=(2^{16}–18)×32 KB, which is approximately 2 GB.
6. On Windows 2000 and later versions, the maximum supported allocation unit size is 64 KB.

   The maximum volume size for FAT16 on Windows 2000 and later versions:

   (Maximum allocation units per volume–Number of reserved FAT entry values)×Maximum allocation unit size=(2^{16}–18)×64 KB, which is approximately 4 GB.
7. The field in the FAT boot sector that specifies the number of sectors on a volume is 32 bit in size. Using the sector size of 512 bytes (available on current

hardware) the maximum possible size of a hard disk drive volume is: (2^{32}×512) bytes=2 terabytes.

8. The field in the NTFS boot sector that specifies the number of sectors on a volume is 64 bit in size. Using the sector size of 512 bytes (available on current hardware) the maximum possible size of an NTFS hard disk drive volume is: (2^{64}×512) bytes=2^{73} bytes.
9. The field in the UDF partition descriptor that specifies the number of logical sectors on the partition is 32 bits in size. In currently available hardware, the allocation unit size used for hard disk drive media is 512 bytes, for CD/DVD media is 2 KB, and for magneto-optical media is 4 KB.

   Note that UDF write support is available only on Windows Vista.

   The maximum volume size for hard disk drive media:

   Maximum allocation units per volume × Allocation unit size = 2^{32} × 512 = 2 terabytes.

   The maximum volume size for CD/DVD media:

   Maximum allocation units per volume × Allocation unit size = 2^{32} × 2048 = 8 terabytes.

   The maximum volume size for MO media:

   Maximum allocation units per volume × Allocation unit size = 2^{32} × 4096 = 16 terabytes.

4.2 File System Limits Imposed by the File System Driver

The following table shows the local file system limits for FAT16, FAT32, NTFS, and UDF that are imposed by the associated Windows file system driver for different versions of Windows.

Table 5—Local file system limits imposed by Windows

Limits	FAT16	FAT32	NTFS	UDF
Maximum file size	2^{32} -1 bytes See Note 1	2^{32} -1 bytes See Note 1	Approx. 256 terabytes See Note 2	Approx. 2 terabytes for hard disk drives Approx. 8 terabytes for CD/DVD media 16 terabytes for magneto-optical media See Note 3
Maximum allocation units per volume in Windows 95 and Windows 98	2^{16}–18 See Note 1	4,177,918 See Note 4	Not supported on Windows 95 and Windows 98	Not supported on Windows 95, and no write support on Windows 98

Limits	FAT16	FAT32	NTFS	UDF
Maximum allocation units per volume in Windows 2000, Window XP, Windows Server 2003, and Windows Vista	2^{16}–18 See Note 1	2^{28}–18 See Note 1	2^{32} See Note 5	No write support on Windows 2000, Windows XP, Windows Server 2003 For Windows Vista: 2^{32} See Note 1
Maximum volume size for Windows 95 and Windows 98	Approx. 2 GB See Note 1	127.5 GB See Notes 4 and 6	Not supported on Windows 95 and Windows 98	Not supported on Windows 95 and no write support on Windows 98
Maximum volume size for Windows 2000, Windows XP, Windows Server 2003, and Windows Vista	Approx. 4 GB See Note 1	File system handles 2 terabytes Format limit is 32 GB	256 terabytes See Notes 7 and 8	No write support on Windows 2000, Windows XP, Windows Server 2003 For Windows Vista: 2 terabytes for hard disk drives 8 terabytes for CD/DVD media 16 terabytes for MO media See Note 1
Maximum files per volume	Approx. 2^{16} See Note 9	Approx. 2^{28} See Note 9	2^{32} See Note 10	Approx. 2^{32} See Note 9
Allocation unit size for Windows 95 and Windows 98	Any power of 2 between 512 bytes and 32768 bytes inclusive	Any power of 2 between 512 bytes and 32768 bytes inclusive	Not supported on Windows 95 and Windows 98	Not supported on Windows 95 and no write support on Windows 98

Limits	FAT16	FAT32	NTFS	UDF
Allocation unit size for Windows 2000, Windows XP, Windows Server 2003, and Windows Vista	Any power of 2 between 512 bytes and 65536 bytes inclusive	Any power of 2 between 512 bytes and 65536 bytes inclusive	Any power of 2 between 512 bytes and 65536 bytes inclusive	No write support on Windows 2000, Windows XP, and Windows Server 2003 For Windows Vista: 512 bytes for hard drive disks 2 KB for CD/DVD media 4 KB for MO media See Note 11
Files per directory	Approx. 2^{16} See Notes 12 and 13	2^{16}–2 See Note 14	No limit	No limit

Maximum disk drives per server	Limited only by available memory
Maximum simultaneous local file opens	Limited only by available memory
Maximum simultaneous file locks	Limited only by available memory

4.2.1.1 Notes for Table 5:

1. The limit imposed by the file system driver is identical to the one imposed by the file system format.
2. The maximum file size on an NTFS volume is limited by the maximum number of allocation units on an NTFS volume, which is 2^{32}. The maximum allocation unit size on an NTFS volume is 64 KB.

   The maximum file size on an NTFS volume:

   (Maximum allocation units per volume–The reserved number FFFFFFFF) × Maximum allocation unit size = (2^{32} -1) × 64 KB = (256 terabytes–64 KB)
3. This is limited by the maximum number of allocation units on a UDF volume, which is 2^{32}. Also, the maximum allocation unit size on a UDF volume is 512 bytes for hard disk drive media, 2 KB for CD/DVD media, and 4 KB for magneto-optical media.

   The maximum file size for hard disk media:

   (Maximum allocation units per volume–The reserved number FFFFFFFF) × Allocation unit size = (2^{32} -1) × 512 bytes = (2 terabytes–512 bytes).

   The maximum file size for CD /DVD media:

   (Maximum allocation units per volume–The reserved number FFFFFFFF) × Allocation unit size = (2^{32} -1) × 2 KB = (8 terabytes–2 KB).

   The maximum file size for magneto-optical media:

   (Maximum allocation units per volume–The reserved number FFFFFFFF) × Allocation unit size = (2^{32} -1) × 4 KB = (16 terabytes–4 KB).

4. In Windows 95 and Windows 98, the CHKDSK and DEFRAG utilities cannot support file allocation tables larger than 16 MB–64 KB in size.

   The maximum number of allocation units that can exist on a FAT32 hard disk drive volume with Windows 95 and Windows 98:

   (File allocation table size)/(Bytes per FAT entry) – (First two reserved allocation unit numbers in the file allocation table) = (16 × 1024 – 64) × 1024/4 – 2 = 4,177,918 allocation units.

   The maximum FAT volume size for Windows 95/98:

   > Maximum allocation units per volume × Maximum allocation unit size=4,177,918×32 KB, which is 127.5 GB.

   Note that this limit is lifted in Windows Me, Windows 2000, and later versions.

5. The file system driver uses 32-bit values to track the location of allocation units.

6. In Windows 2000 and later versions, the operating system does not allow any volumes larger than 32 GB to be formatted with FAT. (For hard disk drive volumes larger than this, NTFS is a more suitable file system.) The operating system will still mount FAT volumes larger than 32 GB. However, in Windows 95 and Windows 98 there is no volume size limit imposed by the Windows format (format.com) utility.

7. On Windows 2000 and later versions, the maximum supported allocation unit size is 64 KB, and the maximum number of allocation units on NTFS is 2^{32}.

   The maximum NTFS volume size for Windows 2000 and later versions:

   > Maximum allocation units per volume × Maximum allocation unit size=2^{32}×64 KB, which is 256 terabytes.

8. The maximum size of an NTFS volume is limited by the maximum size of a disk that can be used on the current versions of Windows. This limitation is external to the file system and is not an NTFS limit.

   The largest hard disk drive that can be used on the current version of Windows is limited to 2^{32} sectors. Using the sector size of 512 bytes (available on current hardware) the maximum possible size of a disk is (2^{32}×512) bytes=2 terabytes.

   A basic volume is limited to one disk. So the maximum size of a basic volume is 2 terabytes.

   A dynamic volume (striped or spanned) can span up to 32 disks. So the maximum size of a dynamic volume is 32×2 terabytes=64 terabytes.

9. This limit is derived from the maximum number of allocation units per volume. For the purpose of this calculation, it was assumed that each file occupies only one allocation unit.

10. The NTFS MFT is limited to 2^{32} files.

11. The UDF specification defines that Block Size must be the same size as the physical sector.

12. This limit is derived from the maximum number of files supported on a FAT 16 volume.

13. The Windows format utility (format.com) limits the FAT16 root directories to 512 directory entries.

14. Because a large number of directory entries on a FAT volume causes file access performance degradation, FAT32 volumes on Windows are limited to 2^{16}–2 directory entries (where 2 accounts for the directory entries for the current directory and the parent directory, also know as "**.**" and "**..**").

5 Resources and Call to Action

5.1 Call to Action

For system administrators and system manufacturers:

- NTFS is recommended as the best choice for hard disk drive media.
- FAT is recommended as the best choice for Flash media.
- UDF is recommended for CD/DVD media.

For device manufacturers:

If you are working on new storage media that is designed to work with Windows and would like to request a file system recommendation from Microsoft, contact: **winhecfs@microsoft.com**

5.2 Feedback

To provide feedback about the content in this paper, please send an e-mail message to **winhecfs@microsoft.com**

5.3 Resources

ISO-13346
http://www.iso.org

ECMA-167
http://www.ecma-international.org/publications/

Universal Disk Format (UDF) specification, Revision 2.50
http://www.osta.org/specs/

Windows Platform Development white papers and resources:
http://www.microsoft.com/whdc/default.mspx

Windows Driver Development Kit:
http://www.microsoft.com/whdc/devtools/ddk/default.mspx

Windows Logo Program for Hardware:
http://www.microsoft.com/whdc/winlogo/default.mspx

WHQL Test Specifications, hardware compatibility tests, and testing notes:
http://www.microsoft.com/whdc/hwtest/default.mspx

6 Appendix—Glossary

The following table describes the terminology used in this paper.

Term	Description
Access control list (ACL)	A list of access control entries (ACEs). Each ACE in an access control list identifies a user and specifies the access rights allowed, denied, or audited for that user.
Allocation unit	The minimum amount of space on a local file system that can be allocated to store a single file. An allocation unit is also known as a cluster, when describing the FAT and NTFS file systems, and as a block, when describing UDF. For the purposes of this paper, the term allocation unit is used regardless of the local file system being described.
Alternate data streams	Files generally have a default data stream. Certain file systems allow additional named streams to be associated with a file. These are called alternate data streams.
B-tree organization	A mechanism for placing and locating entries in an index (in the form of a B+ tree). This organization of records minimizes the number of times a medium must be accessed to locate a desired record, which speeds up this process.
Change journal	A special log that NTFS maintains for every volume, in Windows 2000 and later versions. When any change is made to a file or directory in an NTFS volume, the change journal for that volume is updated with a description of the change and the name of the file or directory.
Data stream	A simple sequence of bytes that constitutes the data in a file.
Incremental-write media	Media that does not support overwriting locations that have previously been written to. After a location on the media has been written to, that location cannot be rewritten and the next write to the media must be to a location beyond the location that was last written.
Metadata clustering	On certain file systems, the metadata that describes the files that are present on the volume is stored in contiguous locations on the disk. The file system keeps track of these contiguous areas that store the metadata. Metadata clustering improves performance and recoverability of data on the volume.
Master file table (MFT)	A collection of MFT records for a given volume.
MFT record	Every file stored on an NTFS volume has a corresponding metadata entry in the Master File Table, which is referred to as an MFT record. The MFT record for a file contains the metadata information for that file, such as it's name, size, time stamps, permissions, location of data content, and so on.
Reparse points	A file or directory can contain a reparse point, which is a collection of user-defined data. The format of this data is understood by the application that stores the data and a file system filter that is installed to interpret the data and process the file.

Term	Description
Sparse file support	A file in which much of the data is zeros and is said to contain a sparse data set. Files like these are typically very large—for example, a file containing image data to be processed. When the sparse file functionality is enabled, the file system does not allocate hard drive disk space to a file except in regions where it contains nonzero data; the file system maintains the zero-data in a special way to save disk space.
Sparing	The file system maintains a list of data blocks that can be used as replacements for bad blocks on the media. When a block on a medium goes bad, the file system maps all access requests for the bad block to a replacement block. This process is called sparing.