**Overview of FAT, HPFS, and NTFS File Systems**

**Summary**

This article explains the differences between FAT, HPFS, and NTFS under Windows NT, and their advantages and disadvantages. It is divided into the following sections:

* FAT Overview
* HPFS Overview
* NTFS Overview

NOTE: HPFS is only supported under Windows NT versions 3.1, 3.5, and 3.51. Windows NT 4.0 does not support and cannot access HPFS partitions. Also, the FAT32 file system is only supported in the Windows 98/95 and Windows 2000.

**More Information**

**FAT OVERVIEW**

FAT is by far the most simplistic of the file systems supported by Windows NT. The FAT file system is characterized by the file allocation table (FAT), which is really a table that resides at the very "top" of the volume. To protect the volume, two copies of the FAT are kept in case one becomes damaged. In addition, the FAT tables and the root directory must be stored in a fixed location so that the system's boot files can be correctly located.  
  
  
A disk formatted with FAT is allocated in clusters, whose size are determined by the size of the volume. When a file is created, an entry is created in the directory and the first cluster number containing data is established. This entry in the FAT table either indicates that this is the last cluster of the file, or points to the next cluster.  
  
  
Updating the FAT table is very important as well as time consuming. If the FAT table is not regularly updated, it can lead to data loss. It is time consuming because the disk read heads must be repositioned to the drive's logical track zero each time the FAT table is updated.  
  
  
There is no organization to the FAT directory structure, and files are given the first open location on the drive. In addition, FAT supports only read-only, hidden, system, and archive file attributes.

**FAT Naming Convention**

FAT uses the traditional 8.3 file naming convention and all filenames must be created with the ASCII character set. The name of a file or directory can be up to eight characters long, then a period (.) separator, and up to a three character extension. The name must start with either a letter or number and can contain any characters except for the following:

. " / \ [ ] : ; | = ,

If any of these characters are used, unexpected results may occur. The name cannot contain any spaces.  
  
  
The following names are reserved:

CON, AUX, COM1, COM2, COM3, COM4, LPT1, LPT2, LPT3, PRN, NUL

All characters will be converted to uppercase.

**Advantages of FAT**

It is not possible to perform an undelete under Windows NT on any of the supported file systems. Undelete utilities try to directly access the hardware, which cannot be done under Windows NT. However, if the file was located on a FAT partition, and the system is restarted under MS-DOS, the file can be undeleted. The FAT file system is best for drives and/or partitions under approximately 200 MB, because FAT starts out with very little overhead. For further discussion of FAT advantages, see the following:

* Microsoft Windows NT Server "Concepts and Planning Guide," Chapter 5, section titled "Choosing a File System"
* Microsoft Windows NT Workstation 4.0 Resource Kit, Chapter 18, "Choosing a File System"
* Microsoft Windows NT Server 4.0 Resource Kit "Resource Guide," Chapter 3, section titled "Which File System to Use on Which Volumes"

**Disadvantages of FAT**

Preferably, when using drives or partitions of over 200 MB the FAT file system should not be used. This is because as the size of the volume increases, performance with FAT will quickly decrease. It is not possible to set permissions on files that are FAT partitions.  
  
  
FAT partitions are limited in size to a maximum of 4 Gigabytes (GB) under Windows NT and 2 GB in MS-DOS. For additional information on this limitation, please see the following article in the Microsoft Knowledge Base:

**HPFS OVERVIEW**

The HPFS file system was first introduced with OS/2 1.2 to allow for greater access to the larger hard drives that were then appearing on the market. Additionally, it was necessary for a new file system to extend the naming system, organization, and security for the growing demands of the network server market. HPFS maintains the directory organization of FAT, but adds automatic sorting of the directory based on filenames. Filenames are extended to up to 254 double byte characters. HPFS also allows a file to be composed of "data" and special attributes to allow for increased flexibility in terms of supporting other naming conventions and security. In addition, the unit of allocation is changed from clusters to physical sectors (512 bytes), which reduces lost disk space.  
  
  
Under HPFS, directory entries hold more information than under FAT. As well as the attribute file, this includes information about the modification, creation, and access date and times. Instead of pointing to the first cluster of the file, the directory entries under HPFS point to the FNODE. The FNODE can contain the file's data, or pointers that may point to the file's data or to other structures that will eventually point to the file's data.  
  
  
HPFS attempts to allocate as much of a file in contiguous sectors as possible. This is done in order to increase speed when doing sequential processing of a file.  
  
  
HPFS organizes a drive into a series of 8 MB bands, and whenever possible a file is contained within one of these bands. Between each of these bands are 2K allocation bitmaps, which keep track of which sectors within a band have and have not been allocated. Banding increases performance because the drive head does not have to return to the logical top (typically cylinder 0) of the disk, but to the nearest band allocation bitmap to determine where a file is to be stored.  
  
  
Additionally, HPFS includes a couple of unique special data objects:

**Super Block**

The Super Block is located in logical sector 16 and contains a pointer to the FNODE of the root directory. One of the biggest dangers of using HPFS is that if the Super Block is lost or corrupted due to a bad sector, so are the contents of the partition, even if the rest of the drive is fine. It would be possible to recover the data on the drive by copying everything to another drive with a good sector 16 and rebuilding the Super Block. However, this is a very complex task.

**Spare Block**

The Spare Block is located in logical sector 17 and contains a table of "hot fixes" and the Spare Directory Block. Under HPFS, when a bad sector is detected, the "hot fixes" entry is used to logically point to an existing good sector in place of the bad sector. This technique for handling write errors is known as hot fixing.  
  
  
Hot fixing is a technique where if an error occurs because of a bad sector, the file system moves the information to a different sector and marks the original sector as bad. This is all done transparent to any applications that are performing disk I/O (that is, the application never knows that there were any problems with the hard drive). Using a file system that supports hot fixing will eliminate error messages such as the FAT "Abort, Retry, or Fail?" error message that occurs when a bad sector is encountered.  
  
  
Note: The version of HPFS that is included with Windows NT does not support hot fixing.

**Advantages of HPFS**

HPFS is best for drives in the 200-400 MB range.

**Disadvantages of HPFS**

Because of the overhead involved in HPFS, it is not a very efficient choice for a volume of under approximately 200 MB. In addition, with volumes larger than about 400 MB, there will be some performance degradation. You cannot set security on HPFS under Windows NT.  
  
  
HPFS is only supported under Windows NT versions 3.1, 3.5, and 3.51. Windows NT 4.0 cannot access HPFS partitions.

**NTFS OVERVIEW**

From a user's point of view, NTFS continues to organize files into directories, which, like HPFS, are sorted. However, unlike FAT or HPFS, there are no "special" objects on the disk and there is no dependence on the underlying hardware, such as 512 byte sectors. In addition, there are no special locations on the disk, such as FAT tables or HPFS Super Blocks.  
  
  
The goals of NTFS are to provide:

* Reliability, which is especially desirable for high end systems and file servers
* A platform for added functionality
* Support POSIX requirements
* Removal of the limitations of the FAT and HPFS file systems

**Reliability**

To ensure reliability of NTFS, three major areas were addressed: recoverability, removal of fatal single sector failures, and hot fixing.  
  
  
NTFS is a recoverable file system because it keeps track of transactions against the file system. When a CHKDSK is performed on FAT or HPFS, the consistency of pointers within the directory, allocation, and file tables is being checked. Under NTFS, a log of transactions against these components is maintained so that CHKDSK need only roll back transactions to the last commit point in order to recover consistency within the file system.  
  
  
Under FAT or HPFS, if a sector that is the location of one of the file system's special objects fails, then a single sector failure will occur. NTFS avoids this in two ways: first, by not using special objects on the disk and tracking and protecting all objects that are on the disk. Secondly, under NTFS, multiple copies (the number depends on the volume size) of the Master File Table are kept.  
  
  
Similar to OS/2 versions of HPFS, NTFS supports hot fixing.

**Added Functionality**

One of the major design goals of Windows NT at every level is to provide a platform that can be added to and built upon, and NTFS is no exception. NTFS provides a rich and flexible platform for other file systems to be able to use. In addition, NTFS fully supports the Windows NT security model and supports multiple data streams. No longer is a data file a single stream of data. Finally, under NTFS, a user can add his or her own user-defined attributes to a file.

**POSIX Support**

NTFS is the most POSIX.1 compliant of the supported file systems because it supports the following POSIX.1 requirements:  
  
  
Case Sensitive Naming:  
  
  
Under POSIX, README.TXT, Readme.txt, and readme.txt are all different files.  
  
  
Additional Time Stamp:  
  
  
The additional time stamp supplies the time at which the file was last accessed.  
  
  
Hard Links:  
  
  
A hard link is when two different filenames, which can be located in different directories, point to the same data.

**Removing Limitations**

First, NTFS has greatly increased the size of files and volumes, so that they can now be up to 2^64 bytes (16 exabytes or 18,446,744,073,709,551,616 bytes). NTFS has also returned to the FAT concept of clusters in order to avoid HPFS problem of a fixed sector size. This was done because Windows NT is a portable operating system and different disk technology is likely to be encountered at some point. Therefore, 512 bytes per sector was viewed as having a large possibility of not always being a good fit for the allocation. This was accomplished by allowing the cluster to be defined as multiples of the hardware's natural allocation size. Finally, in NTFS all filenames are Unicode based, and 8.3 filenames are kept along with long filenames.

**Advantages of NTFS**

NTFS is best for use on volumes of about 400 MB or more. This is because performance does not degrade under NTFS, as it does under FAT, with larger volume sizes.  
  
  
The recoverability designed into NTFS is such that a user should never have to run any sort of disk repair utility on an NTFS partition.

**Disadvantages of NTFS**

It is not recommended to use NTFS on a volume that is smaller than approximately 400 MB, because of the amount of space overhead involved in NTFS. This space overhead is in the form of NTFS system files that typically use at least 4 MB of drive space on a 100 MB partition.  
  
  
Currently, there is no file encryption built into NTFS. Therefore, someone can boot under MS-DOS, or another operating system, and use a low-level disk editing utility to view data stored on an NTFS volume.  
  
  
It is not possible to format a floppy disk with the NTFS file system; Windows NT formats all floppy disks with the FAT file system because the overhead involved in NTFS will not fit onto a floppy disk.

**NTFS Naming Conventions**

File and directory names can be up to 255 characters long, including any extensions. Names preserve case, but are not case sensitive. NTFS makes no distinction of filenames based on case. Names can contain any characters except for the following:

? " / \ < > \* | :

Currently, from the command line, you can only create file names of up to 253 characters.  
  
NOTE: Underlying hardware limitations may impose additional partition size limitations in any file system. Particularly, a boot partition can be only 7.8 GB in size, and there is a 2-terabyte limitation in the partition table.