

MASTER FILE TABLE (MFT) & FRAGMENTATION

Mark E. Donaldson

THE MASTER FILE TABLE

MFT stands for Master File Table. The MFT is the heart of the NTFS file system. It is essentially an index to all of the files on an NTFS volume, containing the file name, a list of the file attributes, and pointers to the fragments. (A contiguous file is considered to have just one fragment and the record in the MFT contains only one pointer.) The MFT stores data in the format of a relational database, that is, each attribute has a specific position within the record. Thus, NTFS can create indexes from the MFT records to sort files by attributes, although at this time, only the file names are indexed.

The data for each file is contained in one record in the MFT, called the "file record". Each file record is one to four KB in size, depending on how the volume was formatted. However, a file can have a large number of attributes, too many to fit in a single record, in which case an additional record is needed. In that case, the first file record is called the "base file record" and it contains the location of the other file record(s) associated with the file.

A file can also become very badly fragmented, filling the file record with pointers to the fragments of the file, in which case an additional record will also be needed. If a file becomes really badly fragmented, it may require many additional records and could eventually cause you to run out of records in the MFT, at which point you would not be able to create new files.

NTFS stores directories in the MFT as it stores any other files. In fact, everything stored on an NTFS volume is a file; the system was designed that way, and it does make the design simpler. So, the directories are files, whose records are simply the names of the files which all share the same directory path.

When an NTFS volume is formatted, a large amount of space (over 10% of the volume capacity) is reserved for the MFT. This reserved space doesn't necessarily fill with MFT records. For example, if you have 10,000 file records (one each for 10,000 files), and each record is 4 KB in size, then you have a 40 MB MFT. But if those 10,000 files are on a 1 GB volume, the space reserved for the MFT is roughly 120 MB in size. The actual MFT space used is displayed by the Diskkeeper Graphic Display as solid green areas.

Another variable that comes into play is the fact that small files (typically less than 4 KB in size) are stored in their entirety in the MFT. So, in cases where a volume contains a large number of small files, the MFT can become quite large, and in fact can expand beyond the area that is reserved for it. This is fairly unusual, however.

As mentioned earlier, most of the MFT is located at the beginning of the volume, but one section is placed in the middle of the volume. This part of the MFT duplicates the first 16 records of the MFT, containing critical data that, if it were lost, could mean everything on the volume would be lost. Storing this data in two widely separated places makes it very unlikely that you will lose the data.

The first nine records in the MFT are the index entries for:

- the MFT itself.
- the second part of the MFT (duplicating the first 16 records).
- the log file (records disk activity, helps in reconstructing the disk if necessary).

MASTER FILE TABLE (MFT) & FRAGMENTATION

Mark E. Donaldson

- the volume file (contains the volume name, NTFS version the disk was formatted under, and a flag that is set when disk repair is needed).
- the attribute definition table.
- the root directory.
- the bitmap file (one bit for each cluster, set to indicate the cluster is allocated to some file).
- the boot file.
- the bad cluster file (bad clusters on the disk are allocated to this file, so they won't be used by any other file).

The other seven records are apparently reserved for future use. You can see that it would indeed be a disaster if this data were lost!

The MFT plays a critical role on an NTFS volume and is crucial for the implementation of the very aspects of NTFS that make it the recoverable, secure, reliable, and efficient file system for client-server and other high-end systems.

MFT FRAGMENTATION

Fragmentation of the Master File Table (MFT) can be a serious problem on NTFS partitions. This is because the MFT is used for every disk I/O. While much of the MFT can be cached, so that an actual disk I/O does not have to be done every time, it is still true that on most systems the MFT is accessed more than any other file. This means that MFT fragmentation is likely to have more impact on the system than fragmentation of any other single file. Steps should be taken to prevent, or at least minimize, MFT fragmentation. But how does the MFT get fragmented in the first place?

CONVERTING FAT TO NTFS

When a FAT partition is converted to NTFS, an MFT is created. If there is a large enough contiguous free space, the MFT is made contiguous, with contiguous pre-allocated expansion space. However, since the pre-allocation comprises about 12% of the partition, this is usually not possible, and the MFT is created fragmented. In addition, the MFT will almost certainly not be at the beginning of the disk, where it is put on partitions that are created with the NTFS format. This means that you have data areas before the MFT and after it, then a secondary copy of critical MFT data, then another data area. So you can see that the converted partition has three data areas instead of the usual two; data fragmentation will thus begin sooner.

The problem for the MFT is that Windows NT will, under some conditions, write files in the space reserved for MFT expansion. If such a file has been written, then as the MFT grows, when it reaches the file, the MFT will have to fragment to get around the file.

Note that the system partition created on installing Windows NT is a FAT partition. If, during installation, you choose to use the NTFS format, the partition will still be created as FAT, and only converted to NTFS after the first boot. This means you get the initial system files written to the beginning of the disk, then, when the conversion is done, the MFT is created. You can avoid this by using a dual-boot system (see the earlier article on dual boot for details). Basically, you install a minimal Windows NT, then create

MASTER FILE TABLE (MFT) & FRAGMENTATION

Mark E. Donaldson

an NTFS partition, preferably on another disk, and do your full installation of Windows NT to the new partition. Afterwards you can reformat the original partition to NTFS and reinstall the minimal Windows NT to retain the benefits of having two Windows NT installations.

MFT RECORD OVERFLOW

A partition can have scattered pieces of the MFT sitting in the data area, outside the normal MFT area. These "MFT record overflow" pieces are actually things like extent lists, security attributes, and extremely long filenames that at some time in the past couldn't fit inside the MFT record. These pieces are not part of a file, but are extensions to the MFT. This means the free space cannot be fully consolidated, and it will be much harder for the partition to be defragmented.

PREVENTING MFT FRAGMENTATION

The only real "fix" at this time is to back up the partition, reformat it, and restore the data. Note that the partition should have a cluster size of at least one kilobyte, because the MFT records are one kilobyte in size; a smaller cluster size will allow the MFT records themselves to fragment. To prevent or minimize MFT fragmentation, you should avoid the following:

- converting FAT partitions to NTFS format (other than by deleting and recreating the partition).
- use of compressed files on an active partition.
- compressing and decompressing entire volumes or directory sub-trees.
- unnecessary security lists.
- extremely long file and directory names (more than 31 characters).

FAT TO NTFS MFT FRAGMENTATION

As time passes, more and more Windows NT users are running into problems because their Master File Tables (MFTs) are fragmenting. This is because the MFT is used for every disk I/O. While much of the MFT can be cached, so that an actual disk I/O does not have to be done every time a file is used, it is still true that on most systems the MFT is accessed more than any other file. This means that MFT fragmentation is likely to have more impact on the system than fragmentation of any other single file. The worst cases occur on partitions that were converted from FAT to NTFS, because the conversion process usually fragments the MFT as it is created.

MOST PARTITIONS

The Boot partition is the one that your BIOS checks to start the boot process, usually C:. The System partition is the one on which Windows NT is installed. Usually this is also the Boot partition. If the partition you want to convert is not Boot or System, you can convert from FAT to NTFS by simply copying the entire partition to a tape or another partition, reformatting the partition as NTFS, and copying the files back. This does not work on System because that's where you have the files used to do the formatting, or on Boot because the reformat would wipe out the boot sector and you would not be able to reboot your machine.

CONVERTING SYSTEM TO NTFS

The system partition created while installing Windows NT is a FAT partition. If you choose during installation to use the NTFS format, the partition is still created as FAT, and only converted to NTFS after

MASTER FILE TABLE (MFT) & FRAGMENTATION

Mark E. Donaldson

the first boot. This means you get the initial system files written to the beginning of the disk, then, when the conversion is done, the MFT is created.

If you are installing Windows NT on a new disk, select to install it to C:, making C: a FAT partition, not NTFS. Do a minimum installation, because you will be deleting these files shortly. When the installation completes, Bring up Disk Administrator (click Start, go to Programs, Administrative Tools, and click Disk Administrator) and create a new partition with NTFS format.

If you already have Windows NT installed on your boot partition, create a new NTFS partition as described above (or select an existing one). Now do your full installation of Windows NT to the new partition and boot into it; this is now your permanent system partition.

CONVERTING BOOT TO NTFS

If Boot is also the System partition, create a new system partition as described above.

Now follow these steps:

- Start Windows NT Explorer.
- Click the C: partition.
- On the Menu Bar, click View, Folder Options.
- Click the View tab.
- In the Advanced Settings box, locate Hidden Files.
- Under Hidded Files, click the "Show all files" button.
- Click Apply, then OK.
- In the C: folder you will see a file called Boot or Boot.ini. Copy this file to your system partition.
- Delete all files on C:
- Copy Boot.ini back to C:
- Reboot to the system partition.
- Delete boot.ini from C:
- Click Start, Run.
- In the Open box, type "convert C: /fs:ntfs" (omit the quotes).
- Copy Boot.ini back to C:

MASTER FILE TABLE (MFT) & FRAGMENTATION

Mark E. Donaldson

WHY THESE METHODS WORK

When a partition is created as NTFS, about 12% of the partition is pre-allocated as the MFT zone, which is expansion space for the MFT. The MFT is placed at the start of the MFT zone. Thus you have a large contiguous expansion space, and the MFT should not fragment unless you fill the partition too full. But when you convert a partition from FAT to NTFS, there are already files at the start of the partition, so the MFT zone has to be placed wherever there is space available. It is very rare to have 12% of a partition as contiguous free space, so the MFT zone is created as dozens or hundreds of fragments. As the MFT extends, it too becomes very fragmented.

Using the method described above, you empty the partition completely, then put back one file. Now all you have is the C: folder and boot.ini. When you reboot, the "next free space" pointer for C: is reset to point to the very first free space, right at the start of the partition. Now when you run the convert command, the MFT zone goes at the start of the disk where it belongs. You may have the C: folder file and boot.ini in the MFT zone, but that only adds two fragments to the MFT; two fragments is not significant.

Incidentally, never use a 512 byte cluster size on an NTFS partition. The MFT records are all 1024 bytes, so the smaller cluster size means MFT records may get fragmented. Don't worry about wasting disk space. First, files that are small enough are stored entirely within their MFT records, and second, disk space is so cheap now that the time you lose because of slow I/O is much more expensive.

RAID AND FRAGMENTATION

RAID originally stood for Redundant Array of Inexpensive Disks; now Inexpensive has been replaced by Independent. They are often referred to as RAID arrays, which is redundant, but useful because Microsoft has an internal-use debugger whose acronym is RAID, so we use the term "RAID array" to avoid confusion. Most RAID devices are RAID-5, and are essentially stripe sets. They fragment just like stripe sets, and are defragmented in the same way.

Let's take for an example a RAID array of four physical disks. When data is written to the array, it is written more or less equally to all four devices. This means writing and reading can be almost four times as fast as to a single disk, because all four devices are active simultaneously. If it takes 1000ms (milliseconds) to write a file to a single disk, the same write will take about 250 to 300ms to write to this RAID array. Reads are similarly faster.

If the data on one of the disks is in two fragments, the read will not take twice as long, but it will take longer than usual. The read/write head will have to perform an additional seek (seek = movement of the read/write head from where it is to the track where the data is) to get to the second fragment, then you wait for the disk to turn until the data comes under the head. How long this takes depends on the hard disk, but it typically averages about 9ms. The longest time required is the sum of the seek time plus one rotation. The average time is one-half of the longest time. The formula to find the access time for a disk is:

1. **Divide RPM by 60 to get the revolutions per second**
2. **Divide 1000 by the result to get the number of milliseconds to do one revolution**
3. **Add seek time in milliseconds**
4. **Divide by 2 to get the average time to get to the data.**

MASTER FILE TABLE (MFT) & FRAGMENTATION

Mark E. Donaldson

This is the average extra time required for each excess fragment. The calculation should be done for the disk that has the most fragments for the file, because your time to complete the I/O is limited by your most fragmented disk.

If the file in our example above has one extra fragment, it will take about 3% longer to read (about 9ms extra time added to a 250ms read). If it has ten extra fragments (all on the same disk), it will take about 30% longer to read (90ms added to 250ms). Now, look at the same file if it is on a plain single disk partition instead of a stripe set. It takes 1000ms to read, and each fragment adds 9ms. One fragment extends the read time only .9%; ten fragments extends the time only 9%!

You see, we are adding the same amount of time in milliseconds whether the file is on a stripe set or not, but the added percentage of time is much worse on a stripe set. Of course, the stripe set figures assume all of the fragmentation is on one disk, but in the real world, fragmentation will be spread across all of the disks. If the fragmentation is spread equally on all disks (best case), the percentage of slowdown will be the same as for a single disk. Any other distribution will be worse. So, at best, the stripe set will be no better off than a single disk. In most cases, the effects of fragmentation will be greater.