How File Fragmentation Occurs On Windows® XP / Windows Server™ 2003

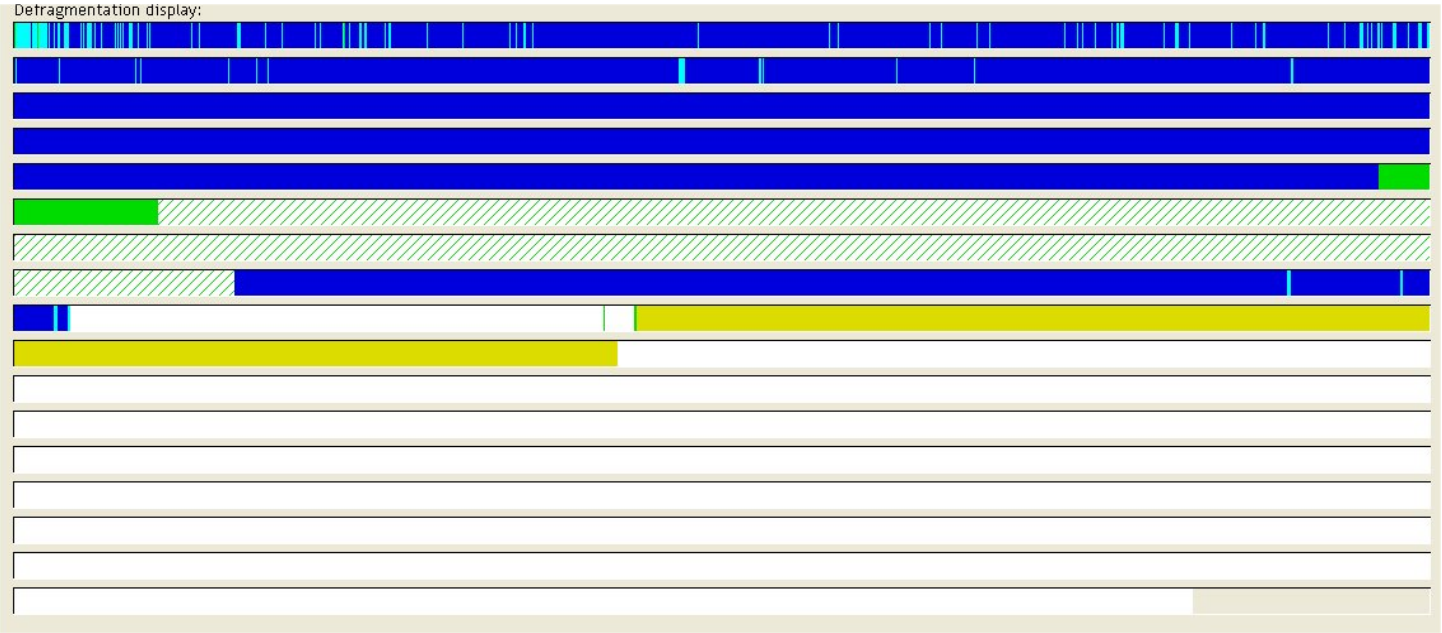
From: The Diskeeper® Development Team

A while ago we did a little research project to find out exactly how fragmentation occurs on Windows XP. It was sparked by some users at a past trade show that mentioned that having all the free space defragmented prevented newly-created files from fragmenting, and we had wanted to research if that was actually true.

Well, it turns out that Windows XP/2003 fragments files whether or not all the free space is defragmented.

Let's take a look at it.

Here's a picture of a lead programmer's D: volume, the volume usually used for running main development operations.



Note: screen capture created with Diskeeper 7.0 SE.

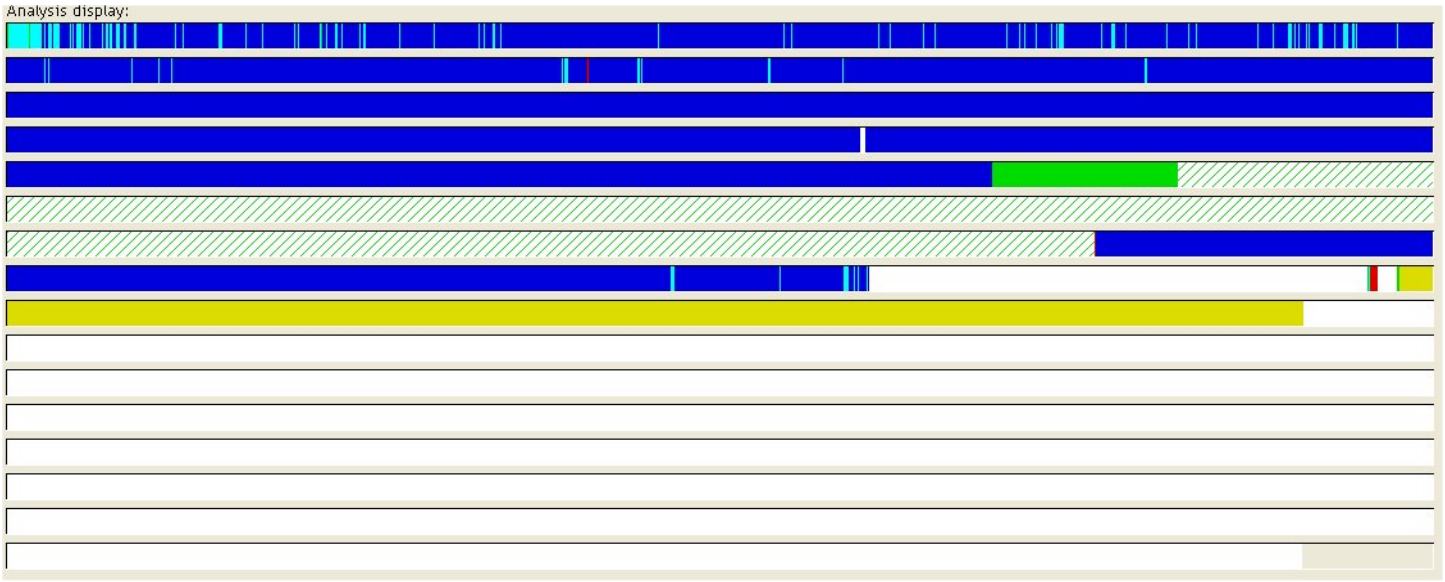
And here's the free space distribution on that volume:

```
There were       3 space(s) of size        1 cluster
There were       2 space(s) of size        4 clusters
There were       1 space(s) of size      844 clusters
There were       1 space(s) of size    14702 clusters
There were       1 space(s) of size    80000 clusters
There were       1 space(s) of size   288285 clusters
```

Well that's pretty well consolidated (we can thank Diskeeper's free space improvement feature for that).

Now, to continue the experiment we launched Internet Explorer and navigated directly to a favorite web site at http://www.cnet.com/.

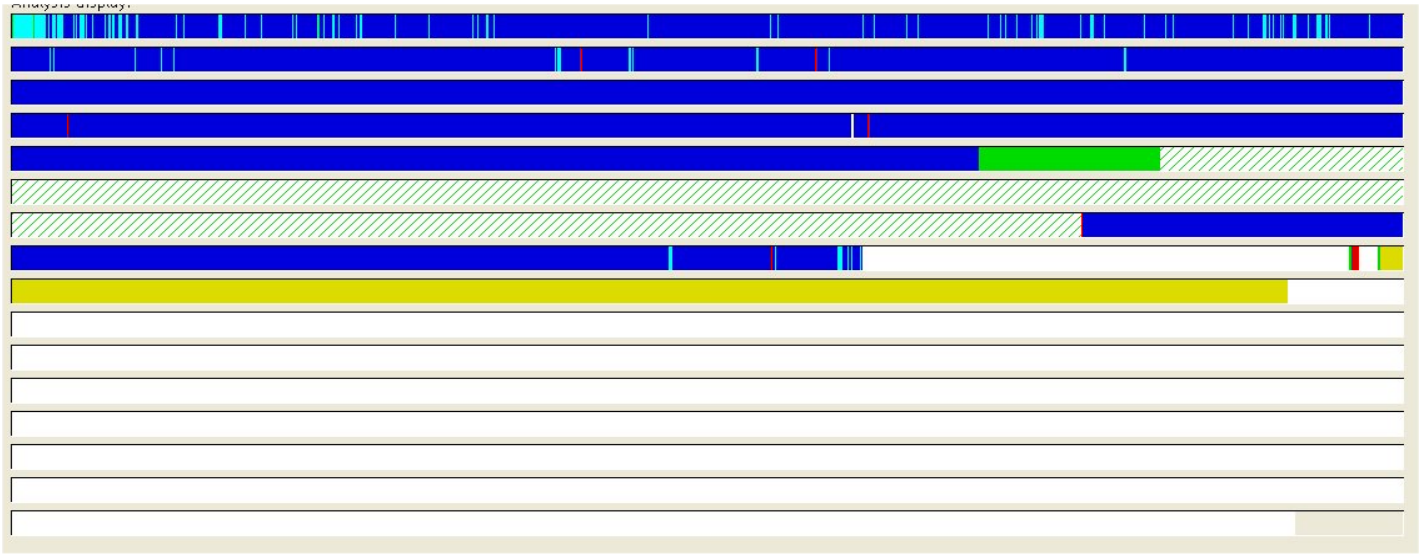
After loading that page, the volume looks like this:



Note: screen capture created with Diskeeper 7.0 SE.

Then we went to another favorite web site at: http://www.ucomics.com/foxtrot/

After loading those two pages, the volume looks like:



Note: screen capture created with Diskeeper 7.0 SE.

Whoa! What happened? Didn't we have the free space pretty well consolidated? Wasn't that supposed to prevent the fragmentation of newly arriving files? Let's look at what really happened.

The first page ended up putting some fragmented files at the end of the MFT zone and then after some NTFS metadata. The second page ended up putting some more files in the area after the metadata and then in some free spaces scattered around the existing files.

Why did that happen? Well, for one thing, these files are files in the Internet Explorer Temporary Files cache, which is a dynamically changing file set, with files being deleted and added on an as-needed basis. When internet temporary cache files get deleted, that makes free spaces! Even though the free space was very well consolidated, the fact that some files got deleted—as the very first activity after total defragmentation—there were enough free spaces created to ruin the "nicely consolidated files" and "nicely consolidated free space".

For another thing, the free space allocation algorithms in XP/2003 can actually contribute to the rate of fragmentation. Let's just take the NTFS situation as an example.

The XP/2003 NTFS file system driver maintains a list of the largest free spaces on the volume. On a volume that started out as consolidated as my volume, ALL of the free spaces were in the list and became candidates for allocation.

When a file gets created, it gets created in the free space that most closely matches the size of data available to write, in other words a "best fit". Additionally, a presumption is made that a newly created file will end up larger than the size that is currently available for the operating system to write, and extra free space, an “overallocation”, is reserved for the file so as to prevent the file from fragmenting (see Microsoft Knowledge Base article Q228198). That presumption is that the file will be 2, 4, 8 or 16 times larger than the currently known data size, depending on how much data is currently available for writing to the file in the operating system’s file cache.

The file data is written to the volume, and the file is closed. Any overallocation is released, returning to the free space pool in the NTFS file system driver if it qualifies as one of the largest free spaces on the volume. In our extremely consolidated case, of course, it will. And, because we were loading pages that had more than one graphic file per page, more than one cache file was being created simultaneously.

Note that the simple business of just saving a few internet temporary cache files will result in the fragmentation of free space by the simple act of releasing the overallocation! This happened despite the fact that we had made the D: volume a 16K cluster NTFS volume! Imagine the possibilities if we’d used a 4K cluster or a 512-byte cluster size!

This simple Internet access ended up with 6 fragmented files, even though the free space was highly consolidated. The fragmentation was actually aggravated by having most of the free space in four very large chunks and five smaller chunks; the smaller chunks being considered better candidates for "best fitting" the internet temporary cache files.

Of course, it can be argued that if those five small chunks weren't there, they wouldn't have been considered candidates. But also note that the internet temporary cache was getting files deleted from it simultaneously and those free spaces were being added to the free space list being maintained in the NTFS file system driver.

The bottom line is this: on an active volume, with a dynamic operating system and dynamic utilities, the disk volume free space situation is constantly changing. This, in conjunction with the algorithms used to allocate the free space, defeat the very best efforts of any defragmenter the very next time you go off and surf the web. This just underscores the necessity to *proactively address fragmentation* with regular automated file defragmentation.

And "consolidating every piece of free space" is no guarantee that newly arriving files won't be fragmented by the file system. That was a maxim on FAT volumes in DOS, but certainly is not applicable to NTFS or FAT volumes under Windows XP/2003.

We invite you to do some experiments and see for yourself.

Get rid of fragmentation on all your systems. Visit diskeeper.com/wpquote and get a no-obligation quote now!