

Virtual Memory in Windows XP

This page attempts to be a stand-alone description for general users of the way Virtual Memory operates in Windows XP. Other pages on this site are written mainly for Windows 98/ME and, while a lot is in common, there are significant differences in Windows XP.

What is Virtual Memory?

A program instruction on an Intel 386 or later CPU can address up to 4GB of memory, using its full 32 bits. This is normally far more than the RAM of the machine. (The 32nd exponent of 2 is exactly 4,294,967,296, or 4 GB. 32 binary digits allow the representation of 4,294,967,296 numbers — counting 0.) So the **hardware** provides for programs to operate in terms of as much as they wish of this full 4GB space as **Virtual Memory**, those parts of the program and data which are currently active being loaded into **Physical Random Access Memory (RAM)**. The processor itself then translates ('maps') the virtual addresses from an instruction into the correct physical equivalents, doing this on the fly as the instruction is executed. The processor manages the mapping in terms of **pages** of 4 Kilobytes each - a size that has implications for managing virtual memory by the system.

What are Page Faults?

Only those parts of the program and data that are currently in active use need to be held in physical RAM. Other parts are then held in a **swap file** (as it's called in Windows 95/98/ME: Win386.swp) or **page file** (in Windows NT versions including Windows 2000 and XP: pagefile.sys). When a program tries to access some address that is not currently in physical RAM, it generates an **interrupt**, called a **Page Fault**. This asks the system to retrieve the 4 KB page containing the address from the page file (or in the case of code possibly from the original program file). This — a **valid** page fault — normally happens quite invisibly. Sometimes, through program or hardware error, the page is not there either. The system then has an 'Invalid Page Fault' error. This will be a fatal error if detected in a program: if it is seen within the system itself (perhaps because a program sent it a bad request to do something), it may manifest itself as a 'blue screen' failure with a STOP code:

If there is pressure on space in RAM, then parts of code and data that are not currently needed can be 'paged out' in order to make room — the page file can thus be seen as an overflow area to make the RAM behave as if it were larger than it is.

What is loaded in RAM?

Items in RAM can be divided into:

- **The Non-Paged area.** Parts of the System which are so important that they may never be paged out - the area of RAM used for these is called in XP the 'Non-Paged area'. Because this mainly contains core code of the system, which is not likely to contain serious faults, a Blue Screen referring to 'Page Fault in Non-Paged area' probably indicates a serious hardware problem with the RAM modules, or possibly damaged code resulting from a defective Hard disk. It is, though, possible that external utility software (*e.g.* Norton) may put modules there too, so if such faults arise when you have recently installed or updated something of this sort, try uninstalling it.
- **The Page Pool** which can be used to hold:
 - Program code,
 - Data pages that have had actual data written to them, and
 - A basic amount of space for the file cache (known in Windows 9x systems as Vcache) of files that have recently been read from or written to hard disk.

Any remaining RAM will be used to make the file cache larger.

Why is there so little Free RAM?

Windows will **always** try to find some use for **all** of RAM — even a trivial one. If nothing else it will retain code of programs in RAM after they exit, in case they are needed again. Anything left over will be used to cache further files — just in case they are needed. But these uses will be dropped instantly should some other use come along. Thus there should rarely be any significant amount of RAM 'free'. That term is a misnomer — it ought to be 'RAM for which Windows can currently find no possible use'. The adage is: '**Free RAM is wasted RAM**'. Programs that purport to 'manage' or 'free up' RAM are pandering to a delusion that only such 'Free' RAM is available for fresh uses. That is not true, and these programs often result in reduced performance and may result in run-away growth of the page file.

Where is the page file?

The page file in XP is a hidden file called pagefile.sys. It is regenerated at each boot — there is no need to include it in a backup. To see it you need to have **Folder Options | View** set to 'Show Hidden and System files', and **not** to 'Hide Protected mode System files'.

In earlier NT systems it was usual to have such a file on each hard drive partition, if there were more than one partition, with the idea of having the file as near as possible to the ‘action’ on the disk. In XP the optimisation implied by this has been found not to justify the overhead, and normally there is only a single page file in the first instance.

Where do I set the placing and size of the page file?

At **Control Panel | System | Advanced**, click **Settings** in the “Performance” Section. On the **Advanced** page of the result, the current total physical size of all page files that may be in existence is shown. Click **Change** to make settings for the Virtual memory operation. Here you can select any drive partition and set either ‘Custom’; ‘System Managed’ or ‘No page file’; then always click **Set** before going on to the next partition.

Should the file be left on Drive C:?

The slowest aspect of getting at a file on a hard disk is in head movement (‘seeking’). If you have only one physical drive then the file is best left where the heads are most likely to be, so where most activity is going on — on drive C:. If you have a second physical drive, it is in principle better to put the file there, because it is then less likely that the heads will have moved away from it. If, though, you have a modern large size of RAM, actual traffic on the file is likely to be low, even if programs are rolled out to it, inactive, so the point becomes an academic one. If you do put the file elsewhere, you should leave a small amount on C: — an initial size of 2MB with a Maximum of 50 is suitable — so it can be used in emergency. Without this, the system is inclined to ignore the settings and either have no page file at all (and complain) or make a very large one indeed on C:

In relocating the page file, it must be on a ‘basic’ drive. Windows XP appears not to be willing to accept page files on ‘dynamic’ drives.

NOTE: If you are debugging crashes and wish the error reporting to make a kernel or full dump, then you will need an initial size set on C: of either 200 MB (for a kernel dump) or the size of RAM (for a full memory dump). If you are not doing so, it is best to make the setting to no more than a ‘Small Dump’, at **Control Panel | System | Advanced**, click **Settings** in the ‘Startup and Recovery’ section, and select in the ‘Write Debug information to’ panel

Can the Virtual Memory be turned off on a really large machine?

Strictly speaking Virtual Memory is always in operation and cannot be “turned off.” What is meant by such wording is “set the system to use no page file space at all.”

Doing this would waste a lot of the RAM. The reason is that when programs ask for an allocation of Virtual memory space, they may ask for a great deal more than they ever actually bring into use — the total may easily run to hundreds of megabytes. These addresses have to be assigned to *somewhere* by the system. If there is a page file available, the system can assign them to it — if there is not, they have to be assigned to RAM, locking it out from any actual use.

How big should the page file be?

There is a great deal of myth surrounding this question. Two big fallacies are:

- The file should be a fixed size so that it does not get fragmented, with minimum and maximum set the same
- The file should be 2.5 times the size of RAM (or some other multiple)

Both are wrong in a modern, single-user system. A machine using Fast User switching is a special case, discussed below.)

Windows will expand a file that starts out too small and may shrink it again if it is larger than necessary, so it pays to set the initial size as large enough to handle the normal needs of *your* system to avoid constant changes of size. This will give all the benefits claimed for a ‘fixed’ page file. But no restriction should be placed on its further growth. As well as providing for contingencies, like unexpectedly opening a very large file, in XP this **potential** file space can be used as a place to assign those virtual memory pages that programs have asked for, but never brought into use. Until they get used — probably never — the file need not come into being. There is no downside in having potential space available.

For any given workload, the total need for virtual addresses will not depend on the size of RAM alone. It will be met by the **sum** of RAM and the page file. Therefore in a machine with small RAM, the extra amount represented by page file will need to be larger — not smaller — than that needed in a machine with big RAM. Unfortunately the default settings for system management of the file have not caught up with this: it will assign an initial amount that may be quite excessive for a large machine, while at the same leaving too little for contingencies on a small one.

How big a file will turn out to be needed depends very much on your work-load. Simple word processing and e-mail may need very little — large graphics and movie making may need a great deal. For a general workload, with only small dumps provided for (see note to ‘Should the file be left on Drive C:?’ [above](#)), it is suggested that a sensible start point for the initial size would be the **greater** of (a) 100 MB or (b) enough to bring RAM plus file to about 500 MB. **EXAMPLE:** Set the Initial page file size to 400 MB on a computer with 128 MB RAM; 250 on a 256 MB computer; or 100 MB for larger sizes.

But have a high Maximum size — 700 or 800 MB or even more if there is plenty of disk space. Having this high will do no harm. Then if you find the actual pagefile.sys gets larger (as seen in Explorer), adjust the initial size up accordingly. Such a need for more than a minimal initial page file is the best indicator of benefit from adding RAM: if an initial size set, for a trial, at 50MB never grows, then more RAM will do nothing for the machine's performance.

Bill James MS MVP has a convenient tool, 'WinXP-2K_Pagefile', for monitoring the actual usage of the Page file, which can be downloaded [here](#).

Note that these aspects of Windows XP have changed significantly from earlier Windows NT versions, and practices that have been common there may no longer be appropriate. Also, the 'PF Usage' (Page File in Use) measurement in **Task Manager | Performance** for 'Page File in Use' include those **potential** uses by pages that have not been taken up. It makes a good indicator of the adequacy of the 'Maximum' size setting, but not for the 'Initial' one, let alone for any need for more RAM.

Should the drive have a big cluster size?

While there are reports that in Windows 95 higher performance can be obtained by having the swap file on a drive with 32K clusters, in Windows XP the best performance is obtained with 4K ones — the normal size in NTFS and in FAT 32 partitions smaller than 8GB. This then matches the size of the page the processor uses in RAM to the size of the clusters, so that transfers may be made direct from file to RAM without any need for intermediate buffering

What about Fast User Switching then?

If you use Fast User Switching, there are special considerations. When a user is not active, there will need to be space available in the page file to 'roll out' his or her work: therefore, the page file will need to be larger. Only experiment in a real situation will establish how big, but a start point might be an initial size equal to half the size of RAM for each user logged in.

Problems with Virtual Memory

It may sometimes happen that the system give 'out of memory' messages on trying to load a program, or give a message about Virtual memory space being low. Possible causes of this are:

- The setting for Maximum Size of the page file is too low, or there is not enough disk space free to expand it to that size.

- The page file has become corrupt, possibly at a bad shutdown. In the Virtual Memory settings, set to “No page file,” then exit System Properties, shut down the machine, and reboot. Delete PAGEFILE.SYS (on each drive, if more than just C:), set the page file up again and reboot to bring it into use.
- The page file has been put on a different drive without leaving a minimal amount on C:.
- There is trouble with third party software. In particular, if the message happens at shutdown, suspect a problem with Symantec’s Norton Live update
- It is also reported that spurious messages can arise if NAV 2004 is installed. If the problem happens at boot and the machine has an Intel chipset, the message may be caused by an early version (before version 2.1) of Intel’s “Application Accelerator.” Uninstall this and then get an up-to-date version from Intel’s site.
- Another problem involving Norton Antivirus was recently discovered by MS-MVP Ron Martell. However, it only applies to computers where the pagefile has been manually resized to larger than the default setting of 1.5 times RAM — a practice we discourage. On such machines, NAV 2004 and Norton Antivirus Corporate 9.0 can cause your computer to revert to the default settings on the next reboot, rather than retain your manually configured settings. (Though this is probably an improvement on memory management, it can be maddening if you don’t know why it is happening.)
- Possibly there is trouble with the drivers for IDE hard disks; in Device Manager, remove the IDE ATA/ATAPI controllers (main controller) and reboot for Plug and Play to start over.
- With an NTFS file system, the permissions for the page file’s drive’s root directory must give “Full Control” to SYSTEM. If not, there is likely to be a message at boot that the system is “unable to create a page file.”