About Safe & Sound

Thank you for purchasing Safe & Sound, a collection of the finest utilities available for diagnosing, repairing and protecting your PC and its valuable data. With Safe & Sound, you can diagnose and repair your system and application software; create a recovery disk and image snapshots of critical sectors of your hard drives; locate and protect against computer viruses; avoid lost data during application crashes; automatically or interactively create backup sets; back up to a protected volume file which makes your data recoverable when it otherwise would not be; and ensure that your PC meets Year 2000 hardware compliancy.

Safe & Sound also places a wealth of information about your PC system at your fingertips with Discover.

From the moment you install Safe & Sound, it begins protecting your PC. First, by guiding you to create a recovery disk that you can use later if problems arise. Second, by activating Bomb Shelter, which immediately protects you against losing unsaved information in the event that an application crashes. Next you'll want to use Retake to create a backup set of your drives' valuable data that you can restore in case any unrecoverable problems arise.

Related Topics

Safe & Sound Overview

Safe & Sound Overview

When you start Safe & Sound, the Safe & Sound Central window appears. This window offers buttons to take you to the primary utilities (PC Checkup, Retake, and Virus Scanner). It also offers a Tools button that displays a menu of other protection and repair tools that you can use.

Safe & Sound is made up of these utility programs:

- **PC Checkup** thoroughly analyzes the state of your PC's hardware and software. It can repair many software and configuration problems for you. When PC Checkup finishes its diagnosis, it produces a full report of any problems. You can let PC Checkup fix them, or perform repairs yourself.
- **Retake** lets you create automatic or interactive backups of selected drives, directories, files, or file types. You can back up to a protected volume file (a separate area on the drive). A protected volume file contains information about each file in every sector to ensure that files can be recovered even if the hard drive's directories and data are severely damaged or lost. You can also create mirror backups that instantly back up data as you save it, make backups after a time delay when the PC is idle, or create manual backups.
- **Discover** shows a wealth of hardware and software information about your PC. You can run benchmark tests to see how your PC's performance measures up against a comparable computer.
- **Bomb Shelter** protects various critical parts of your Windows system from being overwritten by other programs. It also acts as a safety net, allowing you to recover from application crashes (and save your work) rather than being forced to restart your computer and lose any unsaved data.
- **Virus Scanner** lets you scan your computer's memory, boot sector and drives for computer viruses at any time.
- **Instant Update** connects to the Internet and checks the Network Associates website to see if your copy of Safe & Sound needs updating.
- **Year 2000 Checker** lets you test your system hardware for Year 2000 compliancy. If it is not, Year 2000 Checker installs Y2Kfixer.com to make your PC compliant so it handles dates accurately starting on 01-01-2000.
- **McAfee Image** saves an "image" copy of critical disk information in a file, and restores the image later if your disk is corrupted.
- **Rescue Disk** creates a recovery disk that you can use to boot up the PC and begin recovery if you can't start from the system hard drive.

Understanding Windows Memory

This topic explains the advanced subject of how Windows handles memory. You can skip this topic, without learning about memory, and still achieve optimal results with Safe & Sound. Safe & Sound, with its intelligent default settings, is carefully designed to ensure no tweaking is required.

This topic is offered to help you learn about the more crucial details of how Windows handles memory. You may decide to learn about memory to better understand the real-time memory usage gauges that WinGauge shows you so you can squeeze every last ounce of performance out of your system. Or you might simply want to stay informed, or impress your friends and co-workers with all you've learned about the inner workings of your computer.

The Three Main Types of Memory
Windows memory is divided into three main types:

- Physical memory
- Virtual memory
- Linear memory

Physical Memory

Physical memory, also called Random Access Memory (RAM), is the total amount (typically measured in megabytes, or MB) of all the RAM chips that are installed on small SIMM or DIMM boards connected to sockets in the main board inside your computer.

WinGauge offers a Physical Memory gauge available in your choice of display types.

A SIMM, or Single In-line Memory Module, contains a single row of RAM chips. A DIMM, or Dual In-line Memory Module, contains a double row of RAM chips. If you want to increase your computer's physical memory, you purchase RAM as either SIMMs or DIMMs, depending upon the available expansion slots inside your PC. For more information, see your PC manual, or main board manual.

Tip RAM chips on SIMMs or DIMMs are rated at a timing speed in nanoseconds (ns), such as 60 ns or 70 ns. Likewise, the computer is designed to handle RAM at one or more timing speeds. Installing faster RAM than your computer is designed to handle might not harm your computer (the RAM may work fine), but it cannot increase performance beyond the computer's fastest RAM timing speed.

The random access part of RAM's name comes from the fact that the next bit of information can be located--no matter where it is in the RAM--in an equal amount of time.

This means that access of information to and from RAM memory is extraordinarily fast. On the other hand, RAM is volatile, which means that if you turn off the computer or experience a power interruption, the information stored temporarily in RAM is lost unless it is protected with a functional battery.

Physical memory is a fast, temporary “active working place” for your computer, its software, and your information. When you are ready to save information (or when Windows or an application determines it is time to do so), information in RAM is copied to a more permanent form of storage (a 3.5-inch disk, a hard drive, a ZIP drive, a tape cartridge, and so on).

How Windows Uses Physical Memory

Physical memory is the primary memory in your computer, where the computer temporarily places a copy of information so you can actively work with it. So, when you start Windows, portions of the Windows software is copied into physical memory so it can display information on your screen and respond to your actions with input devices like the keyboard and mouse.

Windows takes control of a portion of your computer's physical memory, offering memory management schemes for allocating pieces of memory to Windows applications as you run them. Applications need to be copied into physical memory while you work with them, as do the portions of their documents (spreadsheets, graphic images, sounds, and so on) that you're actively working with.

For example, if you open a spreadsheet document, Windows copies the spreadsheet application into RAM, and the application then copies the document into RAM. If you recalculate values in your spreadsheet, the application performs the calculations and temporarily places the results in RAM. If you save the results, the application copies this information from RAM to a disk or hard drive you specify.

Unlike memory, disks are a more permanent storage media because the information they contain is preserved when power is interrupted to the drive.

Virtual Memory

When a Windows application requests memory, Windows allocates what is called linear memory to that application. We'll discuss linear memory later in Linear Memory.

Depending upon how much physical memory is available, Windows 'fills' an application's linear memory with physical memory or virtual memory. Virtual memory is actually hard disk storage space that Windows reserves in the form of a swap file on your system's hard drive.

Tip Virtual memory usage is important, no matter which version of Windows you're using. When your computer runs low on both RAM and virtual memory, Windows will attempt to supply applications that request memory by finding application segments that are discardable and discarding them, and then reallocating the freed up memory. Windows discards segments arbitrarily, rather than based on least recently used segments, which is a situation best to avoid, as explained in Discardable Memory.

Windows can move virtual memory 4 K at a time, which is more efficient than how it reallocates entire discardable segments (which is 64 K at a time).

Virtual Memory in Windows 3.1x

Windows creates virtual memory when you first install Windows, or when you modify the settings in the Control Panel's Virtual Memory dialog box. The reason this hard disk space is called virtual memory is that Windows uses the disk space to store parts of physical memory (RAM) that are not in active use. This allows Windows to, at any given time, run and store more information and applications than can fit in all of your computer's physical memory.

The advantage of this approach--of letting Windows temporarily and automatically remove parts of memory to disk--is that your applications can rely on Windows to provide them with as much memory as they need. Letting Windows take care of memory management issues, including virtual memory, gives you the best possible performance on your PC. Also, it makes life easier for application developers who can avoid developing their own memory management schemes that rely on the hard disk in some other, more complex, way.

Windows also provides a secondary memory management scheme for configurations that don't rely on a disk-based swap file. This is discussed in the 'Discardable Memory' section later in this topic.

Virtual Memory in Windows 95

In Windows 95, the virtual memory swap file is dynamic. That is, depending upon what you're doing with your computer, the swap file will get larger or smaller. If you use Windows 95, you probably don't need to select virtual memory settings as was required in Windows 3.1x. Instead, you can let Windows 95's 'intelligent system defaults' configure the virtual memory for you. If you're technically advanced, you can still configure virtual memory yourself in Windows 95 if you prefer.

Linear Memory

Of the three types of memory, linear memory is the hardest to understand. Linear memory is artificial memory that is only vaguely related to the other two memory types (physical and virtual memory).

WinGauge offers a Linear Memory gauge (most important if you're using Windows 3.1x) available in your choice of display types.

Linear Memory in Windows 3.1x

When you start Windows, it determines the linear memory by multiplying the free physical memory (RAM) available to it by 4 (or by whatever value is defined in your SYSTEM.INI file's PageOverCommit setting, such as multiplying free memory by 5 if this setting is: PageOverCommit=5).

linear memory = free RAM x 4 (or PageOverCommit=#)

The free physical memory that Windows uses in this calculation is most likely less than the total of your RAM because of DOS-based applications that are using conventional and extended memory.

For the duration of this Windows session, this number (4 times free RAM), is the maximum amount of linear memory available to all applications running under Windows, even to DOS applications running under Windows (in a MS-DOS Prompt window).

Note If you exceed the linear memory amount calculated by Windows 3.1x, no matter how large the virtual memory swap files are, Windows behaves as though your computer has run out of memory. So the Linear Memory gauge in WinGauge is important if you're using Windows 3.1x. WinGauge can alert you when linear memory is running low with the Linear Memory gauge.

Each time an application asks Windows for memory, Windows gives it a portion of this linear memory range. When Windows allocates linear memory to an application, no physical memory is initially associated with the linear memory range. Windows does this to intelligently allocate physical memory to applications as they actually need to use it. So, when an application tries to use its linear memory range, Windows detects an 'error condition,' because the application is trying to use a memory region with no RAM associated with it.

If RAM is available when an application needs to use it, Windows fills the application's linear memory range with RAM. If no RAM is available, Windows searches through all the other ranges of linear memory allocated to other applications that do have RAM associated with them.

Windows takes the least recently used range of linear memory that has some RAM assigned to it, and writes the contents of the RAM to the virtual memory swap file (in case it is needed later) and then reassigns the RAM to the new range, marking the older, less used range as empty.

Should the older information that's been placed in the virtual memory swap file on disk be needed, Windows goes through this whole procedure again, reassigning ranges of linear memory to disk.

This process even applies to your DOS applications. Each time you open a DOS Prompt window inside Windows, a (roughly) 640 K chunk of linear memory is assigned to it. However, if you run a DOS application that only uses 250 K of the 640 K, Windows need only assign 250 K of RAM to the DOS Prompt window.

So, physical memory is RAM, and you can think of linear memory as room. Linear memory is the space within which the RAM is put to use by Windows applications. RAM. Room.

Seeing is Understanding

The DiscoverPro Linear Memory pie chart shows what ranges of linear memory have been assigned to which applications. The DiscoverPro Physical Memory pie chart shows how much, and which parts of, your RAM are in use.

Take a look at the linear memory assigned to a DOS Prompt window in DiscoverPro. To do this, double-click the MS-DOS Prompt icon in the Main group of the Program Manager. Then, start DiscoverPro. Check how much linear memory is taken by the DOS Prompt window's purple region of the pie chart. You'll see this is somewhat more than 640K. This is the linear memory, or room where the DOS application will run.

If you look at your physical memory, before and after you create the DOS Prompt window, you'll see that only around 250K of physical memory (RAM) has been used. So your DOS Prompt window's room is less than half full of RAM. Windows won't fill the rest of the linear memory for the DOS Prompt window with RAM unless you run a program that actually needs more than the 250K to run.

Linear Memory Recap

So linear memory is the space or (room) where applications can run, and physical memory is the usable building materials allocated to the space when and where it's needed.

Or consider another analogy. When you go on vacation, you might make hotel reservations. When you do this, the hotel reserves a room in your name, but it does not actually give you the keys to the room until you arrive at their front desk. If you don't actually show up, the hotel may eventually make the room reserved for you available to another patron who has arrived.

So a hotel reservationist is like Windows--both are in the business of allocating 'rooms.' When you make a hotel reservation, it's like a Windows application requesting RAM before it actually needs to use it. And when you sign in at a hotel's front desk, it's comparable to a Windows application trying to access its assigned range of linear memory.

When you sign in, the hotel selects a room, from those available, for you to stay in. When a Windows application tries to access its linear memory, Windows is alerted and attempts to provide physical memory to the application (that is, hand it the keys to RAM room) for the tasks at hand.

Linear Memory in Windows 95

Windows' linear memory model in Windows 95 consists of three regions rather than one. The original linear memory region found in Windows 3.1xx is now called the system region. Two additional linear memory ranges have been added:

- The private memory region where Windows 32-bit applications run
- The shared memory region where memory mapped files reside and where Windows 32-bit applications share information with each other and with Windows 16-bit applications

Don't be concerned that Windows 95 offers three ranges of linear memory rather than just one. Linear memory is still just room--there's simply more room in Windows 95, and there's no need for a PageOverCommit setting because Windows 95 can grow its linear memory address range dynamically, almost to the full 4 GB (gigabyte) capacity of the 386 processor's architecture. This means that Windows 95 should never run out of linear memory. So if you use Windows 95, you won't need to be concerned about linear memory usage.

It's still just room, which is only useful once Windows allocates physical memory (RAM) to a range of linear memory. If you run out of RAM, you still need to save something that is in RAM to disk (that is, you should save documents that you have made changes to since the last time you saved). Once you save these changes, Windows can use that RAM to fill some other room.

In Windows 95, 16-bit applications still operate exactly as they did in Windows 3.1x, discardable regions and all. For 32-bit applications, however, the discardable memory concept has been discarded. Windows 32-bit application files are accessed as memory mapped files and therefore do not need to be paged out to disk as often. Otherwise, they operate in much the same way as their 16-bit counterparts.

Memory Mapped Files

The Memory Mapped Files concept is simple and elegant. A memory mapped file is simply a file whose entire length has been assigned linear memory (room). When an application accesses any part of the linear memory assigned to the file, the corresponding part of the file is read into RAM, and the RAM is placed at the linear address that was accessed by the application.

If the RAM is needed, Windows can tell if the file region has been modified, and if not, it can simply throw away the contents of the RAM, knowing that the information is still stored on disk. If the file has been modified, the contents of the RAM are written to the file before the RAM can be reused.

For application files, this makes life simple. Windows merely assigns linear memory to the whole application and then starts executing it. As various parts of the application are needed, they are loaded into RAM. When a part of the application has not been used for a while, it can be discarded and reloaded as needed.

This, however, is not as straightforward for data files, where it is frequently more possible to design the application accessing the file so that it is more efficient than the memory mapped file mechanism, which relies on the caching mechanism of the operating system and the (relatively slow) faulting mechanism of the CPU (central processing unit).

What is Free Memory?

The Windows 3.1x Program Manager, and its various counterparts display a value they call free memory. To see this value, you can choose About Windows from the Help menu in the Program Manager.

Unfortunately, this free memory amount does not directly relate to any of the above we've discussed. Free memory is actually the approximate sum of the:

- Usable physical memory (we'll discuss what makes some physical memory unusable later in the 'Locked Memory' section)
- Unused part of the virtual memory swap file
- Discardable memory, which we'll discuss in a moment

Note Windows 95 does not display a free memory value, as Windows 3.1x does. Instead, Windows 95 shows the actual physical memory (RAM) in the About Windows 95 dialog box.

Some Windows applications require that the linear memory (or room) they allocate (or reserve) be associated with physical memory immediately. They may also require that the physical memory they are given never be swapped out to the virtual memory swap file.

This is usually for performance reasons, and sometimes because the contents of the memory must be instantly available to handle some external events.

For example, if you are connected via your modem to an on-line service such as to your Internet Service Provider (ISP) or to CompuServe or America Online, you wouldn't want the part of the application that actually handles the communications swapped out to disk. If this happened, it might take a long time for the disk to be read back into physical memory and some information might get lost in the interim.

Locked Memory

To prevent problems such as those described above, like time-sensitive information being accidentally lost or delayed, Windows allows applications to lock ranges of linear memory. This forces physical memory (RAM) to be associated with the linear memory range immediately, and prevents the linear memory range from being swapped to the virtual memory swap file (the room is locked so RAM can't get out), even if the physical memory is not used for some period of time.

When you exit an application that has locked a range of linear memory, or at any time that the application considers it appropriate, the application can unlock the linear memory.

Since the physical memory that is locked in this fashion cannot be swapped out to the virtual memory swap file, Windows does not consider it usable. Windows does included locked physical memory when calculating the current Free Memory on your computer.

We'll get back to free memory in a moment, but first, you need to understand discardable memory.

Discardable Memory

Both Windows 3.1x and Windows 95 applications are all divided into multiple small parts called segments. The size and contents of the various segments that make up a Windows application vary, and are dictated primarily by the programmers who developed that application.

In order to provide memory management for 286 systems, and in order to handle situations where there is not enough disk space for a large virtual memory swap file, Windows uses this secondary memory management scheme:

When Windows applications are developed, the programmers must indicate, as part of the development process, whether each of the segments that make up a Windows application, is discardable.

When Windows runs out of both physical memory, and available virtual memory in the swap file, Windows searches linear memory for segments of applications that are marked as discardable. The linear memory range occupied by these discardable segments is then simply freed and the contents of that linear memory segment are thrown away (discarded).

Windows does not consider whether a segment is the 'least recently used' (or LRU) when deciding what segment to discard. Furthermore, it discards an entire segment (which might be as much as 64 K), even though Windows may only need an additional 4 K of memory.

When the contents of a discardable segment are needed again, they are reloaded from the original .EXE or .DLL file that contains the entire application. Since Windows doesn't consider how likely a discardable segment will be needed again (by checking for a LRU segment), the odds increase that a discarded segment will be needed again soon.

No matter which version of Windows you use, you'll want to avoid this situation. That is, it's better if Windows does not need to obtain more physical or virtual memory by discarding segments and reallocating their memory.

For example, let's say that the part of an application (segment) that handles the About dialog box for an application is marked as discardable in the application file. When Windows runs out of other types of memory, it simply throws this segment away.

If you then try to access the About dialog box, Windows tries to recreate the segment by allocating some new linear memory, and reading the segment back into the linear memory from the application's file on disk.

Of course, reading the segment back in requires that physical memory, which can store the information, be associated with the linear memory range that has been allocated to hold the segment. If there isn't enough available physical memory, some other segment will need to be discarded first. And round and round we go...

So, the Program Manager's free memory value is, roughly speaking, the total of:

- The usable physical memory
- The available virtual memory swap file space
- The sum of all the discardable segments loaded in memory

It's actually even more complicated than this, but for the purposes of this Help topic, it's as much as we need to consider. This method of calculating free memory makes the resulting Windows 3.1x free memory value irrelevant for all practical purposes.

Your Windows system would become unusable long before you run your free memory down to a dangerously low level. (That being the point where there are not enough discardable segments around to allow Windows to load a needed segment). Your system will have been so slowed down by the loading and unloading of segments and the swapping and unswapping of virtual memory, that it would be spending all its processing time accessing the disk, and shuffling memory.

What's worse is that the free memory value might never hit zero because the system will lock up, unable to load a needed segment long before then. If you've ever seen the error message, Segment Load Failure, you've probably hit this barrier, and free memory was of no help.

Discardable Memory in Windows 3.1x

If you're using Windows 3.1x, the linear memory is more likely to run out, causing Windows to discard segments and reallocate memory. If it happens, you can increase the virtual memory swap file size and increase the PageOverCommit=# setting in the SYSTEM.INI file above it's current size, or if the setting doesn't appear, make it larger than the default of 4.

Tip It's a good idea to keep your Virtual Memory gauge at less than 50 percent full to ensure that Windows does not need to discard segments of applications that are marked as discardable.

Discardable Memory in Windows 95

If you're using Windows 95, the linear memory shouldn't run out, so you may never encounter this situation. If Windows 95 does begin discarding segments, you should free space on the Windows hard drive (the hard drive where the Windows folder is located, and where the virtual memory swap file is created).

WinGauge and Memory

The foregoing is why you won't see a free memory value in Safe & Sound' WinGauge. Instead, WinGauge shows you the key components of free memory so you can decide when to start saving or providing more memory by:

- Closing Windows applications
- Adding more physical memory (RAM)
- Increasing the size of your virtual memory swap file

The most important type of memory is, of course, your physical memory (RAM). As soon as you run out of free (completely unused) physical memory, Windows has to start manipulating your various other memories (linear, virtual, or discardable) to get your applications the physical memory that they need. In other words, your system will slow down slightly as soon as you run out of physical memory.

At that point, Windows will start using your virtual memory. You'll be able to see the virtual memory gauge in WinGauge start creeping upwards. You won't notice a serious performance degradation though, until Windows starts relying more heavily on the virtual memory swap file. The time it takes to access information in the virtual memory swap file (the hard disk's access time) is slower than the speed at which your computer can access physical memory.

WinGauge can show you both how much of your virtual memory swap file is in use, in the Swap File in Use gauge, and how heavy a use Windows is making of this swap file, in the Swap File Requests gauge. If the Swap File Requests gauge is moving upwards constantly, your system is relying heavily on virtual memory and you should consider closing some of your applications, or in the longer term, adding more physical memory (RAM) in the form of SIMMs or DIMMs that contain RAM chips.

WinGauge also shows how much memory you have in your discardable reserves with the Discardable Memory gauge. You're in real trouble when this value drops below 10%, and your physical memory and virtual memory are also heavily used up. Save your work, close some applications, and exit Windows.

Other WinGauge Gauges

Here are some other, more obscure parts of the Windows architecture that you can monitor using the gauges available to you in WinGauge.

The LDT Gauge

Note Windows 95 32-bit applications are designed to share a Local Descriptor Table (LDT), so this is an issue only affecting 16-bit applications that run in Windows 3.1x or Windows 95.

The application segments mentioned above, some of which may be discardable, are used and accessed through a special table called the Local Descriptor Table. This table, called the LDT for short, contains a descriptor that describes the position in linear memory of each segment (which is like a room number).

The LDT's format is fixed and defined by the architecture of the CPU, and although the CPU can have up to 8192 LDTs, each with 8192 descriptors for a maximum of 67,108,864 segments, Windows (by design), can only use one LDT, or 8192 segment. Thus, Windows creates the possibility of running out of room in the LDT.

WinGauge's LDT gauge indicates this, and if you see the LDT warning light activate, which it does when you exceed 90% usage of the LDT, you should immediately close some applications and save your work. If you find that a single application frequently uses up a big chunk of the LDT, you may want to let the developers of this application know that you are encountering this situation.

The CPU Gauge

WinGauge's CPU gauge measures CPU activity by monitoring how frequently the CPU is idle.

The Heap Gauges

The Heap gauges measure how much of the linear memory that has been allocated by Windows applications (not by DOS Prompt windows) is actually in use. A low reading (below 50 percent) indicates that some application is grabbing much more than it actually needs, or that you have memory leaks--portions of unused memory that are being left around by an application.

Global DOS Memory

One last aspect of Windows memory that is of concern is the region known in Windows-speak as 'global DOS memory.' Formerly this was known as conventional memory, and is actually and simply the first megabyte of memory in your system.

This memory is crucial because of the backwards compatibility (the ability to run older DOS programs and drivers) that is designed into Windows. In order to remain compatible with DOS and DOS programs, Windows uses the first megabyte as a DOS communications region, placing critical structures needed by DOS in the first megabyte, along with various buffers needed to communicate with DOS based drivers and the system BIOS (Basic Input/Output System). The BIOS is the built-in firmware that starts up your

PC. BIOS contains buffers for sending information from a program to the hardware device where that information should go and vice versa, such as to and from the keyboard.

Unfortunately, the global DOS memory (the first megabyte) gets pretty crowded, particularly because Windows has a tendency to put in the first megabyte things that don't really need to be there.

The Global DOS gauge shows how much global DOS memory you've got left. If you get much above 90% you may start getting Out of Memory errors from Windows.

The Global Heap and Local Heaps

To understand the rest of what follows you'll need a little background about heaps.

Windows consists of three components:

- Kernel is the program responsible for the multitasking of Windows programs
- User is the program responsible for managing windows, buttons, controls and menus
- GDI is the program responsible for drawing graphics on your screen or printer

One of the many jobs of the Windows Kernel is that of allocating linear memory. The Kernel is in charge of carving off chunks of linear memory and doling them out to applications on request. The Kernel actually gets linear memory from a Virtual Device (VxD) called the Virtual Memory Manager (VMM) which is the part of Windows that created the linear memory room. The Kernel takes big pieces of linear memory as it needs them, and then doles them out to applications when requested.

All the linear memory that has been allocated by the Kernel is called the global heap. Heap simply means a pile of linear memory--that is, a big room. Simply put, the global heap is that region of linear memory that is managed by the Windows Kernel. Applications call the Kernel to allocate parts of the global heap for their own use.

All Windows 16-bit applications, when they start in Windows 3.1x or Windows 95, are automatically allocated 64 K segments of the global heap. These segments are 64 K because that is the maximum amount of memory that a 16-bit application can access conveniently. These smaller 64 K segments, and other similar 64 K memory segments that are allocated by Windows applications, are known as local heaps.

The Kernel has built-in routines that help Windows applications manage these local heaps to make it easy for programmers to keep track of various program information.

To summarize, the global heap is, as far as Windows applications are concerned, all of linear memory. Local heaps are small 64 K segments of the global heap allocated for use by individual Windows applications.

Resources

You've heard a lot about resources, but what are they? And what are system resources and application resources?

Resources are simply objects you (or in this case, the Windows system and its applications) can use. In order to manage the various objects that appear on your screen, the User and GDI programs each maintain various information about those objects. For instance, for each button on your screen the Windows User program has to know:

- Where the button should be located on the screen
- What application owns the button
- Which part of that application must be notified when you click the button.

The GDI program must maintain information about input devices, such as pens used to draw lines on screen, and so on.

The information about the various objects managed by the User and GDI programs is kept in six 64 K segments of memory known as resource heaps. These regions are actually 64K local heaps. Again, these regions are 64 K because that is the maximum size a 16-bit application can deal with efficiently. User and GDI are 16-bit programs because Windows was originally designed to run on the Intel 8088 and 80286, 16-bit processors.

User Resource Heaps
The User Resource Heaps are divided into:

- The Window heap that contains information on windows and controls
- The Menu heap that contains information on drop-down menus
- The Menu String heap that contains the text messages that appear in the menus
- The User Atom heap that contains mostly Window titles

GDI Resource Heaps
WinGauge offers two GDI Heap gauges for tracking GDI resource usage for 16-bit and 32-bit applications.

The GDI Resource heaps are divided into:

- The main GDI Resource heap that contains pens, brushes, fonts and various other information related to displaying graphics
- The GDI Atom heap that contains some font-related information, including font names

The GDI and User Atom heaps are usually discounted because the information in them is subsidiary to the other local resource heaps. So, the other heaps would always fill up before the Atom heaps.

Free System Resources
When you choose the About command from the Help menu in Windows' Program Manager, it displays a Free System Resources number. The number displayed is the lowest free percentage of the resource heaps, not counting the Atom heaps. The percentage free is computed as the available space in the particular heap divided by 64 K.

Applications that show User and GDI numbers are actually displaying the percentage free space in the main GDI Resource Heap, and the lowest percentage free of the three main User Resource Heaps (the User and GDI Atom heaps are not included because the other heaps would necessarily run out first).

You can set WinGauge to display the lowest of the six (or in our case the highest, since WinGauge displays the percentage used rather than the free memory), or to track an individual heap so you can see how that particular one is affected by your applications.

Application Resources
Finally there are application resources. These are completely unrelated to system resources. Application resources are simply static, unchanging data items that are kept as part of the application's .EXE image. Typically icons, bitmaps, strings and fonts are kept as application resources and are loaded into memory as needed.

If an application resource must be used by the system, it changes from an application resource to a system resource when it is loaded into memory. For example an application's icon is an application resource until it is loaded into memory, when it needs to be displayed on the desktop, the icon becomes a system resource. Other application resources are unique to the application and always remain application resources, such as menus or bitmaps.

You can use Discover Pro to peek into application resources for any application that is currently running. You can see what is loaded in memory and what is on hand in the application's disk file.

That's it! Now you know everything there is to know about Windows' memory. Don't feel obliged to remember it all because Safe & Sound is designed to prevent you from running out of almost any critical commodity. Safe & Sound will let you know when it's time to add more RAM to your PC.

Contacting Network Associates

Before contacting Network Associates, check the Whatsnew.txt file for late-breaking information that may help you resolve any issues you are experiencing. If you don't find an answer in the Whatsnew.txt file, prepare for calling Network Associates by jotting down some basic information about your PC computer. For example, you should know the model, such as a Pentium, 586, or 686; how much memory your computer has, how much free disk space; and the version of Safe & Sound you are using.

You can contact Network Associates by mail at:

Network Associates, Inc.
2805 Bowers Avenue
Santa Clara, CA 95051

by phone or fax Monday through Friday from 6:00 am to 6:00 pm (PST) at

Single User Technical Support: **972-278-6100**
Corporate Technical Support: **408-988-3832**
Fax: **408-980-3660**

We also have provide electronic support resources at:

Forums:	http://support.nai.com/Forums/
E-mail:	support@nai.com
Internet:	http://www.nai.com
AOL Keyword:	MCAFEE
CompuServe:	GO MCAFEE

Address Space

The sum total of all possible memory addresses available at a given time. This is 4 GB (gigabytes) on a 386 or later PC in protected mode.

Launch Pad

The Launch Pad is a window where you can place application and document icons so you can conveniently access them.

Benchmarks

A benchmark is a standardized task that tests various devices for measurements, such as speed.

BIOS

The BIOS (or Basic Input/Output System) contains buffers for sending information from an application to the hardware device, such as a printer, where the information should go.

Buffers

A buffer is a temporary storage location for information being sent or received.

Bytes

A byte is eight bits of information composed of zeros and ones, one of which may be a parity bit. Most character sets, such as ASCII, use one byte to represent each character (letter, number, or special symbol).

Cache

A cache is part of the computer's memory used to temporarily store recently accessed information. A cache is designed on the premise that recently used information may be needed again soon. Keeping information available in cache reduces the time it takes for an application to obtain the information again.

Cluster

A cluster is a unit of storage allocation usually consisting of four or more 512-byte sectors.

Conventional Memory

Conventional memory is the first 640 K (kilobytes) of RAM (random access memory).

CPU (Central Processing Unit)

The “brain” of your computer. This is main computer chip that controls all activity that takes place on a computer.

Diagnostics

Diagnostics are tests run to detect faults in a computer system. Diagnostics tests are run to detect faults before they become serious problems so the faults can be corrected.

Directories

Directories are locations within a volume on a drive where you can store files or subdirectories. In Windows, directories are equivalent to folders that appear on the desktop in a drive window.

Discardable Memory

Discardable memory is memory used by an application that it has marked as discardable. Windows can reallocate the discardable memory to a different application if it needs to.

DLLs (Dynamic Link Libraries)

A DLL is an executable code module that can be loaded on demand and linked at run time. DLLs can be shared among multiple applications and independently updated, transparent to the applications. DLLs can also be unloaded when they are no longer needed.

DMA (Direct Memory Access)

DMA is a fast method of moving information from a storage device or LAN interface card directly to RAM which speeds processing time. DMA is direct memory access by a peripheral device that by-passes the CPU to save time.

Expanded Memory

DOS running on the Intel 80286, 80386, or 80486 family of computers can only address one megabyte of memory at one time. Expanded memory is the memory located between the base memory (either 512 K or 640 K) and one megabyte. Expanded memory is reserved by DOS for housekeeping tasks, such as managing information that appears on the screen.

Extended Memory

Memory above one megabyte in 80286 and higher PCs. Extended memory can be used for RAM disks, disk caches, or Windows, but it requires the CPU to run in a special mode (protected mode or virtual real mode).

FAT (File Allocation Table)

The FAT is an index to the location where all the information is stored on a floppy disk or hard drive. The FAT is extremely important because the system uses it to store and retrieve files containing information.

GDT (General Description Table)

The GDT is a table that is basic to the operation of protected mode. This table contains data structures (descriptors) that describe various regions of memory and how they may be accessed. Windows uses the GDT for system devices. *See LDT.*

Global Heap

The Global Heap is the general pool of memory available to Windows applications.

GPF (General Protection Fault)

An error condition caused by an application when it attempts to perform an operation not allowed by the operating system. Windows uses GPFs to determine and control the state of the currently executing application. GPFs that are unexpected by Windows cause a system error message to appear.

HMA (High Memory Area)

The HMA is the first 64 K of extended memory. If you use DOS 5.0, you can save memory by loading DOS into the HMA. Do this by adding the DOS=HIGH setting to your CONFIG.SYS file and restarting your PC.

Interrupt

A temporary suspension of a process caused by an event outside that process. More specifically, an interrupt is a signal or call to a specific routine. Interrupts allow peripheral devices, such as printers or modems, to send a call to the CPU requesting attention.

I/O (Input/Output) Device

An I/O device is any piece of computer hardware that can exchange information with the CPU. Examples of I/O devices include network cards, printers, speakers or other sound devices, or devices connected to the serial or parallel ports of your PC such as external modems.

Kernel

The Kernel is the part of a computer operating system that performs basic functions such as switching between tasks.

LDT (Local Descriptor Table)

The LDT is a secondary data structure table that contains additional information about various regions of memory and how they can be accessed. Windows uses the LDT for programs.

Linear Memory

Linear memory is the currently defined address space of the system that Windows uses to allocate memory to Windows applications.

Local Heap

The Local Heap is a region of memory allocated for local use by an application.

Locked Memory

Locked memory is memory used by an application that cannot be relocated or discarded by Windows.

Mapping

Mapping is the process of assigning physical memory (RAM) to a particular linear address range.

Mode Switch

A mode switch is a transition made by the CPU when changing from one mode of operation to another. For example, switching from real or protected mode, or a transition between different levels of protection. *See Ring 0, 1, 2, 3*.

Modules

A module is a device driver loaded by Windows.

Paging

The process of saving information stored in RAM to the swap file on the system hard drive so Windows can make the RAM available at a different linear address.

Parallel Port

The parallel port is a connector on the back of your PC and on some peripheral devices. With the appropriate driver software installed and a parallel cable connected to the parallel ports on your PC and a peripheral device, the two can communicate with each other. Parallel transmissions have no EIA standard, but most equipment follows a quasi-standard called the Centronics Parallel Standard.

PCI (Peripheral Component Interconnect) Bus

The PCI Bus is a local motherboard specification (that provides connector slots on the motherboard for installing peripheral cards). The PCI Bus, designed by Intel, offers a high performance, peripheral component level interface to the CPU bus.

Physical Memory

Physical memory is the RAM (Random Access Memory) installed in your PC. *See Random Access Memory (RAM).*

Protected Mode

A mode of operation of 80286 or later CPUs which allows access to more than 1 MB of memory.

RAM (Random Access Memory)

RAM (Random Access Memory) is also called physical memory. It is installed in your PC on SIMMs (Single Inline Memory Modules) or DIMMs (Dual Inline Memory Modules). RAM is volatile, extremely high-speed storage used by your computer for processing information.

Real Mode

A mode of 80286 or later CPUs, where the CPU operates substantially like an older 8086 CPU and can address directly only 1 MB of memory.

Resources

Resources are objects that Windows and its applications can use, such as the buttons on the screen that you can click.

Ring 0, 1, 2, 3

Different levels of protection in protected mode, where programs having varying degrees of freedom of operation. Ring 0 (zero) is least protected and has direct access to all hardware in the system.

Sector

A sector is a pie-shaped portion of a hard disk. A disk is divided into tracks and sectors. Tracks are complete circuits and are divided into sectors. Under DOS, a sector is 512 bytes.

Serial Port

A serial port is an input/output port (connector) that allows the transmission of information out at one bit at a time, as opposed to parallel which transmits eight bits, or one byte at a time.

Swap File

The swap file is created by Windows on the system hard disk. It uses the swap file to copy information stored in part of the linear address space so it can reallocate the RAM used at that location to another linear address space.

Swapping

Swapping is the process of saving to disk or restoring from disk the contents of RAM so that the RAM can be used elsewhere in linear memory.

System Resources

System resources are a series of data structures kept by Windows. System resources are managed by the Windows User and GDI programs and maintain information about objects that appear on your screen.

32BDA (32-Bit Disk Access)

32BDA is a process in Windows where the device driver that accesses the disk runs entirely as a 32-bit program at Ring 0 (zero).

32BFA (32-Bit File Access)

32BFA is a process in Windows where the DOS file operations are controlled by a program, or set of devices, that operate entirely as 32-bit programs at Ring 0 (zero).

Unlocked Memory

Unlocked memory is physical memory that Windows can copy to the swap file on disk, and whose linear address can be changed whenever Windows chooses.

UMB (Upper Memory Block)

The UMB is the area in memory between 640 K and 1 MB that have RAM mapped into them by memory managers, such as Network Associates' Netroom or MemMaker. *See Expanded memory.*

V86 Mode (Virtual 8086 Mode)

V86 mode is a mode of operation of 80386 or later CPUs where programs, originally designed to run in real mode, can run as sub-programs to a protected mode control program or operating system.

Video Memory

Video memory, called VRAM, is physical memory installed on your PC's video card that is used for displaying information on the screen.

Virtual Memory

Virtual memory is the amount of memory that exists either as physical memory (RAM) or on the hard drive (in the swap file). When a part of memory that is located in the swap file is accessed by an application, Windows reads the information into RAM.

VMs (Virtual Machines)

Virtual machines (also called Virtual DOS machines or VDMs) are created in Windows 95/98 when you open a MS-DOS Prompt window. The VDM is a software emulation of a separate computer, offering all the services that the DOS application expects of a PC.

VxDs (Virtual Device Drivers)

VxDs are used in Windows to communicate with all physical hardware in the system. This prevents any application from having direct access to a piece of hardware. Instead, it communicates only through the VxD for that hardware.

Windows Registry

The Windows 95/98 Registry file contains user, application, and computer-specific configuration information in a central location that was kept in various .INI files in Windows 3.1. The Registry contains settings that determine how your computer runs.