

The 32-bit Address Space

The following information pertains to processors capable of running 32-bit Microsoft Windows operating systems.

What is a 32-bit address space?

An address space is a logical area or space containing one or more locations that can be individually accessed or addressed. A 32-bit address space describes an area containing a maximum possible 4,294,967,296 locations or addresses. A 32-bit operating system, therefore, would typically be limited to access 4,294,967,296 pieces of computer information, or bytes (4 GB).

Address spaces on 32-bit Windows-compatible hardware

Computers that run Windows operating systems actually utilize two different kinds address spaces: a memory address space and an I/O port address space. The memory address space is where main memory (RAM) is accessed, while devices (network, video, disk, etc.) typically use both memory and I/O address spaces.

Furthermore, memory is accessed through two different kinds of address spaces: physical and virtual. The virtual address space is what applications and processes "see" and use. One of the most important purposes of virtual address space is that it can present to applications and processes more memory than is physically present in the machine. A virtual address may or may not correspond to a real, physical address. The physical address space is where real, physical memory, or memory-addressed devices, are accessed.

As noted above, a 32-bit processor's memory address space is used to access both main memory (RAM) and devices that require memory addresses. Since the devices (called "memory-mapped" devices) may consume a considerable amount of memory address space, much less than the full 4 GB address space may be available for accessing main memory. See [Memory Managed by Windows](#) to see how Windows handles this problem.

However, with special hardware and software, 32-bit processors can address much more than 4 GB. See [Physical Address Extensions](#) for how this is possible.

Physical Address Extensions (PAE)

What are Physical Address Extensions?

Physical Address Extensions, or PAE, are a set of hardware specifications that extend a 32-bit processor's physical memory address space to a maximum of 36 bits, thus providing support for up to 64 gigabytes of memory. Depending on the hardware, the memory address space may be 33, 34, 35 or 36 bits wide. These represent 8, 16, 32, and 64 gigabyte address spaces.

It's worth repeating that PAE defines **hardware** that supports physical address spaces larger than 32 bits. However, taking advantage of these larger address spaces requires more than a PAE-compatible processor and chipset (on the mainboard): it requires a PAE-enabled operating system. Operating systems that are "unaware" of PAE will run on PAE hardware but "see" only a 32-bit (4 GB) address space.

Happily, all 32-bit Windows operating systems from Windows 2000 forward support PAE mode. See [Memory Managed by Windows](#) for more about how Windows supports PAE. See [Unmanaged Memory](#) for information about how RamDisk Plus supports PAE.

Memory Managed by Windows

Non-PAE Systems

Windows operating systems come in two basic editions: 32-bit and 64-bit. Logically, 32-bit editions of Windows run on 32-bit processors, many of which, along with their chipset, implement 32-bit memory and I/O address buses. These non-PAE machines can address a maximum of 4,294,967,296 bytes (4 gigabytes, GB), per address space. (For more about PAE, see [Physical Address Extensions](#).)

On Windows computers, the base 4 GB memory address space is not devoted exclusively to main memory. Video, network, disk, and other devices also require memory address space. Because of this, only a reduced amount of the 4 GB address space is actually available to main memory.

In order to eliminate a host of problems associated with the device drivers that require memory address space, starting with Windows XP SP2 Microsoft has limited the amount of main memory accessible in the 4 GB address space to approximately 3.2 GB. Therefore, even if a non-PAE machine has more than 3.2 GB of main memory installed, only 3.2 GB of it will be accessible to Windows. The remaining 0.8 GB of address space is reserved for memory-mapped devices.

For machines that do not have PAE hardware, this access limit to main memory cannot be overcome. It is a hardware limitation.

PAE Systems

On machines with PAE hardware, all main memory can be accessed -- at least theoretically. For Windows to access the memory, it must also be running in PAE mode. All 32-bit Windows operating systems from Windows NT 4.0 forward support PAE, including 32-bit Windows XP, Vista and 7. To determine if Windows is running in PAE mode, open the System Properties dialog and select the General tab. If PAE is enabled the text "Physical Address Extensions" will appear near the bottom of the dialog.

Depending on the edition and applied service pack, Windows operating systems provide different facilities for enabling PAE. (See "Boot Parameters to Configure DEP and PAE" <http://msdn.microsoft.com/en-us/library/aa468629.aspx>, "/noexecute" <http://msdn.microsoft.com/en-us/library/ms791539.aspx>, "/execute" <http://msdn.microsoft.com/en-us/library/aa468631.aspx>, "/nopae" <http://msdn.microsoft.com/en-us/library/ms791508.aspx>, "/pae" <http://msdn.microsoft.com/en-us/library/ms791485.aspx>.)

On PAE-enabled machines, Windows versions from XP SP2 forward still dedicate the upper 0.8 GB of the first 4 GB of memory address space to memory-mapped devices. However, if Windows is running with PAE mode enabled, then it is capable of accessing whatever amount main memory is installed after the first 3.2 GB, up to a total of 64 GB. This memory simply will be mapped to start above the first 4 GB of address space. In other words, assuming the machine has more than 3.2 GB of main memory, the first 3.2 GB of memory will be accessible in the lower 3.2 GB of memory address space, while the remaining main memory will be accessible at addresses starting at and above 4.0 GB.

Even so, Microsoft has imposed limits on what each edition of 32-bit Windows will access, or manage, even if PAE mode is enabled. Here are the limits:

32-bit Windows Edition	Maximum managed memory (GB)	
	w/o PAE	w/PAE
Windows XP SP2, Windows Vista, Windows 7	3.2	3.2
Windows Server 2003, 2003 R2, 2008, 2008 R2, Standard edition or lesser	3.2	4.0

Notice that Windows XP SP2, Windows Vista and Windows 7 have the same memory limits whether running in PAE mode or not.

Overcoming the Windows Memory Barrier

The fact is, with "extra help" all 32-bit PAE-enabled editions of Windows can support up to 64 GB of main

memory -- the PAE hardware limit. SuperSpeed's RamDisk Plus product provides that "extra help" by supporting what we call "Unmanaged Memory". See [Unmanaged Memory](#) for information about how RamDisk Plus supports memory not managed by Windows.

Unmanaged Memory

With [PAE](#)-mode enabled, all 32-bit Windows editions can potentially access up to 64 GB of main memory. Nonetheless, different Windows editions limit the amount of main memory they support to 4 GB or less.

SuperSpeed's RamDisk Plus features support to utilize main memory that Windows does not.

What is unmanaged memory?

We call the main memory that Windows does not use or manage "Unmanaged Memory", or UM. On a machine running 32-bit Windows XP SP2, Windows Vista, or Windows 7 with 4 GB of main memory installed and with PAE mode enabled, Windows typically limits the amount of main memory available to about 3.2 GB. This leaves about 0.8 GB of main memory unused or unmanaged by Windows.

With RamDisk Plus, that 0.8 GB of main memory unused or unmanaged by Windows (Unmanaged Memory, UM) can now be put to work! What's more, UM can be used right along with the physical memory that Windows manages. In the case described above, RamDisk Plus can create a RAM disk as large as 3.5 GB! On a machine running 32-bit Windows XP SP2 or Vista with 8 GB of main memory installed and with PAE mode enabled, a RAM disk up to 7.5 GB can be created!

Put as Much RAM as You Have and Need to Work

RamDisk Plus thus reclaims the memory resources that would otherwise be lost or unusable. This provides greater versatility in the use of Windows operating systems and applications that can benefit from the increased performance afforded by RAM disks. Now any 32-bit PAE-enabled edition of Windows can access up to 64 GB of main memory, using the bulk of it to enhance system performance!

RamDisk Plus supports two methods to access UM: dynamic and static. Dynamic access, or [dynamic allocation](#), allows UM to be used transparently along with the physical memory managed by Windows. Static access, or [direct address](#), allows specific blocks of physical memory to be allocated for RAM disks.

Dynamic Allocation of Unmanaged Memory

What is dynamic allocation?

Dynamic allocation allows [Unmanaged Memory \(UM\)](#) to be allocated in a fashion similar to how Windows allocates physical memory pages. When a RAM disk is created, memory is first allocated from any unallocated UM, and then from unallocated Windows-managed memory. When a RAM disk is removed, the physical memory allocated to it is returned -- either to the pool of physical pages that Windows manages, or to the UM pool that RamDisk Plus manages.

For example, on a 32-bit, [PAE](#)-enabled Windows XP, Windows Vista or Windows 7 system with 8 GB of main memory, typically about 3.2 GB is managed by the operating system. Ordinarily, a RAM disk of only some 2.5 GB could be created. With RamDisk Plus and the UM feature properly configured, the additional 4.8 GB of UM also would be available. Therefore, a single RAM disk of 7 GB or more (4.8 GB UM, 2+ GB Windows-managed) could be created.

Generally, dynamic allocation is as it offers the simplest and most flexible method to use UM.

Direct Address of Unmanaged Memory

What is direct address?

Direct address allows [Unmanaged Memory \(UM\)](#) to be allocated as a single block of contiguous physical memory pages. A base or starting physical address must be specified in the form of a Page Frame Number (PFN). The number of physical memory pages used for the RAM disk is determined by the RAM disk's size.

Generally, RAM disks created with direct address of UM are used in embedded environments. Direct address does NOT support RAM disks created from memory managed by Windows.