



FreeMeg Allocation Method (Windows 3.1 only)

This list lets you decide how QEMM's **FreeMeg** feature safeguards the first megabyte of memory while other Windows programs are loading.

If you choose **Original**, which is QEMM's default, then FreeMeg allocates nearly all of the first megabyte while Windows programs are loading, to prevent some programs from monopolizing precious first-megabyte memory that other Windows programs may need to load. When using the Original method, FreeMeg first allocates all chunks of first-megabyte memory larger than 32K; then all chunks bigger than 16K; then all chunks bigger than 8K; and so on, until all chunks of first-megabyte memory larger than 512 bytes (or whatever number you set the FreeMeg Block Size to) are taken.

If you choose **Worst-Case**, then FreeMeg allocates all first megabyte memory while Windows programs are loading, then keeps only every other 512-byte block (or whatever number you set the FreeMeg block size to) and frees the rest of the 512-byte blocks. This method makes it impossible for any program to monopolize a large region of first-megabyte memory, but still leaves behind a great many small chunks of memory, in case Windows needs lots of small bits of first-megabyte memory for the data blocks it uses to keep track of programs. However, this method may cause your Windows programs to load more slowly. You probably won't need to use the Worst-Case option, but you can try it if a particular program cannot load with FreeMeg enabled.

If you choose **None**, QEMM disables the FreeMeg feature. This makes it possible for a Windows program to monopolize first-megabyte memory and prevent other Windows programs from loading, no matter how much memory remains on the system.



FreeMeg Block Size (Windows 3.1 only)

The Block Size field indicates the largest size block that QEMM's **FreeMeg** feature will leave available when it safeguards first-megabyte memory while Windows programs are loading. By default, this field is set to 512 bytes. A smaller Block Size value will safeguard even tiny chunks of first-megabyte memory, but may slightly slow down the loading of programs; a higher value may speed up program loading slightly, but leaves more first-megabyte memory at risk. You can try increasing the Block Size value if a particular program cannot load with FreeMeg enabled.



Booting Your System Without QEMM

To reboot your PC without QEMM's memory management follow these steps:

- 1 Reset your system by pressing the **Ctrl**, **Alt**, and **Del** keys simultaneously, by pressing the reset button, or, if necessary, by turning the machine on and off.
- 2 When you hear a beep, hold down the **Alt** key until the boot sequence stops.
- 3 If you are using QEMM's **DOS-Up** feature, you will see a message asking if you want to unload the **DOSDATA** device driver. Press Esc to unload DOSDATA, then immediately press and hold down Alt again until you see the following message:

QEMM: Press Esc to unload QEMM or any other key to continue with QEMM.

- 4 Press the **Esc** key.

QEMM will not load, so programs will not load into High RAM; however, your system will be usable.

Compatibility tab

The Compatibility tab of QEMM Setup lets you review or change certain aspects of QEMM's behavior that may affect QEMM's compatibility with your particular system or configuration. When you select Compatibility by clicking on its tab you see a list of options. If you move the mouse pointer to an option, you see a brief description of what that option does in the **Parameter Information** area near the bottom of the window.

Each of the selections on this screen adds, deletes or modifies a parameter on the QEMM386.SYS device line in CONFIG.SYS. You can see QEMM's device line above the list of options. When you select an option, you will see how it modifies the device line. You can even edit the device line--just click at the point you want to edit.

IMPORTANT: Once you enable or disable any of the compatibility options, the change will not take effect until you reboot your PC.

Related topics

[Remove or Set Page Frame Address](#)

[Find ROM Holes](#)

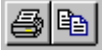
[Exclude Stealthing a Particular ROM](#)

[Reclaim Top Memory](#)

[Enable Suspend/Resume Laptop](#)

[Relocate Extended BIOS Data Area](#)

[Setup QEMM for Troubleshooting](#)



Compression Buffer Size

The size of the MagnaRAM Compression Buffer determines how much memory MagnaRAM can reserve for data compression.

We recommend that you check **Auto** to enable MagnaRAM to select its own buffer size.

You should only need to specify a buffer size manually if you find that MagnaRAM's performance using the automatic buffer size is slow. MagnaRAM will disregard any values that would leave Windows insufficient memory to operate.



Compression Threshold

The Compression Threshold field tells QEMM whether to compress all the data in the MagnaRAM RAM buffer, or just a part of it. The number in the field is a percentage that indicates how much of the RAM buffer should be left uncompressed. The default setting, 0, means that QEMM will try to compress all the data in the buffer. The maximum setting for this field is 100, which means that QEMM will only start compressing data when the buffer is 100% full, and that it will stop compressing when the contents of the buffer have shrunk to less than 100% of its capacity.

Any setting of this field is a tradeoff between more memory and better performance. A bigger value means that MagnaRAM can create slightly less additional Windows memory, but that performance will be improved slightly. A smaller value means a bit more Windows memory and a bit more performance overhead.



QEMM Setup

Contents

[Overview](#)

[Using QEMM Setup](#)

[QEMM Setup: Command and option reference](#)

[Hints](#)

For **Help** using the Online Help, see [Help About Help](#).



Copy ROMs to RAM

This option enables or disables QEMM's ability to speed up ROMs by copying their program code into RAM where it will execute more quickly.

To enable or disable this feature:



To have QEMM copy ROM code into faster RAM, select **Yes**.



To prevent QEMM from copying ROM code into RAM, select **No**.

Yes adds the ROM parameter to the QEMM386.SYS line in CONFIG.SYS; No removes the ROM parameter.

Why you may want to copy ROMs to RAM:

- ◆ If your system does not already speed up ROMs by copying them into faster RAM, enabling this option may speed up some system operations, particularly writes to the screen by programs that use BIOS or DOS video calls (like DOS's COMMAND.COM).

Why you would *not* want to copy ROMs to RAM:



Your system may already copy ROMs to RAM--this feature is already provided if your system has shadow memory.



On some systems, ROMs may not work properly when copied to RAM; floppy disk drives may malfunction on a few systems if the ROM code that controls the floppy disk drives is speeded up. In this case, you can use the QEMM Analysis procedure to help determine which areas of ROM can be copied to RAM (for information see Chapter 9 of the QEMM Reference Manual).



This feature diminishes QEMM's memory pool by the amount of memory taken up by your ROMs - usually about 96K.

QEMM 9 Setup online help. Design for QEMM 9 by Robert Parker. Written by Michael Bolton, Dan Sallitt, Phil Glosserman, Robert Parker, and Kathy Hand.

© 1987-1997 Quarterdeck Corporation. All Rights Reserved.

All other trademarks are the property of their respective owners.



DOS=HIGH

DOS=HIGH is a CONFIG.SYS statement that loads parts of the DOS kernel into the HMA (the first 64K of extended memory). The HMA is available only if you are using DOS version 5 or higher (and is not available for DR DOS 6 users). The amount of DOS that gets moved to the HMA depends on your configuration, but is generally at least 40K.

The most common reason not to enable the DOS=HIGH feature is if you run a program that uses the HMA more efficiently than DOS, like DESQview or DESQview/X. By eliminating the DOS=HIGH statement in CONFIG.SYS you may be able to have more available memory inside DESQview and DESQview/X windows. For information on maximizing the memory inside DESQview and DESQview/X windows, see your QEMM user's guides.

Disabling the Resource Manager for individual applications

If you determine that the Resource Manager conflicts with a single application, you can instruct the Resource Manager not to interfere with that application.

- 1 Go to the QEMM directory and use Notepad to edit the file **RSRCMGR.INI**.
- 2 Under the **[TurnOffApp]** section, add the following line:
program.exe=off
where "program.exe" is the name of the program file for which you want to disable Resource Manager.
- 3 Save the file.

If you later decide to enable services for that program, delete the **entire line** from the .INI file.



Exclude stealthing a particular ROM

You use this option to tell QEMM not to stealth a particular ROM. You should tell QEMM not to stealth a particular ROM only when attempting to solve problems with the StealthROM feature.

To use the feature that excludes stealthing of a particular ROM:



Select **Address** to exclude a particular ROM from being Stealthed. Then, click in the adjacent field and type the starting address of the ROM you want to prevent from being Stealthed.



Select **None** if you have excluded a particular ROM from being Stealthed and you now want to remove this exclusion.

If you specify that a particular ROM should not be Stealthed, QEMM Setup will place the XST=xxxx parameter on the QEMM386.SYS line in your CONFIG.SYS file, causing QEMM not to stealth that ROM. You can get the starting addresses of all stealthed ROMs from the Manifest QEMM Overview screen. In general, video ROMs are located at C000 (or at E000 on Micro Channel systems); system ROMs at F000. If you have a disk ROM (many systems do not), it will generally be located at an address between C800 and E000.

If possible, it is usually more memory-efficient to solve StealthROM problems with the EXCLUDE parameter than with the EXCLUDESTEALTH parameter.

DoubleSpace is the disk compressor that comes with DOS versions 6.0 - 6.20.

If you are using MS-DOS 6's **DoubleSpace** or **DriveSpace**, you can save 31K-49K of memory by using QEMM's **Stealth D*Space** feature to relocate the DoubleSpace or DriveSpace device driver in expanded memory. See Chapter 5 of the QEMM Reference Manual for details.

DriveSpace is the disk compressor that comes with DOS 6.22 (or later).

The EMS page frame is a 64K area, usually in upper memory, used by programs to access expanded memory. QEMM also uses the page frame to enable its StealthROM and Stealth D*Space features.

Edit QEMM device line

This selection lets you manually edit the QEMM device line. If the QEMM parameters do not fit in the visible field on screen, an arrow at the left or right of the field indicates the presence of off-screen parameters. The field will scroll when you use the arrow keys or type



Enable MagnaRAM

This option enables MagnaRAM, the centerpiece of QEMM's memory enhancement system for Windows. Enabling MagnaRAM also activates CacheBack, which borrows unused disk-cache memory and adds it to MagnaRAM's memory pool. (CacheBack is not available under Windows 3.1; see your QEMM 9 manual for more information).



Enable QuickBoot

This selection enables or disables QEMM's feature that speeds up warm reboots (i.e., when you reboot by pressing Ctrl+Alt+Delete).

To enable or disable QuickBoot:



To enable QuickBoot, select **Yes**.



To disable QuickBoot, select **No**.

If you enable QuickBoot, you can also enable the **Timeout** feature and specify a number of seconds in the adjacent field. The timeout feature tells QEMM to post a QuickBoot menu for xx seconds (where xx is a number from 1 to 99) when you warm boot your system. The QuickBoot menu lets you choose which drive to boot from. When the timeout value of xx seconds expires, QEMM automatically reboots the system without your intervention. The default timeout value is 0, which tells QEMM to warm boot without posting the Quickboot menu.

To enable the Timeout feature:



Select **Timeout**.



Click on the seconds field and type the number of seconds to wait for user input before automatically booting using the default boot drive.

DOS-UP tab

The DOS-UP tab controls QEMM's DOS-Up feature, which loads into High RAM certain parts of DOS that would normally load into conventional memory. Depending on how your system is configured, DOS-Up can free between 7-70K of conventional memory for running DOS programs.



Select **Do not use DOS-Up** to disable DOS-Up.



Select **Use all the features of DOS-Up** to have DOS-Up load as much of DOS as possible into upper memory.



Select **Use the specified features of DOS-Up** if you want to choose the parts of DOS that DOS-Up should load into High RAM. Then, click on the features you want to load into High RAM.

IMPORTANT: Once you enable or disable DOS-Up, you must reboot your PC for the change to take effect.

Related topics

[Parts of DOS that DOS-UP moves](#)

Enable or Disable Stealth D*Space

This tab is only displayed if you are using DOS 6's DoubleSpace or DriveSpace feature.

QEMM can use its Stealth technology to move DOS 6's DriveSpace or DoubleSpace driver entirely out of conventional and upper memory, making it appear in the EMS page frame when it is needed. This saves about 31K-49K that would otherwise use up space in conventional memory or upper memory.

To enable or disable QEMM's Stealth D*Space feature:



Select **Use QEMM's Stealth D*Space** to enable Stealth D*Space.



Select **Do not use QEMM's Stealth D*Space** to disable Stealth D*Space.

IMPORTANT: Once you enable or disable Stealth D*Space, you must reboot your computer for the change to take effect.

If you enable Stealth D*Space, QEMM Setup will place the ST-DSPC.SYS driver in your CONFIG.SYS file to relocate the DoubleSpace or DriveSpace driver. ST-DSPC.SYS uses about 3K and can be loaded high. Optimize will add the necessary command to load this driver high if there is room for it in upper memory.



Suspend/resume laptop support

This option enables or disables QEMM's special support for the suspend/resume feature found on many portable computers. Suspend/Resume is a feature that allows you to run the computer on low power when it is not in use, and to restore the system to its previous state when you return to it. Many systems with the suspend/resume feature will work fine without special support from QEMM, but some systems will not return properly from a low power state if a 386 memory manager such as QEMM is active. If your system has a suspend/resume feature that is not working properly with QEMM installed, you should enable QEMM's support for suspend/resume.

To enable or disable QEMM's special support for the suspend/resume feature:



Select **Auto** to enable QEMM's special support for suspend/resume.



Select **No** to disable QEMM's special support for the suspend/resume feature.



Select **Interrupt** if you have tried the Auto selection without success. After selecting Interrupt, click in the adjacent field and specify a hardware interrupt number for your PC's suspend/resume feature. 2, D, 72, 73, and 77 are the numbers most likely to be used by the Suspend/Resume feature. See your hardware documentation or contact the manufacturer for information on the appropriate interrupt number.

If you choose Yes, QEMM Setup places the SUS parameter on the QEMM386.SYS line in the CONFIG.SYS file. This parameter makes QEMM search for the hardware interrupt that suspend/resume is using. If you select Interrupt, Setup adds the SUS:xx parameter, where xx is the interrupt number you specify.



Find ROM holes

This option enables or disables QEMM's feature that finds "ROM holes" --unused areas in the system ROM (between F000 and FFFF)--and makes them available for High RAM or expanded memory mapping. This feature is only available when QEMM's StealthROM feature is not in use.

To enable or disable QEMM's ability to find and use ROM holes:



Select **Yes** to enable QEMM's ability to find ROM holes and make them available for High RAM or expanded memory mapping.



Select **No** to disable this feature.

The feature that finds ROM holes is on by default. If you choose No, QEMM Setup adds the RH:N parameter to the QEMM386.SYS line in CONFIG.SYS.

The most common reason to disable this feature is to troubleshoot floppy disk problems or other conflicts between QEMM and your system. If disabling this feature solves your problem, it may be more memory-efficient to use the EXCLUDE parameter on a section of the system ROM instead of using the RH:N parameter.



Fill Upper Memory with RAM

This option creates or removes High RAM in the upper memory area. When High RAM is present, you can load TSRs, device drivers and parts of DOS into upper memory. By loading these items into upper memory, you will have more conventional memory available for DOS programs.

To create or remove High RAM:



Select **Yes** to create High RAM.



Select **No** to remove High RAM.

Yes causes QEMM Setup to place the RAM parameter on the QEMM386.SYS line in CONFIG.SYS; No causes QEMM Setup to remove the RAM parameter. By default, QEMM's installation creates High RAM.

Help About QEMM Setup's Online Help

For information on using the Windows online help system, press **F1** at any time.



Help toolbar

A small toolbar with printer and clipboard buttons appears in the upper left corner of many secondary windows. When a topic in a secondary window is accompanied by this graphic image, you can print the topic by clicking on the printer or copy the topic to the Windows clipboard by clicking on the "Copy" button.



High RAM

The upper memory area (the area between 640K and 1024K) is normally reserved for use by your system's hardware. On most PCs, there are "holes" in this area--upper memory addresses that are not associated with any physical RAM or ROM chips. QEMM maps memory from outside the first megabyte of RAM into the vacant areas. In addition, QEMM uses advanced techniques to reclaim some parts of upper memory that were previously reserved for use by hardware. High RAM is QEMM's name for the memory mapped into the upper memory addresses. Once memory is mapped into upper memory addresses, QEMM can use that High RAM to load TSRs, device drivers, and parts of DOS. By loading these items into upper memory instead of conventional memory, more conventional memory is available for your other programs.

Hints

Information is available on the following topics:

[QEMM and Disk Compression Software](#)

[QEMM and Bus-mastering Devices](#)

[VIDRAM: Extending Memory for Text-based Programs](#)

[New Parameter Names \(for users upgrading from a QEMM version earlier than 8](#)

[Bootting Your System Without QEMM](#)

[Switching Between MS-DOS 6's Memory Manager and QEMM](#)



Multiple Configurations

QEMM Setup has detected that your CONFIG.SYS file contains multiple configuration paths. Because the QEMM386.SYS device driver may be in more than one of these paths, you need to tell QEMM Setup which path you want to modify, in the event that the changes you specify will alter the QEMM386.SYS device line.

To select an existing configuration path:

Choose the configuraton you want, then select **Continue**.

To create a new configuration path:

Select highlight one of the existing paths and select **Create a new path from the selected existing path**. Then type a unique name for the new path in the field below. The name can be up to 32 characters long and can consist of more than one word. When you choose this option, QEMM Setup will add the new path to your CONFIG.SYS file. See your DOS manual for information on how to modify your CONFIGF.SYS and AUTOEXEC.BAT files for multiple configurations.

If you create a new configuration path, you should run Optimize and select the new configuration path. You will be prompted to run Optimize when you exit QEMM Setup.



PAGEOVERCOMMIT (Windows 3.1 only)

The PAGEOVERCOMMIT=n setting appears in the [386Enh] section of the Windows 3.1 SYSTEM.INI file. It determines the size of linear memory, which is the address space that Windows programs see when they allocate memory from Windows.

If the PAGEOVERCOMMIT statement does not appear in the SYSTEM.INI file, Windows 3.1 defaults to a value of 4, which means that the size of linear memory will be four times the amount of physical memory (installed RAM) on your machine. QEMM will normally set PAGEOVERCOMMIT to 8, which is twice Windows 3.1's default. The allowable values of PAGEOVERCOMMIT are 1 through 20.

If the value of PAGEOVERCOMMIT is too small, then some of the physical or virtual memory on your system will go to waste, because there will be no addresses for Windows 3.1 to give out to programs, even if the memory is available. If PAGEOVERCOMMIT is too big, however, Windows 3.1 wastes precious physical memory to keep track of unneeded linear memory addresses.

The best way for you to determine how big PAGEOVERCOMMIT should be is to monitor MagnaRAM's performance on the QEMM screen. Open a number of programs until physical and virtual memory have been almost used up, and then check the amount of available linear memory. If there is quite a bit of linear memory left, try decreasing PAGEOVERCOMMIT. You want to arrive at a value of PAGEOVERCOMMIT that leaves some available linear memory behind when physical and virtual memory are used up, but does not leave too much.

If you change this setting, you must restart Windows to implement the change.

Parts of DOS that DOS-UP moves

Parts of DOS that DOS-Up can move out of conventional memory are:



DOS resources (FILES, BUFFERS, FCBS, STACKS, LASTDRIVE). The amount of memory that these resources take up varies with your configuration. See Manifest's DOS Overview screen for details.



COMMAND.COM (the DOS command processor). Its size varies in different versions of DOS. It is normally smaller than 5K.



DOS data (the DOS data structures that are not moved out of conventional memory by the **DOS=HIGH** statement). If you do not use DOS=HIGH, DOS-Up will additionally load into upper memory those parts of the DOS kernel that DOS=HIGH would have loaded into the HMA.



DOS=HIGH is a feature of DOS version 5 and later (it is not a DOS-Up feature, but you can enable or disable it from QEMM Setup). DOS=HIGH loads the DOS kernel, buffers and part of COMMAND.COM into the HMA, the first 64K of extended memory. The amount of DOS that gets moved to the HMA depends on your configuration, but is generally at least 40K. We recommend that you use DOS=HIGH unless you routinely run a program (such as DESQview or DESQview/X) that can use the HMA more efficiently than DOS. If you use DOS=HIGH, you can still use the features of DOS-Up.



If you are using DR DOS 6 or Novell DOS 7, you cannot use DOS=HIGH.

DOS-Up makes three changes to your CONFIG.SYS file. The DOSDATA.SYS driver, which loads at the beginning of the CONFIG.SYS, prepares the system for DOS-Up. The DOS-UP.SYS driver loads the DOS kernel, data, and resources into High RAM. And your SHELL statement is modified so that LOADHI.COM can put COMMAND.COM in upper memory. If you have no SHELL statement, DOS-Up creates one for you.

QDPMI tab

The QDPMI tab lets you enable or disable the Quarterdeck DOS Protected Mode Interface for programs that support DPML (e.g., Microsoft's C/C++ Development System for Windows version 7, Borland's C/C++ version 3, and Intel's Code Builder Kit version 1.1). QEMM's DPML host is called QDPMI. Unlike other DPML hosts, QDPMI provides virtual memory in the DOS environment.

To enable or disable QDPMI:



Select **Do not use Quarterdeck's DPML host** to disable this feature.



Select **Use Quarterdeck's DPML host** to enable this feature.

IMPORTANT: Once you enable or disable QDPMI, you must reboot your PC for the change to take effect.

QDPMI uses about 2K of RAM. If you do not have applications that support DPML, you may want to disable QDPMI to free up 2K of memory. Protected-mode programs that are VCPI clients will run under QEMM even if QDPMI is not loaded.

If you enable QDPMI, QEMM Setup will place the QDPMI.SYS device line in your CONFIG.SYS file. You can specify the size in kilobytes of the DPML swapfile, an area on disk that will be used as virtual memory for DPML applications. The default swapfile size is 1024K (1 meg). The advantage of specifying a bigger swapfile is that more virtual memory will be available to DPML programs. It is particularly important to have a large swapfile if you have a low-memory system and a memory-hungry DPML application. The disadvantage of specifying a bigger swapfile is that more of your hard disk may be used up by your DPML program. QDPMI does not use any of your hard disk for a swapfile until the DPML program requests the memory, and the swapfile grows as needed up to the maximum size that you set.

Overview of QEMM Setup

QEMM Setup makes it easy to enable or disable QEMM's optional features, as well as add or delete QEMM's fine-tuning and troubleshooting parameters. QEMM Setup also provides you with hints on using QEMM and lets you view the QEMM READ ME file for late-breaking information and technotes covering a variety of technical issues. QEMM Setup can also assist you in troubleshooting any problems that might occur.

QEMM Setup is organized into five tabbed pages:

- Features
- Compatibility
- QDPMI
- DOS-Up
- Windows

Each page includes a **Reset** button which discards any changes you have made on that page since you last saved your QSetup settings. The **Reset All** button at the bottom of the screen discards changes made to all of the pages. The Windows page also includes a **Defaults** button which restores the settings on this page to their default values. After making changes to any of the pages, you must select **Save** to store your changes.

QEMM Setup gives you help every step of the way. When you select an option from a menu, you will see an explanation of what the option does. If you still have questions, press **F1** or select the **Help** button for assistance.

QEMM Setup: Reference

Tabs on the QEMM Setup screen

Features

Compatibility

QDPMI

DOS-Up

D*Space (only displayed on systems using DriveSpace or DoubleSpace)

Windows

Other topics

Reviewing and editing proposed changes to system configuration files



QEMM and Bus-mastering Devices

Certain **SCSI** disk controller cards (and, less frequently, **ESDI** disk controllers and network cards) use a technique called **bus-mastering** to speed up disk access. This technique can cause a conflict when a memory manager (such as QEMM) attempts to load a device driver or TSR into upper memory.

QEMM automatically supports bus-mastering disk controllers. In the vast majority of cases, QEMM can detect a bus-mastering hard disk controller and will take steps to prevent problems. (This is not true if the card controls something other than a hard drive or if QEMM is not being loaded from the bus-mastering hard drive.)



QEMM and Disk Compression Software

QEMM is fully compatible with current disk compression software and includes special features for both Stacker and MS-DOS's DoubleSpace and DriveSpace. QEMM's Stealth D*Space feature reduces the memory overhead of DoubleSpace or DriveSpace to as little as 3K. If you are using MS-DOS disk compression, this feature will be displayed on the QSETUP main menu; choose L for "Enable or disable Stealth D*Space", and Yes to enable the feature.

Adding the /QD parameter to the STACKER.INI file can reduce Stacker's overhead to as little as 10K. Please see your QEMM user's guides for more information.

QEMM's New Parameter Names

If you are upgrading from a version of QEMM earlier than 8.01, you can still use the old parameter names if you like. Below is a list of the old parameter names, cross-referenced with the new names. Abbreviations are listed in parentheses.

<u>Old Name</u>	<u>New Name</u>
COMPAQ386S (C386S)	COMPAQ386S:Y (C386S)
COMPAQEGAROM (CER)	COMPAQEGAROM:Y (CER)
COMPAQHALFROM (CHR)	COMPAQHALFROM:Y (CHR)
COMPAQROMMEMORY (CRM)	COMPAQROMMEMORY:Y (CRM)
DONTUSEXMS (DUX)	USEXMS:N
DOS4 (D4)	DOS4:Y (D4)
FORCEEMS (FEMS)	FORCEEMS:Y (FEMS)
FORCESTEALTHCOPY (FSTC)	FORCESTEALTHCOPY:Y (FSTC)
IGNOREA20 (IA)	TRAP8042:Y (T8) **
LOCKDMA (LD)	LOCKDMA:Y (LD)
NOCOMPAQFEATURES (NCF)	COMPAQFEATURES:N (CF)
NOEMS	EMS:N
NOFILL (NO)	FILL:N
NOHMA	HMA:N
NOPAUSEONERROR (NOPE)	PAUSEONERROR:Y (PE)
NOROM (NR)	MAPREBOOT:N (MR)
NOROMHOLES (NRH)	ROMHOLES:N (RH)
NOSHADOWRAM (NOSH)	SHADOWRAM:NONE (SH)
NOTOKENRING (NTR)	TOKENRING:N (TR)
NOTOPMEMORY (NT)	TOPMEMORY:N (TM)
NOVDS	VDS:N
NOVIDEOFILL (NV)	VIDEOFILL:N (VF)
NOVIDEORAM (NVR)	VIDEORAM:N (VR)
NOWINDOWS3 (NW3)	WINDOWS3:N (W3)
NOXBDA (NX)	XBDA:N
NOXMS	XMS:N
UOLDDV (ODV)	OLDDV:Y (ODV)
UNUSUALEXT (UX)	UNUSUALEXT:Y (UX)

**** Default has changed.**

QEMM's StealthROM Feature

StealthROM is an exclusive QEMM feature that can typically create an additional 48K to 115K of High RAM on almost any PC. StealthROM hides your PC's ROMs and makes their memory addresses available for High RAM or expanded memory mapping. The advantage of having the additional High RAM is that QEMM can load TSRs, device drivers and selected parts of DOS there instead of in conventional memory. By freeing up conventional memory, you will have more room for running DOS programs.

Depending on your configuration and the installation options you chose, StealthROM may have been enabled on your system when you installed QEMM. When you run the **Optimize** program, Optimize will try to load your TSRs, device drivers and selected parts of DOS into High RAM. If all of them will not fit, Optimize will test your system for compatibility with StealthROM and will determine which StealthROM method is best for your system.

ROM stands for Read-Only Memory--memory that is fixed in content and cannot be changed. The contents of ROM memory are not lost when the power is turned off. ROMs generally occupy addresses in upper memory. The BIOS and video services are among the programs contained in ROM.



Reclaim top memory

This feature enables or disables QEMM's ability to reclaim top memory, adding that memory to QEMM's memory pool. By default, QEMM reclaims unused top memory on Compaqs and other systems on which QEMM recognizes the presence of top memory. This feature typically adds 256K to 384K of RAM to QEMM's memory pool. Manifest's QEMM Memory screen will include a Top Memory row if QEMM is reclaiming top memory on your system.

To enable or disable QEMM's ability to reclaim top memory:



Select **Yes** to enable reclamation of top memory.



Select **No** to disable this feature.

This feature is on by default. If you disable it, QEMM Setup adds the TM:N parameter to the QEMM386.SYS line in CONFIG.SYS.

If QEMM has a problem reclaiming top memory on your system, you may experience a hang or reboot when QEMM386.SYS loads.



Reclaim unused shadow memory

This option lets you enable or disable QEMM's feature that reclaims unused portions of shadow memory. When QEMM reclaims shadow memory, it adds that memory to QEMM's memory pool for general use, typically giving you about 192K more usable RAM. By default, QEMM reclaims unused shadow memory, giving you more expanded or extended memory on systems that have any of the following: Chips & Technologies LEAP, AT386, NEAT, or SCAT ShadowRAM; or NEC, OPTI, PEAK or TOPCAT shadow memory,

To have QEMM reclaim unused shadow memory:



Select **Yes** to reclaim shadow memory.



Select **No** if you do not want QEMM to reclaim shadow memory.

QEMM reclaims shadow memory by default. When you disable the feature that reclaims shadow memory, QEMM Setup adds the SH:N parameter to the QEMM386.SYS line in CONFIG.SYS.

Manifest's QEMM Memory screen will include Shadow RAM information if QEMM is reclaiming shadow memory on your system. On some systems with unusual types of shadow memory, QEMM may have a problem reclaiming the unused portion. A common symptom is continual rebooting when QEMM loads, although other symptoms can occur. Disabling the shadow memory feature is a common troubleshooting technique.



Relocate Extended BIOS Data Area

This selection tells QEMM how to treat the XBDA (Extended BIOS Data Area).

To relocate the XBDA:



Select **No** to tell QEMM not to move the XBDA.



Select **Auto** to have QEMM determine where to most effectively place the XBDA.



Select **Low** to move the XBDA to low conventional memory.



Select **High** to force the XBDA into High RAM.

Below is a more detailed summary of these options:

AUTO is the default. QEMM moves the XBDA into High RAM unless it detects that you have a suspend/resume feature, that you have a machine (like some IBM ThinkPads) that fails with the XBDA in High RAM, or you place the **SUSPENDRESUME (SUS)** parameter on the QEMM386.SYS line in CONFIG.SYS. In these cases, QEMM moves the XBDA into low conventional memory.

If you select No, QEMM Setup adds the XBDA:N parameter to the QEMM386.SYS line in CONFIG.SYS. The XBDA will remain at the top of conventional memory where it will prevent video filling or the use of **VIDRAM** and will decrease the size of windows in DESQview and DESQview/X. You should choose No if you have a system or a program that expects the XBDA to be at the top of conventional memory. The symptom of this problem is usually a system crash, which can occur at boot time or later.

Low gives most of the benefits of moving the XBDA, and so is a less drastic way to try to solve any XBDA-related problem than choosing No.

You may want to select High (to save 1K of conventional memory) if QEMM is loading your XBDA low. If you do this on a laptop PC that has a Suspend/Resume feature, or on an IBM ThinkPad, your system may not work properly.

Moving the XBDA into High RAM lets VIDRAM and video filling work, increases the size of windows in DESQview and DESQview/X, and saves 1K of conventional memory. Moving the XBDA to low memory does not save conventional memory but gives all the other benefits listed above.

To find out where your XBDA is loaded, see Manifest's First Meg BIOS Data screen. If the third line on this screen does not say "0E: Extended BIOS Segment," then you do not have an XBDA.

If you do have an XBDA, check the four-digit hexadecimal address of the XBDA. If this address is 9FC0, then the XBDA has not been moved at all. If the address starts with 0 or 1, the XBDA has been moved to low conventional memory. If the address starts with a letter (A through F), then the XBDA is in High RAM.



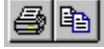
Remove or set address of page frame

This option lets you specify the starting address of the EMS page frame or specify that QEMM should not create a page frame.

To specify the EMS page frame select one of the following:



Select **None** to eliminate the page frame. This will disable the StealthROM and Stealth D*Space features and make expanded memory unavailable for programs. If you do not use Stealth or programs that use EMS, eliminating the page frame will make 64K of upper memory addresses available for High RAM, at the expense of all the benefits of having a page frame. We strongly recommend that you leave the page frame enabled.



Select **Auto** to have QEMM choose the page frame address based on your system configuration.



Select **Address** if you want to specify a particular address for the page frame. Then, click in the adjacent field and enter the 4-digit hexadecimal segment address for the beginning of the page frame. The address must be on a 16K boundary (i.e., its last 3 digits must be 000, 400, 800 or C00). You can specify the starting segment address of the page frame if a different location will consolidate two smaller High RAM regions into one large one, or if you need to place the page frame at the starting address of one of your ROMs to make the ROM work with the StealthROM feature. However, you should not set the page frame address if you do not know how to avoid conflicts between the page frame and ROM, adapter RAM, or video RAM.

Depending on your selection, QEMM Setup places the FR=NONE or FR=xxxx (where xxxx is a hex address) parameter on the QEMM386.SYS line in CONFIG.SYS, or removes the FR parameter from the QEMM386.SYS line.

Features tab

The Features tab lets you review or change certain aspects of QEMM's behavior. When you select QEMM Features by clicking on its tab, you see a list of options. If you move the mouse pointer to an option, the option will become highlighted and you will see a brief description of what that option does in the **Feature Information** area near the bottom of the window.

Each QEMM Feature adds, deletes or modifies a parameter on the QEMM386.SYS device line in CONFIG.SYS. You can see QEMM's device line above the list of options. When you select an option, you will see how it modifies the device line. You can even edit the device line--just click at the point you want to edit.

IMPORTANT: After enabling or disabling any QEMM Feature, you reboot your PC in order for the change to take effect.

The options on the QEMM Features page are:

[Fill Upper Memory with RAM](#)

[Copy ROMs to RAM](#)

[Enable QuickBoot](#)

[StealthROM Method](#)

[Set Size and Type of Disk Buffer](#)

[Reclaim Unused Shadow Memory](#)

Reviewing and Editing Proposed Changes to Configuration Files

Options selected from the various QEMM Setup screens can result in changes to your CONFIG.SYS, AUTOEXEC.BAT, SYSTEM.INI, and WIN.INI files. Changes that result from your selections are not saved until you choose **Save** at the bottom of the QEMM Setup display.

To review proposed changes to these files, select one of the following options from the **File** menu on the menu bar:

Proposed CONFIG.SYS

Proposed AUTOEXEC.BAT

Proposed SYSTEM.INI

Proposed WIN.INI

Proposed MSDOS.SYS (Windows 95 only)

The file you select is displayed and can be reviewed and edited as desired. At any time during the editing process, you can click **Reset** to discard any changes you have made and continue editing.

When you finish reviewing and editing the file, click **OK** to close the editor window or **Cancel** to discard any changes you have made to the file and close the window. If you choose **OK**, you must still select **Save** when you return to QSetup's main screen in order to save your changes.



Set size and type of disk buffer

This option allocates additional memory to buffer disk reads and writes. Buffering may be necessary if you are experiencing problems with QEMM on a system with a bus-mastering hard disk or if there are conflicts between an EMS-using disk utility and QEMM's StealthROM feature.

To set the size and type of disk buffer:



Select **None** to remove any disk buffering.



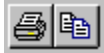
Select **Auto** to have QEMM attempt to determine whether you need a disk buffer to resolve problems with a bus-mastering hard drive. If it detects a bus-mastering conflict with the drive from which QEMM loads, QEMM Setup will add the parameter DB=2 to the QEMM386.SYS line in CONFIG.SYS.



Select **Full** to have QEMM intercept all disk reads and writes to resolve problems with a bus-mastering hard disk. This adds the DB=xxx parameter to the QEMM386.SYS line in CONFIG.SYS. If you select the Full option, you should set the number of kilobytes to reserve for the disk buffer; just click in the box on the right side of the field and type a number. 2 and 10 are commonly-used values. This type of disk buffering eliminates problems with bus-mastering hard disks, but with a penalty in conventional memory and disk performance.



Select **Frame** to resolve conflicts between an expanded memory-using disk utility and StealthROM. This selection tells QEMM to buffer only disk reads and writes into the EMS page frame. It adds the DBF=xxx parameter to the QEMM386.SYS line in CONFIG.SYS. If you select the Frame option, you should set the number of kilobytes to reserve for the disk buffer; just click in the box on the right side of the field and type a number. 2 and 10 are commonly-used values.



Set up QEMM for troubleshooting

If you wish to troubleshoot a particular problem, you can set up QEMM for troubleshooting. When you set up QEMM for troubleshooting, QEMM Setup will place the following ten parameters on the QEMM386.SYS line in CONFIG.SYS:

DB=2 SH:N TM:N TR:N CF:N FILL:N MR:N RH:N XBDA:N BE:N

To set up QEMM for troubleshooting:



Select **Yes** to add the troubleshooting parameters,



Select **No** to remove the troubleshooting parameters.

These are not the only QEMM386.SYS parameters that can solve problems, but they are the easiest to try as part of a one-step troubleshooting process. If your problem goes away after you enable the troubleshooting parameters, you should try eliminating the parameters one by one until you find the parameter that solved the problem, then take all the other troubleshooting parameters off the QEMM386.SYS line. Be sure to reboot whenever you add or delete parameters.



FreeMeg SkipFile List (Windows 3.1 only)

QEMM's **FreeMeg** feature normally safeguards the first megabyte of memory whenever any Windows program loads. However, in case a particular program does not load properly when FreeMeg is active, QEMM keeps a list of programs for which it deactivates FreeMeg. Only the programs on this SkipFile list load without FreeMeg's protection; all other programs will still be unable to monopolize precious first-megabyte memory when they are loading.

To add a program to the SkipFile list, click the **Add** button next to the list, and type or select the name of the program, complete with file extension, into the Add to SkipFile List dialog box. If you do not see the name of the file you want to add to the list, select a new drive or directory

To edit the Skip File list, select the entry on the list that you want to change and click the **Edit** button next to the list. Use the Edit Skip File List dialog box to change the entry.

To delete an entry from the Skip File list, select the entry and click the **Delete** button next to the list.

Stealth ROM

StealthROM is a QEMM feature that creates additional mappable areas at the addresses used by your PC's ROMs. By default, QEMM will turn these areas into High RAM that can be used to load TSRs, device drivers and selected parts of DOS. When StealthROM is enabled, QEMM monitors the interrupts pointing into those ROMs. When those interrupts occur, QEMM maps the appropriate ROM into the page frame and passes the interrupts to the ROM's location in the page frame. In general, the ROMs targeted are your system ROM, video ROM and disk ROM, although certain other ROMs may be "Stealthed" as well.

With the ROMs out of the way, the amount of usable upper memory is greatly increased. Depending on the location of the ROMs, High RAM regions can become quite large and able to accommodate more or larger device drivers and TSRs.



StealthROM method

This selection lets you enable or disable QEMM's StealthROM feature. StealthROM can typically free 48K-115K of upper memory addresses which can then be used for High RAM or expanded memory mapping.

To enable or disable StealthROM:



Select **Mapping** to enable StealthROM using the mapping method.



Select **Frame** to enable StealthROM using the frame method.



Select **None** to disable StealthROM.

If you select Mapping, QEMM Setup will add the ST:M parameter to the QEMM386.SYS line in CONFIG.SYS. If you select Frame, Setup will add the ST:F parameter. If you select None, Setup will remove the ST parameter.

QEMM offers to enable StealthROM during the installation process or the Optimize process if it sees that you need additional High RAM. If you use DESQview or DESQview/X, you should use the StealthROM feature even if QEMM has not enabled it for you.



Switching Between MS-DOS 6's Memory Manager and QEMM

QEMM provides all the functionality of MS-DOS 6's memory manager, and much more. For a list of QEMM's features and how they stack up against DOS 6's memory manager, see "QEMM Benefits and Features" in Chapter 1 of the QEMM Reference Manual.

If you are using MS-DOS 6 and you have run its MemMaker memory utility, you can switch back to QEMM by running QEMM's Optimize program (assuming you have already installed QEMM on your hard disk). To run Optimize, type **OPTIMIZE** at the DOS prompt.

If you ever want to switch back to MS-DOS's memory manager, simply run MemMaker again. If you are using QEMM's **DOS-Up** feature, be sure to run QEMM Setup and disable DOS-Up before you run MemMaker.

If you are using QEMM's **Stealth D*Space** feature and you decide to switch back to MS-DOS 6's memory manager, the **ST-DSPC.SYS** driver will perform the same function as DOS's **DBLSPACE.SYS /MOVE** or **DRVSPACE.SYS /MOVE**. It will allow **DBLSPACE.BIN** or **DRVSPACE.BIN** to be moved into upper memory. You can replace ST-DSPC.SYS with DBLSPACE.SYS or DRVSPACE.SYS, but you will suffer no ill effects if you do not.

TurboLoad (Windows 95 only)

TurboLoad helps Windows programs to load faster. TurboLoad is enabled by default; to enable or disable TurboLoad, simply check or clear the **TurboLoad** check box.



Use Available Cache Memory

This option activates CacheBack, a MagnaRAM feature that "borrows" unused memory from the Windows built-in disk cache. MagnaRAM is able to add this borrowed memory to its compression buffer.

Windows for Workgroups 3.11 and Windows 95 automatically reserve a certain portion of memory for a disk cache called VCACHE, which is separate from the SmartDrive disk cache. Windows always makes slightly over-generous allocations for VCACHE, leaving your system with some small amount of underutilized memory. MagnaRAM can determine when this memory is not being used, "borrow" it from Windows, and assign it to the MagnaRAM compression buffer. This effectively makes more memory available for your applications. The amount of memory MagnaRAM is able to obtain from the disk cache varies from system to system.

Your system must support **32-bit file access** for CacheBack to work. Windows 3.1 does not support 32-bit file access (it supports 32-bit **disk** access, which is not the same thing), but Windows 95 and Windows for Workgroups 3.11 do. Also, some hard drives do not support 32-bit file access; if your hard drive does not, do not be concerned. MagnaRAM relies on other sources for its major performance boosts, and you can still get a satisfactory increase in performance without using CacheBack.



Use Memory Compression

This check box determines whether MagnaRAM memory compression feature is active. If you change this setting, you must restart Windows to implement the change.

We recommend that you do not disable memory compression without disabling MagnaRAM, except for troubleshooting purposes.



Use Resource Manager (Windows 3.1 only)

QEMM's Resource Manager feature lets Windows 3.1 users fit more programs into memory before running out of precious system resources. The Resource Manager feature stores some system resources in a different place in memory, bring them back into the system resources area when they are needed.

If any program does not function properly when the Resource Manager feature is enabled, you can disable Resource Manager by clearing this check box, saving the change, and restarting Windows.

To enable or disable the resource manager, check or clear the **Use Resource Manager** check box.

Related topics

[Disabling the Resource Manager for individual applications](#)

Using QEMM Setup

To make changes to your system configuration using QEMM Setup:

1. Select the tab on which you want to make changes.
2. Make any changes.
3. Click **Save** to save the changes to your system configuration and startup files.
4. Click **Exit** to exit QEMM Setup.
5. **Reboot your computer.** The changes you make with QEMM Setup will not be implemented until you reboot your computer.

VIDRAM is a QEMM program you can use to extend conventional memory when running DOS text-based programs on a system with an EGA or VGA adapter. By using VIDRAM, you can get an additional 64K-96K of memory to run such programs. However, you cannot use EGA or VGA graphics while VIDRAM is on. For information on VIDRAM, see Chapter 6 of your QEMM user's guide.

VIDRAM: Extending Memory for Text-based Programs

QEMM's **VIDRAM** program can extend conventional memory by as much as 96K for running DOS text-based programs. VIDRAM even extends conventional memory for DOS text programs running in Microsoft Windows.

To use VIDRAM, your system must have an **EGA** or **VGA** video adapter or an adapter with EGA or VGA capability (this includes VGA-compatible 8514A video adapters). Your PC must have 640K of conventional memory and the programs that you run while using VIDRAM must not use EGA or VGA graphics.

If your PC has an EGA or VGA video adapter, the 64K memory area just above conventional memory (640K-704K or A000-AFFF hex) is reserved for use by graphics modes. When you run text-based programs, that area is unused, so VIDRAM can appropriate it to extend the contiguous conventional memory for running programs.

It is important to understand that you cannot run EGA or VGA graphics operations while VIDRAM is in use. If you routinely use both large text-based programs and graphics programs, you can turn the VIDRAM feature on when you need it for a text program and off before you run a graphics program. If you are using an 8514A adapter, you can still use 8514 graphics programs while VIDRAM is enabled.

To turn VIDRAM on type **VIDRAM ON** at the DOS prompt.

This command will extend conventional memory into the EGA/VGA graphics area for a total of 704K conventional memory.

To turn VIDRAM off so you can use graphics programs again, type **VIDRAM OFF**.

For more information on VIDRAM, including using VIDRAM with Microsoft Windows, DESQview, and DESQview/X, and extending conventional memory an additional 32K, see your QEMM user's guide.

Windows tab

The Windows tab of QEMM Setup lets you review or change the configuration of QEMM's Windows features. The options that you see depend on the version of Windows you are running.

If you move the mouse pointer to one of the configuration options, the option will appear in highlighted text and you will see a brief description of what that option does in the Information area near the bottom of the window.



You must restart Windows to implement any changes you make on the Windows tab.

Buttons:

Reset resets any changes you have made in this editing session.

Defaults changes all the settings to their installation defaults.

Help displays this page of the Help system.

Other controls:

MagnaRAM (all versions)

Enable MagnaRAM

Use memory compression

Buffer Size

Compression Threshold

Windows 3.1 only

PAGEOVERCOMMIT

Windows 95 only

Enable TurboLoad

Windows 3.1 only

Use Resource Manager

FreeMeg (Windows 3.1 only)

Allocation Method

Block Size

SkipFile List

Windows 95 and Windows for Workgroups 3.11

Use Available Cache Memory

The XBDA (Extended BIOS Data Area) is a RAM region on IBM PS/2s and some PC clones that contains hardware information beyond that contained in the BIOS data area. The XBDA is normally located at the top of conventional memory and can be an obstacle to effective memory management.

A bus-mastering hard drive does its own direct memory access (DMA) without going through the PC's processor or its DMA controller. The most common bus-mastering hard drives are SCSI drives. Because bus-mastering drive controllers transfer information without going through the PC's processor, they circumvent QEMM's memory mapping which works at the processor level.

The frame method leaves the system, video, and disk ROMs in place. QEMM places the EMS page frame so that it lies on top of a ROM's address space. When the ROM at the location of the page frame is needed, QEMM saves the current contents of the page frame and restores the ROM to its original location. The ROM code then executes normally. When the ROM routine is finished, QEMM restores the previous contents of the page frame. The frame method typically provides 48K-64K of extra High RAM and is provided for systems that are incompatible with the mapping method.

StealthROM's mapping method maps system, video, and disk ROMs and any other Stealtable ROMs out of the first megabyte of memory. When the system needs the ROM, QEMM maps the appropriate ROM code into the EMS page frame. The ROM code then has a valid real mode address at which it can execute normally. When the ROM routine is finished, QEMM restores the previous contents of the page frame. This mapping method typically provides 83K-115K of extra High RAM. If your system is not compatible with the mapping method, try the frame method.

Certain PCs devote 384K of RAM to shadow memory. Shadow memory is a hardware feature for speeding up the execution of ROM code by copying that code from ROMs to faster RAM.

Top memory is a kind of memory found on certain systems--notably, some Compaqs and some machines with Micronics motherboards. Top memory is used to speed up ROMs and also used by some pieces of system software.

