A20 is a hardware address line which can be enabled or disabled under program control. The A20 line is important in the management of extended memory and the HMA. Programs usually enable the A20 line when using extended memory and the HMA, and disable it otherwise.

The batch file automatically executed by DOS when the system boots up. This file typically specifies the initial PATH and PROMPT, and loads whatever TSRs the user has specified.

Adapters

Available Screens:

- Video
- Disks
- Ports (Serial and Parallel)
- Micro Channel
- IRQs
- PCMCIA
- Battery

Battery Adapters (AB)

A major concern to laptop owners is battery life. Users often purchase a specific model based on the length of time they can expect to use it without recharging the battery. In order to minimize the problem of power consumption on portable computers, Microsoft and Lotus worked together to develop a specification called Advanced Power Management.

If your laptop computer is equipped with APM, you can check on its status by choosing the Battery topic under the Adapters category. The upper box provides information on the version of APM installed and its status (enabled or disabled.) You can also quickly determine whether the unit is connected to an AC adapter and if the battery is being charged. A second box shows the percentage of remaining battery life, allowing you to estimate the length of time you can work before the battery must be recharged.

DISK Adapters (AK)

The Disk Adapters screen identifies the floppy and hard disk drives installed in your PC. For each drive, Manifest gives the drive characteristics, type (identification number of the disk drive), and calculated capacity.

Drive characteristics include the number of read/write heads, cylinders, and sectors each drive has. Capacity is expressed in kilobytes (KB) or megabytes (MB.)

The Disk Adapters screen reports the formatted capacity of your physical disk drives. It does not reflect how those disks are partitioned into logical drives, nor does it report PCMCIA drives. This information can be found on the DOS Disks screen. You should be aware that if you use a disk drive utility such as SpeedStor the information reported in Disk Adapters may be incorrect.

Micro Channel Adapters (AM)

The Adapter Description Table screen shows you the special adapters installed in your PS/2 or compatible--according to Manifest's Adapter Description Library (MCA.ADL) file. For each adapter, Manifest tells you its slot number, device type number in hexadecimal, identifying name, and the area of memory used.

You can use this screen to display adapter information without the need for your reference disk, resolve conflicts when programs use the same addresses, understand the memory usage of your special adapters, and print additional inventory information about your PC.

NOTE: The MCA.ADL file must be located in the QEMM subdirectory in order for Manifest to read the information contained in the file.

Serial and Parallel PORT Adapters (AO)

The Serial/Parallel Adapters screen shows the industry standard Port I/O adapters present in your PC. It tells you the number of ports, the BIOS port address, and current signal state of the adapter.

Experienced PC users may find the signal states helpful in tracking down problems. Serial ports report the following signal conditions:

CD	--	Carrier Detect
RI	--	Ring Indicator
DSR	--	Data Set Ready
CTS	--	Clear To Send

Parallel ports have the following signal states:

SEL	--	Select
OOP	--	Out of Paper
ACK	--	Acknowledge
NBZ	--	Not Busy

For both serial and parallel ports, Manifest indicates the active status by an * and an inactive status by a -.

Communications Ports and Manifest for Windows

VIDEO Adapters (AI)

The first Adapter screen is the Video Display screen. It provides details about display adapters (video boards) that are installed in your PC. The adapter type is given in industry standard terms (such as CGA, EGA, or VGA) and areas of adapter memory used by the video board are listed.

If the video ROM contains a "signature" identifying the manufacturer and the specific model number, this information will be provided. Also listed is the amount of memory installed on the board, usually 256K to one megabyte. On systems containing two display adapters, information about both is provided.

You can use the Video Display Adapter screen to verify the type of display and adapter installed in your PC, identify areas of upper memory used by your adapter(s), and verify that your PC recognizes two adapters.

The range of memory addresses that the processor can reference. PCs running in real mode have an address space of 1 megabyte. Expanded memory can only be referenced after it is mapped into the processor's address space.

Adjusting the Sizes of Your DESQview Windows

Use the DESQview **Move and Resize commands** to adjust the size of the window and move it around on your screen. When you are happy with the size and placement, press the spacebar to update the figures on this screen. You can now use Change a Program to enter these figures into the .DVP of any program.

An **All Adapters** report includes the following:

Adapters / Video
Adapters / Disks
Adapters / Ports
Adapters / MicroChannel
Adapters / IRQs
Adapters / PCMCIA
Adapters / Battery

An **All Manifest** report includes every category of information provided by Manifest. The All Manifest report can be quite lengthy, particularly if it includes the two Windows configuration files, WIN.INI and SYSTEM.INI.

An **All Overviews** report includes the following categories:

System / Overview
First Meg / Overview
Expanded / Overview
Extended / Overview
DOS / Overview
Adapters / Video
QEMM / Overview
Windows / Overview
Hints / Overview

Analyzing the DOS FILES Display

Microsoft Windows and Files:
Manifest's report of the total number of DOS file handles on your system will often be different inside Microsoft Windows than outside of it. This is because Windows often increases the number of files on the system.   You may also observe that a different number of files is available to a Windows application than to an application running in a Windows DOS window. Manifest reports these different numbers accurately.

Network Files and Manifest:
If you are connected to a network and run Manifest locally from your PC, Manifest reports the files open on your PC, but does not report any files open on the server. If, however, you run your licensed copy of Manifest from the server, it indicates all files open on the server, including the files opened on the server by a node.

Already Open Files:
There are three "standard" files that you will always see as open. DOS opens three devices, AUX, CON, and PRN, as files so that they can be accessed by programs through standard DOS requests. Thus there are typically three fewer file handles actually available to programs than the number specified on the FILES line of the CONFIG.SYS.

Files "Unknown" Display:
A Files display entry with the text "Unknown" indicates that Manifest has detected an inconsistency in the file table. This typically is due to a DOS bug which leaves a file table entry marked as "in use" when it is not. It also means that this entry will remain unavailable until you reboot your system.

FCBs Used:
While DOS keeps careful track of its table of open files, it is less careful with its table of open FCBs. In fact, many times it will leave entries in its FCB table marked as open even after they have been closed. This is usually not a problem. DOS can reuse slots in its FCB table even when there are open FCBs there. This characteristic can be prevented by using the second parameter to the DOS FCB statement. The second parameter is a number indicating how many of the total entries (specified by the first number) in the FCB table should be "protected" from being used when the entry is still marked as open. By default this number is 0, meaning all slots can be reused at any time. Also, even when DOS' FCB table is irretrievably filled up, programs can still open files using FCBs. Fortunately, this confusion is usually academic, since DOS never pays attention to its FCB table. Only certain utilities, such as the SHARE program that comes with DOS, do anything with the FCB table.

If you prefer to add the new information to an existing text file, select **Append to file.**

Assigning Names to Programs

If Manifest cannot determine the name of your program or device, it gives it a name. The name given is the memory address where the program is running.

The Basic Input Output System software which provides low level, basic functionality to your system. This software is embedded in one or more ROMs in the upper memory area.

The data area that is used by your PC's BIOS to keep track of the state of your keyboard, video adapter, and other standard hardware.

A CONFIG.SYS statement that defines the number of disk buffers placed by DOS in memory to enhance disk operations. Each buffer uses about 528 bytes of memory. Increasing the number of buffers tends to speed up file operations at the cost of decreasing available memory.

A special memory area that is used to store configuration information needed at system boot-up. A battery maintains this memory while system power is off.

CMOS List

The CMOS List (of interest to the technical PC user) shows the byte by byte detail of your CMOS memory in tabular form. The columns in the display show the byte offset in hexadecimal, the meaning of the byte, and the current value of the byte in hexadecimal. You get this screen by pressing the F3 key when the CMOS Summary screen is displayed.

Note that most PCs have only 64 bytes of CMOS, so addresses 40 through 7F are a copy of bytes 00 through 3F.

Experienced users may find this screen helpful in resolving differences in Manifest's reporting of installed equipment or ferreting out undocumented use of reserved bytes by your PC manufacturer.

The standard command processor used by DOS. This is the program responsible for displaying your DOS prompt and processing your DOS commands (COPY, DIR, etc.).

The file which defines the initial configuration of DOS. It typically specifies the number of DOS resources to be allocated and the device drivers to be loaded.

To exit the Print menu without printing any reports, click on **Cancel** or press the **Esc** key.

Words or symbols on a command line which follow the program name and specify certain commands or options to the program.

Manifest, when running as a Windows program, may be unable to determine the correct status of the communications ports.   This is because Windows conceals information about the ports from Windows programs.   Manifest can give correct information about the communications ports when it is run from the DOS command line, either inside a Windows DOS window or outside of Windows.

The space taken up in the MagnaRAM compression buffer by compressed physical memory.

The space taken up on the hard disk by compressed virtual memory.

The current size of the MagnaRAM compression buffer.

A ratio indicating how effectively MagnaRAM has been able to compress memory since the beginning of the Windows session.

Manifest for Windows

Contents:

For **Help** using the online Help, press **F1**.

The contiguous memory starting at 0K used by DOS to run programs. This is usually 640K in size.

An optional auxilliary processor used to speed up complex math operations.

Manifest for Windows

Online Help

Designed By

Kathy Hand

Text By

Kathy Hand and Dan Sallitt

Current Selection refers to the Manifest category that is highlighted.

DESQview

Available Screens:

- Overview
- Memory Status

DESQview MEMORY STATUS (VM)

DESQview Memory Status shows information about the amount of memory that is available to DESQview. This display is similar to the Memory Status program which comes with DESQview. This screen is only displayed when Manifest is run in a DESQview window.

You use this screen to determine the largest program that can be run in DESQview, see how much expanded memory is available in which to run programs, and check the current status of common memory.

Terms Relating to the Memory Status Screen

Maximizing Your Memory for DESQview

An operating environment created by Quarterdeck Corporation which allows off-the-shelf DOS programs to multitask within separate windows on the screen. It also provides a menu-driven interface to DOS, a macro facility, and the ability to transfer information from one DOS program to another.

DESQview OVERVIEW (VO)

DESQview Overview gives you information about DESQview and the window in which Manifest is running. This screen only appears if you are running Manifest in DESQview or DESQview/X.

You can use this screen to look up the DESQview version number,   see the Memory Size (in K) allocated to this window (as specified in Change a Program), and determine the row and column positions of your DESQview window.

Adjusting the Sizes of Your DESQview Windows

A CONFIG.SYS statement that tells DOS to load an add-on device driver at system startup.

DOS

Available Screens:

- Overview
- Drivers
- Files
- Environment
- Disks

The data segment used by DOS.

DOS Disks (DK)

The DOS Disks screen provides information on "local disks" or disk drives that are physically installed in your stand-alone computer. Each local drive is listed and described as follows:

Drive letter assigned to each drive.

Type of drive. (Floppy drive, hard drive, optical, **DevDriven**, etc.)

Capacity of drive. (In the case of floppy drives, this refers to the capacity of floppy diskettes used in the drive.)

Currently logged directory. (Not given for floppy drives.)

Additionally, DOS Disks displays a list of any network drives you may be logged onto at the time Manifest is run. Drive letter, drive name, and the currently logged directory is shown for each network drive listed.

You can use the information in this display to verify the number and type of drives installed in your system and to troubleshoot problems involving "invalid drive specification" or other drive-related error messages.

DOS Environment (DE)

The DOS environment is memory set aside for the purpose of assigning names and giving values to those names (variables) which programs can use in a number of ways. Every program in memory, including COMMAND.COM, is given a copy of the environment.

This screen shows you how much memory (in bytes) there is in the environment and how many bytes are as yet unused. The variables set and their values are shown in the lower window.

You can use this screen to:

Find out where COMMAND.COM is located (by way of COMSPEC).

Verify the presence of a variable and its value, i.e. make sure your PATH= variable lists all the subdirectories it should.

See how COMMAND.COM displays its prompt (PROMPT=).

Verify the presence of variables your applications expect to find in the environment and their values.

Determine if you should increase or decrease the number of bytes in the environment.

Aid you in responding to questions about your environment from support personnel.

See, in DESQview, if applications in different windows have different environments set up by batch files.

Software that allows programs to use protected mode while running under DOS. Running in protected mode allows programs access to the larger address space of the 80286 and higher processors.

DOS Files (DI)

DOS Files shows you the total number of files and FCBs specified and the number open. It also lists the names of the open files and FCBs, and when possible, the name of the program that opened the file.

Manifest finds the information about which files and FCBs are open by looking in DOS's files and FCBs tables when this screen is selected. If you press the space bar, Manifest reexamines these tables and updates the display.

You can use this display to determine the adequacy of your FILES= level, determine programs which keep files open for extended periods, identify programs using FCBs, and, in DESQview, to see files opened by several programs at a time.

Analyzing the DOS FILES Display

The two program files which contain the DOS code. These files are called IBMBIO and IBMDOS in PC-DOS, and are called IO and MSDOS in MS-DOS.

DOS Overview (DO)

DOS Overview gives you a summary of the memory used by your version of DOS. Using this screen can help you understand the memory requirement of the devices loaded and DOS resources allocated in CONFIG.SYS as well as find out the amount of memory used by DOS.

Terms Relating to the DOS Screens

Making Sense of the Numbers

Using DOS Resources: Optimally Configuring Your System While Maximizing Your Memory

BUFFERS, FCBS, FILES, LASTDRIVE, and STACKS are resources utilized by MS-DOS based on either statements in the CONFIG.SYS file or default values assigned in the absence of such statements.

DOS Drivers (DR)

The DOS Drivers screen gives you information about the drivers Manifest finds on DOS's list of drives, called the device driver chain. These are standard devices always present in the DOS kernel and other devices as specified in CONFIG.SYS.

This screen lists the actual order in which a driver is loaded into the driver chain. Usually the first device found is the NUL device.

For each driver specified in CONFIG.SYS, Manifest displays the beginning and ending memory addresses used by the driver, its memory size and attributes. Both the name found in the device structure in memory and the program name (if available) are listed.

Note that not all drivers show addresses of memory. The memory used by standard device drivers is embedded in the DOS kernel, and neither address range nor size is given.

Driver Attributes

How Manifest Describes Drivers

The size of the compression buffer that MagnaRAM will create if you choose the Defaults button on the Windows tab of QEMM Setup. This figure is equal to a quarter of the available unlocked physical memory when MagnaRAM initializes.

Some drives, including those created by disk compression utilities such as SuperStor, Stacker, and DoubleSpace, do not "exist" unless a device driver is loaded in the CONFIG.SYS file during bootup. Such drives may be identified as "DevDriven."

A piece of software used to communicate with an installed hardware device (such as a printer, video adapter, disk drive, or memory card). It is either built into DOS or loaded using the DEVICE statement in CONFIG.SYS.

Displaying the Interrupt List

The Interrupt List consists of sixteen screens that list interrupts by their number. The list also shows the name or purpose of the interrupt and the address and name of the program providing the service.You display this list by pressing the F3 key when the Interrupt Owners screen is displayed.

You can display any one of the 16 screens by entering Alt-X where x is the first digit of the hexadecimal interrupt number. For example, Alt-2 will display the screen containing interrupts 20 through 2F.

Driver Attributes

An attribute is a bit-encoded word containing 16 bits of information (9 bits defined; 7 undefined) that indicates highly technical characteristics of a device driver. In Manifest, the bits are indicated by C, S, B, x, R, L, K, N, O, and I :

C--character rather than block device

S--supports standard I/O control calls from DOS

B--supports output until a device is busy

x--undefined

R--supports removable media (i.e. floppy disk and CD ROM drives)

L--supports logical drives

K--clock device

N--NUL device

O--standard output device

I--standard input device

EMB (Extended Memory Blocks)

EMB refers to the remaining extended memory available for allocation as extended memory.

The current standard expanded memory specification which unified the two previous standards, EMS 3.2 and EEMS, and added support for multiple real alternate maps. Additionally, it added various software features which make it easier to write expanded memory programs.

The address of the expanded memory page frame. The page frame is a mappable area, 64K in size, which is the place that almost all EMS programs use for mapping.

Editing Text Files in Manifest

Manifest for Windows includes a text editor so that you can easily review and make changes to your DOS and Windows configuration files. The following files are listed on the drop-down menu which appears when you select **File** from the menu bar:

CONFIG.SYS

AUTOEXEC.BAT

SYSTEM.INI

WIN.INI

Any of these files can be loaded into the editor by selecting it from the menu. When you are done reviewing or editing the file, press **F3** or click the button labeled **Press F3 when done**. If you have made changes to the file, you will be prompted with a message asking if you want to save your changes. Select **Yes** to save your changes, **No** to discard edits made to the file, or **Cancel** to return to the editor.

You can also use the editor to view and edit files other than the ones listed on the File menu. To edit a file other than your CONFIG.SYS, AUTOEXEC.BAT, SYSTEM.INI, or WIN.INI, select **Open** from the File menu. A standard Windows "Open File" dialog box appears, from which you can select the file you want to edit.

Editing the Windows 95 Registry

Windows 95 includes a text file called the registry which contains detailed information about your computer and software. Also included with Windows 95 is a program called the **Registry Editor**, an advanced tool that enables you to change settings in your system registry.

Most users do not need to edit this file directly, and it is recommended that you not edit your registry unless it is absolutely necessary.   If you make an error when editing this file, your computer may become nonfunctional.

For those advanced users who may desire to tweak their systems by editing the registry, Manifest for Windows includes a **Tools** menu from which the Windows 95 Registry Editor can be run.   The Tools menu is only available from within Windows 95.

1. The block of memory which contains the current PATH, PROMPT, COMSPEC, and any other variables defined using the SET command.

2. A program, such as Quarterdeck's DESQview or Microsoft Windows, which resides on top of DOS and whose purpose is to run other programs in ways which enhance the usability and/or functionality of your system.

On QEMM's Analysis screen, Exclude (X) refers to areas of memory that have been used by a program, but which QEMM did not expect to be used.   You should use the EXCLUDE switch to QEMM to prevent QEMM from using these areas.

When used as a parameter to QEMM, EXCLUDE (or X) is used to designate an area of memory that QEMM should not fill with High RAM.

Unmappable (U) pages are those 16K segments of address space into which memory cannot be mapped because the segment is in use by an adapter or has been reserved as High RAM, the memory manager cannot or has been told not to map there, or a TSR or environment such as DESQview has reserved the pages for its own use.

Expanded BENCHMARK (PB)

Expanded Benchmark tests your expanded memory manager for speed in performing EMS calls. The results are measured in microseconds (1 million microseconds = 1 second), and show the maximum, minimum and average times required for the given function.

This screen only appears if your PC has expanded memory. Note that the Benchmark requires 64K of expanded memory to run. If your PC does not have this much available expanded memory, the Benchmark will not run. As you will notice, this is a dynamic display. The test results change until a significant sample has been taken.

If your expanded memory manager does not support one or more of these calls, then the data for the corresponding row will be left blank. This will also occur if you have too little expanded memory available for the open handle tests.

Using the Expanded Benchmark Screen

What the Tests Mean

Expanded HANDLES (PN)

The Expanded Handles screen lists the currently allocated EMS handles. The table lists the handle number, the amount of memory allocated in K, and, if a program has named the handle, its assigned name.

This display appears only if you have expanded memory in your system.

You may use this list to determine which of your application programs make use of expanded memory and the amounts of memory they are using. In some cases (i.e., in DESQview, or by commands to the application itself), you may be able to limit the amount of memory the application is able to acquire if its expanded memory use is excessive for your purposes.

Terms Relating to Expanded Handles

Memory supported by a memory manager which can be mapped into the first megabyte of address space.

Expanded Memory (EMS)

Available Screens:

- Overview
- Pages
- Handles
- Timings
- Benchmark

Expanded Memory and DESQview

Alternate Maps & DESQview:
The presence of alternate maps enables DESQview to better manage high speed communications programs. Without alternate maps, such programs may occasionally lose characters when running in the background. In systems with only one alternate map available, you should make sure you open up your communications program first for the best performance. Additionally, QEMM's alternate maps provide DESQview with the ability to virtualize graphics programs and to implement its protection features. DESQview uses one real alternate map for itself plus one for each window you open.

Mappable Pages & DESQview:
If you are viewing this screen while running DESQview, the number of mappable pages is four because DESQview is controlling the memory. If you want to see the number of mappable pages your PC is capable of or how much conventional memory is backfilled, run Manifest outside of DESQview.

Expanded OVERVIEW (PO)

The Expanded Overview shows the type and amount of expanded memory in your PC. This screen appears only if your system currently has expanded memory, with an expanded memory manager loaded and enabled.

The Expanded Overview screen has two sections. The top section gives you information about your EMS driver. The lower section of the screen accounts for the memory and other resources managed by the driver.

You can use this screen to quickly identify the version number of your expanded memory manager, determine what kind of expanded memory manager you have, and determine how much expanded memory is being used by your programs.

Manifest makes expanded memory manager calls to determine the information it reports. If Manifest hangs when you select the Expanded Overview screen, the problem may be a bug in your EMS driver or in a TSR which intercepts and mishandles an EMS call Manifest makes.

Terms Relating to Expanded Memory

Expanded Memory and DESQview

Expanded PAGES (PG)

Expanded Pages displays a map of the first megabyte of memory as it pertains to use by the expanded memory manager. This display appears only if your computer has expanded memory and an expanded memory manager is loaded and enabled.

Each character in the map represents the status of one 16K memory area, or "page". Each of the 16 rows contains 4 characters, and thus represents 4 pages, a 64K area. Each page is either mappable (+), Page Frame (F), or unmappable
(U).

The rows of the map are identified by their hexadecimal address, where the **n** is to be replaced by the appropriate hexadecimal number appearing along the top of the table.

The mappable pages summary shows the Memory Area range (in hexadecimal), the size of the area, and the range of logical pages assigned to this memory range. The page frame is present to allow EMS 3.2 capabilities. Other entries may or may not appear depending upon your EMS setup.

Terms Relating to Expanded Pages

Using the Expanded Pages Screen to Solve a Problem

Expanded TIMINGS (PM)

Expanded timings provide memory access times for your expanded memory.   The purpose of this display is to show how fast your expanded memory is relative to a PC/XT and to show the relative speeds of different areas of expanded memory.

Speed Test Variances:
There may be up to a 10% variance in the speed reported from one test to another.

Timings & QEMM:
If the speed of your expanded memory is faster than the speed of your ROMs (see First Meg Timings), you can use QEMM's ROM feature to significantly increase the speed of ROMs in your PC.

Memory addressed above 1024K (one Megabyte).

Extended Memory (XMS)

Available Screens:

Overview
XMS

Extended OVERVIEW (TO)

The Extended Overview provides information on XMS, DPMI, and VCPI memory available to your programs. It also provides a map of your extended memory and shows how it is being used.

The Extended Overview screen has two sections. The top section provides the versions of XMS, VCPI, and DPMI installed in your system. Also provided is information regarding the amount of VCPI memory available and whether or not 32-bit DPMI support is enabled.   The bottom section shows each area of extended memory that is used or available, the area's beginning and ending addresses, and its size.

How Memory is Allocated

Undetected Extended Memory Usage

Extended XMS (TM)

XMS defines **three areas of memory** covered by the specification:

HMA (High Memory Area)

UMB (Upper Memory Blocks)

EMB (Extended Memory Blocks)

Individual XMS drivers may or may not support all three.

Terms Relating to Extended Memory

XMS and QEMM

Programs Which Request All Extended Memory

NOTE: Included in your QEMM directory is a program called **XMS.COM**. This program provides detailed information about how your system's XMS memory is being used. Documentation for XMS.COM is also included in your QEMM directory in a text file called **XMS.DOC**.

A file control block. This was the data structure used by DOS 1.XX programs to access files. It is rarely used by today's programs.

A CONFIG.SYS statement which specifies the number of file control blocks kept track of by DOS.

A CONFIG.SYS statement which specifies the maximum number of files which can be open simultaneously.

You can elect to save the information reported by Manifest as a text file. By default, Manifest saves the report in your QEMM directory with the **filename MANIFEST.PRN**. Manifest gives you the option of specifying an alternative directory and/or filename. If necessary, you can use the standard Windows **Browse button [...]** to select a drive and directory for your file.

First Meg

Available Screens:

- Overview
- Programs
- Interrupts
- BIOS Data

First Meg BIOS DATA (FB)

The First Meg BIOS Data screens present data on the area in RAM devoted to use by the ROM BIOS.

This BIOS Data area is an area of conventional memory used by the system ROMs for keeping track of the state of the system--the keyboard, disk drives and video adapter cards. The BIOS data area is 172 bytes of memory located just above the interrupt vectors.

Many application programs also reference, and some change, the information contained in this memory area.

Using the First Meg BIOS Data Screens

PCs with an Extended BIOS Segment

First Meg INTERRUPTS (FI)

The First Meg Interrupts screen presents two detailed views of your PC's interrupt vectors. The Interrupt Owners screen lists the programs, drivers, and ROMs providing interrupt services, followed by the interrupt numbers they service. The Interrupt List screen lists the interrupts by their number, giving also the name or purpose of the interrupt, its address, and the name of the program, driver or ROM providing the service.

These screens only show the most recent program which has the vector. Other programs loaded earlier may also be servicing the interrupt. On a color display, program and driver names are displayed in cyan, while DOS and ROMs are shown in white.

These screens provide technical information of use to programmers and technical support personnel. If you are a non-technical user, your ability to relay this information may help to resolve some difficulty you are experiencing.

Using the Interrupt Owners Screen

Displaying the Interrupt List

Using the Interrupt List

First Meg OVERVIEW (FO)

First Meg Overview shows you a map of the memory areas used in your PC's first megabyte of address space and details how you are currently using memory.

The map has two sections: the area known as conventional memory (usually the first 640K of memory) and the area between conventional memory and the first megabyte boundary, known as the upper memory area. Portions of this area are reserved for hardware adapters and BIOS ROM.

Information on this screen can help you optimize your memory usage.   Unfamiliar terms used here are more fully defined in the sections on Expanded memory and QEMM and in the Glossary.

Using the First Meg Map

Understanding the First Meg Map

Unused Areas

Hints for DESQview Users

First Meg PROGRAMS (FR)

The First Meg Programs screen shows you a map of what is loaded into memory (excluding DOS and device drivers), and the areas of memory available for use. If you are using QEMM, the display may be divided by a double line indicating where conventional memory ends and the upper memory area begins.

The program map of the upper memory area shows where memory resident programs, drivers (i.e., network and mouse drivers) and pieces of DOS have been loaded by QEMM's LOADHI or DOS-UP. It also shows your available memory.

Using the First Meg Programs Screen

Assigning Names to Programs

Why Manifest Isn't Listed

Using Available Upper Memory

The Total Jumbo Physical figure minus the compressed and uncompressed physical memory in use.

The Total Jumbo Virtual figure minus the compressed and uncompressed virtual memory in use.

Total Linear minus the amount of memory that has been allocated from the Windows memory manager.

Available physical memory.

Available virtual memory.

An area of memory used for data related to drawing graphics devices like icons, window borders, and so on.

A pool of memory--a portion of your system's total memory--that is managed by Windows.

Glossary

A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

A

A20
Adapter RAM
Address Space
AUTOEXEC.BAT

B

BIOS
BIOS Data Area
BUFFERS

C

CMOS
Command Line Switch
COMMAND.COM
Common Memory
CONFIG.SYS
Conventional Memory
Coprocessor

D

DESQview
DEVICE
DevDriven
Device Driver
DOS Data
DOS Extender
DOS Kernel
DOS Resource
DOS-UP

E

Environment
Exclude (X)
Expanded Memory
EMS 4.0
Extended Memory

F

FCB
FCBS
FILES

G

GDI Heap
Global Heap

H

Handle
High Memory Area
High RAM
HMA

I

Include (I)
Interrupt Vectors

K

K
Keyboard Buffer

L

LASTDRIVE
LOADHI
Logical Page

M

Mappable
Mappable Pages
MB
Memory Address
Micro Channel Architecture (MCA)

N

Network Adapter

O

OK

P

Page Frame
Parallel Ports
Processor
Protected Mode

Q

QEMM
QEMM Mode
QEMM Status

R

RAM
Rammable
Real Alternate Maps
Real Mode
Reserved
Resources
ROM

S

Serial Ports

Shadow RAM
Split ROM
STACKS
Stealth ROM
System ROM

T

Top Memory
TSR

U

Unassigned Memory
Unmappable Pages
Upper Memory area
Upper Memory Blocks (UMB)
User Heaps

V

VCPI
Video Adapter
Virtual 8086 Mode
Virtual Device Driver (VxD)

X

XMS

High Memory Area.

HMA (High Memory Area)

The HMA is a 64K block of memory starting one byte below the one megabyte boundary (1024K), the beginning of extended memory. Its primary and most beneficial use is in enabling program code, nominally restricted to the conventional memory area, to be loaded and executed in the HMA, thus adding 64K to the addressable DOS memory space.

A shortcut value used to refer to a particular item. For example, DOS files are assigned a handle when opened so that a program need not refer to the full name each time. Expanded memory handles are used to refer to groups of expanded memory pages. XMS handles are used to refer to blocks of extended memory.

A **Hardware** report includes the following categories:

System / Overview
System / Config
System / Autoexec
System / CMOS
DOS / Disks
Adapters / Video
Adapters / Disks
Adapters / Ports

The area, as defined by XMS, from 1024K to 1088K, i.e. the first 64K of extended memory. It is important because it is the only extended memory that can be accessed directly by DOS programs. This area is also referred to as the HMA.

Portions of the upper memory area into which RAM has been mapped. Device drivers and TSRs may be loaded here to minimize conventional memory overhead.

Hint Example

As an example, consider the possible hint "Reduce the number of DOS FILES allocated." The reason for the hint is that your FILES=statement in CONFIG.SYS is set to a large number, say 99. For most purposes, this number is far more than is needed and the memory required to support that many handles is excessive. However, a few applications do require a large number of file handles. They will have altered your CONFIG.SYS file during the installation process (or will have requested that you do so) and will not initialize or function properly if that number of files is unavailable.

If you are unsure why the value is set as it is, you may want to experiment with lower values or check the documentation that accompanied your hardware or software for further information.

Hints

Available Screens:

- Overview
- Detail

Hints DETAIL (HE)

The Detail screens address each of the hints given in the Overview section. Each screen covers one suggestion and has the following format: a WHY section, a WHY NOT section, and a HOW section. The bottom of each screen will indicate if there are more hints.

The details contained in each section vary depending on the topic involved. The WHY section states the reasons which motivate the hint, citing numbers if their inclusion would be helpful. The WHY NOT section provides the counter argument, which might remind you of special situations requiring your current configuration. The HOW section describes the action to take if you agree to the suggestion.

Hint Example

Note: It is important to remember that items listed on Manifest's **Hints** screens are **suggestions only**. While a given hint may be appropriate for a typical computer system, your system may not be typical!

You (or the person who configured your system) may have had a valid reason for setting up your computer in a way that differs from other systems. Don't be afraid to ignore a hint if you feel your system is already optimally configured.

Hints OVERVIEW (HO)

The Hints Overview screen presents you with a list of one-line descriptions of optimizations you may wish to perform.

If Manifest has not found any conditions upon which to comment, the following message is displayed:

> **"Everything checks out OK.   Nothing to suggest."**

Hints for DESQview Users

DESQview users, running Manifest in a window, will see information related to the configuration of the window's memory, not the configuration of the PC's memory. This means that the memory marked as Available is the amount of memory available for use by programs running in the window, not by all programs on the PC.

Secondly, DESQview makes it so that all memory which is not usable by programs in the window will not be included in the window's known memory area. This makes Manifest report that conventional memory ends not at the "real" end of conventional memory (usually 640K), but where the program's memory partition ends.

Finally, all areas marked as Mappable outside of DESQview will be hidden from programs running in DESQview, and will be designated by Manifest as Unused.

How Manifest Describes Drivers

In describing your drivers, Manifest uses standard methods to find its information. Sometimes this information is incomplete.

Program Name:
Manifest may not be able to determine the name of the program file which contains your device driver. In this case, the field is blank.

Memory Area or Size:
Memory areas and sizes are only shown for device drivers loaded in your CONFIG.SYS. Other drivers are either part of the DOS kernel or are part of memory-resident programs, such as Novell's NET3, that hooked themselves into the driver chain. Refer to the First Meg Programs screen for information on the memory usage of memory-resident programs.

Device Memory Usage and LOADHI.SYS:
If you have used LOADHI.SYS to load drivers into the upper memory area, the memory area and size listed in this section indicates only the addresses used by LOADHI.SYS and not the addresses used by the program. To find out how much upper memory area is used by the device, refer to the First Meg Programs display.

Missing Drivers:
The DOS Drivers screen reports on drivers that are present in the DOS driver chain. Some device drivers simply do something when loaded and then remove themselves from memory. These drivers will not appear in Manifest's list.

How Memory is Allocated

Some programs allocate extended memory from the lowest to the highest address first, beginning at 1024K. The map indicates this by indicating Used from the Bottom in its status column.

Other programs allocate extended memory from the top of extended memory downward. When these programs are present, the map indicates their status as Used from the Top. Quarterdeck's QEMM and DOS extenders allocate memory from the top.

Remaining, unallocated memory is called Available.

Using the First Meg Map

You can use the First Meg map to:

Find out the size of the largest program you can run in conventional memory (Available).

Find out how much of conventional memory is being used by COMMAND.COM and TSRs (Program Area).

Determine if there is an upper memory area, designated as Unused, where Manifest finds neither ROMS nor address space that could be mapped.

See what memory, designated as High RAM, is usable by QEMM's LOADHI to load drivers, TSRs or parts of DOS in the upper memory area.

IRQs

The Adapters/IRQs category provides a reference of hardware IRQs that are in use. The screen lists three columns of information: IRQ number (0 through 15), current use of the IRQ, and its default use.

Manifest is only able to detect that an IRQ is in use if it has been "hooked" by a piece of software. For instance, you must load software in order to access a CD-ROM drive. Once that software has been loaded, Manifest will identify the IRQ used by the drive. Manifest will not, however, report that an IRQ is in use if no software has been loaded which accesses it. As a result, you may have fewer IRQs available than this screen indicates. IRQs which have not been hooked by software are listed as Unassigned.

On QEMM's Analysis screen, Include (I) refers to areas of upper memory that have not been used by a program, but have been reserved by QEMM.   You can use the INCLUDE switch to QEMM to allow QEMM to use these areas.

As a parameter to QEMM, INCLUDE (or I) designates an area in which QEMM should create High RAM.

Information Displayed

The upper box provides details on the following items: the type of processor (80386, 80486, etc.), the current mode of the processor, BIOS manufacturer and date, bus type, and whether or not the system is equipped with a math coprocessor. The type of video adapter is listed, as well as details on your mouse, keyboard, and the number of parallel, serial, and game ports.

The lower box shows both total and available memory in each of four categories: Conventional, Expanded (pooled), Extended (pooled), and Upper Memory (High RAM created by QEMM during bootup.) The difference between total memory and available memory represents memory being used by DOS, TSRs, disk caches, network drivers, etc.

The table of locations in memory through which hardware events and software requests may get serviced by a supporting program. The first 1024 bytes (1K) of conventional memory is devoted to storing interrupt vectors.

The abbrevation for kilobyte, equal to 1024 bytes of memory.

The place where the BIOS stores keystrokes as they are typed.

A CONFIG.SYS statement which specifies the drive letter of the last drive that DOS should support.

Two programs included with Quarterdeck's QEMM, used to load programs into High RAM. The LOADHI.COM program loads TSRs and DOS resources into High RAM, while the LOADHI.SYS program loads device drivers into High RAM.

The number assigned by the expanded memory manager to each mappable page. The page frame will always be assigned numbers 0 through 3.

The abbreviation for megabyte, equal to 1,024K or 1,048,576 bytes of memory.

MS Windows

Available Screens:

- Overview
- Global Heap
- Resources
- Programs
- VxDs
- DevCaps

Making Sense of the Numbers

Manifest's report of your system configuration (bottom left of screen) may differ from your CONFIG.SYS. DOS allocates a default value for FILES, etc. regardless of whether you specify it. This default varies, depending upon the version of DOS.

The number of DOS FILES and BUFFERS shown in the memory map may be less than that reported in the left column of the Manifest display. In some versions of DOS the default memory allocated to buffers or files may be embedded in the DOS kernel. Manifest notes the files and buffers included in the DOS kernel.

You may be able to reduce the conventional memory used by DOS: you can increase the memory available to programs by using QEMM's DOS-UP to move DOS resources into the upper memory area.

A portion of the address space through which expanded memory can be accessed.

Mappable (+) pages are 16K segments of address space in the first megabyte into which an expanded memory manager can map and include unused segments above conventional memory in the area nominally devoted to adapter memory.

If Manifest's Expanded Pages screen identifies any pages as Mappable (+) you could create High RAM with them by using the RAM parameter on QEMM. Anywhere that you have created High RAM will show up as Unmappable (U) here.

Maximizing Your Memory for DESQview

Largest Expanded Memory Available is 0K (zero K):
If Memory Status indicates that no expanded memory is available you have either used all of the expanded memory in your system or your expanded memory manager is misconfigured.

You Cannot Open a Large DESQview Window:
If your largest block of expanded memory is not large enough to allow you to open a given program, take a close look at what you are loading in your CONFIG.SYS and AUTOEXEC.BAT files. Remove unnecessary or seldom-used drivers and TSRs, or try loading them in a DESQview window. Also check to see if Stealth ROM is enabled on your QEMM device line in the CONFIG.SYS. If you are loading DOS high (DOS=High in the CONFIG.SYS) try removing this line so that DESQview can load into the HMA. Keep in mind that DESQview requires 7 to 12K per window for its overhead. By changing various settings in DESQview's SETUP program or by altering a specified program's configuration via Change a Program, you can often free up enough additional memory to run a large program. Refer to your DESQview manual for further information.

Available Common Memory Usage:
If the largest available common memory falls below 1500K, you may not be able to open any more windows. However, common memory comes out of conventional memory, so do not increase this figure unnecessarily.

Maximizing Upper Memory:
DESQview can load much of itself into an upper memory area. It can map memory into mappable locations above conventional memory and share High RAM with resident programs. Using this screen, you can determine if there is more memory available in an upper memory area than DESQview can use. With Manifest running as the first and only DESQview window, the difference between the Largest Available Conventional Memory and the Total Available Conventional Memory is the amount of memory that exists in an upper memory area but has not been used by DESQview or anything loaded before DESQview. If this amount is large, more resident programs could be loaded high without impacting conventional memory size inside DESQview.

The largest possible size for the MagnaRAM compression buffer under the present system conditions. This figure is equal to 100% of the available unlocked physical memory when MagnaRAM initializes.

The memory used internally by DESQview for housekeeping, storing marked data, and providing memory to DESQview-specific programs. The amount of Common Memory allocated is specified by DESQview Setup.

A **Memory** report includes the following categories:

System / Overview
System / Config
System / Autoexec
First Meg / Overview
First Meg / Programs
Expanded / Overview
Expanded / Pages
Expanded / Handles
Extended / Overview
Extended / XMS
QEMM / Overview
QEMM / Type
QEMM / Memory
Windows / Overview

A number indicating a location for the processor to look for a piece of memory. There may or may not be a piece of memory actually at that location. To use memory addresses that have no memory associated with them, you must first map memory into them.

A bus design implemented by IBM in its PS/2 line of computers. Its relevance to Manifest is that it allows easier identification of what adapter cards are installed.

A **Minimal** report includes the following categories:

System / Overview
System / Config
System / Autoexec

A **Network** report includes the following categories:

System / Overview
System / Config
System / Autoexec
System / Network
DOS / Disks

A hardware card that provides the interface between your processor and a Local Area Network. This hardware often reserves portions of the upper memory area for its own use.

If you prefer to generate the report as a newly created text file, select **New file.**

On the QEMM Analysis screen, OK (O) refers to areas of memory which may or may not be used by a program but which are used or not used in the manner in which QEMM expected them to be.

PCMCIA

Manifest detects the presence of **PCMCIA** devices if Card Services software version 2.1 or later is running on your machine.

The PCMCIA screen reports the PCMCIA release version, driver revision, vendor signature, total number of sockets on the system, and the range of memory addresses used. Manifest also provides information on the device installed in each socket. If a socket is not currently in use, Manifest reports that socket as Empty.

PCs with an Extended BIOS Segment

Manifest reports the existence of the Extended BIOS segment at offset 0E. In other systems, this location is assigned to a possible fourth parallel port. If your computer displays only 639K of conventional memory, this is an indication that the computer does have an Extended BIOS Segment. (Conventional memory ending below 640K can also signal the presence of virus. Be sure to run an up-to-date virus scanning program on your system if Manifest suddenly reports something other than 640K conventional memory.)

Note: By default QEMM automatically relocates the extended BIOS Data area.

An EMS term describing the mappable area, 64K in size, which is used by most EMS programs for mapping.

Hardware ports used for communication with parallel devices such as printers, accessed through I/O addresses stored in the BIOS data area.

When you are satisfied with your selections, click on **Print** to create the report.

You can send a report to any **printer** device that is installed on your system, including fax devices.

Printing Manifest Reports

You can print a wide variety of reports based on the information provided by Manifest. To display the Print menu either press the F2 key or select Print from the File menu. The menu contains the following choices:

Where to Print:
    Printer
    New file
    Append to file

What to Print:
    Current Selection
    All Adapters
    All Overviews
    All Manifest
    Minimal
    Windows
    Memory
    Hardware
    Network

Filename

Print Button
Setup Button
Cancel Button

The chip which controls your PC and determines its capabilities.

Programs Which Request All Extended Memory

Certain programs using extended memory will request and be allocated all available EMBs. This will also exhaust expanded memory, since they are allocated from the same pool of available memory. If you have applications which do this and they prevent other programs or TSRs from obtaining the memory they need, you should check to see if that application provides some means of limiting the amount of memory it requests. In DESQview, this limiting capability is provided by a field in the program's .DVP file.

The mode of the 80286 and higher processors which allows all of extended memory to be accessed directly. Normally, DOS programs cannot be run in protected mode. However, programs that use a DOS extender can run in protected mode while still maintaining compatibility to DOS. See also Real Mode and Virtual 8086 Mode.

QEMM

Available Screens:

- Overview
- Type
- Accessed
- Analysis
- Memory
- MagnaRAM
- Resources

QEMM Memory (QM)

QEMM Memory gives you an accounting of the memory in your PC (conventional, High RAM, extended, and expanded) before and after QEMM has configured this memory. On some computers there will be a fifth category, either Top Memory (seen most often on Compaqs) or shadow RAM (seen most often on machines using a Chips & Technologies chip set).   This screen appears if QEMM is installed and running at the time you run Manifest.

You use this screen to see a summary of how QEMM has made use of your memory, quickly learn if QEMM can detect the shadow RAM on your PC, find out how much memory is in your PC, and how that memory is being used.

Understanding the Memory Screen

Tips for Maximizing Your Memory

Tips for Maximizing Your Memory

Let QEMM allocate extended memory:

Load device drivers that use extended memory after QEMM.

It is seldom necessary to use the EXTMEM parameter on the QEMM device line. This is only required when loading a driver (usually an old version of a ramdisk) that cannot get memory through an XMS manager such as QEMM. Setting aside raw extended memory through the use of EXTMEM prevents QEMM from allocating it to programs. Such unmanaged memory is usually wasted.

If you believe your PC has more memory than Manifest reports:

Check that your PC's CMOS is properly configured.

Check for defective or loose memory chips.

Run QEMM's Scanmem utility. (See your QEMM documentation for instructions on using Scanmem.)

Why QEMM may not be able to use all of your Shadow RAM:

Shadow RAM which overlaps ROM or adapter RAM cannot be used by QEMM.

This category represents that memory which QEMM controls but which has not been given any function. After QEMM has set up its data, High RAM and other resources, remaining memory is converted to expanded. Since expanded memory is allocated 16K at a time, any memory that is less than 16K, or is fragmented, is left as Unassigned. Some of the QEMM parameters can be altered (RAM, MAPS, ROM, primarily) to increase or decrease the amount of Unassigned memory, to either use it for these new purposes or increase it until it becomes more expanded memory.

Understanding the Memory Screen

Initial:
This column shows the totals of each type of memory when you turn on your computer. The Expanded and High RAM categories always start at OK. The total of this column should equal the amount of memory in your computer.

Unavailable to QEMM:
This column indicates memory which has been made unavailable to QEMM before QEMM gets a chance to use it. Drivers loaded before QEMM which use extended memory will make that memory unavailable to QEMM. Also, machines with Shadow RAM make much of it unavailable for QEMM to use.

Converted by QEMM:
This column shows what QEMM does to your memory. It converts Extended memory (and Shadow RAM or Top Memory if you have any) into Expanded Memory and High RAM. Thus, the column shows some memory being "used" (subtracted from the available) and some being "created" (added to the available). The total of this column indicates how much memory QEMM keeps for its own use and for mapping your ROMs.

Leaving:
This column shows how much memory is left over once QEMM is through initializing itself. These numbers are very important. The remaining conventional memory, usually 640K, indicates how much contiguous memory is accessible for your DOS programs. The remaining extended memory, usually 0K, indicates how much extended memory is left over for other programs to use. Note that if your programs can access memory through the EMS, XMS, DPMI or VCPI interfaces, then they do not need any extended memory left for them by QEMM. The remaining expanded memory is the amount of memory available to programs which support EMS. The remaining High RAM indicates the memory available for loading DOS resources, TSRs, and device drivers into an upper memory area.

How QEMM Uses Memory:
The box at the bottom of the screen details the ways in which QEMM has used memory. The first number is the total amount of memory QEMM is using for itself and for mapping ROMs. This is the same as the total of the third column from the top box. The next numbers indicate the size of QEMM's code and data, the memory reserved for QEMM tasks, the memory reserved for a DMA buffer, the memory reserved for real alternate maps, the memory used for mapping ROMs, and finally a small amount of memory (usually less than 16K) which was unusable for various reasons. These numbers can be changed depending on the setup of your computer and your use of some command line parameters to QEMM386.SYS.   At the bottom of the screen is a number showing how much of your conventional memory area is being used by QEMM. Note that this number is always very small.

Unassigned:
This category represents that memory which QEMM controls but which has not been given any function. After QEMM has set up its data, High RAM and other resources, remaining memory is converted to expanded. Since expanded memory is allocated 16K at a time, any memory that is less than 16K, or is fragmented, is left as Unassigned. Some of the QEMM parameters can be altered (RAM, MAPS, ROM, primarily) to increase or decrease the amount of Unassigned memory, to either use it for these new purposes or increase it until it becomes more expanded memory.

QEMM Overview (QO)

QEMM Overview presents the key elements necessary to determine QEMM's status. This screen appears if QEMM is installed and running at the time you run Manifest.

You will need to refer to this screen if you need technical support from Quarterdeck on QEMM.

In the sections which follow, you may find suggestions about modifying QEMM parameters or command line switches. Details concerning QEMM parameters and how you specify them are contained in the Quarterdeck Expanded Memory Manager user's manual.

Terms Relating to the QEMM Overview

QEMM Resources (QR)

The Windows Resources category gives information on the state of conventional memory and on the operation of the Quarterdeck utilities **Resource Manager** and **FreeMeg**. This screen is only available in Manifest if one or both of these utilities is active.

The Conventional Memory Summary on this screen gives the total memory below the one-megabyte mark that is available in Windows, and the largest contiguous chunk of free memory below one megabyte. Both figures are in bytes. Windows programs often require free memory below one megabyte to load.

The Resource Manager Statistics section gives the percentage of free system resources for both system resource areas: the User heaps, which are used to keep track of the size and location of windows, how and when they respond to keys or mouse clicks, and other window information;and the GDI heap, which is used for data related to drawing graphics devices like icons, window borders, and so on. For each area, Manifest gives both the percentage free with Resource Manager in effect and the percentage that would be free if Resource Manager were not active.

The FreeMeg Statistics section tells you whether or not FreeMeg is active, which method it is using to allocate first-megabyte memory when programs are initializing, and the size in bytes of the smallest block of first-megabyte memory that it will protect.

QEMM Type (QY)

QEMM Type presents you with two views of your first megabyte of memory, as seen by QEMM. This screen appears if QEMM is installed and running at the time you run Manifest.

The first view of your first megabyte of memory, seen through QEMM's eyes, is a map. Each row of the map contains 16 characters representing 64K of address space. Each character (+ * F H h S M X V A R /) represents 4K of memory and has a special meaning, as described in the legend table to its right. Base addresses are indicated in the left column (0n00 at the top through Fn00 at the bottom). To determine a particular 4K area replace the value **n** with the character listed at the top of the screen.

The second view is the same information, displayed in list format. You can display this list by pressing the F3 key.

You use the QEMM Type screen to determine the effect of QEMM's management of memory in your PC.

QEMM's Type Map

QEMM's Type List

QEMM's DOS-UP loads pieces of DOS (including buffers) into upper memory, allowing you to increase the number of DOS buffers you use without using excessive conventional memory.

QEMM's Type List

This screen shows the identical information as the Type Map screen, but lists the regions by Memory Area, indicating the size of each region.

QEMM's Type Map

The following terms are used on QEMM's Type Map:

Mappable (+):
Areas which can be mapped into using EMS function calls. These areas must be 16K in size and aligned on 16K boundaries. Areas of upper memory that are marked as Mappable can be converted to High RAM by adding the RAM switch to the QEMM386.SYS command line.

Rammable (*):
Areas which can be mapped into by QEMM but are too small to be accessed by EMS function calls. You must add the RAM switch to the QEMM386.SYS command line to fill the memory area with expanded memory.

Page Frame (F):
A mappable area, 64K in size, which is the place that almost all EMS programs use for mapping.

High RAM (H):
Areas above conventional memory which have been filled with RAM by QEMM. Any areas above conventional memory marked as Mappable (+) or Rammable (*) can be converted to High RAM (H) by adding the RAM switch to the QEMM386.SYS command line. Once the RAM instruction has been added and the system has been rebooted, High RAM areas may be used by QEMM's LOADHI.COM and LOADHI.SYS programs to load TSRs and device drivers in the upper memory area.

Shrunk High RAM (h):
When Windows is run in 386 Enhanced mode, QEMM "shrinks" (or unmaps the memory from) all unused areas of High RAM, making them available for Windows to load portions of its code high. QEMM does not create these areas of Shrunk High RAM if the line DOS=UMB is present in the user's CONFIG.SYS file.

Stealth Thunk Page (S):
The ROMs on your system are accessed via interrupts. If a piece of software attempts to circumvent the use of interrupts by jumping directly to a memory address (where it believes the ROM to be but which, due to the enabling of Stealth ROM, is no longer valid) your system will fail to operate correctly. You may find that a specific piece of hardware or software fails to work as expected or, even worse, your system may crash.

If QEMM determines, during the Optimize process, that a 4K piece of the ROM is being accessed directly, this 4K "page" is marked as a "Stealth Thunk Page." Stealth Thunk Pages are not mappable areas. No High RAM will be created at these addresses. In addition, QEMM monitors the accesses to such pages in order to allow Stealth's mapping method to work reliably on systems which would otherwise be incompatible with its use.

Mapped ROM (M):
Addresses of Read Only Memory which have been copied to RAM, and then mapped into the original addresses by QEMM's ROM switch. The purpose of mapping ROM is to run ROM code in fast memory.

There are two conditions under which there will be Mapped ROM even when you have not given a ROM switch on the QEMM386.SYS command line:

There is always 4K of Mapped ROM in the system ROM. QEMM automatically uses one area of the system ROM to help it detect reboot requests.

On Compaq computers QEMM will automatically map the video ROM and the system ROM to emulate the speed-up that the Compaq itself imposes but which QEMM overrides. QEMM does this to make optimal use of the COMPAQ reserved memory without slowing down video and other operations.

Excluded (X):
Areas of memory which have been explicitly excluded from use by the exclude switch to QEMM386.SYS. Additionally, QEMM automatically excludes the addresses from 0000 to 0FFF on 80386 systems and from 0000-03FF on 80486's. If one of your adapter cards fails to identify itself properly, QEMM may try to use all or part of the memory addresses used by the adapter as either High RAM or Mappable memory. You must explicitly exclude the conflicting memory areas from QEMM or problems will occur.

Video (V):
Addresses reserved for video display memory.

Adapter RAM (A):
Regions which have RAM mapped into them by adapter cards other than video adapter cards.

ROM (R):
Regions of Read Only Memory which have not been re-mapped by the QEMM386.SYS ROM switch.

Split ROM (/):
Addresses which QEMM386.SYS has detected as having ROM which occupy only a portion of the 4K area. Such areas cannot be re-mapped by the QEMM386.SYS ROM switch.

QEMM MagnaRAM (QM)

The QEMM MagnaRAM category gives a list of information about MagnaRAM, the QEMM utility that compresses data in RAM and on the hard disk so that you have more Windows memory. At the top of this list is the version number of the MagnaRAM VxD file, and information on whether or not MagnaRAM's memory compression is enabled. This screen is only available if MagnaRAM is running.

Also displayed on this screen are:

Compression Ratio
Total Physical
Free Physical
Total Virtual
Free Virtual
Total Jumbo Physical
Free Jumbo Physical
Total Jumbo Virtual
Free Jumbo Virtual
Total Linear
Free Linear

Compressed Physical
Compressed Virtual
Compression Buffer Size
Default Compression Buffer Size
Maximum Compression Buffer Size
Swap-Ins Requested
Swap-Ins Made
Swap-Outs Requested
Swap-Outs Made
Time Saved (seconds)

QEMM Accessed (QC)

The QEMM Accessed screen presents you with two views of the first megabyte of memory, indicating the 4K regions of memory that have and have not been accessed from the time you booted your PC until the time of the display. This screen appears if QEMM is running at the time you use Manifest and Windows is running in Standard mode.

The first view is a map of this information. The second view is a list. The Accessed List screen shows the identical information as the Accessed Map screen, but lists the regions by Memory Area, indicating the size of each region. You alternate between the map and the list by pressing the F3 key.

QEMM Analysis (QN)

QEMM Analysis presents you with two views of your PC's first megabyte of memory, indicating Manifest's assessment of what memory is OK to use, what memory should be excluded, and what should be included. This assessment is determined by whether you have told QEMM to watch what memory was accessed by your programs. This enables you to better optimize the use of your memory.   The QEMM Analysis screen is only available when QEMM is running and Windows is running in Standard mode.

The first view is in map form. The second view of this information is in list format. You can alternate between the map and the list by pressing the F3 key. You use the map to see what memory can be used. However, the list gives you the memory addresses that you can instruct QEMM386.SYS to include or exclude.

IMPORTANT NOTE:
In order for the Analysis screen to be accurate you must precisely follow the steps described in your QEMM manual. If you do not follow the steps, the information presented may not be valid.

The Benefits of Performing an Analysis

Understanding the Analysis Screen

A product which includes an EMS 4.0 driver and utilities to load TSRs, device drivers, and DOS resources into the upper memory area.

Random Access Memory. This is the memory in your machine into which programs and data can be loaded. Its contents are lost when power is turned off. It is distinct from ROM in that the contents of RAM can be changed.

Memory that is fixed in content and cannot be changed. The contents of ROM are not lost when the power is turned off. The programs contained in ROM are responsible for controlling your hardware.

A full description of the EMS state. When real alternate maps are available, information regarding the current state of EMS can be quickly set, improving performance in a multitasking environment.

The processor mode common to all Intel microprocessors and the only mode directly supported by DOS. In real mode, only the first megabyte (1024K) of memory is addressable. See also Protected Mode and Virtual 8086 Mode.

Recovering Lost CMOS Settings

If, for some reason, your PC boots and indicates the need to run your PC's Setup program, there is a good chance that the CMOS has been corrupted. This is most likely to occur after installation of new equipment or when the battery for your PC's CMOS needs to be replaced.

Maverick programs have been known to corrupt CMOS. Having a recent hardcopy of your current CMOS settings can save you a great deal of time if you have to reset your CMOS values.

Reporting Anomalies

Manifest finds the information it reports using a number of recognized techniques. But sometimes it can be inaccurate through no fault of its own. For example:

Manifest may indicate that your keyboard is standard (84 key) when it is really enhanced (101 or 102 key). One cause of this inaccuracy may be a TSR, loaded before Manifest, which does not support the enhanced keyboard. Another cause may be that you are using an enhanced keyboard with a standard keyboard BIOS ROM.

Manifest may report a special high resolution graphics adapter as one of the standard video adapters it most closely resembles (CGA, EGA, VGA, etc.).

Manifest will only report a BIOS that identifies itself in a recognizable way.

Manifest determines the presence of a math coprocessor programmatically and directly. It does not rely on CMOS or the installed hardware byte in the BIOS Data Area.

The designation given to memory areas which have not been given any standard function, but have been reserved for future use. Some of these areas are as yet unused, while others are used for different purposes by different manufacturers.

Components of programs--such as icons, bitmaps, and cursors--which use memory from one of two pools: GDI Resources and User Resources.

A CONFIG.SYS statement which specifies the number and size of the stacks DOS uses for servicing hardware interrupts.

Screen #1 - CPU Identification

You can use the first of the three System Pentium screens as a reference to determine the version of the chip installed in your computer and the speed at which it is running.   The initial screen also gives you a list of Pentium-specific features including:

Floating Point Unit (Math coprocessor.)

Machine Check Exception (Hardware error checking.)

CMPXCHG8B Instruction   (Compare/Exchange 8 Bytes. This new instruction is useful in multiprocessing situations in which two or more Pentiums are installed in one computer.)

A third item, **CPU Counters**, appears on the initial screen, referring to the performance monitoring abilities of the Pentium. The two items (out of approximately forty) that you have instructed the processor to monitor for you are displayed. These tests count the occurrences of certain selectable events internal to the processor.

Screen #2 - CPU Counter Analysis

Pressing the F3 key brings up a second display from which you can analyze or change the items being monitored. Four figures are shown for each item:

Counter Value represents the number of times the event occurred since you began the test.

Clocks Per Event refers to the number of machine cycles that occurred during the event. (As a point of reference, on a 66 Mhz 80486 CPU, 66,000,000 clocks occur per second.)

Events Per Second means the number of times the monitored activity occurred per second.

Counter #1 Events Per Counter #2 Event compares the number of times that event number #1 occurred during the test period versus one occurrence of event #2 (and vice versa.)

From this screen you can press F3 to return to the initial display, restart the tests by pressing Alt-R, or modify either of the two tests by pressing Alt-1 or Alt-2.

Screen #3 - Modify CPU Counter Test

The third screen, Modify CPU Counter Test, displays the current test and the privilege levels being tested. A processor running in real mode is executing at privilege level 0; virtual 86 mode is privilege level 3. A memory manager is always running at privilege level 0.   If you are interested in testing the performance of your program without considering the effects of the memory manager, you should choose privilege level 3. Alternately, to test the memory manager's performance independent of any programs you are running, choose privilege levels 0-2.

From this screen you can disable the current test, change the privilege levels being tested, and choose other events to monitor.

Hardware ports used for communication with serial devices such as modems, accessed through I/O addresses stored in the BIOS data area.

The **Printer Setup** option is not unique to Manifest. Rather, selection of this option brings up the standard Windows Printer Setup dialog box. For information on configuring these options, see your Windows documentation.

A non-standard type of memory, 384K in size, used on some PCs to speed up servicing of BIOS requests. It is typically inaccessible for any other purpose, but QEMM can often convert some of it into expanded memory.

Stealth ROM is a feature of QEMM which relocates a ROM, freeing up the address space the ROM would otherwise use and creating High RAM in its place.   Stealth ROM uses the EMS page frame to access the ROM when necessary.

The number of times that data was actually retrieved from virtual memory.

The number of times that Windows asked for data from virtual memory.

The number of times that data was actually swapped to virtual memory.

The number of times that Windows tried to swap data to virtual memory.

System

Available Screens:

- Overview
- CMOS
- Network
- Pentium - CPU ID

AUTOEXEC.BAT File

Statements in the AUTOEXEC.BAT file are executed by DOS every time you start up your PC. They serve to individualize your PC in a number of ways, including specifying search paths, setting up a meaningful prompt, loading Terminate and Stay Resident programs (TSRs), and executing batch files.

By looking at this file you can review how you have customized your PC and possibly uncover the presence of obsolete utilities, identify TSRs being loaded, and determine overall memory usage.

If a line in your AUTOEXEC.BAT file exceeds the width of Manifest's display that line of text will be continued on the next line following a series of three dots. Pressing F3 puts you in "edit mode" and allows you to scroll to the right and read the entire line.

> Please Note:
>
> If your system is using DOS 3, Manifest displays the AUTOEXEC.BAT on the drive pointed to by your COMSPEC environment variable. In DOS 4 and above, Manifest displays the AUTOEXEC.BAT on the drive from which you booted. Usually there is no difference between these techniques.
>
> Manifest also has a command line switch (/Bn) which tells the program where to find your CONFIG.SYS and AUTOEXEC.BAT files.   Typing "MFT /Ba" tells Manifest to look for the CONFIG.SYS and AUTOEXEC.BAT files in the root directory of drive A, for instance.

System CMOS (SM)

System CMOS consists of multiple screens which show you the contents of your CMOS memory.

The System CMOS display is similar to what is presented in your PC's Setup program, which controls configuration details. The contents of CMOS memory are preserved by battery power when your PC is turned off. When you change your batteries, information here may be lost.

You can use the information displayed here to save your CMOS settings before changing the battery, determine the presence of devices and memory as reported by CMOS memory, and detect discrepancies in installed hardware as maintained in CMOS memory versus other means of hardware determination.

Most of the information in the CMOS has standard meanings. However, some PC manufacturers use the CMOS to maintain information about unusual configurations. You may need to refer to your PC's technical reference manual to learn the exact purpose of the CMOS values.

CMOS List

Recovering Lost CMOS Settings

CONFIG.SYS File

The CONFIG.SYS file customizes your PC when the operating system is initialized. The statements found in this file extend the operating system with software which controls your add-on hardware devices (for example, mouse, network, ramdisk, expanded memory). CONFIG.SYS statements also may increase or decrease DOS resources--i.e. the number of buffers, file control blocks (FCBS), files, and stacks. All these affect memory usage.

By examing this file you can determine overall memory usage, identify add-on hardware and software devices, confirm the presence of required devices, and determine if the DOS resources meet your individual requirements.

If a line in your CONFIG.SYS file exceeds the width of Manifest's display, that line of text will be continued on the next line following a series of three dots. Pressing F3 puts you in "edit mode" and allows you to scroll to the right and read the entire line.

> Please Note:
>
> If your system is using DOS 3, Manifest displays the CONFIG.SYS on the drive pointed to by your COMSPEC environment variable. In DOS 4 and above, Manifest displays the CONFIG.SYS on the drive from which you booted. Usually there is no difference between these techniques.
>
> Manifest also has a command line switch (/Bn), which tells the program where your CONFIG.SYS and AUTOEXEC.BAT files are.   Typing "MFT /Ba" tells Manifest to look for your CONFIG.SYS and AUTOEXEC.BAT files in the root directory of drive A, for instance.

System NETWORK (SN)

If your PC is connected to other computers via a local area network (LAN), the System Network screen will provide detailed information about the networking environment. The information displayed varies depending upon what network software you are running. Manifest recognizes most widely used networking packages.

If you are using Novell's Netware, the System Network screen displays the name and version of the networking software in use, the versions of specific network drivers loaded, and both the total number of users supported by the package and the number of users currently logged in.   By pressing the spacebar you can update this information.

Also provided is your personal login information, including the server you log into by default, your login name and ID, and the date and time you last logged in. If you are connected to a network print server, it is identified.

If you are in Windows and a Winsock program is loaded, the System Network screen will give you the version number of the Winsock, the path to the Winsock DLL file, and the manufacturer’s description of the Winsock, which is included within the DLL file. The System Network screen also shows you the Winsock status area, a character string that Windows sometimes writes into the Winsock to provide useful information for users or technical support. Finally, the screen includes the number of network sockets that the Winsock makes available, and the size of the largest UDP (User Datagram Protocol) datagram that the Winsock can sent or receive.

If you are not loading software to connect to a network, choosing System Network results in the message "No networks found."

Referencing this information can be particularly helpful when you are working by phone with technical support personnel, troubleshooting possible network-related problems.

System OVERVIEW (SO)

System Overview tells you about your PC's hardware configuration. See the topic "Information Displayed" for a detailed list of peripherals identified on this screen. The System Overview screen can be helpful in determining whether your system is able to recognize and utilize newly installed memory or peripherals.

Information Displayed

Reporting Anomalies

What the Numbers Mean

When The Numbers Don't Add Up

System PENTIUM - CPU ID (SI)

In the spring of 1993 Intel added to its family of processors with the release of the long-awaited Pentium. For the first time Intel chose to christen one of its offspring with a word rather than a group of digits in order to set it apart from its older siblings. Not surprisingly, the Pentium's name is not all that is new about this chip. With regard to both speed and features, it far exceeds the capabilities of the 80486 and earlier processors. The Pentium topic appears at the top right corner of the screen when you choose the System category.

NOTE: At the time of its initial release, the Pentium was the only Intel processor featuring **CPU ID**. Manifest identified a processor as a Pentium based on the presence of this feature. Since that time, Intel has shipped 80486 chips capable of providing the same type of identifying information. If Manifest is run on a 486 equipped with CPU ID, the information reported by this feature is displayed.

Screen #1 - CPU Identification

Screen #2 - CPU Counter Analysis

Screen #3 - Modify CPU Counter Test

The ROM which is responsible for initializing and servicing requests to the standard hardware (disk drives, keyboard, standard video adapters) attached to your PC.

A Terminate and Stay Resident program. Such a program initializes itself, hooks up to system resources it wants to monitor, reserves some memory for itself, and then returns to the DOS prompt. Such programs can often be "popped up" over the currently running program by the user. Quarterdeck's LOADHI.COM program can load TSRs into upper memory area.

Refers to the "current state" of QEMM386.SYS. When there is High RAM, ROM mapping, memory back-filling, or video filling, QEMM's mode is ON and may not be turned OFF. Otherwise, QEMM's mode will typically be either AUTO/ON or AUTO/OFF, depending upon whether expanded memory is in use. If QEMM's mode is OFF (without the AUTO), then expanded memory is not accessible until you change the mode using the QEMM.COM program.

Shows you the current status of QEMM386.SYS. QEMM386.SYS may indicate that it is configured with High RAM or Mapped ROM, whether memory is backfilled or video filled, and whether expanded memory is being used.

Terms Relating to Expanded Memory

EMS Driver Name & Version:
The name and version number which identify the expanded memory manager installed. There are memory managers which are not recognizable by Manifest. In these cases the driver name will not be given.

EMS Version:
Indicates the EMS level of support (EMS 3.2, EMS 4.0 and EEMS 3.2). This is the version reported by your EMS software driver.

Page Frame:
The address of the expanded memory page frame. The page frame is a mappable area, 64K in size, which is the place that almost all EMS programs use for mapping.

Number of Mappable Pages:
The number of 16K areas of memory within the first megabyte address space that an expanded memory manager identifies as capable of being mapped. At least four contiguous mappable pages are required to support EMS memory. Many more than four are needed to support multitasking in expanded memory.

Total Expanded Memory:
The total amount of expanded memory in your PC as reported by the expanded memory manager.

Available Expanded Memory:
The amount of expanded memory which is currently not being used. Note that for EMS 4.0 drivers, the part of conventional memory which is mappable (and thus controllable by the EMS manager) is counted twice--once as conventional memory and once as used expanded memory, contained in EMS handle 0. This accounts for the discrepancy between the Total and Available columns even when you are not using EMS.

Total EMS Handles:
The number of EMS handles the expanded memory manager supports. This number affects the memory manager's ability to honor requests for expanded memory. Some memory managers allow this number to be adjusted by a configuration parameter.

Available EMS Handles:
The number of EMS handles that are currently unallocated.

Total Real Alternate Maps:
Shows the number of alternate maps available. A real alternate map is a full description of the EMS state. When real alternate maps are available, information regarding the current state of EMS can be quickly set, improving performance in a multitasking environment.

Terms Relating to the DOS Screens

Kernel:
The memory used by the two system programs (IBMBIO and IBMDOS in PC-DOS; IO and MSDOS in MS-DOS).

Drivers:
The amount of conventional memory used by the device drivers specified by DEVICE=statements in your CONFIG.SYS.

Base Data:
The amount of conventional memory used by DOS resource (FILES, etc.) statements in CONFIG.SYS--including default values set by DOS.

Added Data:
DOS resource memory added by such programs as DOS SHARE and by Quarterdeck's BUFFERS, FILES and DOS-UP programs.

Terms Relating to Expanded Handles

Handle:
The number used to identify each block of EMS allocated to a program. The number will be between 0 and 255

Size:
The amount of memory (expressed in kilobytes) allocated for the handle.

Name:
The name a program has assigned to a particular handle. The name is optional, and is a feature of the EMS 4.0 specification. The name exists strictly for informational purposes. Currently very few programs assign handle names. The EMS program supplied with QEMM allows you to name handles.

Terms Relating to Expanded Pages

Mappable Pages (+):
16K segments of address space in the first megabyte into which an expanded memory manager can map and include unused segments above conventional memory in the area nominally devoted to adapter memory.   If Manifest's Expanded Pages screen identifies any pages as Mappable (+) you could create High RAM with them by using the RAM parameter on QEMM. Anywhere that you have created High RAM will show up as Unmappable (U) here.

Page Frame (F):
A mappable area, 64K in size, which is the place that almost all EMS programs use for mapping.

Unmappable (U):
Those 16K segments of address space into which memory cannot be mapped because the segment is in use by an adapter or has been reserved as High RAM, the memory manager cannot or has been told not to map there, or a TSR or environment such as DESQview has reserved the pages for its own use.

Logical Pages:
The number assigned by the expanded memory manager to each mappable page. The page frame will always be assigned numbers 0 through 3.

Terms Relating to Extended Memory

XMS Version:
The version of the Extended Memory Specification supported by your XMS driver.

Driver Revision:
The revision number of the XMS driver.

High Memory Area Allocated:
Indicates whether the High Memory Area (HMA) is being used or has been allocated. Only one program at a time can use the HMA. Consequently, the first program to access the HMA will be the program which gets to use it.

A20 Enabled:
A20 is a hardware address line which can be enabled or disabled under program control. The A20 line is important in the management of extended memory and the HMA. Programs usually enable the A20 line when using extended memory and the HMA, and disable it otherwise.

Handles Available:
Each extended memory block (EMB) that gets allocated gets a handle associated with it. Thus, there must be an XMS handle available to honor a request for an EMB. Handles Available shows you the number of unused handles. With QEMM providing XMS support, XMS and expanded memory handles are essentially equivalent and are flexibly assigned as needed for either type of memory allocated.

Terms Relating to Windows Memory

The following terms relate to Windows use of memory:

Program Code:
The code of a program's executable files (.EXE's, .DLL's, and drivers.)

Program DGroup:
A program's primary data segment.

Program Data:
Any data segments other than the primary segment.

Module Database and Task Database:
Data structures for keeping track of modules and tasks (components of programs such as .EXE's, .DLL's, and drivers.)

Sentinel and Internal:
Markers that divide the memory in the Global Heap by use. These markers use no memory.

Unknown:
Blocks of memory allocated by programs for purposes other than program code and database functions.

Free:
Available memory within the Global Heap.

Terms Relating to the Memory Status Screen

Common Memory

Common Memory (in Bytes):
The memory used internally by DESQview for housekeeping, storing marked data, and providing memory to DESQview-specific programs. The amount of Common Memory allocated is specified by DESQview Setup.

Total Common Memory:
The amount of common memory available when DESQview starts up.

Total Available Common Memory:
The total amount currently available for use by DESQview.

Largest Available Common Memory:
The largest single block of common memory which is currently available.

Conventional Memory

Conventional Memory (in K):
In Memory Status this refers to the conventional and upper memory area in which DESQview can run programs.

Total Conventional Memory:
The total amount of conventional memory that DESQview manages. This memory may not be contiguous.

Total Conventional Memory Available:
The total amount of conventional memory currently available.

Largest Conventional Memory Available:
The largest contiguous block of conventional memory available.

Expanded Memory

Expanded Memory (in K):
Refers to memory available and managed by an expanded memory manager.

Total Expanded Memory:
The amount of expanded memory managed by the expanded memory manager.

Total Expanded Memory Available:
The total amount of expanded memory usable by DESQview at this moment.

Largest Expanded Memory Available:
The largest block of memory for opening a window using backfilled, mappable memory available at this moment.

Terms Relating to the QEMM Overview

QEMM Status:
Shows you the current status of QEMM386.SYS. QEMM386.SYS may indicate that it is configured with High RAM or Mapped ROM, whether memory is backfilled or video filled, and whether expanded memory is being used.

Mode:
Refers to the "current state" of QEMM386.SYS. When there is High RAM, ROM mapping, memory back-filling, or video filling, QEMM's mode is ON and may not be turned OFF. Otherwise, QEMM's mode will typically be either AUTO/ON or AUTO/OFF, depending upon whether expanded memory is in use. If QEMM's mode is OFF (without the AUTO), then expanded memory is not accessible until you change the mode using the QEMM.COM program.

Page Frame:
Gives the address of the page frame. This entry indicates "None" if there is no page frame.

Stealth Type:
Appears only if you have enabled QEMM's Stealth ROM feature. An "M" indicates you are using the mapping method. An "F" indicates the frame method is in use.

Stealth ROMs:
Lists the starting address and size of all ROMs being relocated by QEMM's Stealth ROM feature.

The Benefits of Performing an Analysis

As you run your favorite programs, QEMM is recording the memory accessed by these programs. Manifest then analyzes the memory accessed and recommends how to best configure your QEMM386.SYS for optimum use of your memory. The end result is that more of your upper memory area may become usable for memory resident programs, device drivers, and DOS resources, leaving more conventional memory for your favorite programs.

An approximation of the amount of additional time that would have been spent accessing virtual memory had MagnaRAM's compression feature not been active.

A non-standard type of extended memory, 384K in size, which is discontiguous with the rest of extended memory. It is addressed at the top of the standard AT 16 megabyte address space.

An approximation of the amount of physical memory that MagnaRAM can effectively create. This figure equals Total Physical multiplied by the Compression Ratio.

An approximation of the amount of virtual memory that MagnaRAM can effectively create. This figure equals Total Virtual multiplied by the Compression Ratio.

The size of Windows’ memory address space, containng both physical and virtual memory.

The total physical memory (installed RAM) on the system.

The total virtual memory (hard disk space that can be used to simulate RAM) on the system.

When the Numbers Don't Add Up

Manifest can assist you in diagnosing memory problems. For example:

If the Total Extended Memory reported is 0 but you think you have some, check that your PC's Setup program is properly configured for extended memory. Next watch the memory count when you power on the machine to see if any bad memory chips are detected during POST (Power On Self-Test) .

If the Total Expanded Memory reported is 0 but you think you have some, check that an expanded memory manager is installed in your CONFIG.SYS file. A device driver or TSR could also be interfering with the expanded memory manager.

Manifest may report that you have less extended memory being used (i.e. Total - Available) than is the case. This does not necessarily indicate a problem, but you may be running a program that does not accurately report its memory usage. You should exercise caution when installing other devices which use extended memory, to insure that no conflicts occur.

If Available Expanded Memory is 0, check that you do not have software using it, and that you are not running Manifest in a DESQview window whose "Maximum Expanded Memory Size" is set to 0K.

Available Memory & QEMM: Even if no expanded memory is being used, the available expanded memory is less than the total extended memory. This difference reflects the amount of memory used by QEMM.

Available Conventional Memory & DESQview: If you are running DESQview, the amount of available memory reflects the amount of memory specified for the window running Manifest. The total conventional memory number indicates the highest memory address accessible to the window, not the total conventional memory in your system.

Regions which have RAM mapped into them by adapter cards other than video adapter cards.

Areas which can be mapped into by QEMM but are too small to be accessed by EMS function calls. You must add the RAM switch to the QEMM386.SYS command line to fill the memory area with expanded memory.

Addresses which QEMM386.SYS has detected as having ROM which occupy only a portion of the 4K area. Such areas cannot be re-mapped by the QEMM386.SYS ROM switch.

UMB (Upper Memory Blocks)

UMB refers to memory mapped into unused areas above 640K and below the one megabyte boundaries. Like the HMA, available UMB memory increases effective DOS memory when used with QEMM.

Understanding the Analysis Screen

OK (O) refers to areas of memory which may or may not be used by a program but which are used or not used in the manner in which QEMM expected them to be.

Exclude (X) refers to areas of memory that have been used by a program, but which QEMM did not expect to be used. You should use the EXCLUDE switch to keep QEMM from using them.

Include (I) refers to areas of upper memory area that have not been used by a program, but have been reserved by QEMM. You can use the INCLUDE switch to allow QEMM to use them.

Understanding the First Meg Map

To better understand the information reported in First Meg Overview, familiarity with some of the other Manifest screens may be helpful. For example:

Memory reported for DOS includes that of the DOS kernel plus the memory added by statements in CONFIG.SYS. (Refer to DOS Overview for details).

Memory reported as Program Area consists of memory used by COMMAND.COM and any TSRs not loaded high. (Refer to First Meg Programs).

Memory reported as Video Text/Graphics or Video ROM shows the memory devoted to and/or used by your video adapter (See System Adapters).

Memory reported as High Ram may have programs loaded there. (See First Meg Programs).

Memory reported as mappable is under control of the expanded memory manager. (See the section on Expanded Memory).

Undetected Extended Memory Usage

Some programs manage and utilize extended memory in ways that cannot be detected by Manifest. This can occur when programs require raw, unmanaged extended memory rather than XMS memory. If Manifest cannot detect such usage, it is likely that other programs will also fail to detect such extended memory usage. To avoid conflicts between programs that use extended memory, it is best not to use programs whose use of extended memory cannot be detected.

Unused Areas

Manifest examines memory and determines how each area has been used. An area that Manifest cannot determine a use for is labeled as unused. Unused areas are seen in systems without an expanded memory manager. In some cases, the expanded memory driver and hardware have not been optimally configured to create areas of High RAM.   QEMM is able to create areas of High RAM out of this unused address space.

Memory that is mapped into unused areas above 640K and below the one megabyte boundaries. Like the HMA, available UMB memory increases effective DOS memory when used with QEMM.

The area between the end of conventional memory (typically 640K) and 1024K reserved for use by system hardware, such as the system ROM, video and other adapters.

Two areas of memory used to keep track of the size and location of windows, how and when they respond to keys or mouse clicks, and other window information.

Using Available Upper Memory

The upper memory area shown by Manifest as Available can be used to load additional TSRs, device drivers, and pieces of DOS or by DESQview to load portions of its code into the upper memory area.

Every program, when loaded, is given an area of memory known as an environment. These areas are shown separately in this map. When Manifest shows small memory fragments of available upper memory area, it is often because a program has released its environment memory. These areas show up as available but, due to their size, are not particularly useful.

Using DOS Resources: Optimally Configuring Your System While Maximizing Your Memory

BUFFERS:
Buffers are blocks of memory that DOS uses to try to minimize disk accesses and therefore speed up file operations. Each buffer is relatively large (about 528 bytes each), and so this is the resource that most significantly affects the memory that DOS uses. Primarily file lookups or directories are affected by the number of buffers. You want to choose a moderate number of buffers. With too few, it will take too long to access files. With too many, you are wasting memory for no good reason. Normally, 20 buffers is a good amount, unless you have some very long directories.

QEMM's DOS-UP:
DOS-UP loads pieces of DOS (including buffers) into upper memory, allowing you to increase the number of DOS buffers you use without using excessive conventional memory.

FCBS:
FCBS=n,m specifies that DOS should reserve n areas of memory, about 53 bytes each, to keep track of file control blocks (FCBs) that are open. m specifies the number of these FCBs that DOS should not close automatically in order to open more than will fit in the table. Relatively few programs use FCBs anymore, and so you usually do not need to specify any extra FCBs in CONFIG.SYS. FCB structures in DOS are usually only significant when DOS's SHARE is loaded.

FILES:
Files are relatively small structures (about 53 bytes) with which DOS keeps track of files that are open. 20 files are usually sufficient, but, in a multitasking environment such as Windows, you may need 40 or more.

LASTDRIVE:
This CONFIG.SYS parameter specifies the highest drive letter that DOS recognizes. The primary reason to specify a letter higher than the highest letter actually in use is to reserve drive letters for the DOS SUBST program. If you do not use SUBST and none of your drives (or logical partitions) uses a letter above E, you do not need to set a LASTDRIVE.

STACKS:
STACKS=n,m specifies that DOS should reserve n areas of memory, each m bytes long, to use as stacks for servicing hardware interrupts. By default, DOS reserves about 3.2K of conventional memory for these stacks, which is a significant amount of memory. DESQview already deals with the problem of choosing a stack for hardware interrupts, and so if you run predominantly in DESQview, you can specify STACKS=0,0 in DOS 3.3 and higher to make DOS not reserve any room for these stacks.

During installation of Microsoft Windows 3.1 the line STACKS=9,256 is added to your CONFIG.SYS file. Some MS Windows users are able to replace this line with STACKS=0,0 and run their MS Windows applications without a problem.   Other users find that using STACKS=0,0 causes overflow errors in MS Windows. If you want to save a few K of memory, try using 0,0. If this causes you to experience overflow errors, change the line back to 9,256.

Using Manifest's Online Help

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

Secondary windows: A small graphic image of a printer and clipboard appears in the upper left corner of most 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 clipboard.

Using the Expanded Benchmark Screen

You use these benchmark tests to:

Compare two or more expanded memory managers.

Measure interrupt latency to determine how quickly interrupts can be handled.

Using the Expanded Pages Screen to Solve a Problem

If you have a problem with a program or device which does not function properly at specific times, try adding an exclude statement for the expanded memory manager or changing the addresses that a specific device uses. Use this screen to determine which areas are controlled by the memory manager and compare these addresses with those used by the devices in question.

It is important that no device be installed in a mappable area. If it is, the device may fail to function, programs which use expanded memory may not work, and/or the computer might lock up.

Using the First Meg BIOS Data Screens

Experienced PC users may find these screens helpful as a database reference of the BIOS data structure and for quick access to port address values or the installed hardware and memory size values.

Experimenters and students may find these screens helpful in confirming the effect of DOS and BIOS calls.

Using the First Meg Programs Screen

You use this screen to:

See what memory your programs are using. Often you will see a separate area for the program's environment and a separate area allocated for the program's data.

Identify programs that are currently loaded in conventional memory but which could be loaded into the upper memory area.

Using the Interrupt List

You can use the Interrupt List to:

Learn what program is using an interrupt. (For example, if you are having keyboard problems, you may want to look at which program is using INT 16).

Check whether the interrupt is in use by another program.

Note: Due to the use of the interrupt vector area of memory as a program stack during power up, few programs make use of any of the later interrupt numbers. Consequently, Manifest does not attempt to identify programs using vectors C0 to FF.

Using the Interrupt Owners Screen

You use the Interrupt Owners screen:

To determine what programs service interrupts and which interrupts they service.

To determine which interrupts are not being serviced (i.e., the interrupt address is zero or the address indicates the ROM BIOS default interrupt handler).

With Manifest in TSR mode, to help you profile a program's interrupt usage.

Virtual Control Program Interface. A programming specification created by Quarterdeck Corporation and Phar Lap Software to allow DOS extended programs to run when loaded after a virtual 8086 mode expanded memory manager such as QEMM. Without VCPI, DOS extenders would be incompatible with these expanded memory managers.

The hardware and ROM software that control the video display characteristics. Examples are CGA, EGA, VGA and Hercules.

A mode of the 80386 and higher processor chip which has the following two characteristics:

Real mode DOS programs can run without modification.

Extended memory can be mapped into the first megabyte of address space.

Virtual 8086 mode allows QEMM to support EMS 4.0 without special hardware and allows DESQview to run graphics programs in the background. See also Protected Mode and Real Mode.

VxDs, used in Windows 386 Enhanced mode and in Windows 95, are drivers that Windows and other programs use to perform various system-level functions. A VxD can be internal (part of a larger Windows file) or external (a separate file).

What the Numbers Mean

Total Conventional, Extended, Expanded and Upper Memory often add up to more memory than you have in your PC. When you convert extended memory into expanded memory by using an EMS manager (such as QEMM or DOS' EMM386) memory is counted twice--once under Expanded Memory (pooled) and once under Extended Memory (pooled.)

Some machines have 384K of memory set up in a non-standard way. That is, it is neither conventional, nor expanded, nor extended memory. Sometimes this 384K of memory is set up as Shadow RAM (such as on machines which use certain Chips & Technologies chip sets.) On other systems it is set up as what we call "Top Memory" (most notably, on some Compaq models.) In either case, it is typically used by your PC to increase performance of your system and video ROMs. Since this memory is not, by default, accessible in the same ways as the three standard memory types, Manifest will usually not include it in its memory listings.

What the Tests Mean

The following is a list of the benchmark tests performed by Manifest and a brief explanation of each. Shorter timings indicate expanded memory manager software with higher performance capabilities. The benchmark timings shown on the Manifest screen are the results of running these tests using QEMM on a 33 Mhz i486 PC.

Timer Interrupt Latency:
An indicator of how long the expanded memory manager disables interrupts at any given time. An acceptable test time is below 500, although the smaller the better.

Map Single Page:
The time required for the expanded memory manager to map a single page of expanded memory. This number has great significance, as it is the function which most EMS applications use repeatedly to access their data.

Real Alternate Map Set:
The time it takes to switch to an alternate map. This number has great significance for multitasking environments, such as DESQview. It indicates the minimum amount of time it takes to map in a whole new program. The maximum in this category added to the maximum timer interrupt latency indicates the "worst case" for running communications programs in DESQview. If the numbers add up to less than 500, you should be able to run 9600 baud communications in a DESQview window. When performing this test, Manifest should be run from DOS, not from within DESQview.

Save All - nn Pages:
The time required for the expanded memory manager to save all EMS pages. The number of mappable pages at the time of the test is shown in place of nn. When comparing expanded memory managers, you should configure them so that **nn** is the same in order to get a fair comparison for this test and the next two tests.

Map Multiple - nn Pages:
The time required for the expanded memory manager to map all available EMS pages. The number available, which is the pages that do not conflict with Manifest's memory, is shown in place of nn.

Restore All - nn Pages:
The time required for the expanded memory manager to restore all EMS Pages. The number of mappable pages at the time of the test is shown in place of nn. The sum of the times of the Save All and Restore All tests indicates how long it would take DESQview to map in a whole new program if there were no real alternate maps available.

Save Page Frame:
The time required to save the state of the EMS Page Frame.

Restore Page Frame:
The time required to restore the state of the EMS Page Frame. The sum of the previous two tests indicates how long it takes a TSR or device driver using expanded memory to preserve the current page frame mapping around its own EMS use. Thus, it affects the performance of expanded memory disk caches and similar programs.

Open Handle - 4 Pages:

The time required to open an EMS Handle which has 4 EMS pages. Neither this timing nor the next three timings are very significant, since they are not made very frequently. However, the calls might possibly generate large timer interrupt latencies if the expanded memory manager is not well-written.

Close Handle - 4 Pages:
The time required to close an EMS Handle that has 4 EMS pages.

Open Handle - 31 Pages:
The time required to open an EMS Handle that has 31 EMS pages. This test will only succeed if there is sufficient expanded memory.

Close Handle - 31 Pages:
The time required to close an EMS Handle that has 31 EMS pages.

You can create a report consisting of a single Manifest category, all categories, or a select group of categories by selecting the appropriate item under What to Print.

The **Where to Print** options dictate the manner in which the report is generated: as a file, fax, or hard copy from a printer.

Why Manifest Isn't Listed

Manifest excludes itself from its reports. Note that nowhere is there a reference to Manifest in the program map. The report's purpose is to give you information about your programs as you normally use them. Manifest will report itself only if it is left resident in memory.

A **Windows** report contains the following categories:

System / Overview
System / Config
System / Autoexec
First Meg / Overview
DOS / Overview
Windows / Overview
Windows / SYSTEM.INI
Windows / WIN.INI

MS Windows DevCaps (WE)

DevCaps (or "DEVice CAPabilitieS") refers to the capabilities of the Windows printer and display drivers currently in use on your system. You can toggle back and forth between the Display Capabilities screen and the Printer Capabilities screen by pressing F3.

The information on these screens may not interest the casual user. If you are a programmer, however, or an experienced user, particularly a user of high-end video display equipment or printers, these screens can help you fully utilize the power of that equipment or solve problems involving its use.

MS Windows GLOBAL HEAP (WG)

The Windows Global Heap category gives you information on Windows' global heap, the pool of all the memory on your system that Windows manages for its own use and the use of programs. The global heap grows as necessary, based on requests for memory. Manifest gives you four ways of looking at the global heap: to see the Summary screen, press the **F3** key or click on the **F3-Summary** button; for the Details Screen, press **F4** or click on the **F4-Details** button; for the Fixed Map screen, press **F5** or click on the **F5-Fixed Map** button; and for the Free Map screen, press **F6** or click on the **F6-Free Map** button.

The Summary screen under the Windows Global Heap category provides a breakdown of memory within the heap, listing the amount of memory currently used in ten categories. You can also view a Details screen which lists each block of memory within the global heap. The Details list tells you the type of data using the memory, the full 32-bit address of the block of memory, the amount of memory in bytes, and the program that "owns" (or has allocated) the memory.

The Fixed Map screen lists all the areas that are fixed at particular memory addresses, alternating with areas that are moveable to different places in the heap. Similarly, the Free Map screen breaks the global heap down into address ranges that are free for use and ranges that are already allocated. Both these screens list the type of memory, the full 32-bit address of the block of memory, and the amount of memory in bytes.

Terms Relating to Windows Memory

MS Windows OVERVIEW (WO)

Manifest makes it easy for you to get basic information about how your copy of Microsoft Windows is configured. The MS Windows Overview display shows the **version** of MS Windows found on your hard drive and the **directory** where your MS Windows files are located.   If Manifest is run from a directory other than the one containing these files, it will search your PATH for a copy of MS Windows.

Also displayed on this screen are:

Computer type (MS Windows may recognize your specific make of computer or identify it simply as an MS-DOS system.)

Type of display (EGA, VGA, etc.)

Type of mouse (Brand and whether serial or bus mouse.)

Keyboard (AT, enhanced, or vendor-specific.)

Keyboard layout (US or other country, dvorak, etc.)

Language (and variation.)

Codepage (Alternate character set required for some foreign languages.)

Network (Software package and version.)

MS Windows PROGRAMS (WM)

The **Windows Programs** category tells you how memory is being used by your programs.   This category displays information in two ways--as a list of **Tasks** and as a list of **Modules**.   To toggle between the two, press the **F3** key or click on **F3 for Modules/F3 for Tasks**.

The **Tasks** screen contains the following entries for each task and lists the amount of memory used by each.

Program Code:
The code of the program's executable files (.EXE's, .DLL's, and drivers.)

Program Data:
Data segments used by the program.

Other Data:
Additional memory allocated by the program for a wide variety of uses.

Program Resources:
Cursors, bitmaps, icons, dialog boxes, etc.

Total Windows Memory:
Total memory used by the task.

The **Modules** screen provides a detailed listing of all modules (including .EXE's, DLL's, drivers, and fonts) currently being used by tasks.   Each entry contains the name of the module, its handle (or identifier), the number of "users" of the module, and the executable path (drive, directory, and file name).

MS Windows RESOURCES (WR)

The Windows Resources category gives you information on Windows resources, which are graphics elements and window management data that Windows stores in special areas in memory. Two of these areas, the **User heaps**, are used to keep track of the size and location of windows, how and when they respond to keys or mouse clicks, and other window information. A third area, the **GDI heap**, is used for data related to drawing graphics devices like icons, window borders, and so on.

The Resources Summary screen lists twelve categories of resources. For instance, icons, bitmaps, and cursors are resources. The Summary screen tells you how many of each resource type are in use and the total memory used by each category, in bytes. The Details screen lists each resource, the handle used by Windows to refer to the resource, the size of the resource in bytes, and the program which "owns" (or has allocated) the resource. To toggle between the Summary and Details screens, press the **F3** key or click on **F3-Details/F3-Summary**.

Terms Relating to Windows Memory

SYSTEM.INI File

One of two initialization files used by MS Windows, the SYSTEM.INI file is created in your MS Windows directory during installation. The entries contained in this file tell MS Windows about your specific hardware configuration. Typically the file will be edited automatically when you tell MS Windows, via Setup, about changes to your system. You may, on occasion, want to edit the file manually in order to fine tune your MS Windows environment.

By using the Manifest text editor, you can view this file as well as make changes in the text. You may, for example, want to experiment with an EMMExclude statement in the 386 Enhanced section during the course of troubleshooting an MS Windows-related problem. You should exercise caution when altering this file, however. Always make a backup copy of the file before making any changes. MS Windows 3.1 includes a help file called SYSINI.WRI (in MS Windows Write format) that you may want to refer to before altering your SYSTEM.INI.

MS Windows VxDs (WV)

The Windows VxDs category lists all of the VxDs (Virtual Device Drivers) currently active in Windows. VxDs, used in Windows 3's 386 enhanced mode and in Windows 95, are drivers that Windows and other programs use to perform various system-level functions. A VxD can be internal (part of a larger Windows file) or external (a separate file).

The Windows VxDs screen gives the name of all active VxDs, the earliest version of Windows that the VxD claims to support, and the VxD's ID number, a unique number assigned by Microsoft to registered VxDs. Also provided is the 32-bit address of the DDB (Device Data Block), a data structure that gives information about the VxD; information about whether the VxD supports a protected mode program interface (PM-API) and a virtual-8086 mode program interface (-V86); and the number of dynalink services (a kind of program interface call) that the VxD offers.

WIN.INI File

The second of two initialization files used by MS Windows, the WIN.INI is also created in your MS Windows directory during installation. The entries in this file customize your MS Windows environment according to your preferences.   Typically the file will be edited automatically when you make changes to MS Windows via the Control Panel. It is seldom necessary to edit the file.

By using the Manifest text editor, you can view this file as well as make changes in the text. As with the SYSTEM.INI file, you should exercise caution when altering this file. Always make a backup copy of the file before making any changes. MS Windows 3.1 includes a help file called WININI.WRI (in MS Windows Write format) that you may want to refer to before altering your WIN.INI.

The Extended Memory Specification created to allow DOS programs to utilize High RAM and extended memory, including the High Memory Area, in a cooperative way.

XMS and QEMM

If you are using QEMM, XMS support is provided, as well as the conversion of extended memory into expanded memory. In this case, available memory shown on Manifest's Expanded Memory screen is essentially the equivalent to the available memory shown on Manifest's Extended Memory screen.