

## ***Tips for Configuring Windows 3.1***

This chapter presents several tips for configuring Microsoft Windows 3.1, both for gaining optimal performance and for setting up Program Manager and File Manager.

*Related information*

- *Windows User's Guide*: Chapter 3, "Program Manager"; Chapter 4, "File Manager"; Chapter 14, "Optimizing Windows"
- *Windows Resource Kit*: Chapter 4, "The Windows Initialization Files," and Chapter 5, "Windows 3.1 and Memory Management"
- *Glossary terms*: upper memory area, upper memory blocks, Expanded Memory Specification, Extended Memory Specification, paging, virtual machines

*Contents of this chapter*

---

Tips for Optimizing Performance.....	258
Checklist for Optimizing Performance.....	258
Trading Speed for Hard Disk Space.....	262
Optimizing Use of the UMBS.....	263
Adjusting EMM386 for Optimal Performance.....	266
Tips for Using Third-Party Memory Managers.....	269
Optimizing with SYSTEM.INI.....	271
Tips for Configuring the Windows Shell.....	272
Customizing Program Manager.....	272
Customizing File Manager.....	277

---

## **Tips for Optimizing Performance**

---

This section presents a summary of tips for achieving the best performance for Windows 3.1, plus specific information about using EMM386.EXE.

Before making changes to your system configuration, create a backup boot disk, as described in “Creating a ‘Clean Boot’ for Troubleshooting” in Chapter 13, “Troubleshooting Windows 3.1.” For information that will help you solve specific performance problems with your system, see “Troubleshooting Windows Configuration” in Chapter 13.

## **Checklist for Optimizing Performance**

Here is a summary of items related to optimizing your hardware and software configuration for both Windows applications and non-Windows applications running under Windows 3.1.

For information about the parameters discussed in this chapter for HIMEM.SYS, EMM386.EXE, SMARTDRV.EXE, and RAMDRIVE.SYS, see Chapter 14, “Optimizing Windows,” in the *Windows User’s Guide*.

**Tip** To see a list of switches that can be used with Microsoft system drivers that have .EXE filename extensions, at the command prompt, type the driver name and the */?* switch, then press *ENTER*. For example, type **emm386 /?** at the command prompt to see a list of the switches you can use with EMM386.EXE.

## **Configuring Hardware for Optimal Performance**

- Install additional extended memory.
- Install the largest hard disk that you can afford, as discussed in “Trading Speed for Hard Disk Space” later in this chapter.
- Set the optimal hard disk interleave for your system, using third-party software such as Steve Gibson’s SpinRite. Hard disks are often formatted with the wrong interleave at the

dealer or factory. Some utilities can correct your interleave without reformatting the hard disk.

- If your system has a memory expansion board and the board can be configured as either expanded or extended memory, configure all of this memory as extended memory, then use EMM386.EXE to emulate expanded memory only as needed by non-Windows applications that require such memory to run. In such a case, place the expanded memory driver first in CONFIG.SYS, followed by the lines that load HIMEM and EMM386.

For example:

```
device=c:\emmbord.sys  
device=c:\dos\himem.sys  
device=c:\dos\emm386.exe 1024 x=C000-C7FF
```

For information about configuring the memory in your add-on board, see its documentation. Some expanded memory boards may be incompatible with EMM386.EXE.

### **Configuring MS-DOS for Optimal Performance**

- Upgrade to MS-DOS version 5.0 and load MS-DOS into the high memory area (HMA) by setting **dos=high** or **dos=high,umb** in CONFIG.SYS. For more information, see “Loading MS-DOS in the HMA” later in this chapter.
- Make sure the most recent versions of HIMEM.SYS, EMM386.EXE, RAMDRIVE.SYS, and SMARTDRV.EXE are stored in the directories specified in the command lines that load these drivers in your CONFIG.SYS and AUTOEXEC.BAT files. (Placing any of these drivers in the root directory will not improve performance.)
- Load HIMEM.SYS in CONFIG.SYS and make sure the line that loads HIMEM comes before commands that load any other applications or drivers that use extended memory.
- Load SMARTDRV.EXE in AUTOEXEC.BAT and allocate the largest amount of memory possible, because the SMARTDrive disk caching driver can produce the biggest Windows 3.1 performance improvement.
- Set **files=30** in CONFIG.SYS unless you have a software application that requires a higher setting.
- Set **buffers=10** in CONFIG.SYS if you use SMARTDRV.EXE. Using a higher number of buffers with SMARTDrive will decrease efficiency.

If you are not using SMARTDrive, set **buffers=20** in CONFIG.SYS. More buffers may improve disk access times, but will use more conventional memory.

- Install RAMDrive to use as the TEMP environment variable if you have a diskless workstation or if you have limited hard disk space and ample extended memory (more than 6 MB).

Performance improves for Print Manager and for applications that frequently create .TMP files, because access is faster for a RAM disk than for a physical disk.

- Load EMM386.EXE if you are running non-Windows applications that require expanded memory or if you want to allocate UMBS for loading memory-resident TSRs and drivers. For more information, see “Optimizing Use of the UMBS” later in this chapter.
- Load only the necessary TSRs, drivers, or software programs. Wherever possible, load TSRs and device drivers into the UMBS.
- If the environment space is set by the **shell** command in CONFIG.SYS, you can specify a smaller environment.
- Remove any commands for mouse drivers in AUTOEXEC.BAT and CONFIG.SYS if you only use the mouse in Windows, and don’t want mouse support while running non-Windows applications in 386 enhanced mode.

### ***Maintaining Optimal Hard Disk Performance***

- Delete unnecessary application and system files, including .BAK and .TMP files. Make sure that Windows is not running when you delete .TMP files. For a description of the minimum Windows files, see Appendix C, “The Windows 3.1 Disks and Files.”
- Run **chkdsk** frequently to find lost clusters. If lost chains or clusters are detected, then run **chkdsk /f** and delete \*.CHK files.
- Use a utility program to compact your hard disk regularly. A fragmented hard disk greatly impacts Windows performance, especially if SMARTDrive is installed or if you are using a temporary swap file. Make sure Windows is not running whenever you run the disk compacting utility.

### ***Configuring Windows for Optimal Performance***

- Use a color or a pattern for the desktop background instead of wallpaper if you need to free memory for running applications.
- For standard mode Windows, set the application swap file to

the fastest hard drive by setting the **SwapDisk=** entry in the **[NonWindowsApp]** section of SYSTEM.INI.

- For 386 enhanced mode, create a permanent swap file on your fastest hard drive by choosing the 386 Enhanced icon in Control panel and filling in the options in the Virtual Memory dialog box.

The only reason not to use a permanent swap file is if hard disk space is at a premium. For details about permanent swap files, see “Virtual Memory and 386 Enhanced Mode” in Chapter 5, “Windows 3.1 and Memory Management.”

- If you do not create a permanent swap file for 386 enhanced mode, set your temporary swap file to the fastest hard drive by setting a value for **PagingDrive=** or **PagingFile=** in the **[386enh]** section of SYSTEM.INI.
- Run Setup and choose the lowest-resolution display driver that will meet your needs. For VGA-compatible displays that can use an extended mode driver, such as Video 7 or 8514, using the standard VGA driver will ensure faster display performance (but lower resolution and less color support).

### ***Optimizing for Non-Windows Applications***

- Allocate the expanded and extended memory required by non-Windows applications by using the PIF Editor to create a custom PIF, as described in Chapter 7, “Setting Up Non-Windows Applications.”
- Turn off all entries in the Monitor Ports option in the PIF Editor Advanced dialog box. These options are not required for most non-Windows applications.
- Set **FileSysChange=no** in the **[386enh]** section of SYSTEM.INI to avoid sending update messages to File Manager from non-Windows applications.
- Set **NoEMMDriver=yes** in the **[386enh]** section of SYSTEM.INI if your non-Windows applications do not need expanded memory. This ensures that the most memory possible is available in virtual machines.

### Optimizing for TSRs

- Include only the TSRs in AUTOEXEC.BAT that are required for the network, or that must be present for other purposes.
- If a TSR is required only with a single non-Windows application, start the TSR in the virtual machine after launching that application. For more information about creating batch files for running TSRs from Windows, see Chapter 7, “Non-Windows Applications,” in the *Windows User’s Guide*.
- For all TSRs that you want to be available in Windows, add their .EXE names to the file WINSTART.BAT, which is executed whenever Windows runs in 386 enhanced mode. These TSRs are not available in the virtual machines (VMS), leaving more room in conventional memory for the non-Windows applications. For more information about WINSTART.BAT, see Chapter 14, “Optimizing Windows,” in the *Windows User’s Guide*.

The following table shows where a TSR is available (and using memory), depending on where it is launched.

<i>Where TSR is launched</i>	<i>TSR visible in Windows?</i>	<i>TSR visible in virtual machines?</i>
From MS-DOS all virtual machines		Yes      Yes,
From WINSTART.BAT	Yes	No
In a single virtual machine machine	No	Only that virtual

### Optimizing for Windows on a Network

- Set **lastdrive** in CONFIG.SYS to **J** or **K** if you can forgo access to some of the network disk drives.
- For a diskless workstation, use a RAM drive as the TEMP variable.
- Do not use a network drive for a temporary swap file, and avoid using a net drive for a permanent swap file if at all possible.

- If a user has problems running Windows or other applications with network software loaded in the upper memory area or in the high memory area, try disabling the network's HMA option. Consult your network documentation for more information.
- Some memory managers, such as EMM386, 386MAX, and QEMM, can load networks and other device drivers in the upper memory area. If the network driver is loaded in the upper memory area and a user has problems running Windows, try loading the network in conventional memory.

### ***Trading Speed for Hard Disk Space***

To optimize Windows performance, you must make trade-offs between speed and hard disk space, depending on your system's hardware capabilities. In either case, make sure that as much free memory as possible is available.

#### ***To ensure maximum free memory in your system:***

---

- Remove unnecessary TSRs from AUTOEXEC.BAT and CONFIG.SYS.
- Load MS-DOS high and use all possible elements of MS-DOS 5.0 memory management, such as using EMM386.EXE to manage UMBS.

#### ***To ensure optimal speed performance:***

---

- Free as much conventional and extended memory as possible.
- Install SMARTDrive for disk caching.
- Set the TEMP variable to the fastest hard disk or to a RAM disk.
- Create a permanent swap file to use in 386 enhanced mode.
- Set **SwapDisk=** in the **[NonWindowsApp]** section of SYSTEM.INI to the hard disk with the most free space.
- Use third-party software to organize the location of files on your hard disk so that the Windows files are located near the file allocation table (FAT).



**Related information**

- *Microsoft MS-DOS User's Guide and Reference*: "Running Programs in the Upper Memory Area" in Chapter 12

**To ensure maximum hard disk space:**

---

- Create a RAM drive to use as the `TEMP` variable.
- Delete unnecessary files after exiting Windows.
- Decrease the size of your permanent swap file, or use a temporary swap file in 386 enhanced mode. Or prevent swapping by typing **none** in the New Settings Type edit box of the Virtual Memory dialog box, or by setting **paging=no** in the **[386enh]** section of `SYSTEM.INI`.
- Follow the steps in Appendix C, "The Windows 3.1 Disks and Files," to create a minimum Windows "footprint."

## Optimizing Use of the UMBS

- *Windows Resource Kit*: "Expanded Memory: A Technical Discussion" in Chapter 5, "Windows 3.1 and Memory Management"; "Troubleshooting EMS Memory Problems" in Chapter 13, "Troubleshooting Windows 3.1"

On 80386 and higher PCs that use MS-DOS 5.0, you can free conventional memory by loading memory-resident programs such as device drivers, TSRs, and network software into the upper memory blocks (UMBs). You can optimize the way your system uses the UMBs by following the guidelines in this section:

- Load MS-DOS 5.0 in the HMA with the **umb** switch.
- Load `EMM386.EXE` with the **noems** or the **ram** switch.
- Find the best order for loading drivers and programs into UMBs.
- Adjust the command line for `EMM386.EXE` in `CONFIG.SYS` to set aside fewer UMBs for expanded memory, so that there is more room for loading other programs into the UMBs. For details about how to do this, see "Adjusting `EMM386` for Optimal Performance" later in this chapter.

## Loading MS-DOS in the HMA

You must load MS-DOS 5.0 in the HMA and use the **umb** switch to

link the UMBS to conventional memory if you want to load programs into the UMBS.

---

**To load MS-DOS 5.0 in the HMA:**

---

- Add these lines to CONFIG.SYS, in this order:

```
device=<path>\himem.sys  
device=<path>\emm386.exe ram | noems  
dos=high,umb
```

The parts of MS-DOS 5.0 that are loaded into the HMA include most of the kernel and COMMAND.COM, code pages, and the MS-DOS buffers (from the **buffers=** command in CONFIG.SYS). To remain compatible with software created for earlier versions of MS-DOS, about 19K of MS-DOS 5.0 code stays in conventional memory.

You can only load one program at a time into the HMA, so if you load MS-DOS 5.0 into the HMA, then you cannot load a network redirector, for example, into it as well. If you must choose what to load into the HMA, always choose to load MS-DOS 5.0, because it uses the HMA efficiently and leaves little behind in conventional memory.

---

**Note** Use the **device** command to load HIMEM.SYS and EMM386.EXE, since loading them with **devicehigh** has no effect.

---

### **Loading EMM386.EXE to Use the UMBS**

After loading HIMEM.SYS in CONFIG.SYS, load EMM386.EXE with the **ram** switch or the **noems** switch, plus any other parameters as required for your system. The **ram** switch provides both UMBS and a 64K EMS page frame; **noems** provides UMBS only and no page frame. If none of your applications require EMS, then use **noems**, which will provide an extra 64K in the UMBS.

If you load EMM386 with the **noems** switch and find that some programs fail to run or display out-of-memory messages, it's because they require expanded memory to run. If this happens, replace the **noems** switch with the **ram** switch to make expanded memory available again.

### Setting the Load Order for Drivers in UMBS

You must experiment to find the best order for loading drivers and TSRs in the UMBS.

*To find the best order for loading drivers into UMBS:*

1. In CONFIG.SYS, load the largest device drivers with the **devicehigh** command. Use the **device** command to start any drivers that are smaller than the largest memory-resident program loaded from AUTOEXEC.BAT.
2. In AUTOEXEC.BAT, start the largest memory-resident programs with the **loadhigh** command.
3. After rebooting, use the MS-DOS **mem /c** command to see what is loaded into the UMBS and what is in conventional memory. If everything didn't fit into the UMBS, change the order in which the programs are loaded in CONFIG.SYS and AUTOEXEC.BAT.

With **devicehigh** and **loadhigh**, each device and TSR is loaded into the largest available slot, so the order in which programs are loaded can affect how many programs can fit into the UMB. Also, how and when a program allocates memory determines what size UMB it needs. Some programs need a UMB as large as the program size while running, and some need a UMB as large as the program file size.

The following table lists some Microsoft device drivers and memory-resident programs that can run in the UMBS. This table shows whether the program's file size or size while running determines the UMB size the program needs.

<i>Program</i>	<i>UMB size needed</i>	<i>Program</i>	<i>UMB size needed</i>
ANSI.SYS	File size	MOUSE.COM	
DISPLAY.SYS	File size	MOUSE.SYS	File
DOSKEY.COM	Size while running	PRINTER.SYS	
DRIVER.SYS	Size while running	RAMDRIVE.SYS	
	File size		

Figure 6.1

Example of UMBs in  
the upper memory area

For a VGA or V7VGA  
display adapter,  
a network card  
at D000-D7FF,  
and an EMS 3.2  
page frame  
at E000-EFFF

EGA.SYS	Size while running	
	File size	SETVER.EXE
	File size	
FASTOPEN.EXE	Size while running	SHARE.EXE Size
while running		
MIRROR.COM	Size while running	

**Note**

You cannot use **loadhigh** or **devicehigh** to load HIMEM.SYS, EMM386.EXE, or SMARTDRV.EXE.

If your mouse comes with both MOUSE.SYS and MOUSE.COM files, you can load the mouse driver at any point after HIMEM.SYS and EMM386.EXE. Use **devicehigh** in CONFIG.SYS to load MOUSE.SYS, or use **loadhigh** in AUTOEXEC.BAT to load MOUSE.COM.

For example, suppose you have an 80386 computer with 2 MB of extended memory that you want to configure as follows:

- 512K of expanded memory for a non-Windows CAD application
- As much free conventional memory as possible

Your CONFIG.SYS file should contain the following commands:

```
device=c:\dos\himem.sys
device=c:\dos\emm386.exe 512 ram
dos=high,umb
devicehigh=c:\dos\mouse.sys
```

For more information about determining memory requirements

**Flowchart 1.5**  
**Clean Boot Disk**

1. Create a backup boot disk, as described in “Creating a ‘Clean Boot’ for Troubleshooting” in Chapter 13, “Troubleshooting Windows 3.1.”

and load order for programs, see the *Microsoft MS-DOS Getting Started* booklet and see page 519 of the *Microsoft MS-DOS User’s Guide and Reference*.

## Adjusting EMM386 for Optimal Performance

The EMM386.EXE driver takes a conservative approach when scanning the UMBS for free space, because it must remain compatible with software that was created for earlier versions of MS-DOS. For example, EMM386.EXE does not map over any unused ROM. If you have space below C800 or above DFFF that you know is unused and you want it used for the UMBS, you must explicitly instruct EMM386.EXE to include it by using the **i=** switch to specify a range of segment addresses.

For more information about EMM386, see pages 287–290 and 605–609 of the *Microsoft MS-DOS User’s Guide and Reference*; see also Chapter 14, “Optimizing Windows,” in the *Windows User’s Guide*.

### Adjusting EMM386 to Provide More UMBS

You can adjust EMM386 to allocate space in the UMBS, especially if you start EMM386 with the **ram** switch to make expanded memory available to programs. First, carry out Procedure 1 below; then, if you still cannot load programs and device drivers into UMBS, carry out Procedure 2.

**Procedure 1**

To allocate additional space for use as UMBS:

2. Add the **i** switch in front of the **ram** or the **noems** switch in the **device=emm386.exe** line of CONFIG.SYS.

The **i** switch specifies a range of addresses to be allocated in the UMBS. The value you specify for the **i** switch depends on your computer and monitor type.

If your computer is not an IBM PS/2 and does not have a monochrome monitor, add the **i=e000-ffff** and the **i=b000-b7ff** switches before the **ram** or **noems** switch. For example:

```
device=c:\dos\emm386.exe i=e000-ffff i=b000-b7ff ram
```

If your computer is not a PS/2 and has a monochrome monitor, add the **i=e000-ffff** switch before the **ram** or **noems** switch. For example:

```
device=c:\dos\emm386.exe i=e000-ffff ram
```

If your computer is an IBM PS/2 and does not have a monochrome monitor, add the **i=b000-b7ff** switch before the **ram** or **noems** switch. For example:

```
device=c:\dos\emm386.exe i=b000-b7ff ram
```

If your computer is an IBM PS/2 with a monochrome monitor, see Procedure 2.

3. If your computer fails when you restart it after this change, the memory range you specified for EMM386 is probably being used by hardware or video display drivers. In this case, use your backup boot disk to restart your computer. Then remove the **i** switches you added for EMM386 in CONFIG.SYS.

If you start EMM386 with the **ram** switch, follow Procedure 2.

4. After your computer starts, use the MS-DOS **mem /c** command to check whether the programs were loaded into UMBS successfully.

If you still cannot load programs into UMBS, your system might be using much of the UMBS for hardware code. Some computers use the UMBS for ROM shadowing, which can improve your computer's speed. (In ROM shadowing, the computer copies hardware code from the slower read-only memory into faster RAM.) To find out more about how your computer uses the UMBS, see the documentation that came with your system.

If you cannot load programs into UMBS, and you are starting EMM386 with the **ram** switch, carry out Procedure 2.

---

**Note** Use Procedure 2 only if your non-Windows EMS programs use expanded memory according to LIM EMS version 4.0. Programs that follow LIM EMS 3.2 need the full 64K page frame and are unable to use expanded memory if you follow this procedure.

---

**Procedure 2**

**To set aside fewer UMBS for expanded memory:**

---

1. Type **emm386** at the command prompt.

The information that appears shows current memory-management activities. Find the line that reads *Page frame segment*, and write down the hexadecimal address that appears on that line. For example:

```
Page frame segment . . . E000 H
```

2. Add the **p0** switch in front of the **ram** switch in the **device=emm386.exe** line of CONFIG.SYS.

The **p0** switch should specify the address you found in Step 1. For example, if the page-frame segment was E000h, the command would be:

```
device=c:\dos\emm386.exe p0=e000 ram
```

3. If your computer fails when you restart it, insert your backup boot disk in drive A and restart your computer. Then remove the **p0** switch you added to CONFIG.SYS in Step 2. Restart your computer and repeat Steps 1 and 2, checking the page-frame segment address carefully.
4. After your computer starts, use the MS-DOS **mem /c** command to check whether the programs were loaded into UMBS successfully.

If your programs still do not run in UMBS, contact Microsoft Product Support Services for assistance.

### **Freeing Conventional Memory in 386 Enhanced Mode**

When you run non-Windows applications in 386 enhanced mode, Windows allocates translation buffers to pass data between the applications and drivers executing in real mode, as described in Chapter 5, "Windows 3.1 and Memory Management." On a stand-alone PC, Windows allocates two translation buffers for each

virtual machine (8K total); on a networked PC, Windows allocates six buffers (24K total).

If it finds space, Windows places the translation buffers in the UMBS; otherwise, they are placed in conventional memory, where they take up space in every virtual machine.

To find out if you still have UMBS free while running EMM386, at the command prompt type **mem /p** to see what is free in the hexadecimal addresses between A000 and FFFF. Then add an **x=** switch to the EMM386 line in CONFIG.SYS to exclude the unused UMB ranges.

If you don't have at least 28K of free UMB, you can still partially optimize the use of this area. Because each translation buffer is 4K, each 4K chunk you exclude will save conventional memory. For example, this line will let Windows load one buffer into the UMB, freeing 4K of conventional memory:

```
device=c:\dos\emm386.exe noems x=df00-dfff
```

Excluding memory, however, can change how MS-DOS 5.0 loads programs into UMBS, so you may need to experiment to get the optimal configuration.

### ***Tips for Using Third-Party Memory Managers***

For complete compatibility, Windows 3.1 and MS-DOS 5.0 both provide HIMEM.SYS and EMM386.EXE as memory management drivers. The overriding objective Microsoft has for these drivers is to avoid hanging any computer during or after memory configuration. Optimization, while important, is secondary.

You can also use a third-party memory manager to manage UMBS in an 80386 computer. Such memory managers have different objectives, and there are pros and cons to using them in place of the built-in Microsoft memory managers.

To allocate UMBS through a third-party memory manager, you need both a UMB provider and a UMB loader:

- A UMB provider is the memory manager that makes UMBS



available. For MS-DOS 5.0, the UMB provider is the combination of EMM386.EXE plus HIMEM.SYS. In QEMM386 and 386MAX, for example, the UMB providers are QEMM386.SYS and 386MAX.SYS, respectively.

- A UMB loader loads a program into the proper UMB. The UMB loader in MS-DOS 5.0 is in the MS-DOS kernel, accessed through the **devicehigh** and **loadhigh** commands. The equivalents in 386MAX are 386LOAD.SYS and 386LOAD.COM; in QEMM386, they are LOADHI.SYS and LOADHI.COM.

The arguments for using third-party memory managers include:

- Most third-party memory managers have a wider range of options for loading into the UMBS. For example, you can specify a particular UMB for loading a program.
- Third-party memory managers typically require 3K to 5K of conventional memory, while HIMEM.SYS with EMM386.EXE may take up to 10K. However, some memory managers also require overhead in the UMBS, but neither HIMEM.SYS nor EMM386.EXE require any.

The arguments against using third-party memory managers include:

- Third-party memory managers can be more difficult to configure properly than EMM386.EXE, and they are often geared toward more sophisticated users.
- Third-party memory managers are supplemental to HIMEM.SYS and EMM386.EXE, so they require additional costs and resources to support. Unless you absolutely require 100 percent optimization, a third-party memory manager may be superfluous.

If you decide to use a third-party memory manager, follow these guidelines:

- Don't use either HIMEM.SYS or EMM386.EXE.
- Use **dos=high** to load MS-DOS 5.0 into the HMA.
- Don't use the **dos=umb**, **devicehigh**, or **loadhigh** commands.
- Don't use the third-party commands that relocate the MS-DOS buffers into the UMBS. MS-DOS 5.0 automatically puts its buffers into the HMA, saving the UMBS for other programs.
- Use both the third-party UMB provider (such as 386MAX.SYS) and the third-party UMB loader (such as 386LOAD.SYS and 386LOAD.COM).
- Use the third-party commands to relocate MS-DOS file handles and drive space (from the **files** and **lastdrive** commands in

your CONFIG.SYS) into the UMBS.

You can also use third-party memory drivers as alternatives to HIMEM.SYS, SMARTDRV.EXE, and RAMDRIVE.SYS. However, the memory drivers provided with Windows 3.1 have been specifically designed and tested to provide maximum compatibility between MS-DOS 5.0 and Windows 3.1.

## Optimizing with SYSTEM.INI

Many entries in SYSTEM.INI can be set to optimize performance for your system. Some entries are for specific hardware configurations. For details about any of these entries, see Chapter 4, “The Windows Initialization Files.”

The **[standard]** section of SYSTEM.INI contains these entries for configuring Windows performance in standard mode: **FasterModeSwitch=** (for specific Zenith and Olivetti systems), **Int28Filter=**, and **Stacks=**. Also, the **SwapDisk=** entry in the **[NonWindowsApp]** section of SYSTEM.INI affects performance for running non-Windows applications in standard mode.

The **[386enh]** section contains entries for optimizing performance in 386 enhanced mode. These entries specify values for virtual memory and paging:

<b>LRUentries=</b>	<b>MinUnlockMem=</b>	<b>PagingFile=</b>
<b>MaxBPs=</b>	<b>MinUserDiskSpace=</b>	<b>PerVMFILES=</b>
<b>MaxPagingFileSize=</b>	<b>PageOverCommit=</b>	<b>PSPIncrement=</b>
<b>MaxPhysPage=</b>	<b>Paging=</b>	<b>SysVMV86Locked=</b>
<b>MinTimeSlice=</b>	<b>PagingDrive=</b>	<b>XlatBufferSize=</b>

These entries in **[386enh]** specify values for controlling UMBS and EMS management for VMS:

<b>AlEMSLocked=</b>	<b>IgnoreInstalledEMM=</b>	<b>ReserveVideoROM=</b>
<b>EMMExclude=</b>	<b>LocalLoadHigh=</b>	<b>SysVMEMSLocked=</b>
<b>EMMInclude=</b>	<b>NoEMMDriver=</b>	<b>SysVMEMSRequired=</b>
<b>EMMPageFrame=</b>	<b>ReservedHighArea=</b>	<b>UseableHighArea=</b>
<b>EMMSize=</b>	<b>ReservePageFrame=</b>	<b>XMSUMBInitCalls=</b>

These entries in **[386enh]** specify values for XMS management and VMS:

<b>AllXMSLocked=</b>	<b>SysVMEMSLimit=</b>	<b>SysVMXMSRequired=</b>
<b>MapPhysAddress=</b>	<b>SysVMXMSLimit=</b>	

These entries in **[386enh]** specify values for DMA buffers:

<b>DMABufferIn1MB=</b>	<b>EISADMA=</b>	<b>MaxDMAPGAddress=</b>
<b>DMABufferSize=</b>	<b>HardDiskDMABuffer=</b>	<b>MCADMA=</b>

The Program Manager is installed automatically by Setup as the main Windows shell. Use the menu commands in Program Manager to control the appearance and behavior of items in the window. Use the applications and accessories in the Program Manager groups to control the configuration and appearance of all elements of Windows, including the drivers, memory management, and peripherals such as printers and multimedia devices.

Several entries in various sections of SYSTEM.INI are specifically related to optimizing Windows performance on a network.

<i>Section</i>	<i>Network-related entries</i>	
[Boot]	CachedFileHandles=	Network.drv=
[Standard]	Int28Filter=	NetHeapSize=
[NonWindowsApps]	NetAsynchSwitching=	
[386enh]	FileSysChange=	Network=
	InDOSPolling=	PSPIncrement=
	Int28Critical=	ReflectDOSInt2A=
	NetAsynchFallback=	TimerCriticalSection=
	NetAsynchTimeout=	TokenRingSearch=
	NetDMASize=	UniqueDOSPSP=
	NetHeapSize=	

## ***Tips for Configuring the Windows Shell***

This section describes how you can customize Program Manager and File Manager, the principal shells for managing Windows services and files. For information about the system registration database that supports many of these services, see “Managing the Registration Database” in Chapter 11, “Compound Documents in Windows 3.1.”

### ***Customizing Program Manager***

You might choose to use another shell (by changing the **shell=** entry in the **[boot]** section of SYSTEM.INI). However, Setup for most Windows applications looks for Program Manager and cannot complete installation unless Program Manager is the current Windows shell. This is because Program Manager facilitates many services, including the addition of new entries in the system registration database.

## W

In Windows 3.1, Program Manager makes fewer demands on system memory resources than in earlier Windows versions. This means that you can have more groups defined without using up memory, and that the window can refresh itself faster. Other changes in Program Manager in Windows 3.1 include: You can also specify an entry for **TaskMan.exe=** in the **[boot]** section of SYSTEM.INI if you want to use a third-party task manager.

For details about the Program Manager initialization file, see the entries for PROGMAN.INI in Chapter 4, “The Windows Initialization Files.” For details about working with any items in Program Manager, see Chapter 3, “Program Manager,” in the *Windows User’s Guide*.

---

### *New in Program Manager in Windows 3.1*

---

- Context-sensitive help is available for all items.
- Menu settings and window positions are saved automatically, unless you choose otherwise.
- When you use File Run to launch an application, Windows searches for executable programs using every listing in the **programs=** entry of WIN.INI. If you type a filename in the Command Line box, Windows uses the registration database and the **[extensions]** setting in WIN.INI to find the associated program.

### **Defining Program Manager Groups**

The Main, Applications, Accessories, Games, and Startup groups are created in Program Manager automatically during Setup. You can move icons from one group to another by dragging, and you can add to any group by dragging the icon for a file or application from File Manager to the Program Manager group. To create a new group in Program Manager, choose New from the File menu and complete the dialog box. Choose Properties from the File menu to define the appearance and behavior of any group item.

Figure 6.2

New Program Object dialog box

**W** Drag icons into the Startup group for any application or file that you want to run each time you run Windows. (This means you no longer have to edit the **run=** and **load=** entries in WIN.INI to define the startup applications; these entries are independent from the Startup group.)

Information about the groups defined in Program Manager appears in the **[groups]** section of the PROGMAN.INI file.

*Tip* If you want to rename the Startup group to customize your desktop, add a **startup=other group name** entry to the **[settings]** section of PROGMAN.INI.

### **Defining Properties for Program Groups**

The Properties command on the Program Manager File menu provides options for controlling the appearance of icons and the behaviors related to launching applications from Program Manager. Make a group window active, or select a group item, then choose Properties to define:

- The icon caption.
- The program and (if you want) the document filename to be launched when you double-click the icon.
- W** · The working directory for the application. By default, this is the directory that contains the application.
- W** · The shortcut key you want to reserve for switching to the application when it's running.
- Whether you want the application to appear as an icon or in a window whenever you run it.
- The icon you want to represent the program item.

Any changes you make take effect immediately. The **[group]** section in PROGMAN.INI and the related .GRP file are updated to save

the changes.

For information about customizing the group items, and controlling other aspects of how items appear on the desktop, see Chapter 5, “Control Panel,” in the *Windows User’s Guide*.

### **Setting Up Shared Program Manager Groups**

W

Program Manager groups can be shared over the network by making them available to users as read-only files. A user cannot change the contents of a shared group. The Move and Delete commands on the File menu in Program Manager are not available for shared groups. A user can choose the Properties command to review file properties of items in a shared group, but cannot change them.



To share a Program Manager group, you must mark the group file (\*.GRP) as read-only and place it in a shared network directory. At the workstation, you or a user can create a group and use the group file from the shared network directory.

**To set up a shared Program Manager group:**

---

1. Create a group file by using Program Manager to create a group and add program items to it.
2. Copy the group file (.GRP) to a shared network directory and mark it as read-only.
3. From each workstation, use Program Manager to set up the group. In the Group File box in the Program Group Properties dialog box, specify the location and name of the group file in the shared network directory.

Figure 6.3

Program Group Properties dialog box

If you don't want to use Program Manager to set up the group, you can modify the PROGMAN.INI file for the workstation. In the **[groups]** section of the PROGMAN.INI file, add a unique variable plus the name and path of the group file. For example:

```
group8=z:\winshare\programs.grp
```

If you want a shared Program Manager group to be set up when users install Windows for their workstations, you must create a PROGMAN.INI file that points to the shared group, and then have that PROGMAN.INI file copied to the user's WINDOWS directory during Setup.

For more information, see "Modifying .INF Files for Custom Installations" in Chapter 2, "The Windows Setup Information Files."

### **Configuring Program Manager for a Network Workgroup**

W

If you are the system administrator for a network, you might want to restrict user actions for a workgroup that uses Windows on a network. You can add a **[restrictions]** sections to PROGMAN.INI to:

- Remove the Run command from Program Manager, by setting **NoRun=1**.
  - Remove the File menu from Program Manager, by setting **NoFileMenu=1**.
  - Not allow workgroup users to exit Windows, by setting **NoClose=1**.
  - Not allow workgroup users to save settings, by setting **NoSaveSettings=1**.
  - Not allow workgroup users to create, delete, or rename groups or group-items, or change properties for items or groups, by setting values for **EditLevel=**.

The Windows 3.1 File Manager provides the efficiency in Windows 3.1 File Manager and directories and to take advantage of drag-and-drop features for creating compound documents.

For example, if you want users to have a standard work environment where they cannot run programs from Windows other than ones you have set up in Program Manager groups, and you do not want any of their window or icon positions saved when Windows is restarted, specify the following entries in PROGRAM.MAN.INI:

```
[restrictions]
NoRun=1
NoSaveSettings=1
EditLevel=4
```

If you do not want users to be able to exit Windows, specify the following entry in PROGRAM.MAN.INI:

```
[restrictions]
NoClose=1
```

If you don't want users to run any programs from Windows other than ones that you set up in Program Manager groups, specify the following entries in PROGRAM.MAN.INI:

```
[restrictions]
NoRun=1
EditLevel=4
```

---

**Note** Some applications allow users to start other applications, for example File Manager. So if you do not want users to start other applications, make sure you do not include them in Program Manager groups.

---

For more information, see the description of the **[restrictions]** section in "PROGRAM.MAN.INI: The Program Manager Initialization File" in Chapter 4, "The Windows Initialization Files."

## Customizing File Manager

For details about working with any items in File Manager, see Chapter 4, "File Manager," in the *Windows User's Guide*.

---

### New in File Manager for Windows 3.1

---

- The window updating, drive searching, and background

processing capabilities have been enhanced. Also, you can cancel time-consuming processes (such as complete searches of a CD-ROM drive).

- You can view more than one directory tree at a time, and browse the file system in a single window. Also, more information appears, such as disk space usage, network share names, and more.
- Context-sensitive help is available for all items.
- Several inconsistent behaviors have been changed, so that File Manager provides the capabilities you expect and need for file management. For example, menu settings and window positions are saved automatically, unless you choose otherwise.
- You can open and print files by dragging icons in File Manager, and drag icons to embed and link one file in another, as described in Chapter 13, “Integrating Your Windows Applications,” in the *Windows User’s Guide*.
- For networks, the previous network connections are made automatically when you run Windows, and related menu items appear automatically to assist you with network activities related to file management.
- You can use long file names on network servers, if the driver for your network supports this feature.

### **Defining File Manager Actions**

To use File Manager services or to define the behavior and appearance of information in the window:

- Choose File and Disk menu commands for file, disk, and drive management activities.
- Choose Tree and View menu commands to arrange the appearance of directories and files in the File Manager windows.
- Choose Confirmation from the Options menu to specify whether you want warning messages to appear when you delete files or perform other actions.
- Choose other Options menu commands to define the font, status bar, and other behaviors for running and quitting File Manager.

W

### **Associating Extensions with File Types**

If the filename extension for a file is associated with a file type defined in the registration database, you can:

- Open the file by double-clicking its icon in File Manager.
- Print the file by dragging its icon to the Print Manager.
- Place the file as a group item in any Program Manager group.

W

W

- Embed the file as a packaged object by dragging its icon to an application icon or to another document icon.

Many filename extensions are predefined when you install Windows. You can use the File Manager to add or change the association of a filename extension with any file type that is defined in the registration database.

---

**To associate a filename extension with a file type:**

---

1. In the File Manager window, select the icon for the file you want to associate with a file type.
2. Choose Associate from the File menu in File Manager.
3. If the extension of the selected file doesn't appear, type the filename extension you want in the Files With Extension box.
4. Choose the application in the Associate With list that you want associated with the extension. If the application name isn't in the list, type an executable program filename in the box, or click Browse and select a file from the list. Then click the OK button.

Figure 6.4

Associate dialog box

When you change the associated filename extension in the File Manager Associate dialog box, Windows updates the related entry in the registration database and updates the list in the **[extensions]** section of WIN.INI.

---

**Note** If you want to use drag-and-drop in File Manager to create packages, the server application must have been written for Windows 3.1. Otherwise, you must use the Object Packager to create the package, as described in Chapter 13, "Integrating Your Windows Applications," in the *Windows User's Guide*.

---

---

**More About Memory Management**

---

Gookin, Dan. *The Microsoft Guide to Managing Memory with DOS 5*. Microsoft Press, 1991.

"Running Programs in the Upper Memory Area" in Chapter 12 of the *Microsoft MS-DOS User's Guide and Reference*.

