More Information About Microsoft Windows Resource Kit for the Microsoft Windows Operating System Version 3.1

This document contains important information that is not included in the Microsoft Windows Resource Kit manual.

Using Write to View This Document

If you enlarge the Write Window to its maximum size, this document will be easier to read. To do so, click the Maximize button in the upper-right corner of the window. Or open the Control menu in the upper-left corner of the Write window (press ALT+SPACEBAR), and then choose the Maximize command.

To move through the document, press PAGE UP or PAGE DOWN or click the arrows at the top and bottom of the scroll bar along the right side of the Write window.

To print the document, choose the Print command from the File menu.

For Help on using Write, press F1.

To read other online documents, choose the Open command from the File menu.

Contents

This document contains additional information on the following topics:

1.0 Adding third-party drivers to a shared copy of Windows

If your users require Windows drivers which are not supplied
in the Windows 3.1 package you will probably want to add these
to the shared copy of Windows they will be using. For example,
you might have display adapters which are distributed with
their own Windows display driver. In this case you should
have received from the hardware vendor a disk containing both
the drivers and a file named OEMSETUP.INF. Windows Setup
uses this file to understand how to identify and install the
accompanying Windows components.

There are two methods for integrating these files into a
shared copy of Windows:

Method 1: Renaming OEMSETUP.INF

The simplest method is to simply copy all the files provided
by the hardware vendor into the shared Windows directory.
The only other step that is required is to rename the hardware-
vendor's OEMSETUP.INF file to a unique name, so that it won't
conflict with the similar files provided by other vendors.
You should give it a name of the format OEMnnnnn.INF, where
nnnnn is any number; an example would be OEM1.INF.

When users run Setup (or SETUP /N) from the shared Windows
directory, Setup will automatically read in all OEMnnnnn.INF
files and display the hardware options they describe along with
those listed in the Windows SETUP.INF file.

Method 2: Editing SETUP.INF

If you are already modifying the SETUP.INF file in the shared
Windows directory (for example, removing the hardware options
that don't apply to your configuration) then you may want to
integrate the options described in the OEMSETUP.INF file directly
into your modified SETUP.INF. You would then copy the other

vendor-supplied files, such as drivers or fonts, into the
shared Windows directory, but not their OEMSETUP.INF file.

When doing any modifications to SETUP.INF be sure to test out
the changes thoroughly before making the file available to your
end users.

2.0 Background on Stacks = 9,256

With the introduction MS-DOS 3.2, a new feature was added that
allowed the user to control the amount of stack space that device
drivers got when processing HW interrupts. This is the STACKS=
command in CONFIG.SYS. The line has the following form:

    STACKS = <number of stacks> , <size of stacks>

The default value for MS-DOS 3.2 and 5.0 is STACKS = 9,128 or 9
stacks that are 128 bytes in size.

Prior to this, HW interrupts were processed using what ever stack
was in use at the time the interrupt occurred. This was a problem
since some applications did not have much stack space available
at certain phases of their execution which would lead to stack
overflow. When this occurred, usually program code or data got
overwritten and often random behavior would be seen.

When a HW interrupt occurs, the next available stack is switched to
and control passes to the device's interrupt handler. The size of
the stack is important here since the interrupt handler can actually
be a chain of TSRs or drivers that have `hooked' this interrupt or
the interrupt handler can call system functions that are in turn
hooked. Each instance of hooking will consume some amount of stack
space. The minimum is 6 bytes and can often be quite large in poorly
written software. The net result is that as more TSRs, drivers and
system utilities are added to a system, the demand for stack space
increases.

Stack overflow can occur when the chain-of-calls uses more stack space
than was allocated. The symptoms of this are quite random. Often
yielding bizarre behavior or outright hangs. Its often the case
that invalid code is executed.   If you have a memory manager
installed, you will see an error or exception #6. This leads many to
believe that the memory manager is broken when, in fact, it is simply
reporting the problem. These are difficult problems to track down
since they surface randomly.

Our testing has indicated that in many simple cases, more than 100 bytes
of stack space are used. This leaves very little margin of safety.
Thus, stacks = 9,256 is much more reliable than the default. We have
found a number of cases of instability that were cured by adding the
stacks = 9,256 line.

A number of people and products recommend using a STACKS=0,0 line in
your CONFIG.SYS. This will turn off the STACK feature and revert to
the behavior of MS-DOS prior to version 3.2. While this will often cure the
symptoms, it is risky since it forces the interrupt handler to use the current stack.

Thus, the currently running application dictates whether there is
enough stack space for the system to use. You can protect yourself
from this by using the STACKS = 9,256 line in your CONFIG.SYS.

3.0 Setting Up Windows over an asynchronous connection.

When installing Windows over an asynchronous connection, the connection
may be broken during the file copying procedure. This occurs when Windows
is trying to find the mouse. There are three possible solutions:

1. Use a network adapter connection if one exists.
2. Copy the expanded files from the source to a local harddisk using the
    asynchronous connection. Then, run setup from the local drive.
3. Use SETUP /I to ignore hardware detection. Then, when prompted
    choose NO MOUSE.

4.0 RAMDrive Suggestions

There are two main reasons Windows creates temporary files, printing and
insufficent memory available to do a task. By creating a RAMDrive you
may be limiting the memory available to the applications, causing them to
create temporary files (unless you have plenty of memory for what you are
doing and the RamDrive).

Some applications will use temporary files for certain functions no
matter how much memory is available.

If you are printing from a graphics program such as PageMaker, Corel
Draw, Micrograx Designer, or Publisher, a temp file of 3MB or more can
easily be created when printing. In this situation a RAMDrive of 4MB or
more would be needed.

In Standard mode, when running a non-Windows application, Windows will need to
swap programs out of memory to the hard disk. Windows does this in the
directory pointed to by the TEMP environment variable unless the SWAPDISK =
line is used in the SYSTEM.INI. If the TEMP variable points to a RAMDrive,
and considering other temporary files may be there, a RAMDrive of 5MB or
more may be required.

Remember, creating a RAMDrive will take memory away from Windows.
Unless you have more than 6 MB of memory, it is suggested you do not run
Windows with a RAMDrive.

5.0 Documentation Notes:

5.1 Page 239 - Extended Memory and 386 Enhanced Mode

The third paragraph states:

...by creating virtual machines up to 640K in size, or the size defined by the **CommandEnvSize=** entry in the **[NonWindowsApp]** section of SYSTEM.INI.

This should read:

...by creating virtual machines up to 640K in size, **less** the size defined by the **CommandEnvSize=** entry in the **[NonWindowsApp]** section of SYSTEM.INI.

5.2 Page 240 - Expanded Memory and 386 Enhanced Mode

Windows 3.1 will inherit the settings of EMM386.EXE. Thus, if you use the EMM386 **x =** parameter to exclude ranges in the UMA, Windows 3.1 will also exlcude those ranges.

5.3 Page 241 - Placing Translation Buffers in the Upper Memory Area

The second paragraph states:

...To compound the problem, the translation buffers can be allocated either in the upper memory area or in conventional memory, but never half-and-half.

It should be noted this applies to each specific translation buffer, not the group. Each individual buffer must be contingous, and thus, either all in the UMA or all in conventional memory. However, as a group, some of the buffers may reside in the UMA and some may reside in conventional memory.

5.4 Page 246 - Virtual Memory and 386 Enhanced Mode

The flag **LocalLoadHigh=** listed as controling paging is incorrect. This flag controls how Windows 3.1 uses the upper memory area. For more information, see **LocalLoadHigh=** on page 204.

5.5 Page 247 - Swapping Pages to a Network Drive

The sentence that reads:

...If you must page to a network drive, use a permanent swap file.

Should read:

... If you must page to a network drive, use a **temporary** swap file.

This change also applies to **Page 262 - Optimizing for Windows on a Network.**

On page 247 it also states:

Do not set the value for PagingDrive= or PagingFile= in SYSTEM.INI to

a Novell network drive, because Novell networks are not compatible
with MS-Net Redirector.

It should be noted that this is only true for a permanent swap file. For a
temporary swap file these values work correctly.

5.6 Page 258 - Checklist for Optimizing Performance

It should be noted, that the fourth bullet point applies to situations where you have
non-Windows applications that require EMS memory that you are loading
prior to loading Windows. If the non-Windows application is loaded through
Windows, you don't need to load EMM386.EXE.

The following text is only valid for running in 386 enhanced mode.:

> 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.

When running in 386 enhanced mode, in most cases the expanded memory driver
may not be necessary. Please refer to documentation for your specific expanded
memory card for details.

When running in Standard mode, you must have an expanded memory card and
driver installed to provide EMS memory to your non-Windows applications.

5.7 Page 259 - Configuring MS-DOS for Optimal Performance

Windows will optimize the SMARTDRV.EXE settings based on your configuration.
In most cases, this will be the optimal setting for your system. Should you wish to
alter these settings, you should refer to **SMARTDrive 4.0: A Technical
Discussion** starting on page 252.

5.8 Page 263 - Optimizing Use of the UMBs

Looking at the example at the bottom of the page:

device=<path>\himem.sys   This line loads the XMS driver
device=<path>\emm386.exe ram | noems   Allows for access to the UMA
dos=high,umb   This line loads MS-DOS into the

HMA and links the UMB to
conventional memory.

Although in this example MS-DOS is loaded high, it is not required to load
MS-DOS into the HMA in order to access the UMA.

5.9 Page 269 - Freeing Conventional Memory in 386 Enhanced Mode

As noted above, Windows 3.1 will inherit the excludes from EMM386. In order to free
more UMBs for Windows 3.1 use, you may need to experiment in the order of loading
as outlined on **Page 264 - Setting the Load Order for Drivers in UMBs**.

5.10 Page 444 - Windows Driver Library Files

Some of the sound drivers listed on page 444 did not adequately complete compatibility
testing prior to the release of the Windows Resource Kit. These drivers can be obtained
by following the instructions on page 443. The following sound drivers are included on
the Windows Resource Kit disk:

Source	Device driver from the Windows Driver Library
Creative Labs	SoundBlaster Pro
Media Vision	Pro Audio Spectrum
Artisoft	Sounding Board

To install these drivers, follow the instructions on page 444.

5.11 Page 472- BANFORUM / CLMFORUM

Forum ID	Section vendor name	Products	Phone
ADOBE	5 Adobe	Illustrator	(415) 962-6076
	5 Adobe	Streamline	(415) 962-6076
	7 Adobe	Type Library/PC	(415) 962-6076
	17 Adobe	ATM for Windows	(415) 962-6076
	17 Adobe	Type Align/Windows	(415) 962-6076
ACAD	11 Auto Desk	AutoCAD for Windows	(415) 332-2344
BANFORUM	**15 Banyan**	**Windows Support**	**(508) 898-1000**
BLYTH	2 Blyth	Omnis 5/Windows	(415) 571-0222
. . .			
CENTRAL	9 Central Point	Windows Apps	(503) 690-8090
CLMFORUM	**9 Computer Lang Mag**	**Windows Technician**	**(415) 905-2200**
DBADVISOR	4 DB Advisor	Windows & Graphics	(619) 483-6400

5.12 Page 475 - MULTIVEN

PC Magazine Forum ID	Section vendor name	Products	Phone
. . .			
MULTIMEDIA	6 Multimedia Corp.	Windows Multimedia	(408) 737-7575
	12 Multimedia Corp.	MPC - Multimedia PC	(408) 737-7575
MULTIVEN	**2 Meridian Data**	**Windows CD ROM Kit**	**(408) 438-3100**
	4 Video Logic	**DVA-4000 Card**	**(617) 494-0530**
	8 Authorware	Authorware Pro Win	(415) 595-3101
	9 Ntergaid	Hyper Writer 3/Win	(203) 368-0632

	10 HSC Software	HSC Quick Show	(310) 392-8441
NORUTL	9 Peter Norton Group	Norton Anti-Virus/W	(408) 253-9600

. . .

5.13 Page 476 - PCVENE / VIRUS FORU

PC Magazine Forum ID	*Section vendor name*	*Products*	*Phone* *(continued)*
. . .			
	15 Bloc Publishing	Wind-Ease	(305) 445-0903
PCVENE	**4 Farallon**	**Apple Talk PC**	**(510) 596-9100**
SBSALDFOR	2 Aldus	Pagemaker for Windows	(206) 628-2320
. . .			
VENTURA	4 Ventura Software	Ventura Pub/Windows	(800) 822-8221
	14 Ventura Software	Form Base	(800) 822-8221
	14 Ventura Software	Ventura DB Publish	(800) 822-8221
VIRUS FORU	**2 McAfee Associates**	**Windows Scan Virus Scan**	**(408) 988-3832**
WINADV	1 New Uploads		
	2 Microsoft	Accessories	
	3 Microsoft	Batch Files	
	4 Microsoft	Comm Ports	

. . .

5.14 Page 516 - Windows Fails to Run in 386 Enhanced Mode
Page 518 - Windows Runs Slowly in 386 Enhanced Mode

In both instances, a reference is made to verifying free disk space. This only applies
if you are running Windows 3.1 with a temporary swap file. In addition, 386 enhanced
mode will still run, however it will be limited in its virtual memory capabilities.

6.0 Utilities:

6.1 Graphics Viewer, TopDesk Tour and/or Network Assistant

These three utilities are written in Microsoft Visual Basic version 1.0. If you were
involved in the beta program of Micorosoft Visual Basic, you may have a pre-
release version of VBRUN100.DLL. If you encounter errors while loading these
three applications, delete VBRUN100.DLL from your system. Then, reinstall the
WRK utilities into the same directory as the previous WRK installation.

6.2 System Resource Monitor

Some applications may dynamically change system resources while performing certain
functions. If you are running System Resource Monitor Always On Top, in a few cases,
this may interfere with normal operation of the application. Examples of this are Norton
Desktop for Windows Screen Saver and MicroGraphix Draw. If you encounter
interference, simply close System Resource Monitor or clear the Always On Top option.

6.3 VPD.386

Included in the Windows Resource Kit is a file called VPD.386. This file provides
support for the virtual printer device. If you are attempting to print to the same

port from a Windows application and a non-Windows application **at the same time**, you may encounter a port conflict. If this is the case, copy VPD.386 from the DRIVERS directory on the Windows Resource Kit disk to your WINDOWS\SYSTEM directory. Then add the line **device=vpd.386** to your SYSTEM.INI file in the **[386enh]** section. Exit and restart Windows.

7.0 Windows 3.1 Disk Layouts

Please refer to the file LAYOUTS.WRI located in your Windows Resource Kit directory.