



GRID[®]

386^{MAX}
TM

A Memory Manager for 386 Systems

User's Guide
and Reference

386^{MAX}™ User's Guide
A Memory Manager for 386 Systems

June 1990

COPYRIGHT © 1990 GRiD Systems Corporation
47211 Lakeview Boulevard
Fremont, CA 94538-6599
(415) 656-4700
FAX (415) 683-9888

Manual Name: 386^{MAX} User's Guide
Order Number: 22161-40
Issue Date: June 1990
Revision: B

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopy, recording, or otherwise, without the prior written permission of GRiD Systems Corporation.

The information in this document is subject to change without notice.

Neither GRiD Systems Corporation nor this document makes any expressed or implied warranty, including, but not limited to the implied warranties of merchantability, quality, or fitness for a particular purpose. GRiD Systems Corporation makes no representation as to the accuracy or adequacy of this document. GRiD Systems Corporation has no obligation to update or keep current the information contained in this document.

Under no circumstances will GRiD Systems Corporation be liable for any loss or other damages arising out of the use of this manual.

GRiD and GRiDCASE are registered trademarks and GRiDCASE 1500 Series is a trademark of GRiD Systems Corporation.

386^{MAX} is a trademark of Qualitas, Inc. and is used in this user's guide via a license agreement.

Other brand and product names are trademarks or registered trademarks of their respective holders.

CONTENTS

ABOUT THIS BOOK

Manual Organization	vii
Related Publications	viii
Notational Conventions	viii

CHAPTER 1: INTRODUCTION

Product Features	1-1
Technical Specifications	1-3
Where To Find Additional Information	1-3
Product Support	1-4

CHAPTER 2: INSTALLATION

Fast Start	2-1
Installing The Software	2-1
Default Settings	2-4

CHAPTER 3: BASIC OPTIONS

When To Use Options	3-1
How To Use Options	3-2
Typical Memory Map	3-2
List Of Basic Options	3-4
EMS=nnnn	3-4
EXT=nnnn	3-5
FRAME=xxxx	3-5

NOFRAME	3-6
PRO=d:\path\filename.ext	3-6
SWAP=nnn	3-6
AUTO	3-8
Examples	3-9

CHAPTER 4: UTILITIES

LIST	4-1
LOADHIGH, LOADLOW, MAXHI, MAXLO, OPENHIGH, and CLOSEHIGH	4-2
How High MS-DOS Memory is Used	4-3
Loading Programs in High MS-DOS Memory	4-3
Examples	4-4
Notes	4-5
MAPMEM and MAPEMS	4-7
Memory Resident Programs	4-7
How to Read the Display	4-8
EMS Memory Usage	4-8
ROMSCAN	4-10
TIMEMEM	4-12
Enabling And Disabling 386 ^{MAX}	4-14

CHAPTER 5: ADVANCED OPTIONS

How to Use Options	5-1
INCLUDE=nnn or xxxx-yyyy	5-1
EXCLUDE=xxxx-yyyy	5-2
HMAMIN=nn	5-2
XMSHNDL=nnn	5-2
NOCOMPROM	5-3
NOHIGH	5-3
NOLOW	5-3
NOROM	5-4
NOWARMBOOT	5-5
PRGREG=x	5-5
RAM=xxxx-yyyy	5-5

RESETKEYB	5-6
ROM=xxxx-yyyy	5-6
SHADOWRAM	5-6
SHADOWROM	5-7
USE	5-7
VIDMEM=xxxx-yyyy	5-7
Using VIDMEM with GRiD Desktop Computers	5-8
Using VIDMEM with GRiD Laptop Computers	5-9
VGA, EGA, CGA, MONO	5-9
AMRS=nnn	5-10
DMA=nnn	5-10
Advanced Examples	5-11
GRiD Desktop Examples	5-11
GRiD Laptop Examples	5-12

CHAPTER 6: 386DISK — A VIRTUAL DISK FOR 386^{MAX}

CHAPTER 7: COMMON QUESTIONS

APPENDIX A: OPTION SUMMARY

APPENDIX B: UTILITY PROGRAM SUMMARY

APPENDIX C: MESSAGES

Installation Messages	C-1
Device Driver Option Errors	C-5
Critical Error Messages	C-9
Utility Program Error Messages	C-11
386DISK Messages	C-14

APPENDIX D: USING 386^{MAX} WITH NETWORKS

Banyan, Novell, and 3COM Networks	D-1
Banyan Example	D-2
Novell Example	D-3
3COM Example	D-3
Autoexec.bat File	D-4
Config.sys File	D-4
3COM Network Cards	D-5
IBM Token Ring	D-6
Network Device Names	D-6
Etherlink/Plus	D-6
Novell ELS	D-7
Starlan	D-7
TOPS	D-7

GLOSSARY

List of Figures

Figure 2-1. Memory Usage Display	2-3
Figure 3-1. Typical Memory Map for GRiD Computer	3-3
Figure 4-1. Memory Map Displayed by LIST Command	4-2
Figure 4-2. Memory Map for Resident Program Usage	4-7
Figure 4-3. EMS Memory Usage Table	4-9
Figure 4-4. ROM Locations in High Memory	4-10
Figure 4-5. Memory Access Times Table Displayed by TIMEMEM Command	4-12

ABOUT THIS BOOK

This manual describes how to install, configure, and use 386^{MAX}™ on your GRiD computer system.

Manual Organization

The manual is organized as follows:

Chapter 1 lists 386^{MAX} product features and technical specifications.

Chapter 2 explains how to install 386^{MAX} and describes the default settings for 386^{MAX}.

Chapter 3 explains how to customize the operation of 386^{MAX} using basic options.

Chapter 4 explains how to use the utilities provided by the *386max.com* program.

Chapter 5 explains how to use the advanced memory management options provided by 386^{MAX}.

Chapter 6 describes the use of the 386DISK virtual disk program.

Chapter 7 provides answers to commonly asked questions.

Appendixes A-C summarize all 386^{MAX} device driver options, utility program commands, and error and informational messages. **Appendix D** provides information on using 386^{MAX} with networks.

A postage paid Customer Response Card is provided at the end of this manual. Owners and users are encouraged to use the card to comment on the usefulness and readability of the manual.

Related Publications

You may want to refer to the following books for further information while using 386^{MAX}.

The *User's Guide* supplied with your GRiD computer.

MS-DOS User's Guide and Reference recommended in the user's guide supplied with your computer, for reference information on using MS-DOS with your computer.

Notational Conventions

The following conventions are used to distinguish key elements of text in this manual:

- | | |
|----------------|---|
| bold | Used for commands, options, switches, and literal portions of syntax that must be entered exactly as shown. |
| <i>italics</i> | Used for filenames, variables, and placeholders that represent the type of text to be entered by the user. |
| monospace | Used for sample command lines, program code and examples, and sample sessions. |
| keycaps | Used to identify keys or key sequences on the computer keyboard. |

CHAPTER 1: INTRODUCTION

The 386^{MAX} program is a memory manager for 80386- and 80386SX-based GRiD laptop and desktop computers with extended memory. The 386^{MAX} program makes most of the memory in the first megabyte available to MS-DOS programs. You get more fast memory to run large applications such as spreadsheets and CAD/CAM.

Some GRiD Systems laptop and desktop computers are configured with shadow RAM, which is used to store system and video BIOS programs, and is not usable by 386^{MAX}. For additional information about shadow RAM, refer to the user's guide supplied with your computer.

NOTE: Some features of 386^{MAX} are not applicable to GRiD laptop computers or are used differently for laptops. These differences are identified throughout the manual.

Product Features

Using 386^{MAX} gives you these advantages:

- Allows your programs to take advantage of most of the first megabyte of memory not used for some other purposes such as display or network adapters for relocating Terminate and Stay Resident (TSR) programs.
- Fully emulates the Lotus-Intel-Microsoft (LIM) Expanded Memory Specification (EMS) Version 4.0 using the 80386 microprocessor's hardware paging tables and all available extended memory. No separate memory board is needed. Provides access to memory beyond the first 1 MB (megabyte) as either extended or EMS memory, or any combination of the two. The ratio between EMS and extended memory can be changed at system start-up time by the **ems** or the **ext** option.

Also supports advanced LIM EMS Version 4.0 features such as physical page mapping throughout the first 640 kB (kilobytes) of MS-DOS memory, allowing multitasking programs to swap your applications in and out.

- Valuable low MS-DOS memory is freed up for direct use by MS-DOS applications. The 386^{MAX} program does this by loading memory-resident programs above the first 640 kB.
- Reduces the time to type a file to a VGA screen by up to 40 percent! The 386^{MAX} program automatically remaps slow ROMs to fast RAM (many VGA ROMs and BIOS ROMs). This technique is called ROM caching.
- Fully supports the new Extended Memory Specification (XMS) from Microsoft. This specification allows programs to access the 64 kB of memory just above the system BIOS at 1 MB, called the High Memory Area (HMA). It also allows programs to allocate and deallocate both extended memory (called Extended Memory Blocks) and high MS-DOS memory (called Upper Memory Blocks in XMS). For more information, refer to the `hmamin` option in Chapter 5.
- Supports the Virtual Control Program Interface (VCPI) specification, which allows 386^{MAX} to coexist with certain 386 protected mode programs such as Paradox/386 from Borland and Interleaf Publisher from IBM. (Windows/386 is not yet supported.)
- Supports the Virtual DMA Services (VDS) specification, which allows programs that need to manage the DMA controller themselves to coexist with programs such as 386^{MAX}. In particular, SCSI hard disk and other bus masters can work with 386^{MAX} by following the VDS specification.
- For GRiD laptop computers, expands standard MS-DOS memory up to 736 kB, unless your system contains application ROMs, a built-in VGA display, or a VGA expansion cartridge.
- Automatically ensures correct operation of Direct Memory Access (DMA) of remapped memory.
- Allows hardware interruption of all EMS and extended memory usage. With this feature, your system runs more smoothly when using background communications and other interrupt-intensive programs.

- Uses only about 3 kB of valuable standard MS-DOS memory. Other memory managers use up to 64 kB.

Technical Specifications

The 386^{MAX} program requires the following hardware and software:

- Any 80386-based GRiD laptop or desktop computer
- A minimum of 256 kB of extended memory
- GRiD[®] MS-DOS 3.21 or later

The 386^{MAX} program uses approximately 3 kB of conventional memory and 60 kB or more of extended memory. The actual amount of extended memory used varies depending upon the amount of memory to be managed. The 386^{MAX} program conforms to the Lotus-Intel-Microsoft (LIM) Expanded Memory Specification (EMS) Version 4.0. It supports up to 15.33 MB of EMS memory.

The 386^{MAX} program is implemented as a device driver so that other device drivers can take advantage of EMS memory. It runs in protected mode at the topmost addresses in extended memory. When 386^{MAX} is active, MS-DOS programs run in a special mode of the 80386 called Virtual 8086 Mode.

Where To Find Additional Information

For the most current information on enhancements to this product, display the *readme.max* file found on your distribution diskette. This file describes those features added to 386^{MAX} since this user's guide was printed. To display this file located on drive A, enter the following line at the MS-DOS prompt:

```
type a:readme.max | more
```

This command uses the MS-DOS utility programs **type** and **more** to display the *readme.max* file. The file is displayed one page at a time; press any key to see the next page.

Also, refer to the Glossary for the definitions of some of the terms used in this manual.

Product Support

For technical support of the 386^{MAX} program supplied for your GRiD Systems laptop or desktop computer, contact an authorized GRiD Systems representative, call the GRiD Resource Center (GRC) at 1-800-284-GRID (4743), or write to: GRiD Systems Corporation, GRiD Resource Center, P.O. Box 5003, Fremont, California 94537-5003, for further information. In Europe, contact your local GRiD Systems representative or distributor.

CHAPTER 2: INSTALLATION

This chapter explains how to install 386^{MAX} and describes the default settings.

Fast Start

If you need to get started right away, copy the files *386max.sys* and *386max.com* onto your hard disk, and place a line similar to the following in your *config.sys* file:

```
device=d:\path\386max.sys
```

where *d:\path* refers to the disk drive and path where you have copied *386max.sys*.

The 386^{MAX} program assumes default settings, which are just right for most users who want the maximum amount of expanded (EMS) memory. The options, commands, and displays are available if you need them to customize your system. If you are more technically inclined, you may want to look over the options available and explore ways to maximize your system.

Installing The Software

To install 386^{MAX}, copy the files *386max.sys* and *386max.com* from the distribution diskette to any directory on your hard disk and include a line similar to the following in your *config.sys* file:

```
device=d:\path\386max.sys options
```

where *d:\path* refers to the disk drive and path where you have copied *386max.sys*, and *options* refers to additional options and associated parameters listed in Chapter 3 and Chapter 5. If no options are present in the device driver line, 386^{MAX} assumes default settings, which are described in the following section.

NOTE: For GRiD laptop computers, include a line similar to the following in your *config.sys* file:

```
device=d:\path\386max.sys nolow options
```

The **nolow** option shown above is fully described in Chapter 5. The **nolow** option prevents 386^{MAX} from interfering with application ROMs in some GRiD laptop computers.

You can edit or create the *config.sys* file using any word processor you are familiar with or the **edlin** program supplied with MS-DOS. For more information about creating the *config.sys* file, refer to the *MS-DOS User's Guide and Reference*.

The placement of the above line in your *config.sys* file is important. Device drivers that manage your hard disk (such as partitioning it into multiple drives) or that provide similar capabilities should be installed first. Then 386^{MAX} should be installed, followed by any device drivers that reference extended memory (some of which is used by 386^{MAX}) or EMS memory (all of which is created by 386^{MAX}) such as disk caches, RAM disks, or the like.

For example, when using a RAM disk such as **ramdrive** or **vdisk**, install the *386max.sys* driver first in your *config.sys* file before the device driver for the RAM disk. Also, for **ramdrive**, be sure to specify the **/e** switch (to use extended memory); and, be sure to specify how much extended memory to reserve for **ramdrive** or **vdisk** by using the **ext=nnnn** option with 386^{MAX} (see Chapter 3). For example:

```
device=386max.sys ext=512  
device=ramdrive.sys 512 /e
```

NOTE: Although RAM disks and disk caching can be set up in expanded memory (EMS), they are both faster when implemented in extended memory.

Unless you disable the EMS part of 386^{MAX} by specifying **ems=0**, be sure to remove other EMS managers from your *config.sys* file. Such files typically have names that contain the letters *emm*.

NOTE: If you have difficulty installing 386^{MAX} and do not have an MS-DOS disk handy from which to reboot the system, you can abort the installation without having to boot from a separate MS-DOS disk. At the beginning of the 386^{MAX} installation, if the **Alt** key is pressed, 386^{MAX} displays the following error message and does not install:

Alt-key pressed

(This key must be pressed and held down **before** the 386^{MAX} copyright notice appears and for a moment after it appears.)

Default Settings

For GRiD desktop computers and VGA laptop computers, if no options are present on the device driver line in your *config.sys* file, 386^{MAX} assumes the following default settings:

- Allocates all but 64 kB of extended memory, except for 64 kB of XMS, as EMS memory
- Backfills all memory from the top of main memory to the bottom of display adapter memory
- Topfills all memory above display adapter memory and tells MS-DOS that it is available for its use
- Swaps all slow ROMs for fast RAM
- Uses the 64 kB segment at E0000 (hexadecimal address) as the EMS page frame

These standard settings should work for all possible system configurations of GRiD computers. However, depending on your particular system configuration, you may be able to gain additional memory for use by MS-DOS by changing the standard options. Also, if you are using a RAM disk or disk cache, you may want to implement these features in extended memory. Refer to Chapters 3 and 5 for information on how to customize your memory configuration.

If you are running DESQview by Quarterdeck, you can provide more memory in which to run your programs by installing the device driver *qext.sys* (found on your DESQview distribution diskette) in your *config.sys* file.

Make sure that this driver is installed **before** *386max.sys*. Also, be sure to remove the keyword **unshift** if it appears on the *386max.sys* line. The effect is to reduce DESQview's low MS-DOS memory needs by about 63 kB, and thus increase the amount of available memory for your programs by the same amount.

NOTE: If you are using 386^{MAX} in conjunction with a multitasking program such as Windows/286 or DESQview 2.01, be sure to read the description of the **include** option in Chapter 5.

If you are using 386^{MAX} in conjunction with a network or special purpose high-resolution graphics adapters, be sure to read the description of the **ram** option in Chapter 5.

CHAPTER 3: BASIC OPTIONS

This chapter describes the basic options recognized by 386^{MAX} when it is installed in your *config.sys* file. More advanced options are described in Chapter 5.

When To Use Options

Unless you tell it otherwise, 386^{MAX} automatically provides default settings that make full use of extended memory as EMS memory (see Default Settings in Chapter 2). However, you generally should use some options in order to optimize your memory usage.

NOTE: For GRiD laptop computers, you are not specifically required to use any of the device driver options unless you have application ROMs or a VGA expansion cartridge. If you have application ROMs, use the **nolow** option (see Chapter 5). If you have a VGA cartridge, use the **vga** option (see VIDMEM in Chapter 5).

Additionally, if you want to use a RAM disk or a disk cache program, you may want to implement it in extended memory rather than EMS memory, since such programs work faster using extended memory. To do this, you need to use a device driver option.

How To Use Options

To use the device driver options to customize 386^{MAX} to your requirements, first display your *config.sys* file using your word processor or the MS-DOS editor *edlin*, and locate the line where 386^{MAX} is installed. It should look similar to the following line:

```
device=d:\path\386max.sys
```

Simply edit the device driver line to change or add any of the 386^{MAX} options described below. If you are using your own word processor, remember to save the file as a plain, unformatted ASCII file.

Restart your computer to cause your changes to take effect.

Typical Memory Map

Several of the 386^{MAX} options discussed in this chapter and in Chapter 5 concern how to treat special areas of memory in your GRiD computer. A typical memory map of a GRiD computer is shown in Figure 3-1 to help you get a better picture of how memory inside your computer is allocated, and what areas of memory 386^{MAX} affects.

NOTE: For a more complete memory map, refer to the user's guide supplied with your computer.

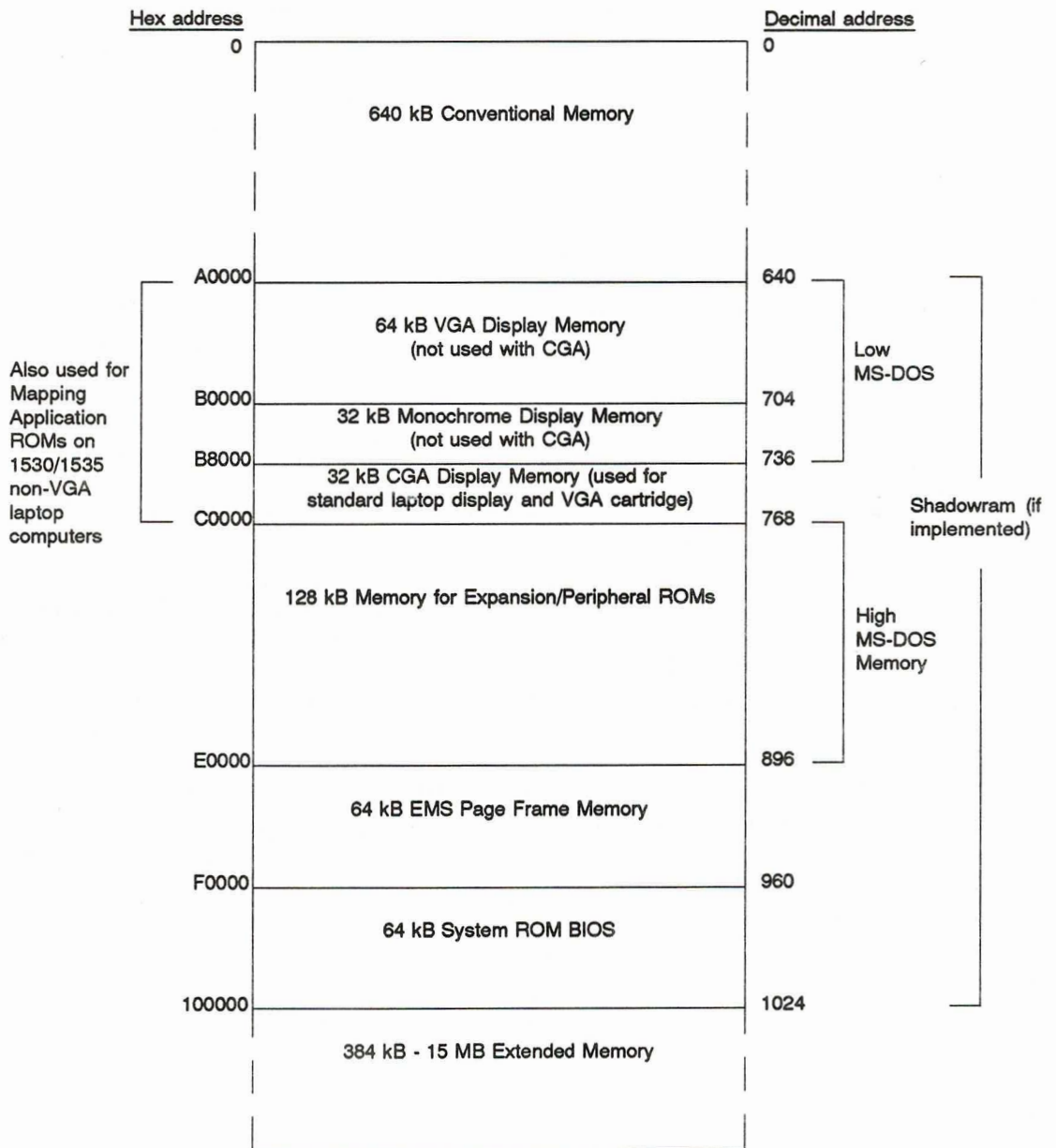


Figure 3-1. Typical Memory Map for GRiD Computer

List Of Basic Options

This section describes the basic device driver options that you can specify with the 386^{MAX} program.

The first two options each tell 386^{MAX} how much extended memory to convert to EMS memory, but they represent different views of the same question. Use the **ems** option if you know exactly how much EMS memory you need (such as for an EMS-based disk cache) and want all the rest used as extended memory. Conversely, use the **ext** option if you know exactly how much extended memory you need (such as for a RAM disk) and want the rest used as EMS memory.

NOTE: The **ems** and **ext** options cannot be used together.

EMS=*nnnn*

This option tells 386^{MAX} to use *nnnn* kilobytes of extended memory as EMS memory. If *nnnn* is less than the total amount of extended memory, the remaining extended memory is available for other use (such as a RAM disk).

If this option is omitted, all extended memory (up to the smaller of either the total available extended memory or the total extended memory less the **ext=** value) is used as EMS memory.

Enter *nnnn* in decimal kilobytes; for example, to use 1,024 kB (1 MB) of extended memory as EMS memory, specify **ems=1024**. The value must be a multiple of 16 kB; it is rounded down if it is not.

To cause all memory to be allocated to extended memory and none to EMS memory, specify **ems=0**.

NOTE: The **ems** and **ext** options cannot be used together.

EXT=nnnn

This option tells 386^{MAX} to leave at least *nnnn* kilobytes of extended memory as extended memory. The remaining memory is automatically used as EMS memory.

If this option is omitted, all extended memory (up to the smaller of either the total available extended memory or the **ems=** value) is used as EMS memory, except for 64 kB for the High Memory Area (HMA). If none of your programs use the HMA, and you want to allocate all memory for EMS usage, specify **ext=0** on the *386max.sys* line. See Chapter 5 for more information on HMA.

Enter *nnnn* in decimal kilobytes; for example, to leave at least 1,024 kB (1 MB) of extended memory as extended memory, specify **ext=1024**. The actual value used may be up to 15 kB higher than the specified value because EMS memory is allocated in multiples of 16 kB only, and any remainder is left as extended memory.

NOTE: The **ext** and **ems** options cannot be used together.

FRAME=xxxx

This option tells 386^{MAX} where to locate the EMS page frame. The EMS page frame is a 64 kB “window” that is used to gain access to EMS memory.

Normally, you do not need to specify this option; 386^{MAX} automatically searches for an available 64 kB window in high MS-DOS memory. The first frame checked is at the highest available address. If this frame is not available, locations lower in memory on 16 kB boundaries are tested.

The frame is normally located beginning at memory location E0000 (or D0000, if location E0000 is occupied).

To select a different location for the EMS page frame, use **frame=xxxx**. Enter the value *xxxx* in hexadecimal paragraphs; for example, specify **frame=d000** to use segment D0000 as the EMS page frame. This value must be on a 16 kB boundary; that is, the value must be *x000*, *x400*, *x800*, or *xC00*.

If there is no room for a 64 kB EMS page frame, 386^{MAX} will install a frame in conventional memory. Generally, you should allow 386^{MAX} to select the EMS page frame automatically.

NOFRAME

If 386^{MAX} cannot find a contiguous 64 kB EMS page frame in high MS-DOS memory, it attempts to install the frame in low MS-DOS memory. If you are using applications such as AutoCad that can take advantage of EMS memory without using the EMS page frame, use the **noframe** option on the *386max.sys* command line. This option allows 386^{MAX} to install without an EMS page frame and to add more high MS-DOS memory.

You can also use **ems=0** to prevent 386^{MAX} from installing an EMS page frame at all.

PRO=d:\path\filename.ext

(Used only on GRiD desktop computers.) This option collects into a single separate file any of the options described in this chapter or in Chapter 5. To use this option, create a file anywhere on your hard disk and place each option on a separate line in that file. Then add the **pro** option as follows to the device driver line in your *config.sys* file:

```
device=d:\path\386max.sys pro=d:\path\filename.ext
```

where the second occurrence of *d:\path* indicates the drive letter and path on that drive where the file is located, and *filename.ext* indicates the filename of the profile.

You may list device driver options on the **device=** line in your *config.sys* file or in the file pointed to by this option; however, they cannot appear in both places.

SWAP=nnn

(Used only on GRiD desktop computers.) This option swaps the first *nnn* kilobytes of conventional memory (which presumably is slow) with the same amount of extended memory starting at 1 MB (which presumably is fast). The *nnn* value must be a multiple of 4 kB and should be a multiple of 64 kB. This option is needed only with 386 accelerator cards installed in systems with 16-bit memory on the system board.

The goal of the memory swapping feature is to put the fastest memory into the region most frequently accessed. For example, if your 386 add-in card contains fast (32-bit) extended memory starting at address 1 MB, and your system board contains slow (16-bit) conventional memory, this option will work well.

Memory is identified by access time (for example, 120 nanoseconds) as well as the width of the data path (8-, 16-, or 32-bits) by which it is accessed. With 80386 systems, memory speed is driven more by the data path width than it is by the access time of the memory chips. Because 16-bit memory is not very closely coupled with the 80386 processor, it often takes much longer to read and/or write it. On the other hand, 32-bit memory is almost always the memory most tightly coupled to the processor and is thus the fastest memory in the system.

Naturally, the most frequently accessed memory in your system is the first 640 kB, and the lower memory of that region is accessed more frequently than the higher memory. The fast 32-bit memory starting at 1 MB can be swapped with the slow 16-bit (but more frequently accessed) memory starting at 0 MB using the **swap** option.

After you have used the **swap** option, you can use the **timemem** utility command to display a table showing memory access times. See Chapter 4 for more information on **timemem**.

After the memory has been swapped, there is 16-bit memory starting at 1 MB. If you want to avoid this memory being used for EMS memory, simply specify **ext=nnn**, where *nnn* is the same value as in the **swap=nnn** option. In this way, 386^{MAX} will not map in any of the slow 16-bit memory for EMS use. Moreover, so that the 16-bit memory does not completely go to waste, you might use **vdisk** to set up a RAM disk in extended memory of *nnn* kilobytes. Be sure to specify **/e** on the **vdisk** line so that **vdisk** uses extended memory.

AUTO

The initial state of 386^{MAX} is normally **on**. In certain circumstances, it may be necessary to disable 386^{MAX} temporarily in order to run protected mode programs, and then reenable it afterwards. To accomplish this, the **auto** option is available.

The **auto** option causes 386^{MAX} to automatically activate itself when EMS needs to be used; it automatically deactivates itself when EMS is not being used. This option is recommended for ease of use and because it results in an increase in system performance.

If desired, you can later explicitly activate 386^{MAX} by using the **on** option with the *386max.com* utility program; refer to Enabling And Disabling 386^{MAX} in Chapter 4.

If, when running certain programs, you encounter an error message such as:

A Privileged operation exception has occurred at address *xxxx:yyyy*

try installing 386^{MAX} with the **off** or **auto** option. Then before running the program that signaled the above message, disable 386^{MAX} using **386max off**. Be sure to reenable 386^{MAX} afterwards with either **386max on** or **386max auto**.

NOTE: If the **auto** option is specified on the device driver installation line, low MS-DOS memory cannot be backfilled to increase the amount of memory available to MS-DOS. Thus, if you have omitted the **nolow** option, 386^{MAX} installs itself as if you had specified **nolow**.

Examples

This section provides examples of common ways in which 386^{MAX} may be configured. The examples below use options that are described in this chapter. In the following examples, assume you have 4 MB of memory in your system. The actual division of memory into its separate uses varies depending on your system configuration. For example, extended memory may be used for ROM mapping, for EMS memory, for extended memory, for high MS-DOS memory fill, and finally for low MS-DOS memory fill.

1. To invoke 386^{MAX} using all of the extended memory as EMS memory, place the following line in your *config.sys* file:

```
device=d:\path\386max.sys
```

2. If you want to reserve 1,024 kB (1 MB) of extended memory for a RAM disk, for example, and use the remaining 2.4 MB for EMS memory, ROM mapping, memory fill, and so on, use the following line in your *config.sys* file:

```
device=d:\path\386max.sys ext=1024
```

3. To allow only enough EMS memory to handle an EMS-based disk cache program, for example, use the *ems* option. If the disk cache uses 512 kB of EMS memory, use the following line:

```
device=d:\path\386max.sys ems=512
```

4. If you are using a 386 accelerator card and have set your system switches to disable all but the first 256 kB of system board memory (presumably the accelerator board is backfilling memory from 256 to 640 kB), use the following line in your *config.sys* file:

```
device=d:\path\386max.sys swap=256
```

5. If you have collected your basic and advanced options into a profile, use a configuration such as the following:

```
device=d:\path\386max.sys pro=d:\path\386max.pro
```

The following example applies to a GRiD laptop computer with 2 MB of RAM.

6. If you want to save part of extended memory for a RAM disk or a disk cache, use a configuration such as the following:

```
device=d:\path\386max.sys norom nolow ext=512
```

This configuration provides 640 kB of standard memory, 512 kB of extended memory for a RAM disk or disk cache, and 752 kB of EMS memory. The ROM BIOS is not swapped into extended memory, and low MS-DOS memory is not backfilled to increase the amount of standard memory available to MS-DOS. For more information on the **norom** and **nolow** options, see Chapter 5.

CHAPTER 4: UTILITIES

This chapter describes the utility functions provided by the *386max.com* program. These utility functions can be invoked at any time by giving the command **386max** followed by one of the utility option names.

NOTE: In order to use the **386max** command, you must be in the directory where that program (*386max.com*) resides, or you must use the **path** command to set the path to that directory.

The commands may be typed in uppercase or lowercase letters.

LIST

The **list** command displays a memory map of the memory usage on your system.

To use the **list** command, enter one of the following commands at the MS-DOS command line:

```
386max list
386max /l
386max
```

This command displays a memory map similar to the one that is displayed at system start-up when **386^{MAX}** is installed. A typical memory map for a GRiD computer with 2 MB of RAM is shown in Figure 4-1.

MS-DOS memory. The programs **maxhi** and **maxlo** implement the same functions as **386max loadhigh** and **386max loadlow**, but take up much less memory. Without exception, you can use **maxhi** in place of **386max loadhigh** and **maxlo** in place of **386max loadlow**.

How High MS-DOS Memory is Used

With **386MAX**, all memory above the display adapters not already in use as ROM or RAM is available to MS-DOS. This high MS-DOS memory can be used:

- by memory-resident programs instead of using valuable low MS-DOS memory. Memory-resident programs can use high MS-DOS memory via the **loadhigh** command;
- by applications requesting more memory from MS-DOS. This use is enabled by the **openhigh** command. When this command is used, MS-DOS is told about the extra memory and uses it automatically when applications request more memory.

Loading Programs in High MS-DOS Memory

To begin loading memory-resident programs in high MS-DOS memory, enter one of the following commands at the MS-DOS command line or in a batch file:

```
386max loadhigh
386max high
maxhi
```

To return to loading programs in low memory, use one of the following commands:

```
386max loadlow
386max low
maxlo
```

The first step is to determine what memory-resident programs are being used currently and how much memory they occupy. To display a list of such programs along with their resident memory needs, use the **386max mapmem** command (described later in this chapter.) In your *autoexec.bat* file, find the program names that appear in the MAPMEM listing. Then surround the call to each memory-resident program with a call to **386max loadhigh** and **386max loadlow**.

NOTE: To surround the call to each memory-resident program, place **386max loadhigh** before the call and **386max loadlow** after the call.

If your memory-resident programs take up more space than is available in high MS-DOS memory, you will have to select those that best fit the available space. Some juggling may be required. Unless there is a specific order in which the programs must be loaded, a good first attempt is to select the one that takes up the largest amount of resident space to be loaded first.

Not all memory-resident programs are well-behaved enough to be loaded into high MS-DOS memory. If some program does not operate the same as it does when loaded into low MS-DOS memory, that one might not be a candidate for relocation.

NOTE: *command.com* cannot be relocated to high MS-DOS memory.

Because space in high MS-DOS memory is limited, you should ensure that each program fits in the remaining space before attempting to load it. Some programs do not check that there is sufficient memory above them before going resident. If there is not enough memory for the program's dynamic memory needs, the program might not realize it and install anyway. Quite likely, the system will crash shortly thereafter.

Examples

Utilizing high MS-DOS memory is made easier by loading memory-resident programs from your *autoexec.bat* file. For example, to load a program such as *print.com*, use calls similar to the following:

```
386max loadhigh
print /d:lpt1 /b:6000 /q:32
386max loadlow
```

where the text to the right of **print** is the argument for the program.

Multiple programs may be loaded into high MS-DOS memory between calls to **386max loadhigh** and **386max loadlow**. For example, to load several memory-resident programs supplied with MS-DOS into high MS-DOS memory, use calls similar to the following:

```
386max loadhigh
append /e
fastopen c:=100
mode lpt1=com1
print /d:lpt1 /b:6000 /q:32
graphics
386max loadlow
```

When **loadhigh** is in effect, programs can be loaded into high MS-DOS memory. When **loadlow** is in effect, high MS-DOS memory is closed off from use. This means that any unused high MS-DOS memory is unavailable to programs running in low MS-DOS memory. Generally, very few programs can take advantage of high MS-DOS memory, so little is lost. If you have programs that can use that extra memory, use **386max openhigh** to make it available again. To close off high MS-DOS memory again, use **386max closehigh**.

However, we suggest you proceed with caution before using **openhigh**. Several commercially available programs (such as Windows 2.0 and Software Carousel) get confused when they detect memory allocation chain entries extending above the display adapters. Unless you know that some specific programs can use high MS-DOS memory, we recommend that you avoid using the **openhigh** and **closehigh** commands.

Notes

1. Only memory-resident programs called from the MS-DOS command line (or through a batch file such as *autoexec.bat*) can be loaded into high MS-DOS memory. While this area is generally 96-128 kB on a typical system, a certain number of programs will fit nicely into high MS-DOS memory.

Device drivers loaded from your *config.sys* file cannot be placed into high MS-DOS memory.

2. If ROM or other RAM is present in high MS-DOS memory (such as found on network or high-resolution graphics adapters), not only is less memory available for memory-resident programs, but there is a chance that memory is fragmented (split into two or more noncontiguous blocks). This can further reduce available memory. While all fragments are available for use, a contiguous block is naturally more useful. To see how high MS-DOS memory is mapped, use the **mapmem** command described later in this chapter.
3. For any program (memory-resident or not) to load in high MS-DOS memory, a certain minimum amount of memory must be available to *command.com*. This amount varies with the version of MS-DOS, but can be as large as 19 kB. The space is taken out of the highest part of high MS-DOS memory.

Although this minimum amount of memory must be available **before** loading each program, it is not necessary that the memory-resident program leave enough memory **afterwards** for *command.com*. If insufficient memory remains to load *command.com* after installing a memory-resident program, 386^{MAX} automatically tells MS-DOS to begin loading programs in low memory.

If there is insufficient room in high MS-DOS memory in which to load the program, MS-DOS displays the message:

Program too big to fit in memory

and you may:

- call a program that requires less memory to load, or
 - return to loading programs in low MS-DOS memory by entering **386max loadlow**.
4. High MS-DOS memory is not available for use by MS-DOS if the **nohigh** option is specified on the device driver line in your *config.sys* file (see Chapter 5).

MAPMEM and MAPEMS

This section describes how to use the **mapmem** and **mapems** commands. The first command is used to display where memory-resident programs are allocated in memory. The second command is used to display how EMS memory is allocated.

Memory Resident Programs

To display the memory map, enter one of the following commands at the MS-DOS command line:

```
386max mapmem  
386max /m
```

A table similar to that shown in Figure 4-2 is displayed.

MEMORY MAP for RESIDENT PROGRAMS					
Name	Hex Start	Hex End	Hex Owner	Decimal Length	Text or Interrupt Numbers
DOS & drvrs	09B2	0E23		18,176	15 19 40 D3
COMMAND.COM	0E23	0EF7	0E24	3,376	22 23 24 2E
	0EF7	0EFB	-Avl-	48	
COMMAND.COM	0EFB	0F10	0E24	320	COMSPEC=C:\COMMAND.COM PROMP
	0F10	0F15	-Avl-	64	T=\$
APPEND.EXE	0F15	0F24	0F25	224	COMSPEC=C:\COMMAND.COM PROMP
APPEND.EXE	0F24	1038	0F25	4,400	T=\$2F DF FE
SUPERPCK.COM	1038	1048	1049	240	COMSPEC=C:\COMMAND.COM PROMP
SUPERPCK.COM	1048	1400	1049	15,216	T=\$13 20 21 27 62 67 CE F0
PKLOAD.COM	1400	1410	1411	240	COMSPEC=C:\COMMAND.COM PROMP
PKLOAD.COM	1410	17D1	1411	15,360	T=\$09 10 16
PS.COM	17D1	17E1	17E2	240	COMSPEC=C:\COMMAND.COM PROMP
PS.COM	17E1	2A1F	17E2	74,704	T=\$01 02 03 06 0D F2 F4 F6 F8 FC
	2A1F	2A2F	-Cur-	240	
-High DOS Mem-	2A2F	A000	-Cur-	482,560	
	C400	E000	-Avl-	114,672	

Figure 4-2. Memory Map for Resident Program Usage

Studying the chain of memory-resident programs enables you to maximize your use of high MS-DOS memory by installing memory-resident programs in a sequence that provides the best fit in the available memory.

The first step in this process is to display the current list of memory-resident programs using **386max mapmem**. After studying the table displayed on your system, you can use the **386max loadhigh** and **loadlow** commands (described earlier in this chapter) to reallocate memory-resident programs.

How to Read the Display

The first column in the table contains the resident program name (if any is available). The next two columns list the starting and ending addresses in hexadecimal 16-byte paragraphs.

The fourth column lists the paragraph of the owner of the block. A typical resident program owns two or more blocks—a small one containing its environment and a large one containing the resident program. Additional blocks may be owned by the program if it allocated memory before terminating and becoming resident.

The fifth column lists the length of each block in decimal bytes. The last column displays initial text in the block or interrupt numbers intercepted by the program. Typically, if some text appears, it identifies an environment block.

Lines with the name "RAM" or "ROM" are used to skip over display adapter memory and other RAM and ROM present in high MS-DOS memory. Lines whose Hex Owner is listed as "-Avl-" are unallocated and are available for use. If the Hex Owner is listed as "-Cur-", then it corresponds to the program currently running (that is, *386max.com*). This memory is returned to MS-DOS upon termination.

EMS Memory Usage

When programs allocate EMS memory, they are given a "handle" by which they are identified to the EMS memory manager. Each active handle and the memory allocated to it is displayed in the upper part of the MAPEMS table. In the lower part, the EMS page map is displayed. This map lays out how each 16 kB block of memory in the first megabyte is treated by the EMS manager. If a block is marked "DOS", "Include", or "Frame", the memory at that location can be swapped in and out. For reference, ROM and RAM in high MS-DOS memory is also displayed in this table although, strictly speaking, those blocks are not touched by the EMS memory manager.

EXPANDED MEMORY USAGE					
Handle	Memory	Name	EMM Version 4.0	Frame = E000	
		42 Mappable Physical Pages			
0	576				
1	512				
2	512				
Avail	640				
Total	2240 KB				

EMS Page Map																	
#	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	1 MB
###	[Pattern of memory usage symbols]																
#	Unused	— 0000-1000		A000-B000		B800-C000		C400-E000									
■	DOS	— 1000-A000															
▒	Include	— B000-B800															
░	Frame	— E000-F000															
■	ROM	— C000-C400		F000-10000													

Figure 4-3. EMS Memory Usage Table

To display the EMS memory map, enter one of the following commands at the MS-DOS command line:

```
386max mapems
386max /e
```

ROMSCAN

(Used only on GRiD desktop computers.) The **romscan** command displays a table showing the location of all read only memory (ROM) present in high MS-DOS memory.

The addresses in the display can be matched against the corresponding addresses in the Memory Access Times table (using the **timemem** command described later in this chapter) to see where ROM appears and how fast it is. As with the **mapmem** command described earlier, this information may clarify how high MS-DOS memory is being used.

To display the ROM locations table, enter one of the following commands at the MS-DOS command line:

```
386max romscan
386max /r
```

A table similar to that shown in Figure 4-4 is displayed.

```
C>386max romscan
386MAX -- Version 4.09. -- A Memory Manager for 386 Systems
(C) Copyright 1987-90 Qualitas, Inc. All rights reserved.
```

ROM/RAM in High DOS Memory					
RAM	High DOS	ROM	EMS	Unused	Length in KB
8800-C400	C400-E800	F800-10000	E800-F800	A000-B800	96
					48
					144
					64
					32
Legend: RAM = Physical RAM present -- avoid High DOS = Already in use as high DOS -- ✓ if overlap ROM = Read-Only Memory EMS = EMS-mappable addresses Unused = No physical RAM present at this time					

C>

Figure 4-4. ROM Locations in High Memory

NOTE: In order to help resolve conflicts with uninitialized RAM in high MS-DOS memory, the **romscan** command displays a table of high MS-DOS memory usage as if 386^{MAX} had not been installed. If this table shows that some physical RAM has been overlaid with high MS-DOS memory, be sure to resolve the conflict. In almost any case, you should use a **ram=** statement to protect the conflicting area. The only exceptions to this rule are Monochrome Display Adapters (MDAs) whose RAM responds to the entire B000-B800 range. If you do not use high-resolution monochrome graphics, it is always safe to recover some of this address space.

Generally, the proper **use=** statement is dependent upon (1) whether or not you run programs from the monochrome screen and (2) your BIOS. If you never run programs from monochrome, the statement **use=b100-b800** should suffice. Otherwise, the stating value for **use=** may have to be increased to B200, B300, or B400. The actual number depends on your BIOS.

TIMEMEM

The **timemem** command displays a Memory Access Times table, showing the access times for the various types of memory in your computer.

The Memory Access Times table supports the memory swapping option (**swap**) discussed in Chapter 3, and lets you make sure your fastest memory is mapped where MS-DOS programs can make efficient use of it. To display the Memory Access Times table, enter one of the following commands at the MS-DOS command line:

```
386max timemem
386max /t
```

A table similar to that shown in Figure 4-5 is displayed.

Note that the memory from A0000h to C0000h is slower than other memory. Those addresses correspond to display RAM, which is slower than conventional memory.

If the **timemem** utility encounters a Parity Error when timing memory, it reports the address of the 4 kB physical memory page containing the bad RAM chip. This address is independent of any memory swapping or DMA transfers that may have rearranged the mapping of linear to physical addresses. With the knowledge of how memory chips are organized in your system along with the above address, you should be able to identify the row containing the bad chip.

Because parity errors are sometimes intermittent, it may require several iterations of **timemem** to catch the error. You may want to create and run the following batch file. It will run continuously until a parity error occurs.

```
echo off
:LOOP
386max timemem >timemem
if not errorlevel 1 goto LOOP
type timemem
```

Enabling And Disabling 386^{MAX}

In certain circumstances, it may be necessary to disable 386^{MAX} temporarily in order to run protected mode programs, and then reenable it afterwards. To accomplish this, the program recognizes three states: **on**, **off**, and **auto**. These states affect the availability of the features of 386^{MAX}. These features can be divided into two groups:

- **Group A:** EMS support and ROM caching
- **Group B:** Low MS-DOS memory fill (from 256 to 640 or 704 kB), high MS-DOS memory fill (**loadhigh**), and low MS-DOS memory swap (**swap=**).

The **on** state means that the features of Group A are available at all times. The **off** state means that the features of Group A are unavailable until turned back **on**. The **auto** state means that the features of Group A are available only when EMS memory is being used. The default initial state is **on**.

These states also affect the availability of the features in Group B. These features are available only if the initial state is **on**. If you specify an initial state other than **on**, none of the features in Group B are available.

The initial state may be specified with one of the keywords **on**, **off**, or **auto** on the device driver line. If you are not using any of the features in Group B, you may change the current state using the command **386max on**, **off**, or **auto**. If you have installed 386^{MAX} using the **auto** option on the device driver line, you can later use any of the commands **on**, **off**, or **auto** to change the operating mode of the device driver.

The **auto** command causes 386^{MAX} to automatically activate itself when EMS memory needs to be used; it automatically deactivates itself when EMS memory is not being used. This command is recommended for ease of use and because it results in an increase in system performance. To use **auto**, enter the following command at the MS-DOS command line:

```
386max auto
```

To explicitly disable 386^{MAX}, enter the following command at the MS-DOS command line:

```
386max off
```

To reenable 386^{MAX}, enter the following command:

```
386max on
```

If, when running certain programs, you encounter an error message such as:

A Privileged operation exception has occurred at address xxxx:yyyy

try specifying an initial state of **off** or **auto**. Then before running the program that caused this message, call **386max off**. Be sure to reenable 386^{MAX} afterwards with either **386max on** or **386max auto**.

If you choose to run with **auto**, you may find that the keyboard locks up occasionally. The keyword **resetkeyb** (see Chapter 5) can be used to overcome this problem.

CHAPTER 5: ADVANCED OPTIONS

In addition to the commonly used device driver options discussed in Chapter 3, 386^{MAX} includes a number of advanced options that provide additional control of how 386^{MAX} manages memory. This chapter describes the advanced options recognized by 386^{MAX} when installed in your *config.sys* file.

How to Use Options

You can enter these options along with any others on the device driver installation line in your *config.sys* file or the profile pointed to by the 386^{MAX} **pro** option (see Chapter 3 for information on using **pro**.)

INCLUDE=nnn or **xxxx-yyyy**

To take greater advantage of multitasking systems such as Windows 2.0 (/286 mode only) and DESQview 2.01, you may want to add additional swapping regions using the **include** option. This option can take one of two forms:

- If you would like 386^{MAX} to select the addresses in high MS-DOS memory to use for swapping, use **include=nnn**, where *nnn* specifies the size of the region. This value must be a multiple of 16 kB and is specified in kilobytes. For example, to specify an additional 128 kB of high MS-DOS memory to use for swapping, use **include=128**.
- If you have specific addresses to use for swapping, use the form **include=xxxx-yyyy**. In this form, *xxxx* is the starting segment and *yyyy* is the ending segment. Both segments are expressed in units of paragraphs in hexadecimal. Both parts must be a multiple of 16 kB (that is, each value must be of the form *x000*, *x400*, *x800*, or *xC00*).

Note that these two forms of the **include** option can be used together. For example, **include=b000-b800, 128** includes the region normally reserved for a monochrome display adapter and requests that an additional 128 kB of high MS-DOS memory be used for swapping. The comma-delimited items can be specified in any order. This option can be specified multiple times.

EXCLUDE=xxxx-yyyy

To decrease the size of the swapping region, use the **exclude** option. If you do not use programs that take advantage of the additional swapping region available in LIM EMS 4.0, you might see a performance improvement by excluding some or all of the region from 64 kB to 640 kB. To exclude the entire region, use **exclude=1000-a000**. Note that *xxxx* and *yyyy* are expressed in units of paragraphs in hexadecimal. Both parts must be a multiple of 16 kB (that is, each value must be of the form *x000*, *x400*, *x800*, or *xC00*). This option can be specified multiple times.

HMAMIN=nn

The Extended Memory Specification (XMS) from Microsoft requires an XMS driver that allows you to specify the minimum amount of the High Memory Area (HMA) a program may request. Use this option to control this amount.

Also, when extended memory is allocated to programs, the memory is assigned a handle in the same way as EMS memory allocation is done. The default number of handles is 32. If your needs exceed this number, use the following option:

XMSHNDL=nnn

The minimum value is 2, and the maximum value is 255. Each handle occupies 6 bytes of extended memory in 386^{MAX}.

Because the HMA requires 64 kB of extended memory, by default, 386^{MAX} allocates a minimum of 64 kB to extended memory. If none of your programs use the HMA, and you want to allocate all memory for EMS usage, specify **ext=0** on the *386max.sys* line.

On the other hand, if you plan to use the HMA and are also running other extended memory programs, be sure to take into consideration the HMA's 64 kB in the amount of extended memory remaining in your system.

NOCOMPROM

In order to maximize the amount of high MS-DOS memory, 386^{MAX} automatically attempts to compress the system BIOS to a smaller size. The memory recovered can be as much as 32 kB, in which case the system BIOS starts at F800 instead of F000. If you do not want this automatic optimization, specify the **nocomprom** option.

NOHIGH

This option tells 386^{MAX} not to fill in memory above the display adapters. By specifying this option, more memory may be available for use as EMS or extended memory, but no additional memory is available for memory allocations by MS-DOS.

Moreover, if this option is used, you cannot load memory-resident programs above the display adapters.

NOLOW

If possible, 386^{MAX} normally increases the amount of conventional memory available to MS-DOS from 640 kB up to 704 kB. It does this by using some of the extended memory to fill in the area between where the normal 640 kB of system RAM stops and where the memory for the display is mapped.

This option tells 386^{MAX} not to use some of its extended memory to increase the amount of memory available to MS-DOS. By specifying this option, more memory may be available for use as EMS or extended memory, but no additional memory is available for direct use by MS-DOS.

For GRiD laptop computers, you should use this option under the following circumstances:

- Use this option if you have GRiD application ROMs (any ROMs containing programs or operating systems) installed in your computer. Application ROMs are mapped into the area that 386^{MAX} would try to fill, and it would cause a conflict.
- Use this option if you do not have application ROMs or a VGA display, but you want to have the maximum amount of extended memory. In other words, you don't want to give up any extended memory to increase the amount of standard memory available to MS-DOS.

NOROM

For GRiD desktop computers, this option tells 386^{MAX} not to swap slow ROMs for fast RAM. Ordinarily, 386^{MAX} times each ROM it finds, and if the ROM proves to be much slower than extended memory, it copies the ROM to the faster memory and uses that instead. 386^{MAX} uses this process to speed up a VGA display by up to 40 percent.

If you do not want 386^{MAX} to move ROMs to fast memory, use the **norom** option. You can see which ROMs (if any) 386^{MAX} found to be slower than extended memory by typing **386max list** at the MS-DOS command line. See Chapter 4 for more information.

For GRiD laptop computers, this option tells 386^{MAX} not to copy the ROM BIOS (Basic Input Output System) into extended memory. By specifying this option, you gain an additional 60 kB of memory to use as extended or EMS memory.

When you omit this option, 386^{MAX} copies the ROM BIOS into faster extended memory, where it is executed. This approximately doubles the speed of the BIOS code; however, this does not always significantly increase overall system performance. Additionally, this uses 64 kB of extended memory.

If you need as much extended memory as possible, then use the **norom** option. If you want to increase the speed of the BIOS and can afford to lose 64 kB of extended memory, then omit the **norom** option.

NOWARMBOOT

To ensure catching system reboot, 386^{MAX} always remaps at least the top 4 kB of system ROM into RAM. Generally, all ROMs are timed to see if extended memory is faster than the ROMs (and almost always the ROMs are very slow). If RAM is faster than ROM, the ROM is sped up. However, even if system ROM is as fast as RAM, the top 4 kB of system ROM is still remapped to ensure properly rebooting the system. If, for some reason, this is not a good use of memory on your system, you can disable it by specifying the keyword **nowarmboot** on the device driver line that invokes *386max.sys* or in a profile.

PRGREG=x

This option specifies the region of high MS-DOS memory to be used to store the 386^{MAX} device driver, where *x* can be 1-9. Regions in high MS-DOS memory are consecutively numbered from lowest to highest. As an example, if high MS-DOS memory is fragmented into three areas from B000-B800, C400-DC00, and DE00-E000, and you want 386^{MAX} to load into the region between DE00 and E000, specify **prgreg=3** as an option on the device driver line in your *config.sys* file.

RAM=xxxx-yyyy

Use this option to mark paragraphs *xxxx* through *yyyy* as RAM. Both values are entered in hexadecimal and are in units of 16-byte paragraphs. This option should be used in case there is RAM in your system (such as on a network or high-resolution graphics adapter) that is not found by 386^{MAX} at start-up.

During installation, 386^{MAX} always checks for all RAM above the display adapters. Use this option in the unlikely case that some RAM is not found when 386^{MAX} is installed. This can occur because the RAM was not enabled at that time. This option may appear multiple times.

RESETKEYB

In certain rare cases, you may find that 386^{MAX} installs correctly, but the keyboard is locked. To overcome this problem, use this option.

ROM=xxxx-yyyy

Use this option to mark paragraphs *xxxx* through *yyyy* as ROM. Both values are entered in hexadecimal and are in units of 16-byte paragraphs. Use this option if the specified ROM doesn't have the usual signature, or if you want to disable automatic remapping for some other ROMs but not for this one.

If this option is used, the automatic remapping mechanism is disabled for all ROMs. In any case, the specified ROM is not remapped to other memory unless that other memory is somewhat faster than the ROM.

To see the ROM addresses currently mapped to fast memory, use the **386max list** command described in Chapter 4. This option may appear multiple times.

SHADOWRAM

This option tells 386^{MAX} to recover the memory not used by the system for shadowing as extended memory.

Shadowram is the area of memory between 640 kB and 1 MB (A000:0h-FFFF:Fh). It is only implemented on certain machines using specific chipsets: 80286-, 80386-, and 80386SX-based desktops and 80386SX-based laptops. It can sometimes be remapped as extended memory or used for EMS. If used as shadowram, it can "shadow" the system BIOS, "shadow" the VGA BIOS and other adapter BIOSes, and be used for standby modes in laptops. The advantage of shadowing BIOS is performance. In many systems, the ROMs used for BIOS are accessed rather slowly. By copying those ROMs to RAM, they execute much faster, sometimes giving great performance improvements. This is not always the case, but generally shadowed ROM BIOSes do improve performance. (This is especially true where an 8-bit VGA card has been used.)

SHADOWROM

To speed up video calls, the system BIOS on some systems automatically relocates the EGA/VGA ROM at C000 to fast RAM at E000 or E800. This performance improvement has the side effect of fragmenting and reducing the amount of high MS-DOS memory. 386^{MAX} automatically backs out of this optimization to increase the amount of high MS-DOS memory as well as speeds up video calls by remapping the ROM in place at C000, thus giving you the same performance improvement as the BIOS technique as well as increasing the amount of high MS-DOS memory. If you do not want 386^{MAX} to perform this improvement, specify the **shadowrom** option to retain the BIOS technique of shadowing the ROM in the E000 block. Using **shadowrom** will reduce as well as fragment the amount of high MS-DOS memory.

USE

This option overrides reserved high MS-DOS memory that the program has found and otherwise would skip over. This option tells 386^{MAX} not to reserve paragraphs from *xxxx* through *yyyy*, but instead to use them for high MS-DOS memory. For example, some 80386 systems have ROMs that are decoded at multiple addresses, only one of which is actually used.

Both values are entered in hexadecimal and are in units of 16-byte paragraphs. This option may appear multiple times.

VIDMEM=*xxxx-yyy*

This option tells 386^{MAX} exactly which memory addresses are used by the video memory in your computer. This option can be useful because 386^{MAX} then allows other video memory areas outside this range to be used to increase the size of conventional MS-DOS memory.

Using VIDMEM with GRiD Desktop Computers

This option should be used for your primary display adapter only. If you have a secondary high-resolution graphics adapter and its RAM isn't found by 386^{MAX} at start-up, use the **ram** option to tell 386^{MAX} to avoid those locations; do not use **vidmem** for this purpose.

Normally, 386^{MAX} automatically detects the most commonly used display adapters. In the rare case that 386^{MAX} misidentifies your primary display adapter as starting higher in memory than it actually does, 386^{MAX} may overlap part of the display adapters with low MS-DOS memory. This misidentification is not fatal. You can detect it if you notice that your programs are not using the full display adapter memory. Typically, this means that high-resolution graphics modes in your primary display adapter don't display data correctly.

To help 386^{MAX} better identify your display adapter, use the **vidmem** option to specify the starting and ending addresses of your display adapter. For example, some EGA boards, when run in monochrome mode, don't properly identify themselves as an EGA, and instead appear as a Monochrome Display Adapter (MDA). In this case, use **vidmem=a000-c000**.

Before experimenting with this option, you might first specify the **nowlow** option (see Chapter 5) to disable low MS-DOS memory fill altogether. If that clears up graphics display problems, then specify a value for **vidmem**. Typical values are **a000-c000** (640 kB for a length of 128 kB) and **b000-c000** (704 kB for a length of 64 kB).

The minimum starting value is the current size of low MS-DOS memory. The maximum starting value is B800 (736 kB). The starting and ending values both must be a multiple of 4 kB (low-order hex digits must be 00).

The default values for the display adapters recognized by 386^{MAX} are as follows:

vidmem=a000-c000 (EGA and VGA)
vidmem=b800-c000 (CGA)
vidmem=b000-c000 (Mono)

If the memory range of the display adapter in your system matches one of the above ranges, you do not need to use the **vidmem** option. If your system has a different adapter, and you encounter difficulty with the display of graphics data, particularly in high-resolution modes, you may need to use this option. In that case, consult the manual for your display adapter to find the lowest starting and highest ending addresses of its memory.

Note that the starting and ending addresses in the previous list are conservative to reduce conflicts. If you are certain of the full memory range of the display adapter in your system, and it is narrower than the above values, using **vidmem** can increase either or both of the available low and high MS-DOS memory. For example, a true MDA has memory at B000-B100, and a true CGA has memory at B800-C000.

Using VIDMEM with GRiD Laptop Computers

If you have a GRiD laptop computer, by using this option and also omitting the **nolow** option, you can increase the size of conventional MS-DOS memory to 736 kB.

NOTE: This option can be used only if you have no application ROMs and are not using the VGA expansion cartridge or built-in VGA display. A VGA computer is unable to increase base RAM above 640 kB since VGA memory starts just above 640 kB.

To use this option on a GRiD laptop computer with no application ROMs and no VGA display, specify **vidmem=b800-c000**. This causes the previously reserved video memory at B000-B800 to be released and backfilled for use as conventional MS-DOS memory.

VGA, EGA, CGA, MONO

These options provide a simplified way of specifying special video regions previously set aside by the **vidmem** option. These options have the same effect as using a **vidmem** statement and, without exception, can be used to replace the corresponding **vidmem** value. The regions specified by each option are as follows:

vga or ega	vidmem=a000-c000
cga	vidmem=b800-c000
mono	vidmem=b000-c000

AMRS=*nnn*

This option can be used to improve the performance of multitasking programs. The value *nnn* specifies the number of Alternate Map Register Sets (AMRS) that are used by multitasking programs to switch between your applications rapidly. The number of AMRS may range from 0 to 255. The default value is zero. If you use a multitasking program, specify a value *nnn* equal to the number of programs you wish to run at the same time. Each AMRS consumes 4 kB of extended memory.

DMA=*nnn*

The 386^{MAX} program uses a separate buffer to speed up DMA transfers. The size of this buffer is ordinarily 64 kB if your hard disk uses DMA; 16 kB otherwise. Under certain circumstances, this buffer might be too small, in which case the following error message is displayed on the screen:

DMA transfer limit exceeded: use DMA=*nnn*

This option allows you to specify the size of the DMA transfer buffer. The value *nnn* is in units of kilobytes and can range from 8 to 128. For GRiD laptop computers, if the error occurs, enter the value **dma=20**.

Advanced Examples

Following are examples of the use of advanced options for both GRiD desktop computers and GRiD laptop computers.

GRiD Desktop Examples

1. In case your EGA ROM can't be cached, but the system BIOS can be, use the following configuration:

```
device=d:\path\386max.sys rom=f000-10000
```

2. In case the memory in your high-resolution graphics adapter (at, say, C700-CA00) is not enabled when 386^{MAX} is installed, use the following configuration:

```
device=d:\path\386max.sys ram=c700-ca00
```

3. In case your system has a ROM that is decoded at multiple locations (say, C800-CC00 and D800-DC00), and only the first location is valid, use the following configuration:

```
device=d:\path\386max.sys use=d800-dc00
```

4. If your system has a true CGA, you can recover 96 kB of low MS-DOS memory as follows:

```
device=d:\path\386max.sys vidmem=b800-c000
```

5. To increase the amount of available EMS memory, you can disable memory fill below and above the display adapters as follows:

```
device=d:\path\386max.sys nolow nohigh
```

6. To maximize the amount of available EMS memory, you can disable memory fill as well as disable swapping ROM to faster memory as follows:

```
device=d:\path\386max.sys nolow nohigh norom
```

7. To increase the amount of EMS swapping memory for multitasking systems, include memory above the display adapters as follows:

```
device=d:\path\386max.sys include=128
```

GRiD Laptop Examples

1. The basic configuration that works for any GRiD laptop computer is as follows:

```
device=d:\path\386max.sys norom nolow
```

Assuming you have 2 MB of RAM in your system, this configuration provides 640 kB of standard memory and the rest as EMS memory. The ROM BIOS is not swapped into extended memory, and low MS-DOS memory is not backfilled to increase the amount of standard memory available to MS-DOS.

2. If you do not have application ROMs or a VGA display installed with your GRiD laptop computer, and you want to increase the amount of standard memory available to MS-DOS, you can use the configuration shown below. Additionally, the ROM BIOS is swapped into extended memory to speed it up.

```
device=d:\path\386max.sys
```

3. If your system is not using a VGA display and does not contain application ROMs, you can recover an additional 96 kB of low MS-DOS memory (for a total of 736 kB of MS-DOS memory) as follows:

```
device=d:\path\386max.sys norom vidmem=b800-c000
```

4. To increase the amount of EMS swapping memory for multitasking systems, include memory above the display adapters as follows:

```
device=d:\path\386max.sys norom include=128
```

5. To maximize the amount of EMS swapping memory for multitasking systems, include memory above the display adapters and the unused video memory as follows:

```
device=d:\path\386max.sys norom include=a000-b800, 128
```

CHAPTER 6: 386DISK — A VIRTUAL DISK FOR 386^{MAX}

The 386DISK program simulates a disk drive on your system in a manner similar to the DOS VDISK program. It can place the data of the simulated disk in either EMS or extended memory. This virtual disk is the only one supported in conjunction with 386^{MAX}, although others will work.

A virtual disk must be installed in your *config.sys* file. It can be installed multiple times to simulate multiple disks, each identified by a separate disk drive letter.

The syntax for installing 386DISK in the *config.sys* file is shown below.

```
device=d:\path\386DISK.SYS disksize sectorsize  
    diretries option
```

where:

- *disksize* is the size of the virtual disk in kilobytes. The default is 64 kB.
- *sectorsize* is the desired number of bytes in each disk sector. The default is 128 bytes. The minimum sector size is 128 bytes; other possible values are 256, 512, 1024, and 2048.
- *diretries* is the desired number of directory entries in the root directory. The default is 64.
- *option* specifies the medium in which the disk data is stored. Use */ems* for expanded memory, */ext* for extended memory, or */xms* to use extended memory through the XMS driver. The default is */ems*.

To ensure that the system can address the entire disk, the sector size may be adjusted upwards. The number of directory entries may be adjusted upwards to fill an entire sector. The disk size may be adjusted downwards in case there is not enough memory available in the chosen storage medium.

In contrast to 386DISK, VDISK contains another parameter to its /e or /x switch, called transfer size, which attempts to ensure that interrupts are not disabled for too long a time. Because 386DISK in conjunction with 386^{MAX} does not have this problem, this transfer size parameter is unnecessary.

Following is an example of the use of 386DISK:

```
device=c:\386DISK.SYS 1024 512 200 /xms
```

This example installs a 1 MB (1024 kB) disk with 512-byte sectors, and 200 root directory entries in extended memory using the XMS driver.

NOTE: See Appendix C for messages that may be generated by the 386DISK program.

CHAPTER 7: COMMON QUESTIONS

This section answers commonly asked questions about 386^{MAX}. For answers to questions not covered here, see the *readme.max* file on the distribution diskette.

Which versions of Windows does 386^{MAX} work with?

Windows/386 runs in the protected mode of the 80386 processor and does not allow any another program to enter that mode. To run Windows /386 with 386^{MAX} in the system, be sure to disable 386^{MAX} first. Refer to Enabling And Disabling 386^{MAX}, in Chapter 4, for more details.

Windows 3.0 only works with 386^{MAX} in real mode. Since 386^{MAX} runs in protected mode, it prevents Windows 3.0 from using either standard or enhanced mode because both of those modes require the protected mode of the chip.

Windows/286 also works with 386^{MAX}.

The 386^{MAX} program does support the Virtual Control Program Interface (VCPI) specification, which allows 386^{MAX} to coexist with certain 386 protected mode programs such as Paradox/386 from Borland and Interleaf Publisher from IBM. Windows/386, however, is not currently supported.

Why does CHKDSK not report more available memory after installing 386^{MAX}?

The `chkdsk` command reports contiguous low MS-DOS memory only. So even if 386^{MAX} has created additional high MS-DOS memory, only the `mapmem` option of *386max.com* reports on that memory.

Why does the system crash when I request MS-DOS 4.0 to use EMS memory?

If GRiD MS-DOS 4.0 detects an EMS manager in the system, and you request MS-DOS to use EMS memory, it reserves two EMS mappable physical pages for its exclusive use. However, MS-DOS 4.0 doesn't properly identify the pages it reserves. If you use the default settings for 386^{MAX}, MS-DOS "steals" the top 32 kB of system board memory. This error, along with several other unexplained errors, can cause the system to crash. (This applies to both the June 17 and August 3, 1988 releases.)

Therefore, we do not recommend that MS-DOS 4.0 use EMS memory. All other EMS applications appear to work correctly. In particular, the following commands and switches enable MS-DOS's use of EMS memory:

```
buffers=nn /X
fastopen /X
device=vdisk.sys /X
```

We recommend that you do not use the /X switch with any of these commands. Each of these commands appears to work correctly without the switch. Following are more details on these problems.

BUFFERS

The **buffers=** command in the *config.sys* file can be told to use EMS memory for its scratch storage with the /X switch. When this is in effect, MS-DOS uses the second of the two reserved pages to swap buffer memory. When disk activity occurs, MS-DOS maps its buffer memory into the last page (the top 16 kB of system board memory), checks for a match in the buffers, and then returns to its caller **without restoring the previous mapping**. This sets to zero the top 16 kB of system board memory. This situation occurs first when loading *command.com* before the *autoexec.bat* file is processed. The system locks up shortly thereafter.

To overcome this problem, define at least two EMS mappable pages above the display adapter. If your system has a monochrome adapter, use **include=32**; otherwise, use **include=b000-b800**.

FASTOPEN

The **fastopen** command also uses EMS memory when the /X switch is specified. At least one bug in the program causes it to crash. **Fastopen** can be patched to

overcome this problem, by even then, because of another bug, it still will not use EMS memory if eight or more EMS mappable pages are defined (other than the two reserved pages). Instead, **fastopen** reports "Expanded memory not available."

Normally, 386^{MAX} defines all but the first 64 kB of conventional memory as EMS mappable to support multitasking programs. To reduce the EMS mappable pages for 386^{MAX}, and thus allow **fastopen** to use EMS memory, specify **exclude=1000-a000**. This option eliminates all EMS mappable conventional memory. As a result, multitasking programs such as Windows, Omniview, and DESQview will not multitask.

Although using the /X switch is not recommended with **fastopen**, the following steps can be taken to patch **fastopen** to avoid crashing the system when /X is specified. In conjunction with using **exclude=1000-a000**, this patch allows **fastopen** to use EMS memory, but multitasking programs will not be supported.

1. Backup the original file:

```
copy fastopen.exe *.old
```

2. Patch the file:

```
copy fastopen.exe f
```

```
debug f
```

```
-u 2C65 15
```

```
xxx:2C65 B80058      MOV  AX,5800
```

```
xxx:2C68 CD67      INT  67
```

```
-e 2C66 1
```

```
-u 2C65 15
```

```
xxx:2C65 B80158      MOV  AX,5801
```

```
xxx:2C68 CD67      INT  67
```

```
-w
```

```
Writing 3FAE bytes
```

```
-q
```

3. Copy the file back to its original name:

```
copy f fastopen.exe
```

Why does my network adapter no longer work?

When 386^{MAX} is installed, it searches around for high MS-DOS memory already in use. Some network adapters (as well as high-resolution graphics adapters) do not enable their memory until the adapter is used. If 386^{MAX} does not find the memory in use, it maps in some from extended memory so you can relocate memory-resident programs. To get your adapter working, use the **ram** option described in Chapter 5. This option tells 386^{MAX} what addresses your adapter is using and not to map high MS-DOS memory into those locations.

Why does the command 386max list report some EMS memory in use although no EMS programs are resident?

This behavior is normal. As part of the EMS 4.0 specification to allow multitasking environments to swap programs in and out, some or all of conventional memory is placed into the EMS memory pool. This memory is then allocated to a special Operating System Environment handle and mapped into the addresses of conventional memory.

How can I get more memory available for DESQview 2.01?

When DESQview is loaded into memory, it looks around in high MS-DOS memory for additional mappable EMS pages, which it uses to relocate parts of itself out of conventional memory. When 386^{MAX} is installed, it assigns all of high MS-DOS memory to relocating memory-resident programs. Thus, you must modify the default behavior of 386^{MAX} to use high MS-DOS memory differently via the **include** option as described in Chapter 5. The more included pages in high MS-DOS memory available to DESQview, the more conventional memory is available to run your programs under DESQview. A typical setting for this option is

```
include=96
```

which assumes that you have a VGA monitor. Remember that specifying additional included pages reduces the amount of memory available to relocate memory-resident programs.

For GRiD laptop computers, a typical setting for this option is

```
include=a000-b800, 128
```

which assumes that you do not have a built-in VGA display, a VGA expansion cartridge, or any application ROMs.

Does 386^{MAX} work on GRiD computers with 80286 processors?

No. The 386^{MAX} program makes great use of capabilities first introduced in the 80386 processor. In particular, backfilling, memory swapping, and EMS support are all implemented using the page remapping feature in the 80386 and 80386SX. Such features are not available in the 80286 processor.

Why does high MS-DOS memory appear to be split into several pieces?

Quite likely, there is an adapter board in your system that contains some ROM and/or RAM. The addresses occupied by the ROM/RAM are the middle of the high MS-DOS memory area. The 386^{MAX} program skips over these addresses, but the adapter board still has fragmented high MS-DOS memory. Sometimes the adapter board can be removed, or the addresses used by the ROM/RAM can be recovered by telling 386^{MAX} to use them via the `use` option. Typical addresses that you might find in use are C8000-CA000 for a hard disk controller ROM, CE000-D0000 for a Bernoulli Box ROM, and D0000-E0000 for some high-resolution graphics adapters RAM.

How do I install the Adaptec software driver with 386^{MAX}?

Adaptec ships a software driver that must be loaded in the `config.sys` file before the `386max.sys` device line. Copy the file `scsiha.sys` onto your boot disk. Insert the line

```
device=d:\path\scsiha.sys /v386 /b16
```

in your `config.sys` file **before** the `386max.sys` line.

How do I install 386^{MAX} with an EMS adapter already in the system?

To gain the maximum out of 386^{MAX}, we recommend that you convert all of the memory on the EMS adapter to extended memory. That way, 386^{MAX} can manage the card's memory in the same way as it manages the system board memory. Consult the manual for the EMS adapter to convert its EMS memory to extended memory. If you wish to leave the adapter as is, be sure to install the adapter's EMS memory manager **before** installing 386^{MAX}, and put `ems=0` on the device driver line for 386^{MAX}.

Why does the TIMEMEM display show very slow memory in the A0000-C0000 region?

Those memory addresses correspond to video RAM, which is notoriously slow. Because that RAM is specially coupled to the video hardware, there is no way 386^{MAX} can speed it up.

Why is there still a small amount of available extended memory even though I asked that all extended memory be converted to EMS memory?

After room is made for 386^{MAX} program storage and low MS-DOS memory backfilling, EMS memory is allocated in blocks of 16 kB. This may leave a small amount of memory (4, 8, or 12 kB) unallocated, which is left as extended memory.

How do I use the Western Digital FASST SCSI controller with 386^{MAX} ?

This controller ships with a software driver called `sstbios.sys`. Copy this device driver to your boot disk. Insert the line

```
device=d:\path\sstbios.sys /we
```

in your `config.sys` file **before** the `386max.sys` line.

What causes the error message “Packed file is corrupt”?

If you see this message, then you must marginally increase the amount of conventional memory used by DOS. This step is necessary to work around a known bug in the program used to pack the file in question. Determine the current starting address of available conventional memory using the **386max mapmem** display. Increase the **files** or **buffers** statement in *config.sys* until the starting entry in the last row in low MS-DOS memory is greater than or equal to 0FF0h.

Why don't Supercalc 5 and Superproject work properly with 386^{MAX} ?

These programs may not function properly if 386^{MAX} backfills or recovers the monochrome display area. It may also be necessary to force 386^{MAX} not to mark the first 640 kB as EMS mappable as well. Therefore, do not use the options **include=b000-b800** or **use=b000-b800**.

You may have to use the option **exclude=1000-a000**. You will also have to stop 386^{MAX} from backfilling (adding low MS-DOS memory) using the **nolow** option.

How can I get Paradox 3.0 to work with 386^{MAX}?

Because of a bug in this program, be sure that 386^{MAX} does not backfill low MS-DOS memory above 640 kB. In particular, you may need to add the **nolow** option to the *386max.sys* line in your *config.sys* file.

Does the Weitek coprocessor work with 386^{MAX}?

Yes. The mechanism 386^{MAX} uses to detect the presence of a Weitek coprocessor is to ask the BIOS via the equipment check interrupt (INT 11h). Unfortunately, some BIOS do not adhere to this published standard, so the option **weitek=force** on the *386max.sys* line or profile is used to force 386^{MAX} to recognize the chip. Use this option only if you have a Weitek coprocessor in your system and 386^{MAX} doesn't seem to recognize it. If you also want to enable the Weitek coprocessor (with **weitek=on**) when 386^{MAX} initializes, be sure that the forcing command precedes the enabling command.

Why does my Ventura Publisher program fail when 386^{MAX} is installed?

When used with the Professional Extension, Ventura can fail if the EMS page frame is above E000. Among other ways, the page frame can be placed this high if the system ROM has been compressed (see NOCOMPROM in Chapter 5). To overcome this problem, force the EMS page frame to E000 or lower with an explicit **frame=** command.

Do I need to install 386^{MAX} differently for a GRiD laptop computer?

For non-VGA GRiD laptop computers, you should specify the **nowlow** option on the device driver line in your *config.sys* file. By specifying this option, 386^{MAX} will not fill unused low memory for use by MS-DOS.

Why does Word Perfect 5.1 hang when 386^{MAX} is installed?

Word Perfect attempts to use an advanced feature of LIM 4.0 that 386^{MAX} cannot support since the feature must be implemented in hardware. Word Perfect fails to properly handle the return code indicating that this feature is not supported, and hangs the computer. On Word Perfect Versions 5.1 and subsequent, there is a */32* command line option to avoid this problem.

APPENDIX A: OPTION SUMMARY

This appendix summarizes all 386^{MAX} device driver options, in alphabetical order. These options are described in Chapters 3 and 5.

These options may appear on the `device=d:\path\386max.sys` line in your `config.sys` file, which invokes 386^{MAX}.

- **amrs=nnn** — Specify *nnn* Alternate Map Register Sets (AMRS) that are used by multitasking programs to switch between applications rapidly.
- **auto** — The 386^{MAX} program automatically turns EMS support on or off as needed. Low and high MS-DOS memory is unavailable in this mode.
- **cga** — Same as specifying **vidmem=b800-c000**.
- **dma=nnn** — Specify the size, *nnn*, of the DMA transfer buffer.
- **ega** — Same as specifying **vidmem=a000-c000**.
- **ems=nnnn** — Reserve *nnnn* kilobytes of extended memory as EMS memory.
- **exclude=xxxx-yyyy** — Mark paragraphs *xxxx* to *yyyy* as not available for EMS memory swapping.
- **ext=nnnn** — Reserve *nnnn* kilobytes of extended memory as extended memory.
- **frame=xxxx** — Use paragraph *xxxx* as the EMS page frame.
- **hmamin=nn** — Specify the minimum amount, *nnn*, of High Memory Area (HMA) that a program may request under the Extended Memory Specification (XMS).

- **include=xxxx-yyyy** — Mark paragraphs *xxxx* to *yyyy* as available for EMS memory swapping.
- **include=nnn** — Mark *nnn* kilobytes of high MS-DOS memory as available for EMS memory swapping.
- **mono** — Same as specifying **vidmem=b000-c000**.
- **nocomprom** — Do not automatically compress system BIOS to smaller size.
- **nohigh** — Do not topfill memory above the display adapters.
- **nolow** — Do not backfill memory below the display adapter for use as conventional memory.
- **noframe** — The 386^{MAX} program installs without an EMS page frame.
- **norom** — Do not swap fast RAM for slow ROMs or swap ROM BIOS into extended memory.
- **off** — EMS support and ROM caching are not available until turned back on.
- **on** — EMS support and ROM caching are available.
- **prgreg=x** — Specify region of high DOS memory to store 386^{MAX} device driver.
- **pro=d:\path\filename.ext** — Read in options to the **device=** line from this profile.
- **ram=xxxx-yyyy** — Mark paragraphs *xxxx* to *yyyy* as RAM.
- **resetkeyb** — When leaving or entering protected mode, reset the keyboard controller.
- **rom=xxxx-yyyy** — Mark paragraphs *xxxx* to *yyyy* as ROM and cache if slow.

- **shadowram** — Remap shadowram as extended memory or use for EMS.
- **shadowrom** — Retain BIOS technique of shadowing ROM in E000 block.
- **swap=nnn** — Swap *nnn* kilobytes of fast for slow memory.
- **use=xxxx-yyyy** — Mark paragraphs *xxxx* to *yyyy* as available for high MS-DOS memory use.
- **vga** — Same as specifying **vidmem=a000-c000**.
- **vidmem=xxxx-yyyy** — Mark paragraphs *xxxx* to *yyyy* as video memory. Other unused video memory is then released for use as conventional memory.
- **weitek=force** — An option used on the *386max.sys* line or profile to force 386^{MAX} to recognize the Weitek coprocessor.
- **weitek=on** — Enables the Weitek coprocessor after it has been recognized by using the **force** command.
- **xmshndl=nnn** — Overrides the default number of handles (32) for the Extended Memory Specification (XMS).

APPENDIX B: UTILITY PROGRAM SUMMARY

This section summarizes the command options available for the *386max.com* utility program. These options are detailed in Chapter 4.

Each command must be preceded by **386max** when entered at the MS-DOS command line. For example: **386max list**.

Note that multiple command options can be entered on the same line to display multiple tables. For example, to display all tables, use **386max /e/l/m/r/t**.

- **auto** — The 386^{MAX} program automatically turns EMS support on or off as needed for EMS functions only.
- **closehigh** — Close off high MS-DOS memory from MS-DOS program memory allocations.
- **list or /l** — Display table of extended memory usage.
- **loadhigh or high** — Begin loading programs in high MS-DOS memory.
- **loadlow or low** — Begin loading programs in low MS-DOS memory.
- **mapems or /e** — Display table of EMS memory usage.
- **mapmem or /m** — Display table of memory-resident program usage.
- **off** — EMS support and ROM caching are not available until turned back on.
- **on** — EMS support and ROM caching are available.
- **openhigh** — Open up high MS-DOS memory for MS-DOS program memory allocations.

- **romscan** or **/r** — Display table of ROM locations.
- **timemem** or **/t** — Display table of memory access times.

APPENDIX C: MESSAGES

For your convenience, all informational and error messages issued by 386^{MAX} and 386DISK are described in this appendix.

Installation Messages

This section contains error messages that you may receive when your computer starts up and attempts to install 386^{MAX}.

Alt-key pressed

What happened: You pressed the **Alt** key to abort the 386^{MAX} installation. This feature is useful in case you run into trouble during installation and do not have a DOS disk from which you can reboot the system.

What to do: The system will continue to reboot until the MS-DOS prompt is displayed. Then retry installation.

Installing EMS support with short page frame.

What happened: No contiguous 64 kB block exists above the display adapter for the EMS frame. The 386^{MAX} program has installed with an EMS page frame smaller than the normal 64 kB.

What to do: No action necessary. Some EMS 4.0 programs (such as Windows 2.0) can still use EMS memory even without a full page frame. If this is not satisfactory, either remove the extra RAM or ROM in high MS-DOS memory or specify **ems=0** and do not use EMS memory.

Insufficient Extended Memory Available
Allocate Less extended memory to other programs.

- What happened: The **ems=** or **ext=** values exceed the available extended memory or low MS-DOS memory fill.
- What to do: Be sure to use the **norom** option to reduce the memory requirements. Additionally, try using the **nowow** option.

Memory manager NOT installed.
Press any key to continue . . .

- What happened: An error (such as one of those in this list) has occurred. The 386^{MAX} program terminates without installing itself.
- What to do: Edit the *config.sys* file to fix the error, and restart your computer.

No page frame available.

- What happened: No room is available in the address space above the display adapter for even a short EMS page frame.
- What to do: Reconfigure your hardware to make room for an EMS page frame, or turn off EMS usage with **ems=0**.

Preceding EMS manager found.

- What happened: Only one EMS manager can be in the system at a time and a preceding one was found.
- What to do: Remove the preceding EMS manager or disable the EMS manager in 386^{MAX} with the option **ems=0**.

To give DOS more memory, we're rebooting the system . . .

What happened: In rare circumstances, the device driver may need to reboot the system to advise MS-DOS of additional low memory being added to MS-DOS. Ordinarily, this message will not occur even if 386^{MAX} adds low memory to DOS.

What to do: No action needed.

Unable to enter protected mode -- error code xx.

What happened: The system BIOS failed to handle the call to enter protected mode. An error code of 80 or 86 indicates that the BIOS doesn't support this function. The 386^{MAX} program terminates without installing itself.

What to do: Contact the GRiD Resource Center at 1-800-284-GRID (4743).

Unable to move code to extended memory -- error code xx.

What happened: The system BIOS failed to move the 386^{MAX} program to extended memory. The error code indicates the type of problem:

- 01 - a parity error occurred while entering protected mode
- 02 - an exception interrupt occurred
- 03 - gate address line A20 failed
- 07 - a parity error occurred after entering protected mode
- 08 - memory to be used for 386^{MAX} is not present or not stable

These errors generally indicate a memory chip or system board failure and not errors in 386^{MAX}. Your computer's power-on self-test routines may not even find anything wrong with its memory. An error code of 80 or 86 indicates that the BIOS doesn't support this function. The 386^{MAX} program terminates without installing itself.

What to do: Contact the GRiD Resource Center at 1-800-284-GRID (4743).

Unable to time extended memory -- error code xx.

What happened: While attempting to determine the locations of fast and slow memory, 386^{MAX} ran into a problem. The error code indicates the type of problem:

- 01 - a parity error occurred while entering protected mode
- 02 - an exception interrupt occurred
- 03 - a gate of address line A20 failed
- 07 - a parity error occurred after entering protected mode
- 08 - memory to be used for 386^{MAX} is not present or not stable

These errors all point to hardware problems in your computer. Your computer's power-on self-test routines may not even find anything wrong with its memory. The 386^{MAX} program terminates without installing itself.

What to do: Contact your local GRiD Systems representative or the GRiD Resource Center at 1-800-284-GRID (4743).

Virtual machine already in use.

What happened: A preceding device driver is already operating in Virtual 8086 Mode.

What to do: Since 386^{MAX} also runs in Virtual 8086 Mode, only one such driver can be active at a time. Remove one of the device drivers.

Wrong machine type.

Must run on an 80386.

What happened: The 386^{MAX} program runs only on systems using an 80386 processor. It terminates without installing itself.

What to do: You can't run 386^{MAX} on this computer.

Failure of hard disk transfer into non-linear memory.

What happened: During initialization of 386^{MAX}, the program attempts to detect a SCSI hard disk bus master. If one is found, this message is displayed and processing continues. Subsequently, you may find that multitasking programs, among others, may fail to work.

What to do: If this happens, contact your SCSI hard disk vendor for a solution. In particular, you will need to ask them to provide you with a device driver or ROM that either supports the Virtual DMA Services (VDS) specification or performs double buffering. For this feature to work, there must be either an EMS page frame or some high MS-DOS memory.

Device Driver Option Errors

This section contains error messages that you can receive because you specified invalid options on the device driver installation line.

DMA transfer limit exceeded: use DMA=nnn

What happened: The DMA transfer buffer is too small.

What to do: Reboot your system from a floppy disk, and edit the *config.sys* file to include the DMA value shown in the message on the *386max.sys* line. Refer to Chapter 5.

EMS= and EXT= both specified -- use at most one of them.

What happened: Both the *ems=* and *ext=* device driver options were specified.

What to do: Edit the *config.sys* file to remove one of the device driver options. For more information, refer to Chapter 3.

EMS= size must be a multiple of 16KB -- rounding down.

What happened: The `ems=` value was not a multiple of 16.

What to do: The 386^{MAX} program automatically rounds down to the nearest 16 kB multiple. You can edit the `config.sys` file to specify a value that is a multiple of 16.

FRAME value must be on a 16KB boundary.

What happened: The `frame=` value is not on a 16 kB boundary (the possible values are `x000`, `x400`, `x800`, and `xC00` where `x` is a hex digit that refers to a segment in high memory).

What to do: Edit the line to specify another value and restart the system.

FRAME value overlaps accessory hardware.

What happened: The EMS page frame starting at the `frame=` value overlaps some RAM or ROM that is above the display adapter.

What to do: Edit the `config.sys` file to specify another value and restart the system.

Invalid address or length.

What happened: A device driver option such as `include` has an invalid address or length parameter.

What to do: Refer to Chapter 5 for more information about the option's limits. Edit the option in the `config.sys` file and restart the system.

Invalid line ending in profile

What happened: A line in the profile does not end with zero or more blanks followed by a CR, LF, or semicolon.

What to do: Edit the profile and restart the system.

Missing separator.

What happened: An option such as **ext=1024** requires an equal (=) sign to separate the keyword and value, or an option such as **include=xxx-yyy** requires a hyphen to separate the two parts of the address.

What to do: Edit the *config.sys* file to correct the problem and restart the system.

Unable to access profile.

Whap happened: The profile specified by **pro=** on the **device=** line in your *config.sys* file cannot be found or read.

What to do: Be sure to specify the full drive and path of the file. Edit your *config.sys* file to specify another value and restart the system. If this file resides on removable media (such as a floppy disk or disk cartridge), do not remove the disk while 386^{MAX} is being installed. This message also can occur if some disk error is encountered while trying to read the file. Try typing the file to the screen from the MS-DOS command line to verify that the disk sectors that contain the file are valid. Restart the system after correcting the problem.

Unknown keyword in profile.

What happened: The first non-blank keyword on some line in the profile cannot be deciphered.

What to do: Refer to the list of valid keywords in Chapters 3 and 5. Edit the profile to correct the problem and restart the system.

Unknown keyword on **DEVICE=** line in **CONFIG.SYS**.

What happened: A keyword on the **device=** line in your *config.sys* file is not valid.

What to do: Refer to the list of valid keywords in Chapters 3 and 5. Edit the *config.sys* file to correct the problem and restart the system.

Critical Error Messages

This section contains critical error messages that indicate a serious system error. You will normally need to restart your system to continue.

A stack fault has occurred at address *xxxx:yyyy*.

What happened: This error might occur under normal circumstances when a program attempts to place data onto the stack, or a hardware interrupt (such as a timer tick) occurs when there is insufficient room on the stack.

What to do: This message results from a bug in application code, not a bug in 386^{MAX}. The address mentioned in the message may be of help in finding the faulty code.

A Privileged Operation Exception has occurred at address *xxxx:yyyy*.

What happened: This error might occur under normal circumstances when a program attempts to execute a privileged operation such as entering protected mode, or to read, write, or execute off the end of a code or data segment. This error is caused by an application program bug or an attempt to use resources to which it does not have access when running in Virtual 8086 Mode.

What to do: If this error persists, invoke 386^{MAX} in automatic mode (using the **auto** option), or disable 386^{MAX} using the **off** option before executing the program that generated the above message.

- A double fault has occurred**
- An invalid TSS fault has occurred**
- A segment not present fault has occurred**
- A page fault has occurred**
- An internal system error has occurred**
- A DMA page fault has occurred**
- A TSS page fault has occurred**
- A page not present fault has occurred**
- A MAPTAB fault has occurred**

Each of the above messages is followed by

at address *xxxx:yyyy*.

Press any key to restart your computer.

- What happened:** These error messages are issued when an internal error occurs within 386^{MAX}. If the internal error is severe enough, pressing a key might have no effect. Moreover, it might not be possible to reboot the system using **Ctrl-Alt-Del**.
- What to do:** Contact your local GRiD Systems representative or the GRiD Resource Center at 1-800-284-GRID (4743) if you encounter any of these error messages. Be sure to have the description of what happened, the address mentioned in the error message, and the version number of 386^{MAX} that you are using.

Utility Program Error Messages

This section contains messages that you may receive as a result of using the utility program *386max.com*.

Extended Memory in Use - - Unable to change to ON or AUTO state

What happened: You started up 386^{MAX} in the **off** state, and a resident extended memory program was installed that allocated memory from the top down. You cannot change 386^{MAX} into the **on** or **auto** state.

What to do: Remove the resident program that is using extended memory. To avoid this situation from happening, start up 386^{MAX} in the **on** (default) state, and then disable 386^{MAX} using **386max off** in your *autoexec.bat* file.

386MAX.SYS device driver not loaded from CONFIG.SYS.

What happened: This message occurs when *386max.com* has been called but the device driver was not successfully installed. Perhaps an error on the command line of *386max.sys* caused it to fail to be installed.

What to do: Review any messages that might have been displayed when the device driver was invoked during processing of your *config.sys* file. Edit this file to fix the error and restart your computer.

Expanded memory in use -- unable to turn OFF.

What happened: An attempt was made to disable 386^{MAX} while some program was still using EMS memory. This also could happen if you specify the **off** or **auto** utility options but 386^{MAX} was installed without the **nolow** option.

What to do: Before disabling 386^{MAX}, you must first remove any resident programs using EMS memory. If you installed 386^{MAX} without the **nolow** option, reinstall 386^{MAX} specifying the **nolow** or **auto** option.

Function unavailable while OFF.

What happened: Most of the functions recognized by the utility program are unavailable when 386^{MAX} is disabled.

What to do: Enable 386^{MAX} with **386max on** or **386max auto** and retry the operation.

Incorrect 386MAX version number.

What happened: The *386max.com* and *386max.sys* programs are interrelated to the extent that they require the same version numbers.

What to do: Ensure that both the *386max.sys* and *386max.com* programs are copied to your hard disk.

Loading programs in HIGH memory . . .

What happened: Programs are now being executed in the memory above the display adapter memory.

What to do: No action is needed.

Loading programs in LOW memory . . .

What happened: Programs are now being executed in the memory below the display adapter memory.

What to do: No action is required.

Loading programs in LOW memory after release . . .

What happened: Programs are now being executed in the memory below the display adapter memory. One of the previous programs loaded into high memory terminated without leaving enough room for MS-DOS to execute *command.com*, so *386max.com* automatically shifted to loading programs into low memory.

What to do: No action is required.

Memory allocation chain error.

What happened: An error has occurred while searching through the memory allocation chain.

What to do: Restart your computer. If the error persists, call the GRiD Resource Center at 1-800-284-GRID (4743).

No room in high MS-DOS memory.

What happened: The *386max.com* program was called to load programs into high memory, but there is not enough room to load even *command.com*.

What to do: If you need to load more programs into high MS-DOS memory, review the list of programs already loaded there using the **386max mapmem** command.

No high DOS memory available.

- What happened: Either **nohigh** was specified, or the program could not find any room above the display adapters for high MS-DOS memory. In either case, there is no high MS-DOS memory in which to load programs.
- What to do: No action is required.

Unrecognized parameter on command line -- Valid options are AUTO, CLOSEHIGH, LIST, LOADHIGH, LOADLOW, MAPEMS, MAPEMS, OFF, ON, OPENHIGH, ROMSCAN, or TIMEMEM.

- What happened: The option on the *386max.com* command line is not recognized.
- What to do: Use a valid option. Refer to Chapter 4 for the options supported on GRiD computers.

386DISK Messages**Drive *d*: *nnnn* KB of *xxx* memory, *nnn* bytes per sector, *nnn* directory entries.**

- What happened: The disk is installed successfully as drive letter *d*.
- What to do: No action is required.

Wrong machine type: must run on an 80386.

- What happened: You tried to run this program on a non-80386 computer.
- What to do: This program can only run on a computer with an 80386 processor.

This program requires DOS 3.00 or later.

What happened: You tried to run the program on a computer with the wrong version of DOS.

What to do: Run this program only on a computer with DOS 3.00 or later.

386MAX.SYS device driver not loaded from CONFIG.SYS.

What happened: The program was installed before 386MAX.SYS.

What to do: Install this program **after** *386max.sys* in your *config.sys* file.

Disk size too small.

What happened: There isn't enough room on the specified size disk for the boot sector, file allocation table, directory, and data.

What to do: Increase the size of the disk and try again.

No EMS driver installed.**No XMS driver installed.**

What happened: Storing data in EMS or XMS memory was chosen (or the default was used), but there is no EMS or XMS driver in the system.

What to do: Install the appropriate driver in your system.

Error from EMS driver: errortext
Error from EXT driver: errortext
Error from XMS driver: errortext

What happened: An error occurred while communicating with the EMS or extended memory system. This message is followed by another message with more information about the specific error.

What to do: Use the error message text to determine the problem and correct it.

Invalid switch: use /EMS, /EXT, or /XMS.

What happened: The characters found following a slash were not from the above list.

What to do: Use a valid switch—either */ems*, */ext*, or */xms*—in your *config.sys* file.

Too many command line arguments.

What happened: More than three numeric arguments were found on the command line.

What to do: You can have a maximum of three arguments on the command line in your *config.sys* file.

APPENDIX D: USING 386^{MAX} WITH NETWORKS

Banyan, Novell, and 3COM Networks

This section discusses and gives examples of the memory mapping for resident programs in Banyan, Novell, and 3COM network environments.

The hardware configurations used in the examples below are as follows:

- Desktop computer with 80386 microprocessor
2 MB RAM
3COM Etherlink II network adapter
Etherlink II card with the following settings: I/O = 300,
RAM Address = C800, IRQ = 2
- GRiDCASE 1530 computer
2 MB RAM
MICOM 5210 Ethernet cartridge
Ethernet cartridge with the following settings: I/O = 300,
RAM Address = C800, IRQ = 2

The software configuration is as follows:

- DOS 3.3
386^{MAX} 4.09
3+Share 1.31
Netware/386 3.0
Vines 3.10

Following are examples of how to set up the *autoexec.bat* file in Banyan, Novell, and 3COM network environments so that the appropriate drivers are loaded into high memory.

NOTE: Once you have edited the *autoexec.bat* file, you must restart your system in order for the changes to take place.

Banyan Example

On a Banyan network, the following lines should be included in the *autoexec.bat* file for Vines version 3.2 and an Ethernet cartridge:

```
ban /nc
intr2ban
386max loadhigh
epcbfs
386max loadlow
```

The first command tells the system not to enable the network adapter card. The ethernet cartridge driver is **intr2ban**. The Vines DOS redirector is **epcbfs**, which is being loaded into high memory. Before it is loaded into high memory, conventional memory has about 474 kB, and high MS-DOS memory has about 176 kB available. After the DOS redirector is loaded, conventional memory has about 532 kB, and high MS-DOS memory has about 118 kB.

For Vines version 4.0 and an Etherlink II card, the following lines should be included in the *autoexec.bat* file:

```
ban /nc
3c503ban (or 3c523ban for Etherlink/MC cards)
386max loadhigh
redir4
386max loadlow
```

NOTE: If you are unsure of the proper names for the drivers, load *ban.com* without the **/nc** option and look at the memory map by typing **386max mapmem**.

Novell Example

On a Novell network, the following lines should be included in the *autoexec.bat* file:

```
386max loadhigh
net3
386max loadlow
```

Before **net3** is loaded into high memory, conventional memory has about 530 kB and high MS-DOS memory has about 147 kB. After loading **net3** into high memory, conventional memory has about 570 kB, gaining about 40 kB, and high MS-DOS memory has about 107 kB.

3COM Example

On a 3COM network, the following lines should be included in the *autoexec.bat* file:

```
386max loadhigh
msredir
386max loadlow
```

Before loading the MS Redirector (**msredir**) into high memory, conventional memory has about 410 kB and high MS-DOS memory has about 147 kB available. After loading the MS Redirector into high memory, conventional memory has 459 kB, gaining about 49 kB, and high MS-DOS memory has about 98 kB.

The following examples show the typical contents of an *autoexec.bat* file and a *config.sys* file for operation with a 3COM network.

Autoexec.bat File

```
echo off
path = \3com;\386max;
echo 3+ Workstation Startup batch file - Ver. 1.3.1
\3drivers\runminds mindspro mindsbuf mindseth
      mindsidp mindsspp mindslgl
\3com\netbios
\3com\minses12
\386max\386max HIGH
\3com\msredir /z:5680 /s:5 /l:7 /b:5
\386max\386max LOW
\3com\setname $$3com$$
\3com\prtsc
\3com\mminder
\3com\recvmsg
rem if exist START.BAT Start
```

Config.sys File

```
device=\386max\386max.sys shadowram use=b000-b7ff
      ram=c600-cc00 ems=0
device=\3drivers\pro.sys 12 2 2
device=\3drivers\buf.sys
device=\3drivers\eth503.sys /t:2
device=\3drivers\idp.sys
device=\3drivers\spp.sys
device=\3drivers\lgl.sys
device=\ansi.sys
buffers=20
files=30
lastdrive=j
```

3COM Network Cards

Some 3COM network cards use RAM buffer areas in high MS-DOS memory. Following is a list of the known 3COM cards and their respective requirements in high MS-DOS memory.

3COM ETHERLINK	No buffer RAM.
3COM ETHERLINK PLUS	No buffer RAM.
3COM ETHERLINK II	16 kB of buffer RAM at a dip switch selectable address.
3COM TOKENLINK	16 kB of buffer RAM at a dip switch selectable address.
3COM TOKENLINK PLUS	No buffer RAM.

If your card is listed as using 16 kB of buffer RAM, you need to add a **ram=** option to the *386max.sys* device line in your *config.sys* file. The appropriate range depends upon the starting address determined by the dip switch settings, as shown below.

If your starting address is:	Add to the <i>386max.sys</i> device line:
C000	ram=c000-c400
C400	ram=c400-c800
C800	ram=c800-cc00
CC00	ram=cc00-d000
D000	ram=d000-d400
D400	ram=d400-d800
D800	ram=d800-dc00
DC00	ram=dc00-e000

IBM Token Ring

In order for this network to function properly, you must provide 386^{MAX} with the RAM and ROM locations of the network board by adding a **ram=** option to the *386max.sys* device line. With the default network settings, the appropriate settings are **ram=cc00-ce00, d800-dc00**.

The network RAM and ROM can be configured at different locations and the associated **ram=** statement must reflect any such changes. (The ROM address is jumper switch configurable. The RAM address can be configured with a command line switch to the **DXMC0MOD** driver in the *config.sys* file. See the Token Ring manual for details.)

To maximize the use of high MS-DOS memory, we recommend that the network RAM and ROM be relocated such that they occupy contiguous memory locations, and the **ram=** statement on the *386max.sys* line be adjusted appropriately.

Network Device Names

If your network needs to know about device driver names used by 386^{MAX} (perhaps because the network allows access only to those names entered in some table), at various times we use the following names:

emmxxx0, qmmxxx0, and 386max\$

Etherlink/Plus

If you have configured your Etherlink/Plus network adapter to map its memory into the A000 segment, use the following option:

vidmem=a000-c000

Novell ELS

The program `net$os` runs in protected mode and is not compatible with 386^{MAX}.

Starlan

If you have configured your AT&T Starlan network to map its memory into the A000 segment, use the following option:

```
vidmem=a000-c000
```

If you map its memory into any other segment, use a

```
ram=
```

on that segment to tell 386^{MAX} not to use those addresses.

TOPS

The DMA setting on the card must be disabled. Use the following option on the TOPS device driver invocation:

```
/dma=none
```

GLOSSARY

Conventional Memory

Memory addressed in the first megabyte of address space. MS-DOS typically uses the first 640 kB of conventional memory for user applications. Conventional memory normally refers to RAM only, not ROM.

Expanded Memory

Memory to which access is obtained by programs adhering to the LIM EMS specification (also called EMS memory). Expanded memory is mapped in and out of conventional memory by a software driver such as 386^{MAX}. The amount of EMS memory in your system can be as much as 15.3 MB.

Expanded memory is always RAM. Version 4.0 of EMS allows, and 386^{MAX} supports, mapping not only in a 64 kB window (called the EMS page frame) but also mapping of additional 16 kB blocks in the first 640 kB of conventional memory.

Extended Memory

Memory addressed at 1 MB and above on systems containing an Intel 80286 or 80386 processor. GRiD computers can contain as much as 15.3 MB of extended memory. MS-DOS cannot normally make use of extended memory, except with an EMS emulator such as 386^{MAX}.

High MS-DOS Memory

Conventional memory in the space between where the display memory stops and where the system ROM begins. The amount of this memory may be reduced by other ROM or RAM in the same address range. The process of filling in high MS-DOS memory and advising MS-DOS of its presence is called "topfilling". The 386^{MAX} program can use this memory as additional swapping space when you are using a multitasking operating environment; refer to the **include** option in Chapter 5.

- Kilobyte/Megabyte** Units of measurement for storage; commonly abbreviated to kB and MB, respectively. A kilobyte is 1,024 bytes; a megabyte is 1,024 kB or 1,048,576 bytes.
- Low MS-DOS Memory** Conventional memory in the space between where the normal 640 kB of system RAM stops to where the memory for the display is mapped. The process of filling in low MS-DOS memory and advising MS-DOS of its presence is called "backfilling". The 386^{MAX} program can use this memory to increase the amount of memory available to MS-DOS programs; refer to the **nowlow** option in Chapter 3 and the **vidmem** option in Chapter 5.
- Paragraph** Unit of measurement for storage. A paragraph is 16 bytes.
- RAM** Random Access Memory. Memory that can be read from as well as written to. This type of memory is used throughout your computer system for temporary storage of information.
- ROM** Read Only Memory. Memory whose contents are constant and cannot be changed. The BIOS code is stored in ROM; additionally, you may have other application ROMs installed in your computer. Application ROMs are ROMs that contain application programs or operating systems such as MS-DOS.

GRiD Publications Customer Response Card

GRiD Systems wants to supply you with the finest possible guides and manuals. Therefore, your feedback is important. To help us maintain the highest quality, please use this form to comment on the usefulness and readability of this publication. Suggest additions and deletions, and list specific problems, if any. Be sure to include page numbers where applicable.

386^{MAX}™ User's Guide

June 1990

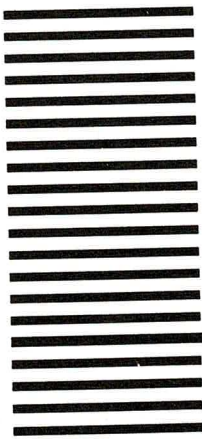
Thank you for your cooperation!
GRiD Systems Corporation



GRID Systems Corporation
 Attention: Publications Manager
 P.O. Box 5003
 Fremont CA 94537-9986

POSTAGE WILL BE PAID BY ADDRESSEE

BUSINESS REPLY MAIL
 FIRST CLASS PERMIT NO. 161 FREMONT, CA USA



NO POSTAGE
 NECESSARY
 IF MAILED
 IN THE
 UNITED STATES



fold here

tape here

GRiD Systems Corporation
47211 Lakeview Boulevard, Fremont, CA 94538-6599