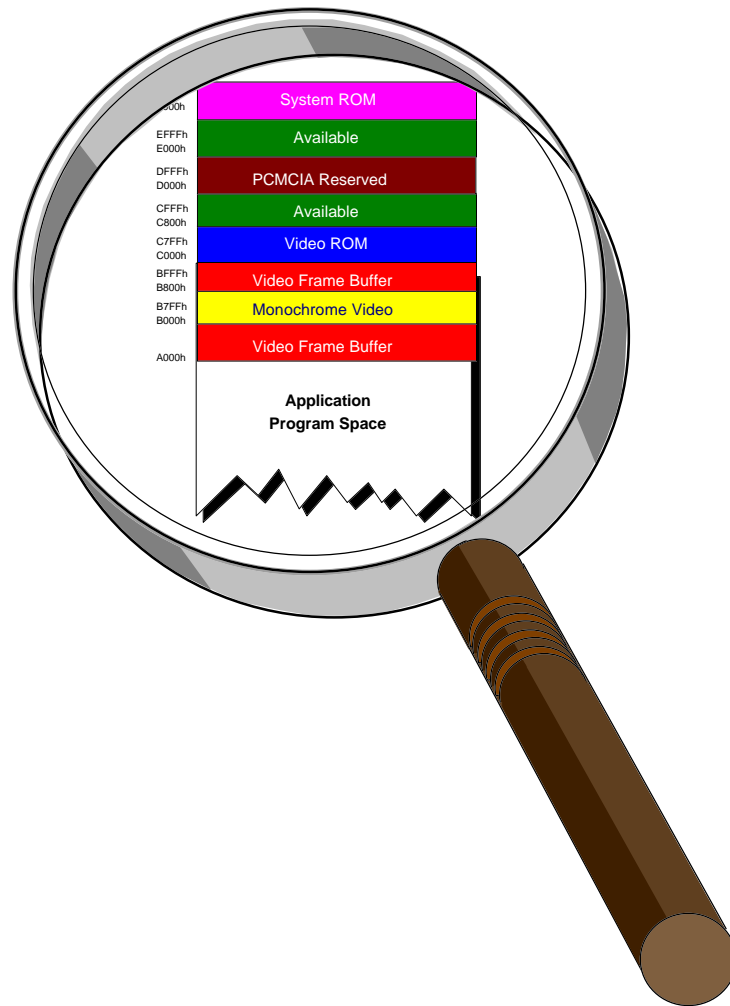




***PCMCIA Memory Optimization White Paper***



***How To . . .***  
*Assess, Recover, and*  
*Optimize Memory in*  
*Compaq PCMCIA-based PCs*

**NOTICE**

**Important:** The information in this guide is subject to change without notice.

COMPAQ COMPUTER CORPORATION SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN; NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL.

©1994 Compaq Computer Corporation.

All rights reserved. Printed in the U.S.A.

Compaq, Deskpro, Systempro, LTE, Fastart

Registered United States Patent and Trademark Office.

QVision, ProLinea, ProSignia, PageMarq, Contura, Concerto, LTE Elite, LTE 5000 are trademarks of Compaq Computer Corporation.

The software described in this guide is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

***Compaq PCMCIA Memory Optimization White Paper***

*Version 1.2*

**Compaq Computer Corporation**

**February 1996**

## Table of Contents

### **SECTION I** - Includes Compaq Contura, Compaq Concerto, Compaq LTE Elite, Compaq Aero families

<b>INTRODUCTION</b>	<b>6</b>
<b>INTRODUCTION</b>	<b>6</b>
<b>INTENDED AUDIENCE</b>	<b>6</b>
<b>OBJECTIVES</b>	<b>7</b>
<b>TERMS AND CONCEPTS</b>	<b>8</b>
<b>MEMORY CONFIGURATION TERMS AND CONCEPTS</b>	<b>8</b>
<b>PCMCIA TERMS AND CONCEPTS</b>	<b>9</b>
PCMCIA COMPONENTS	10
MS-DOS PCMCIA Services and Drivers	10
MS-Windows Drivers	11
<b>ATTAINING THE MOST BASE MEMORY</b>	<b>12</b>
<b>TACKLING THE MEMORY DEFICIT</b>	<b>14</b>
REMOVE UNNECESSARY DRIVERS	15
RECOVER UNUSED MEMORY AREAS	17
UTILIZE PCMCIA RESERVED WINDOW	17
CHANGE DRIVER LOADING ORDER	19
<b>SPECIFIC EXAMPLE</b>	<b>20</b>
<b>IBM TOKEN RING CARD</b>	<b>20</b>
RUNNING PCCARD	21
CONFIGURING THE CARD	37
<b>ADVANCED TECHNIQUES</b>	<b>37</b>
<b>THIRD-PARTY MEMORY MANAGERS</b>	<b>37</b>
<b>ROM CUT TABLE</b>	<b>37</b>
<b>POINT ENABLERS</b>	<b>37</b>

## SECTION II - COMPAQ LTE 5000 FAMILY ONLY

<b>INTENDED AUDIENCE</b>	<b>27</b>
OBJECTIVES	27
<b>SERVICES AND DRIVERS</b>	<b>28</b>
MS-DOS PCMCIA SERVICES AND DRIVERS	28
MS-WINDOWS DRIVERS	29
<b>TACKLING THE MEMORY DEFICIT</b>	<b>30</b>
CARDWIZARD MEMORY CONSIDERATIONS	31
REMOVE UNNECESSARY DRIVERS	31
<b>ATTAINING THE MOST BASE MEMORY</b>	<b>32</b>
OPTIMIZE THE PCMCIA RESERVED WINDOW	33
<b>SPECIFIC EXAMPLE</b>	<b>34</b>
<b>IBM TOKEN RING CARD AND LTE 5000 FAMILY</b>	<b>34</b>
DOS/WINDOWS 3.1 SUMMARY	34
CONFIGURATION STEPS FOR IBM TOKEN RING PC CARD AND ENABLER	34
ENABLER SUPPORT	35
WINDOWS 95	36

## INTRODUCTION

### *Introduction*

The inclusion of PCMCIA (Personal Computer Memory Card International Association) technology in PCs delivers a variety of benefits. Besides providing an industry standard interface for third party cards (PC Cards), PCMCIA allows users to easily swap cards in and out of a PC as needed, without having to deal with the allocation of system resources for those devices. These useful features -- hot swapping and automatic configuration, as well as card slot power management and other PCMCIA capabilities -- are supported by a variety of software components on the PCMCIA-based PC. In most cases, the software aspect of PCMCIA remains relatively transparent to the user. For some MS-DOS environments, however, the memory overhead from these software components conflicts with the requirements of other applications. These memory shortages are primarily caused by large MS-DOS applications and device drivers coupled with the memory requirements that PC Card drivers demand. Memory shortages are further intensified by the requirements of Card and Socket Services which provide the features of hot plugability and automatic configuration. All PCMCIA-based PCs using full Card and Socket Service implementations -- including Compaq PCs -- experience these memory conflicts to some degree.

This paper endeavors to assist you in assessing, recovering, and optimizing memory in your Compaq PCMCIA-based PC. Through a series of well-defined steps, it demonstrates how you can recover the most memory possible, permitting you the freedom to run your chosen applications.

### *Intended Audience*

This paper focuses on identifying and resolving memory resource issues, and therefore, must assume a certain level of expertise. For example, a familiarity with using memory managers and hexadecimal arithmetic is assumed. Even though PCMCIA technology and various configuration possibilities are discussed throughout, specific PCMCIA concepts are beyond the scope of this paper. For more information, please refer to:

If you require information regarding...	Refer to...	Published by:
...general information on PCMCIA...	Compaq PCMCIA White Paper	Compaq Computer Corporation
...which PC Cards have been tested in Compaq PCs...	Compaq PCMCIA Tested Solutions List	
...optimized network configurations...	Compaq PCMCIA Third Party Network PC Card Configurations	

## **Objectives**

After reading this paper, you will be able to:

- Assess the memory requirements of your specific configuration.
- Locate and remove unnecessary drivers.
- Recover previously unusable memory.
- Configure popular PCMCIA cards utilizing the smallest amount of memory possible.

The next section provides background information and terminology that you may find useful for performing the above actions. If you already have experience with Compaq's PCMCIA implementation, you can skip to the actual instructions which begin in the section titled "Attaining the Most Base Memory" on page 10.

## TERMS AND CONCEPTS

### ***Memory Configuration Terms and Concepts***

Compaq PCMCIA-based PCs utilize a segmented memory map. This means that areas of memory are divided into segments that are used for specific functions. For example, under MS-DOS, the first 640 kilobytes (KB) of memory are designated as base memory which is utilized by your applications. The next 384 kilobyte segment is referred to as upper memory and is used by peripheral devices, ROMs, and drivers. Current versions of MS-DOS, its required drivers, and other user-specific drivers position portions of themselves into base memory, using program space that would otherwise be available to MS-DOS applications. Subsequently, some applications may not have enough memory to execute. To alleviate this problem, you probably have installed a memory manager, such as EMM386.EXE. A memory manager locates unused pieces of upper memory called upper memory blocks (UMB) and reclaims them for your use. This allows you to force necessary software, that would otherwise occupy space in base memory, into UMBs. Memory managers help to provide the most base memory possible for actual applications. You can easily assess how much of each kind of memory you currently possess by running the MS-DOS utility *MEM*. For example, *MEM* can display the amount of used and free memory in your system by using the following syntax.

**MEM** [/CLASSIFY | /DEBUG | /FREE | /MODULE modulename] [/PAGE]

#### **/CLASSIFY or /C**

Classifies programs by memory usage. Lists the size of programs, provides a summary of memory in use, and lists largest memory block available.

#### **/DEBUG or /D**

Displays the status of all modules in memory, internal drivers, and other information.

#### **/FREE or /F**

Displays information about the amount of free memory left in both conventional and upper memory.

#### **/MODULE or /M**

Displays a detailed listing of a module's memory use. This option must be followed by the name of a module, optionally separated from /M by a colon.

#### **/PAGE or /P**

Pauses after each screenful of information.

If you are not familiar with using *MEM*, refer to the *MS-DOS User's Guide*. In this paper, you may be required to use *MEM* or a similar program to assess your memory utilization.

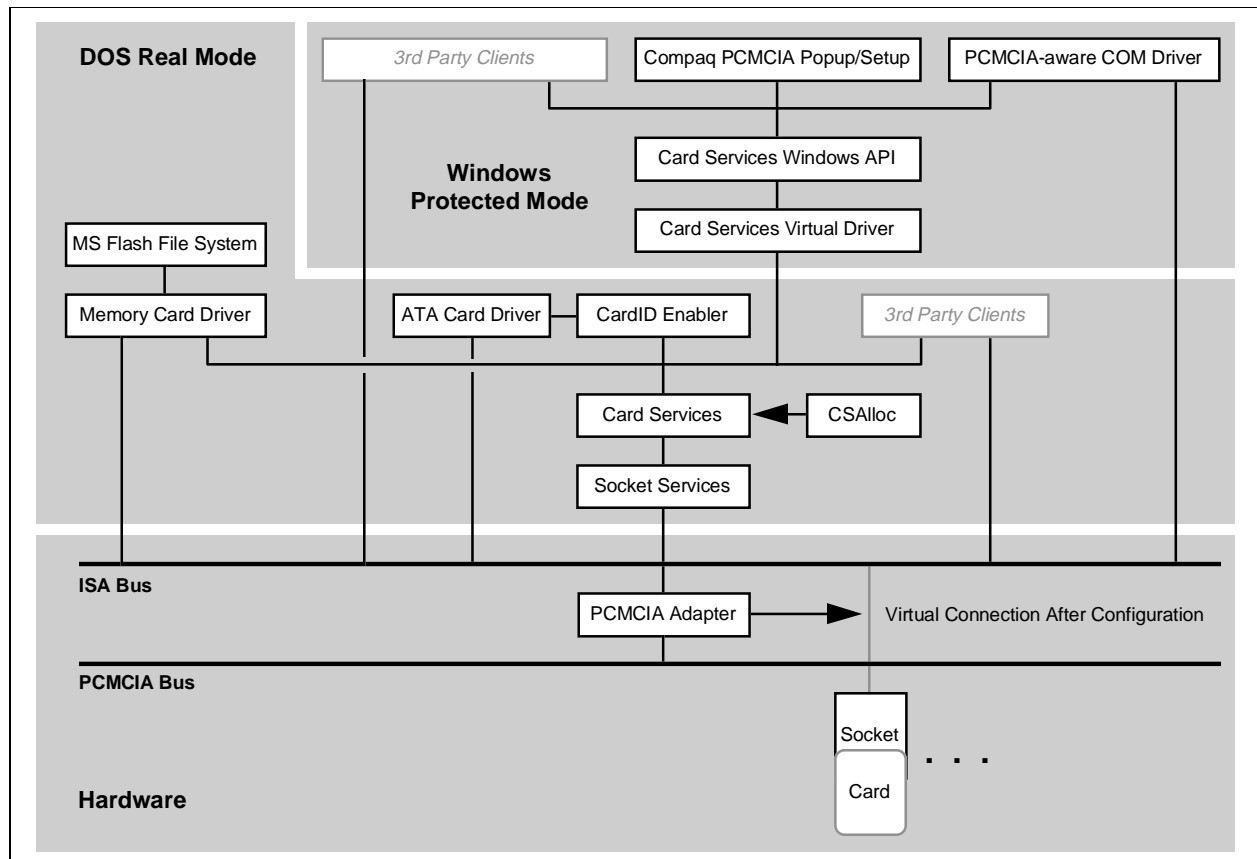


**PCMCIA Terms and Concepts**

When optimizing memory, it is helpful to understand how Compaq PCMCIA-based PCs support hot plugability, automatic configuration, and other PCMCIA-specific features. Card and Socket Services, as well as other card drivers, provide the outstanding features for which PCMCIA is distinguished; however, this software also adds to overhead in the memory optimization equation. In some cases, this additional overhead may conflict with application memory requirements. Also, keep in mind that PC Cards still require other systems resources just like an ISA/EISA card. These may include one or more of the following:

- Memory Addresses
- System Interrupts
- Direct Memory Access Channels
- Input/Output Ports

How exactly are PCMCIA cards supported? PCMCIA support is accomplished with a layered approach. The following illustration shows how a Compaq PCMCIA-based PC interfaces with a PC Card.



### PCMCIA Components

As illustrated in the graphic on the previous page, there are many specific PCMCIA components. This section lists those components by full name, filename, function, and for MS-DOS drivers, memory footprint.

#### *MS-DOS PCMCIA Services and Drivers*

The following table lists the most commonly utilized MS-DOS PCMCIA services. These drivers are loaded at the time of system initialization (boot) by being included in the CONFIG.SYS file. The load order of the drivers is very important since the PCMCIA architecture is layered. For example, the Socket Services driver must be loaded before the Card Services driver can function. The drivers are listed in this table in the order they are typically required to be loaded.

Full Name	Filename	Function	Approximate Memory Footprint	Notes
Socket Services	SSVLSI.EXE or SSCIRRUS.EXE	Provides a standard software interface to PCMCIA host controller chips and isolates the socket hardware from higher level software.	3,760 (4K)	Must be loaded first.
Card Services	CS.EXE	Manages system resources and configuration conflict issues.	39,392 (38K)	Requires socket services (SSxxxx.EXE).
Card Services Resource Allocation	CSALLOC.EXE	Initializes Card Services resource table at boot time.	This driver does not require any memory once initialized.	Requires Socket and Card Services.
Card Services Super Client	CARDID.EXE	Configures PC Cards which do not have CS client drivers and "exception" cases.	20,320 (20K)	Requires Socket and Card Services.
Memory Card Driver	MEMDRV.EXE	Block device driver for memory cards.	17,184 (17K)	Required for SRAM or Flash card support
ATA Card Driver Support	ATADRV.EXE	Provides support for ATA/ IDE mass storage cards such as rotating media or Sundisk-style cards.	6,496 (6K)	Relies on CARDID to configure ATA cards.
Microsoft Flash File System	MS-FLASH.SYS	Provides file system support for flash memory array cards.	70,240 (69K)	Requires MEMDRV.EXE
DoubleSpace for Flash File System	DBLFLASH.EXE	Provides DoubleSpace data compression support for the Microsoft Flash File System.	13,504 (13K)	Requires MS-FLASH.SYS
PCMCIA Power Management Driver	PCMSMIX.EXE	Provides support for standby hibernation and ring resume for certain PCs.	This driver does not require any memory once initialized.	Not used on all Compaq PCMCIA-based PCs.

### **MS-Windows Drivers**

The following table lists the required files for hot insertion/removal of communications I/O, memory, and removable drive cards within Windows.

<b>Full Name</b>	<b>Filename</b>	<b>Function</b>	<b>Notes</b>
Communications driver	SSCOMM.DRV	Card Services-aware version of standard Windows COMM.DRV.	Replaces COMM.DRV
Card Services Serial Port Virtual Driver	CSVCD.386	Card Services-aware version of standard Windows VCD.386	Replaces *vcd
Card Services Windows API	SSWINCS.DLL	Provides the Card Services interface to Windows applications.	
Card Services Virtual Driver	CS.386	Provides the Card Services interface to Windows virtual sessions.library (DLL) required to support hot plugability within Windows.	
Card Event Monitor	CPQEVENT.EXE	Displays pop-ups on insertion / removal of PC Card.	

## Attaining the Most Base Memory

Before you begin optimizing memory, a little up front work can eliminate later frustration. To assist you in planning the steps necessary for optimizing your particular configuration, a chart like the one located on this page may be used to assess your memory configuration needs.

PCMCIA-related software drivers require portions of memory and it makes sense to load these drivers into upper memory. However, you may still need upper memory for other drivers that your configuration requires. This can make upper memory scarce, and force device drivers to use conventional memory. Accordingly, by assessing your requirements now and recording them, you can avoid having to gather the information later while you are in the midst of optimization. For PCMCIA-related driver memory footprints, use the previous charts in “*PCMCIA Components*.” For other configuration specific driver memory footprints, use *MEM* as described in “*Memory Configuration Terms and Concepts*.” Once you have all the settings, it may be helpful to record them in a chart like the one shown below.

Subsequent pages provide additional information about ascertaining your memory requirements which can be used to fill in the chart.

Driver Module Name (your configuration specific driver modules)				Memory Required		
Total Memory Required:						
PCMCIA Card	PCMCIA Reserved Memory required?	If so, how much?	Does card require driver?	If so, what is its memory footprint?	Memory Address	Notes
<b>Totals Where Applicable:</b>						

Before reconfiguring your Compaq PCMCIA-based PC, it is recommended that you document as much information about your present configuration as possible. For example, when changing any setting, whether it is an IRQ, I/O address, memory address, or DMA channel, the change should be noted. The reason is you may be required to identify that change elsewhere, as in a *device=*, or a memory *include* statement in the *CONFIG.SYS* file.

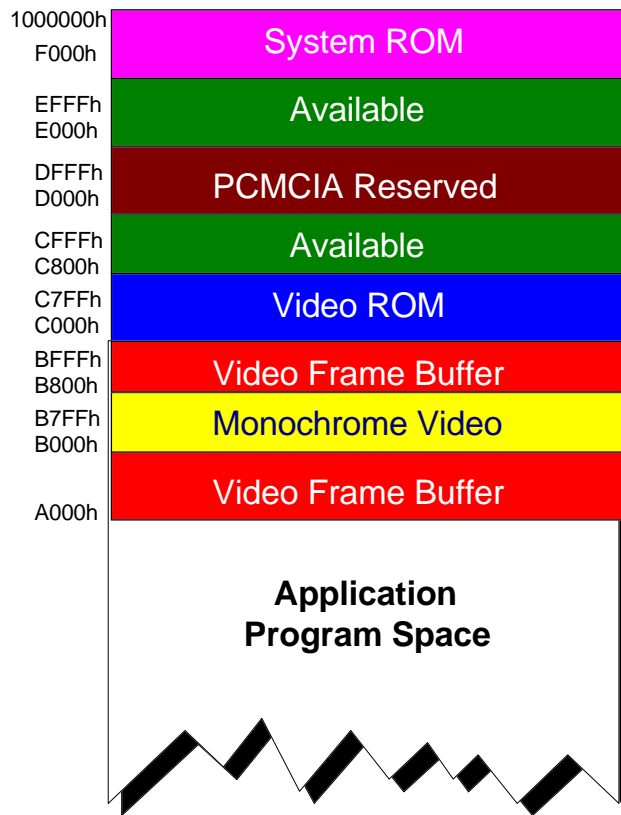
The following chart lists which files must be updated when reconfiguring memory or settings to support modems or network controllers:

	PCMCIA Modem	PCMCIA Ethernet	PCMCIA Token Ring
<b>Editing Required?</b>			
<b>PCCARD</b>			Yes
<b>CONFIG.SYS</b>	Yes	Yes	Yes
<b>CSALLOC.INI</b>	Yes	Yes	Yes
<b>SYSTEM.INI</b>	Yes	Yes	Yes
<b>PROTOCOL.INI</b>		*	*
<b>NET.CFG</b>		*	*

*\*Note: Whether editing is required with these files depends on your network environment. Client drivers for network cards should be executed from the same directory that the NET.CFG or PROTOCOL.INI file resides in. It may also be necessary to include the appropriate memory configuration parameters per your environment.*

### Tackling the Memory Deficit

A PCMCIA-equipped Compaq PC out of the box uses nearly 180KB of memory to provide basic PCMCIA support. This memory is normally available in non-PCMCIA-based PCs. Despite this fact, there are several steps you can take to customize your environment to fit your specific requirements. Once complete, you can regain up to 115KB of previously unavailable memory. This is usually more than sufficient for even the largest MS-DOS applications. So how do you do it?



The steps required to optimize memory do depend on which PC Cards you wish to install; however, there are several generic steps you can take. These steps include:

- Remove Unnecessary Drivers
- Recover Unused Memory Areas
- Utilize PCMCIA Reserved Window Memory
- Change Driver Loading Order

As an overview, the graphic to the left depicts a typical memory map for a PCMCIA-equipped Compaq PC in its default configuration. Notice that the *PCMCIA Reserved* window is located in the middle of a large available block of upper memory. Some of this memory can not only be reclaimed, it can also be moved to another location providing you with a larger contiguous area. Also, notice that there is a region dedicated to *Monochrome Video*. If you are not utilizing a monochrome video controller, this area can

also be used. In the following pages, you will be shown how to do this.

**Remove Unnecessary Drivers**

The first step in regaining memory is to remove any drivers that are not required in your configuration. However, how do you know which drivers are required? The following illustration answers that question for various types of PC Cards which may need to be supported.

		Drivers							
		SS	CS	CSALLOC	CARDID	ATADRV	MEMDRV	MS-FLASH	DBLFLASH
<b>PC Cards</b>	<b>Modem/Fax</b>	●	●	●	●				
	<b>Network<sub>1</sub></b>	●	●	●	1				
	<b>ATA (IDE)</b>	●	●	●	●	●			
	<b>SRAM</b>	●	●	●			●		
	<b>Flash memory</b>	●	●	●			●	●	2
	<b>Aero diskette drive</b>	3	3	3	3				
	<b>Other<sub>4</sub></b>	●	●	●					

NOTES:

1. Network cards also require a network driver from card vendor. CARDID is needed only if network driver is *not* Card Services compliant.
2. Needed only if data compression is desired for flash memory cards.
3. Needed only if diskette drive will be inserted or removed while system is on (“hot swapped”).
4. Other cards (SCSI, Audio) also require a Card Services compliant driver from card vendor.

Are there any drivers that you currently are not using? If so, remove them by inserting the word *REM* (which stands for *remark*) or a single semi-colon (“;”) at the beginning of the appropriate statement in your *CONFIG.SYS* file. This will prevent the driver(s) from being loaded the next time the system is booted and will therefore reclaim the memory that the driver(s) would have used.



If you are currently not using PCMCIA cards *at all*, you can add a *remark* to all PCMCIA driver references and delete the X=D000-DFFF entry on the EMM386 command line in your *CONFIG.SYS* file, disabling PCMCIA entirely. After doing so, you will regain all the upper memory usually available in non-PCMCIA-based PCs; however, you will not have PCMCIA capabilities.

Step	Action	What is Gained?	What is Lost?	Notes
1.	Remark out unnecessary drivers from the <i>CONFIG.SYS</i> file	For example, remarking out: <ul style="list-style-type: none"> <li>• ATADRV.EXE</li> <li>• MEMDRV.EXE</li> </ul> will recover in excess of 20k of memory.  Remarking out: <ul style="list-style-type: none"> <li>• MS-FLASH.SYS</li> <li>• DBLFLASH.EXE</li> </ul> will recover 83k of memory	These drivers must be loaded later if you add an ATA, SRAM, or Flash Memory card.	MS-FLASH.SYS/DBLFLASH.EXE are already remarked out in the original <i>CONFIG.SYS</i> and therefore not included in the original memory savings estimates.



### Recover Unused Memory Areas

Are you using a monochrome controller in an expansion base? If no, the second step in regaining memory is to recover the monochrome region.

Step	Action	What is Gained?	What is Lost?	Notes
1.	Recover the monochrome (B000-B7FF) region by adding the appropriate Include statement to your memory manager. If you are using EMM386 include I=B000-B7FF.	This will recover 32k of memory.	The ability to utilize monochrome cards, for example, in an Elite SmartStation.	If you have any problems entering Windows after making this change you may need to add the following to the [386enh] section of the SYSTEM.INI file.  DEVICE=C:\DOS\MONOU MB.386

### Utilize PCMCIA Reserved Window

The next step is to make use of the otherwise *reserved* memory not being used by PCMCIA cards. Out of the box, Compaq PCMCIA-based PCs reserve a 64KB window from D000 through DFFF for cards to reside in. Card Services requires 4KB of that memory; however, most cards that you install do not require the balance of 60KB. In fact, some cards do not require any memory in this area other than the 4KB required by Card Services. To determine how much memory, if any, your card(s) requires, consult the card's user's guide or call the manufacturer.

Step	Action	What is Gained?	What is Lost?	Notes
1.	Determine how much memory each of your PCMCIA cards require in order to initialize. For example, PCMCIA modems usually do not require any memory and most PCMCIA Ethernet adapters require only 8 to 12KB.	If your memory requirements are relatively small, you can usually move the PCMCIA window out of the D000 range allowing for larger contiguous UMB's. A good place to move this range to is the C800 area.		Remember that, at a minimum, the PCMCIA reserved area will need to be 4KB for Card Services use.

Step	Action	What is Gained?	What is Lost?	Notes
2.	Modify your <i>CONFIG.SYS</i> by moving the PCMCIA card initialization area out of the D000 range and reducing the amount of memory used from 64KB to a smaller amount. For example, a modem (with the 4KB CS overhead) and Ethernet Card will usually only require 16KB or less. Hence, you can change the default EMM386.EXE statement of <i>X=D000-DFFF</i> to <i>X=C800-CBFF</i> .	The previous example will yield a 48KB gain of UMBs.	If you are unable to locate how much memory your card requires to initialize, it may take several trial and error attempts to assess how much memory it requires.	In this example, you may even be able to recover more memory since some Ethernet Cards require less than 12KB.
3.	Modify <i>CSALLOC.INI</i> by adding or editing the MEMEXCLUDE entry to exclude the newly defined C800-CBFF range:  MEMEXCLUDE=CC00-EFFF	This will tell Card Services to avoid using any memory above CC00h to configure PCMCIA cards.		If the 16KB excluded on the EMM386 line is modified from the example (X=C800-CBFF), make sure to modify the MEMEXCLUDE statement to match.
4.	Modify the [386Enh] section of the <i>SYSTEM.INI</i> to include an EMMEXCLUDE statement that matches the range excluded on the EMM386.EXE line in the <i>CONFIG.SYS</i> file. For this example, the line must read: EMMEXCLUDE=C800-CBFF			

*\*Note: You must reduce or increase the PCMCIA Reserved window in 4KB increments. Depending on the network controller card and the NOS, it may be necessary to edit the PROTOCOL.INI or NET.CFG files. Make sure to place the appropriate memory reference in these files. The syntax may vary from card to card. In the example illustrated above, the statement references the C800h range. It is also important to execute the client driver for the network cards from within the same directory that the PROTOCOL.INI or NET.CFG file resides in.*

**Change Driver Loading Order**

Finally, shifting the load order of *your* specific drivers (but not the PCMCIA drivers themselves) may allow you to load more of them into upper memory. To do this, you can manually adjust the order through trial and error by changing the execution order in the *CONFIG.SYS* file.

Alternatively, you can try to force specific drivers into specific regions of upper memory by using the */L:* option for *DEVICEHIGH (CONFIG.SYS)* and *LOADHIGH (AUTOEXEC.BAT)*.

Note that load order of the PCMCIA drivers themselves should not be modified. The required load order is documented in the table *MS-DOS PCMCIA Services and Drivers* on page 8 and should be maintained.

When using *LOADHIGH*, the following syntax should be used:

```
LOADHIGH [drive:][path]filename [parameters]
LOADHIGH [/L:region1[,minsize1][;region2[,minsize2]...] [/S]]
        [drive:][path]filename [parameters]
```

```
/L:region1[,minsize1][;region2[,minsize2]]...
```

Specifies the region(s) of memory into which to load the program. Region1 specifies the number of the first memory region; minsize1 specifies the minimum size, if any, for region1. Region2 and minsize2 specify the number and minimum size of the second region, if any. You can specify as many regions as you want.

Consult your MS-DOS User's Guide for more information on *LOADHIGH* and *DEVICEHIGH*.

## Specific Examples

### ***IBM Token Ring Card***

Many customers have required assistance properly configuring an IBM Token Ring Card within the LTE Elite. This example not only demonstrates how to maximize memory for that environment, it also shows you how to avoid possible traps when configuring the card. As stated in the table on Page 11, token ring cards necessitate changes to:

- PCCARD
- CONFIG.SYS
- CSALLOC.INI
- SYSTEM.INI

In addition, PROTOCOL.INI or NET.CFG, depending on your network environment, may also require changes.

To properly configure the IBM Token Ring Card to function in the Elite, you must change the system resources, in addition to performing any memory optimization required. Thus, before you begin, it is recommended that you backup all the files that will be edited (CONFIG.SYS, SYSTEM.INI, CSALLOC.INI, , PROTOCOL.INI, and NET.CFG).

### Running PCCARD

The first step in properly configuring the IBM Token Ring Card is to execute PCCARD. PCCARD is a utility used to change default resource settings. For simplicity, the following table illustrates how to maneuver within PCCARD.

Step	Action
1.	<ul style="list-style-type: none"> <li>• Type <b>PCCARD</b> from within the CPQDOS sub-directory and press <b>ENTER</b>.</li> <li>• Press <b>ENTER</b> to bypass the WELCOME Screen.</li> </ul>
2.	<p>There are four available menus:</p> <ol style="list-style-type: none"> <li>1. File <ul style="list-style-type: none"> <li>Edit Configuration</li> <li>Save Configuration</li> </ul> </li> <li>2. Utility <ul style="list-style-type: none"> <li>Resource Allocation</li> </ul> </li> <li>3. Display--Color/Monochrome/LCD</li> <li>4. Help--About</li> </ol>
3.	<p>To edit the configuration of a modem, network card, ATA card, or miscellaneous card parameters:</p> <ul style="list-style-type: none"> <li>• Press the <b>Alt</b> and <b>F</b> keys simultaneously (you may also use the mouse) to open the file menu of card configuration.</li> <li>• Choose <b>Edit Configuration</b>.</li> </ul>
4.	<p>To edit network card parameters specifically, press the <b>Alt</b> and <b>N</b> keys simultaneously to open the Network Card Information Window. You can move between each field by pressing the <b>TAB</b> key.</p>
5.	<ul style="list-style-type: none"> <li>• Once the changes are complete, select <b>OK</b> by pressing the <b>ENTER</b> key twice.</li> <li>• Then select <b>Save Configuration</b> from the FILE menu.</li> <li>• Finally, exit PCCARD by selecting <b>Exit</b> within the FILE menu.</li> </ul>

### Configuring The Card

To properly function in a Compaq PC, the IBM Token Ring Card must be configured to use IRQ 9 rather than its default of IRQ 5. However, by default CS.EXE also uses IRQ 9 to notify you of card insertions and removals. Consequently, CS.EXE must be reconfigured to use another IRQ. The following steps demonstrate how to do this, as well as the other procedures necessary for optimized configuration. (The example below modifies CS.EXE to use IRQ 10, ie. 'A' in hex.)

**Step 1:** Use PCCARD as demonstrated previously to change the Network Card Information:

System Resource	From...	To...
Port address	300	A20
IRQ	5	9
Memory 1	Ensure that the default address is CC00 (CC00 is the shared ram address).	
Memory 2	Ensure that the default address is C800 (C800 is the MMIO).	
Fast token ring speed (16mb/s)	Select the appropriate ring speed. An X in the field enables 16mb/s and no X in the field indicates 4mb/s.	

Ensure that you save the new configuration as instructed previously in "Running PCCARD."

*Please note that in the text file examples in the following table, line length constraints have forced some lines to wrap. These are indented to indicate that the text is all part of a single line in the actual file. Inline explanatory statements are in italics, and changed text is highlighted.*

Step 2	Step 3	Step 4
Edit the <i>CONFIG.SYS</i> file adding the <b>IRQ A</b> (10) to the <i>CS.EXE</i> line. In order to maximize available base memory, in some cases it is necessary to load <i>SMARTDRV.EXE</i> after all network drivers, though prior to launching Windows.	Edit the following lines in the <i>SYSTEM.INI</i> file.	Edit the <i>CSALLOC.INI</i> within the CPQDOS directory to match the other files.
<p style="text-align: center;"><b>CONFIG.SYS Example</b></p> <pre> DEVICE=C:\DOS\HIMEM.SYS <b>DEVICE=C:\DOS\EMM386.EXE NOEMS</b>   <b>I=B000-B7FF X=C800-D0FF I=D100-EFFF</b> DEVICEHIGH=C:\DOS\SETVER.EXE FILES=60 BUFFERS=20 DOS=HIGH,UMB LASTDRIVE=E (<i>This statement could change if using vlm's</i>) FCBS=4,0 STACKS=9,256 SHELL=C:\DOS\COMMAND.COM C:\DOS\ /E:512 /p REM *** Begin PCMCIA Drivers, DO NOT Change Order *** DEVICEHIGH=C:\CPQDOS\SSVLSI.EXE           </pre>	<p style="text-align: center;"><b>SYSTEM.INI Example</b></p> <pre> [386ENH]  <b>ADD:</b> <b>DEVICE=C:\DOS\MONOUM</b> <b>B.386</b>  <b>CHANGE:</b> <b>EMMEXCLUDE=D000-DFFF</b>  <b>TO:</b> <b>EMMEXCLUDE= C800-</b> <b>D0FF</b> (<i>This range should reflect the exclude statement on the EMM386 line in your CONFIG.SYS</i>)           </pre>	<p style="text-align: center;"><b>CSALLOC.INI Example</b></p> <pre> <b>MEM=C800-D0FF</b> RIO=170-177,2E8-   2EF,370- 377,3E8-   3F7,3F0-3F7 IOINCLUDE=3F0- 3F7 <b>MEMEXCLUDE=B0</b> <b>00-B7FF,C000-</b> <b>C7FF,D100-EFFF</b>           </pre>

<pre> <b>DEVICEHIGH=C:\CPQDOS\CS.EXE /IRQ A</b> DEVICE=C:\CPQDOS\CSALLOC.EXE REM DEVICEHIGH=C:\CPQDOS\ATADRV.EXE DEVICEHIGH=C:\CPQDOS\CARDID.EXE   C:\CPQDOS\CARDID.INI REM DEVICEHIGH=C:\CPQDOS\MEMDRV.EXE REM DEVICEHIGH=C:\CPQDOS\DBLFLASH.EXE REM DEVICEHIGH=C:\CPQDOS\MS-FLASH.SYS REM *** End PCMCIA Drivers *** DEVICEHIGH=C:\DOS\POWER.EXE DEVICEHIGH=C:\DXM\DXMA0MOD.SYS DEVICEHIGH=C:\DXM\DXMC0MOD.SYS ,CC00 </pre>		
---	--	--

## Advanced Techniques

### ***Third-party memory managers***

The techniques described here should function with third-party memory managers as well; however, Compaq only supports and tests the utilities that are supplied with your Compaq PC. When using third-party memory managers, ensure that you use the proper syntax.

### ***ROM Cut Table***

The following procedure, though not extraordinarily complex, requires experience and a certain comfort level with the MS-DOS utility *DEBUG*. Therefore, if you are not comfortable with *DEBUG*, do not attempt this procedure.

Compaq system ROMs include code that is executed only at power-on initialization or reboot. This is called the Power-On Self-Test (POST) code and occupies a memory segment above 0F000h. Since this code is not part of an active system, in some cases the ROM space that it occupies can be reclaimed as upper memory. There is a table in Compaq ROMs, called the *Cut Table*, which contains information that describes this memory. However, note that though the table may exist, in many cases, it is empty and cannot provide additional memory.

The procedure described here will allow you to determine if you have any additional usable memory. If you do, you may be able to identify up to 28KB of additional upper memory (though typically 16KB). However, keep in mind that some PCs will not have any additional upper memory above 0F000h. To find out if your particular PC has additional upper memory, you must:

- Examine a data structure in the System ROM
- Determine how much memory, if any, is available



The following steps assume a working knowledge of the *DEBUG* utility and basic hexadecimal arithmetic. To access the Cut Table, do the following:

Step	Action																					
1.	Use <i>DEBUG</i> to examine the 16-bit word in the ROM memory area at address F000:801E. This word is the offset (F000h is the segment) of the variable length Cut Table in ROM.																					
2.	<p>Examine the table using the offset obtained in step 1 (with a segment value of F000). Does the first word at this location contain the value 1234h?</p> <ul style="list-style-type: none"> <li>If <b>Yes</b>, a <i>Cut Table</i> of the following format exists and you may proceed to Step 3. <table border="1" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: center;">Offset</th> <th style="text-align: center;">Size</th> <th style="text-align: center;">Contents</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0000</td> <td style="text-align: center;">Word</td> <td>Cut Table signature = 1234h</td> </tr> <tr> <td style="text-align: center;">0002</td> <td style="text-align: center;">Byte</td> <td>Number of Table Entries</td> </tr> <tr> <td style="text-align: center;">0003</td> <td style="text-align: center;">Byte</td> <td>01 - ID value indicating segment address</td> </tr> <tr> <td style="text-align: center;">0004</td> <td style="text-align: center;">Word</td> <td>Segment to be used in subsequent operations</td> </tr> <tr> <td style="text-align: center;">0006</td> <td style="text-align: center;">Byte</td> <td>02 - ID value indicating POST cut address</td> </tr> <tr> <td style="text-align: center;">0007</td> <td style="text-align: center;">Word</td> <td>Offset of last byte of POST code</td> </tr> </tbody> </table> </li> <li>If <b>No</b>, there is no <i>Cut Table</i> in this particular ROM and therefore no additional upper memory available.</li> </ul>	Offset	Size	Contents	0000	Word	Cut Table signature = 1234h	0002	Byte	Number of Table Entries	0003	Byte	01 - ID value indicating segment address	0004	Word	Segment to be used in subsequent operations	0006	Byte	02 - ID value indicating POST cut address	0007	Word	Offset of last byte of POST code
Offset	Size	Contents																				
0000	Word	Cut Table signature = 1234h																				
0002	Byte	Number of Table Entries																				
0003	Byte	01 - ID value indicating segment address																				
0004	Word	Segment to be used in subsequent operations																				
0006	Byte	02 - ID value indicating POST cut address																				
0007	Word	Offset of last byte of POST code																				
3.	<p>Examine the byte at offset 0002 in the <i>Cut Table</i>. Is it 00?</p> <ul style="list-style-type: none"> <li>If <b>Yes</b>, there are no <i>Cut Table</i> entries and therefore no additional upper memory available.</li> <li>If <b>No</b>, then the table described above contains at least one entry. Each entry consists of a Byte/Word pair. The byte describes the type of entry and the word is the value associated with the entry, either an offset or a segment address.</li> </ul>																					
4.	Examine the entries at offset 0003 and offset 0006. A typical non-empty cut table contains 2 entries. The first entry will be of type 01 and the word will contain the segment address of the ROM (F000h). The second entry will be of type 02 and the word will contain the offset of the last byte of POST code which is the last byte of available memory in the F000h segment.																					
5.	Convert the segment:offset form to a paragraph address by shifting the segment value left 4 bits (append a zero on the right to yield a 5 digit hexadecimal value) and adding in the offset. By truncating the least significant digit of the result, you will obtain an address which can be used in the EMM386 <i>include</i> command.																					

For example:

Step	Action
1.	Follow the first three steps described previously to ascertain that a <i>Cut Table</i> exists and contains two entries. You find a 01 at offset 0003, indicating a segment value, and a 02 at offset 6, indicating the address of the last byte of the POST code. Assume that the Type 01 table entry is F000h, and the Type 02 table entry is 67FFh.
2.	Convert the segment:offset, F000:67FFh, to the paragraph address F67FFh and truncate the least significant digit to obtain F67Fh. The new available range is therefore F000h-F67Fh. Add an <i>include F000-F67F</i> to the EMM386 command line in the <i>CONFIG.SYS</i> file to allocate it. In this example, you would have allocated an additional 28K of upper memory.

**Important:** The value described in the *Cut Table* may vary from PC to PC, and even from ROM version to ROM version. This is because changes and/or additions are made to the basic functionality the ROM provides from time to time. For this reason, you must be careful when configuring multiple machines. What may be available and work on a particular PC may not on another. Ensure that you examine the *Cut Table* (or at least the ROM version to ensure they are the same) for each PC that you plan to use, then optimize each individual configuration accordingly.

### Point Enablers

A point (direct) enabler is a driver supplied by a card vendor that bypasses Card and Socket Services and interfaces with the hardware directly. It configures specific cards and specific controllers only. Once an enabler has initialized the controller and card, it no longer requires memory. For installations with memory shortages, this may seem like a quick cure, but once installed, point enablers negate all the features of Card and Socket Services. Lost features include hot plugability, automatic configuration, power management, and standby/hibernation support. Cards must physically be installed at power-on and cannot be removed until the system is shut down (much like installing an ISA controller).

Besides losing all the features of Card and Socket Services, a point enabler may not always function properly. Whether the enabler can locate your card depends on which PCMCIA controller it was written for. For example, in LTE Elites and Contura 400s, many currently available enablers support only slot one (the bottom slot). In all instances, Compaq recommends that you follow the previous steps to optimize your memory. Use a point enabler only as a last resort, and with a full awareness of its limitations.

## **SECTION II**

### ***PCMCIA Memory Optimization White Paper***

### ***Compaq LTE 5000 Family of Portable PCs***

### **Intended Audience**

This paper focuses on identifying and resolving memory resource issues as they relate to the Compaq LTE 5000 family, and therefore, must assume a certain level of expertise. (When necessary, refer back to Section I of this White Paper.) For example, a familiarity with using memory managers and hexadecimal arithmetic is assumed. Even though PCMCIA technology and various configuration possibilities are discussed throughout, specific PCMCIA concepts are beyond the scope of this paper. For more information, please refer to the following documents published by Compaq:

If you require information regarding...	Refer to...	Availability:
...general information on PCMCIA...	Compaq PCMCIA White Paper	Call the Compaq Support Center at 1-800-345-1518, select the PaqFax option and request catalog 4600  On-line access from following services: InterNet: <a href="http://www.compaq.com">http://www.compaq.com</a> <a href="ftp://ftp.compaq.com">ftp://ftp.compaq.com</a>  Compaq Forums on: CompuServe, Prodigy, and America On-line
...which PC Cards have been tested in Compaq PCs...	Compaq PC Card Solution List	
...optimized PC Card configurations...	Compaq PC Card Solution - Premier PC Card Configurations	

### **Objectives**

After reading this paper, you will be able to:

- Assess the memory requirements of your specific configuration.
- Locate and remove unnecessary drivers.
- Recover previously unusable memory.
- Configure a popular PC Card utilizing the smallest amount of memory possible.

### MS-DOS PCMCIA Services and Drivers

The following table lists the MS-DOS PCMCIA services provided in the Cardsoft 3.14 software package. These drivers are loaded at the time of system initialization (boot) by being loaded in the CONFIG.SYS file. The order in which the drivers are loaded is very important since the PCMCIA architecture is layered. For example, the Socket Services driver must be loaded before the Card Services driver can function. The drivers are listed in this table in the order they are typically required to be loaded.

Full Name	Filename	Function	Approximate Memory Footprint	Notes
Socket Services (SS)	SSCIRUS2.EXE (REQUIRED)	Provides a standard software interface to PCMCIA host controller chips and isolates the socket hardware from higher level software.	4,992 (5K)	Must be loaded first.
Card Services (CS)	CS.EXE (REQUIRED)	Manages system resources and configuration conflict issues.	44,896 (44K)	Requires socket services (SSxxxx.EXE).
Card Services Resource Allocation	CSALLOC.EXE (REQUIRED)	Initializes Card Services resource table at boot time.	This driver doesn't require any memory once initialized.	Requires Socket and Card Services.
Card Services Super Client (CARDID)	CARDID.EXE (REQUIRED)	Configures PC Cards which do not have CS client drivers and "exception" cases.	24,288 (24K)	Requires Socket and Card Services
Memory Card Driver	MTDDRV.EXE	Block device driver for memory cards.	10,480 (10K)	Required for SRAM or Flash card support
SRAM MTD Driver	MTSRAM.EXE	SRAM low level memory card driver.	3,200 (3K)	Requires MTDDRV.EXE
Flash Translation Layer	FTL.EXE	Provides file system translation layer for flash cards.	31,424 (31K)	Requires MTDDRV.EXE
Flash MTD Driver	MTAA.EXE	Device driver for AMD Type A flash cards.	3,120 (3K)	Requires MTDDRV.EXE and FTL.EXE
Flash MTD Driver	MATB.EXE	Device driver for AMD Type B flash cards.	2,970 (3K)	Requires MTDDRV.EXE and FTL.EXE
Flash MTD Driver	MTI1.EXE	Device driver for Intel Series 1 flash cards.	4,992 (5K)	Requires MTDDRV.EXE and FTL.EXE
Flash MTD Driver	MTI2P.EXE	Device driver for Intel Series 2 flash cards.	3,744 (4K)	Requires MTDDRV.EXE and FTL.EXE

Flash MTD Driver	MTATM.EXE	Device driver for Atmel flash card	4,496 (5K)	Requires MTDDRV.EXE and FTL.EXE
ATA Card Driver Support	ATADRV.EXE	Provides support for ATA/ IDE mass storage cards such as rotating media or Saundisk-style cards.	11,520 (11K)	Relies on CARDID to configure ATA cards.

**SUMMARY:** If all PCMCIA drivers were loaded they would occupy 148KB of memory. However, the default installation of drivers (CS, SS, CARDID) occupies 73KB of memory.

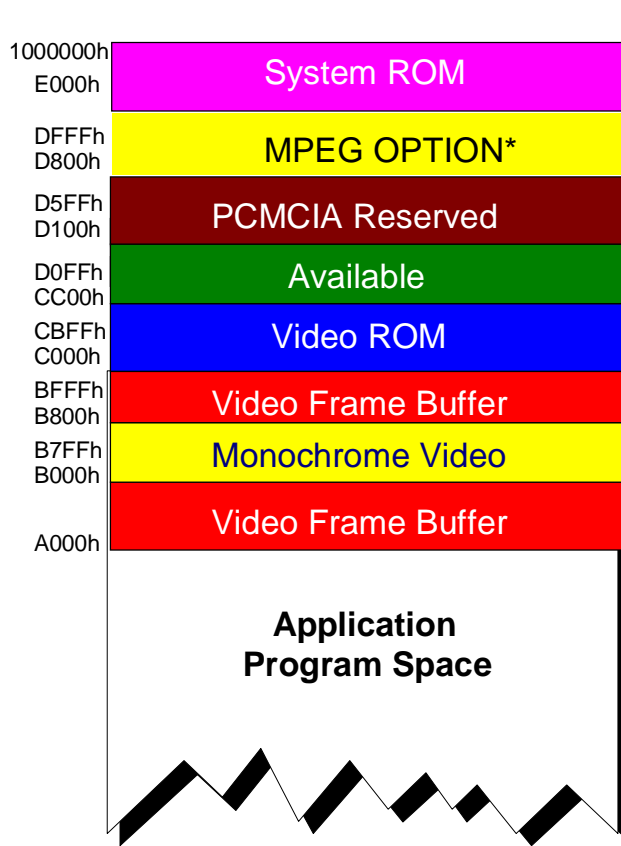
### *MS-Windows Drivers*

The following table lists the required files for hot insertion/removal of communications I/O, memory, and removable drive cards within Windows.

Full Name	Filename	Function	Notes
Communications driver	CPQCOMM.DRV	Card Services-aware version of standard Windows COMM.DRV.	Replaces COMM.DRV
Card Services Serial Port Virtual Driver	SSVCD.386	Card Services-aware version of standard Windows VCD.386	Replaces *vcd
Card Services Windows API	SSWINCS.DLL	Provides the Card Services interface to Windows applications.	
Card Services Virtual Driver	PCCARD.386	Provides the Card Services interface to Windows virtual sessions library (DLL) required to support hot plugability within Windows.	

## Tackling the Memory Deficit

A PCMCIA-equipped Compaq PC out of the box uses nearly 90KB of memory to provide basic PCMCIA support. This memory is normally available in non-PCMCIA-based PCs. Despite this fact, there are several steps you can take to customize your environment to fit your specific requirements. This is usually more than sufficient for even the largest MS-DOS applications. So how do you do it?



The steps required to optimize memory depend on which PC Cards you wish to install; however, there are several generic steps you can take. These steps include:

- Deactivate Or Remove Unused PC Card Drivers
- Optimize PCMCIA Reserved Memory Window
- Change Loading Order Of Drivers

As an overview, the graphic to the left depicts a typical memory map for a PCMCIA-equipped Compaq PC in its default configuration. Notice that the *PCMCIA Reserved* window is located in the middle of a large available block of upper memory. Some of this memory can not only be reclaimed, it can also be moved to another location providing you with a larger contiguous area.

\*MPEG Option - only used if PC has over 16MB and is running WIN/DOS

## Card Wizard Memory Considerations

Compaq preinstalls a PC Card utility, called CardWizard, on the LTE 5000 family. This utility has the ability to recognize different PC Card requirements and alter configuration parameters to support these requirements. CardWizard also has the ability to *dynamically* load the appropriate PCMCIA drivers when a card is inserted into the PC Card slot on either the PC or the expansion base. (Compaq's default configuration includes, but does not activate, PC Card drivers, thereby reducing the default memory footprint.) This is important to note because once the PCMCIA drivers are activated by CardWizard, the drivers remain active after PC Card removal, and continue to consume memory resources.

**Example:** While using Windows 3.1 an Intel 20MB Series 2 flash card is inserted. CardWizard will not configure the card automatically, but if the CORRECT button is selected and then FLASH is chosen the following drivers will be added to the CONFIG.SYS:

MTAA.EXE (3K)  
MATB.EXE (3K)  
MTI1.EXE (5K)  
MTI2P.EXE (4K)  
MTATM.EXE (5K)  
MTDDRV.EXE (10K)  
FTL.EXE (31K)

After rebooting the system an additional 60K will assigned to PCMCIA support. These drivers will stay resident even if support for the card is no longer needed.

### Remove Unnecessary Drivers

The first step in regaining memory is to remove any drivers that are not required in your configuration. However, how do you know which drivers are required? (Refer to the MS-DOS PCMCIA section of Section II)



If you are currently not using PC Cards *at all*, you can add a *remark* to all PCMCIA driver references and delete the X=D100-D5FF entry on the EMM386 command line in your *CONFIG.SYS* file, disabling PCMCIA entirely. After doing so, you will regain all the upper memory usually available in non-PCMCIA-based PCs; however, you will not have PCMCIA capabilities.



## Unnecessary drivers -continued-

All the drivers loaded by CardWizard may not be needed, and using the example from the previous page please examine the following:

```
;DEVICEHIGH=C:\CARDWIZ\MTAA.EXE (3K)
;DEVICEHIGH=C:\CARDWIZ\MATB.EXE (3K)
;DEVICEHIGH=C:\CARDWIZ\MTI1.EXE (5K)
DEVICEHIGH=C:\CARDWIZ\MTI2P.EXE (4K)
;DEVICEHIGH=C:\CARDWIZ\MTATM.EXE (5K)
DEVICEHIGH=C:\CARDWIZ\MTDDR.V.EXE (10K)
DEVICEHIGH=C:\CARDWIZ\FTL.EXE (31K)
```

If the CONFIG.SYS was modified as above the Intel flash card would still function normally, however, a memory savings of 16K would be achieved. If the Intel flash card were a temporary solution, and there was no longer a need for support, all the above drivers could be remarked out. This solution is presented below, and it would free up all previously allocated memory to flash.

```
;DEVICEHIGH=C:\CARDWIZ\MTAA.EXE (3K)
;DEVICEHIGH=C:\CARDWIZ\MATB.EXE (3K)
;DEVICEHIGH=C:\CARDWIZ\MTI1.EXE (5K)
;DEVICEHIGH=C:\CARDWIZ\MTI2P.EXE (4K)
;DEVICEHIGH=C:\CARDWIZ\MTATM.EXE (5K)
;DEVICEHIGH=C:\CARDWIZ\MTDDR.V.EXE (10K)
;DEVICEHIGH=C:\CARDWIZ\FTL.EXE (31K)
```

## Attaining the Most Base Memory

Please refer to Section I of the original Memory Optimization White Paper for further items that can help increase the available amount of base memory. In addition to the items documented in Section I please note:

1. PCCARD.EXE has been replaced with CONFIG.EXE
2. Memory changes made in the CONFIG.SYS must manually be changed in the CSALLOC.INI and the SYSTEM.INI.

### Optimize the PCMCIA Reserved Window

The next step is to move the otherwise *PCMCIA Reserved* Memory window. Out of the box, Compaq PCMCIA-based PCs reserve a 16KB window from D100 through D5FF for cards to reside in.

Step	Action	What is Gained?	Notes
1.	Determine how much memory each of your PC Cards require in order to initialize.	If your memory requirements are relatively small, you can usually move the PCMCIA window out of the D000 range, allowing for larger contiguous UMB's.	Remember, at a minimum, the PCMCIA reserved area will need to be 4KB for Card Services use.
2.	Modify your CONFIG.SYS by moving the PCMCIA card initialization area from the D100 range. Hence, the default EMM386.EXE line could be modified from X=D100-D5FF to X=CC00-D0FF or X=D800-DFFF (if MPEG option did not interfere.)	This will ultimately give you more contiguous UMB's between D100-DFFF or CC00-D7FF respectively.	The exclusion range can be moved to where ever there is available memory. According to our card needs the exclusion range could be as little as 4K such as X=D800-D900. To see what memory areas are free look at the memory map on page 5.
3.	Modify CSALLOC.INI by adding the MEMINCLUDE entry to exclude the newly defined memory range. Example of memory exclusion range of CC00-D0FF:  MEMINCLUDE=CC00-D0FF MEMEXCLUDE=C000-CBFF,D100-DFFF,E000-EFFF	This will tell Card Services to use the memory CC00-D0FF to configure PC Cards.	If the 16KB excluded on the EMM386 line is modified to the example X=CC00-D0FF, make sure to modify the MEMEXCLUDE entry to match memory settings.
4.	Modify the [386Enh] section of the SYSTEM.INI to include an EMMEXCLUDE statement that matches the range excluded on the EMM386.EXE line in the CONFIG.SYS file. For this example, the line must read: EMMEXCLUDE=CC00-D0FF		

*\*Note: Normally the PCMCIA Reserved window will be increased or decreased in 4KB increments..*

## SPECIFIC EXAMPLE

### IBM Token Ring & LTE 5000 Family

Listed immediately below is a Summary of the configuration steps needed to get an IBM Token Ring Card to configure in a DOS / Windows 3.1 environment. (This fix was only tested on Netware clients and is not a general overall fix.) The specific steps follow the Summary:

#### DOS/Windows 3.1 Summary

- Steps 1-4: The purpose of these steps is to increase the memory range for use by the PCMCIA services. The IBM card requires a 8K block of memory followed by a 16K block, and by default we only enable a 16K block. This change will allocate a 28K block for the services. As you can see, the card requires 24K to operate, but because it can only start on a 16K boundary, we have to allocate 28K to get it to initialize.
- Steps 5-8: When you edit the CONFIG.SYS and change memory allocation, the CSALLOC.INI does not automatically get updated. It requires a manual change, and these steps simply change the file to reflect the changes in the CONFIG.SYS.
- Steps 9-17: The IBM card requires a specific IO address to operate. Since this causes a conflict with the internal sound card, the addresss must be changed. The hardware is designed to take this into account, but the PCMCIA software is shipped out with a 10-bit decode which will not free up this address. Changing the software from a 10--bit to 12-bit decode will free up the address.
- Steps 18-22: Changes in the CONFIG.SYS and CSALLOC.INI only affect DOS, so you have to manually change a Window initialization file in order to reflect these changes. These steps simply carry out this process, and they update the amount of memory available for use in Windows.

#### **Configuration Steps for IBM Token Ring PC Card and Enabler:**

1. Exit out of Windows completely to a DOS prompt
2. Type **CD\** and press <ENTER>
3. Type **EDIT CONFIG.SYS** and press <ENTER>, using arrow keys move to the EMM386 line and change the exclusion D100-D5FF to **D100-D7FF**.
4. Press <Alt-F> then <X> for exit, should give an error message, press <Y> for yes
5. Type **CD\CARDWIZ** and <ENTER>
6. Type **EDIT CSALLOC.INI** and press <ENTER>, using the arrow keys change MEM=D100-D5FF to **MEM=D100-D7FF**.
7. Now move down and change MEMEXCLUDE=C000-D0FF,D600-DFFF,E000-EFFF to **MEMEXCLUDE=C000-D0FF,D800-DFFF,E000-EFFF**
8. Press <Alt-F> then <X> for exit, should give an error message, press <Y> for yes
9. Type **CONFIG/ALL** and press <ENTER>
10. Press <Alt-U> then <B>
11. A screen called Binary Application should now be showing giving you 3 options, press <Alt-O> for open
12. Use arrow keys and highlight CSALLOC.EXE and press <ENTER>
13. Use arrow keys and highlight /IODECODE and press <Alt-E> for edit
14. Type the letter **C** and press <ENTER>

15. Press <ENTER> twice, and you should now see the Binary Adaptation screen again
16. Press <Alt-S> to save the configuration
17. Press <ESC>, and then press <Alt-F> then <X>
18. Type **CD\WINDOWS** and press <ENTER>
19. Type **EDIT SYSTEM.INI** and press <ENTER>
20. Using the arrow keys scroll down to the section labeled [386Enh]
21. In this section you should see the line **EMMEXCLUDE=D100-D5FF**, now change this line to **EMMEXCLUDE=D100-D7FF**
22. Press <Alt-F> then <X> for exit, should give a warning message, press <Y> for yes
23. Reboot computer, you should now be able to load the Token Ring drivers.

### Enabler Support

IBM provides an enabler called POINTTR.EXE with the IBM Token Ring card. It can be loaded from a command line or it can be loaded in the CONFIG.SYS file. Our recommendation is to load it from the CONFIG.SYS. In order to use the enabler in the CONFIG.SYS, the following must be met:

1. **It must be the first line in the CONFIG.SYS**
2. **You cannot load ANY Card & Socket services**

Compaq's analysis on the enabler was limited to Novell 3.12. According to IBM, the enabler should support most DOS-based NOSs such as Novell, Banyan, etc. IBM also said it should work in Windows for Workgroups.

The platform used with the enabler was an LTE 5200 running DOS 6.2. The enabler was used in and out of the expansion base.

#### **The command line for use out of the base:**

```
DEVICE=C:\NETWARE\POINTTR.EXE SA RS=16 MMIO=D200 SRAM=D400,16 IRQ=B
```

SA - Socket 1, you would use SB for socket 2

RS - Ring speed, if you wanted 4mps RS=4

MMIO - Memory I/O address, it must start on a 16K boundary

SRAM - Shared memory, it must start on a 16K boundary, and the 16 stands for the byte size

IRQ - The interrupt you want to use, and it uses hexadecimal numbering logic so B=11

#### **The command line for the card in one of the PC Card slots in the base would be the following:**

```
DEVICE=C:\NETWARE\POINTTR.EXE SA RS=16 MMIO=D200 SRAM=D400,16 IRQ=B PCIC=03E2
```

It uses the same parameters, but the difference is the socket designation and the PCIC. The socket in the case was really 3, but due to the fact the enabler sees the second controller in the base as a separate controller, it identifies that it has socket 1 & socket 2. With this in mind, socket 3 will equal A, and socket 4 will equal B. The PCIC line stipulates the memory address of the second controller, and in order **to access the controller and slots in the base you must include the line above.**

**To configure the card , load LSL, TOKENCS, IPXODI, and NETX. The following NET.CFG was used:**

```
Link driver TOKENCS
  INT 11
  MEM #1 D2000
  MEM #2 D4000
```

**NOTE: The enabler will not support power management, standby, or hibernation.**

The drivers used were the latest from the IBM bulletin board @ 919-517-0001. The file to download is TRCC.ZIP. With the enabler, approximately 602K will be available before loading the network drivers, and after loading drivers approximately 540K will be available. These memory window changes reflect the removal of the PCMCIA exclusion range, and specifically take into account Compaq's default CONFIG.SYS.

### **Windows 95**

The following information is based on results achieved when testing the drivers in Windows 95 with a NOVELL and NT network, and transferring a 30MB file across the network.

The drivers that currently ship with Windows 95 do not fully support the IBM Token Ring card. IBM is aware of the problem and has Beta drivers available.

Note: According to IBM, customers are referred to Technical Support for assistance; however, open access to the drivers is not available, and a special password is required to access the file. Some of IBM's OEM partners may have access to an OEM bulletin board where they can download the file IBMTOK.ZIP.

### **If the card does not work after installing the drivers:**

1. Go to the Control Panel and START SYSTEM
2. Double click NETWORK ADAPTERS, and then double click IBM AUTO 16/4 TOKEN RING.
3. A pop-up Window with 2 tabs will appear. Click on the one labeled RESOURCES.
4. Click the CHECK BOX and *deselect* AUTOMATIC CONFIGURATION.
5. Click the MANUAL CONFIGURATION button.

### **The following resources can be used to configure the card:**

Memory Range: 000CC00-000CDFF

Memory Range: 000D000-000D3FF

Input/Output: 0A20-0A23

Interrupt: 11

Click OK and the machine should now give you successful configuration tones and the card should be configured.

**Note:** During the original installation the first memory window will allow you to change the increments in 8K blocks. The second memory window will allow 16K block changes. If you want to change the memory configuration in the *future*, you must use SYSTEM to remove the driver from the system and reconfigure the card as a first time insertion. This requirement is due to the fact that when an attempt is made to reconfigure the card, Windows 95 allows only 4K blocks on the second memory address. Upon initial configuration, however, the drivers are in control, and they have the ability to establish 8K, 16K, 32K, or 64K blocks of memory.