

CardSoft™

Technical Reference Guide

PC Card Compliant Software

Revision 2.00
March 1997

Copyright (c) 1992-1997, SystemSoft Corporation. All Rights Reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form, or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of SystemSoft Corporation.

Disclaimer

SystemSoft Corporation provides this document and the programs "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

This document could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in future revisions of this document. SystemSoft Corporation is under no obligation to notify any person of the changes.

The following trademarks are used in this document:

CardID is a trademark of SystemSoft Corporation.

CardSoft is a trademark of SystemSoft Corporation.

CardWorks is a trademark of SystemSoft Corporation.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

PCMCIA is a trademark of Personal Computer Memory Card International Association.

SYSTEMSOFT is a registered trademark of SystemSoft Corporation.

**CardSoft
Technical Reference
Document Revision: 2.00
Part Number 2-020-10**

March 1997

Table of Contents

Preface	viii
Reference Documentation	viii
Chapter 1	
CardSoft PC Card Software Suite	
Socket Services	1-1
Card Services (CS.SYS)	1-1
Card Identification (CARDID.SYS)	1-1
SRAM Card Driver (MTRAMEXE)	1-1
IDE/ATA Support (ATADRV.EXE and ATAINIT.EXE)	1-1
Card Services Power Management Enabler (CS_APM.EXE)	1-1
Windows Support Drivers	1-2
CardSoft Utilities and Commands	1-2
Windows Support Installation Utility (SETUP.EXE)	1-2
Configuration Utility (CONFIG.EXE)	1-2
Resource Allocation Utility (CSALLOC.EXE)	1-2
Card Information Utility (CARDINFO.EXE)	1-2
Chapter 2	
Using the DOS Configuration Utility	
Using CSALLOC	2-1
When to Run CSALLOC	2-2
How to Run CSALLOC	2-2
Using CARDINFO	2-3
When to Run CARDINFO	2-3
How to Run CARDINFO	2-3
Sample CARDINFO Information	2-4
Card Information Using Windows	2-5
Obtaining Card Information Using CardSoft	2-6
Using the Configuration Utility	2-7
Configuration Utility Help	2-7
Running the Configuration Utility	2-7
Accessing Menu Items	2-7
Editing PC Card Configuration Settings	2-8
Modem Card Configuration	2-9
COM Port Assignment Order	2-9
COM Port Definition	2-10
COM Port Order	2-10
Audio	2-10
Ring Indicate	2-10
Network Card Information	2-11
ATA Card Configuration	2-11
Saving PC Card Configuration	2-12
Changing the Configuration Utility Display Mode	2-12
Using Advanced Configuration Features	2-12
Accessing Menu Items	2-13
Card Recognition/Configuration Process	2-13
Creating a New Card Library File	2-14
Opening a Card Library File	2-14
Opening a Second Card Library File	2-14
Viewing a Card Library File	2-14
Creating New Card Library Records	2-14
Viewing a Card Library Record	2-15
Deleting Records From a Card Library File	2-15
Copying Records From a Card Library File	2-15
Saving a Card Library File	2-15
Closing a Card Library File	2-15
Exiting the Configuration Utility	2-15

Editing Card Recognition Parameters.....	2-15
Recognition Algorithms.....	2-17
CIS Manufacturer/Product Data	2-17
CIS Literal Bytes	2-17
Generic Recognition by Card Type	2-17
Custom Code	2-17
Defining the Card Recognition Configuration	2-18
Configuration Algorithms	2-18
Reject This Card.....	2-18
Do Not Configure, or Use Custom Code only.....	2-18
Standard Configuration.....	2-18
Specify Configuration Parameters	2-18
Using Custom Code	2-18
Defining Specific Card Recognition Parameters.....	2-19
CONFIG Tuple.....	2-19
CFTABLE_ENTRY Tuple	2-20
Defining Parameters for Executable Files.....	2-22
Dumping the Parameter Values Into CONFIG.LST.....	2-23

Chapter 3

Configuring Driver Parameters

DEVICEHIGH Line Parameters.....	3-1
Socket Services.....	3-1
Card Services (cs.sys)	3-47
CSALLOC	3-47
SRAM Driver (MTRAM.EXE).....	3-48
Memory Technology Driver (MTDDR.V.EXE).....	3-48
IDE/ATA Device Driver (atadv.exe).....	3-48
Configuration Parameters	3-49
ATA Driver Modes	3-49
DEVICEHIGH Line Parameters	3-50
Disk Partitioning Utility (atainit.exe).....	3-51
ATAINIT Return Code	3-51
Language Parameter.....	3-51
Flash File System Executables	3-52
Card-Specific Memory Technology Drivers	3-52
MTAA.EXE, MTAB.EXE, MTI1.EXE, MTI2P.EXE, MTMGDD.EXE, MTHB2.....	3-52

**Chapter 4
Customizing Your System**

Using Keywords to Modify System Resource Assignments 4-1
 MEM, IO, and IRQ Lines in CSALLOC.INI..... 4-2
 IOEXCLUDE..... 4-2
 IOINCLUDE..... 4-2
 IRQEXCLUDE..... 4-2
 IRQINCLUDE 4-3
 MEMEXCLUDE 4-3
 MEMINCLUDE 4-3
 RIO 4-3
 RIRQ..... 4-4
 RMEM..... 4-4
 SIO..... 4-4
 SIRQ..... 4-4
 SMEM 4-5
 Using CONFIG to Modify Resource Assignments 4-5
 Editing Modified Resources List 4-6
 Using a Customized .INI File 4-7
 System Beeps..... 4-7
 Configuring System Beeps..... 4-8
 Resolving Card-Specific Problems Using CardSoft 4-9
 Procedure 4-9
 Reference 4-10
 Canceling All Settings 4-11
 Customizing CONFIG.SYS 4-12
 Required DEVICEHIGH Lines..... 4-12
 DEVICEHIGH Lines Required for ATA Card Support..... 4-12
 DEVICEHIGH Lines Required for SRAM Card Support..... 4-12
 DEVICEHIGH Lines Required for Flash Memory Card Support..... 4-12
 If You Have DOS 6.0 4-13

Appendix A

CardID

The [Common] Section..... A-1
 The [Libraries] Section..... A-4
 Comment Record A-4
 Card Services Resource Record A-4
 Card Recognition and Configuration Record..... A-5
 Card Recognition Fields A-6
 Card Configuration Fields..... A-8

Appendix B

Card Services Vendor-Specific API Definitions

Subcomponent IDs B-1
 RegisterSpecialClient..... B-2
 SpecialGetStatus B-3
 Suspend B-4
 Resume B-5
 Socket Suspend..... B-6
 Socket Resume..... B-7
 Ring Resume..... B-8
 BeepControl..... B-9
 GetWindowInfo B-10

Appendix C

Socket Services Vendor-Specific API Definitions

Subcomponent IDs C-1
 Vendor Specific Functions C-1
 Set/Reset DMA Mode..... C-1
 Re-Initialize Adapter After Resume..... C-2
 Set/Reset Ring Indicate Mode C-2

Appendix D

SystemSoft *CardSoft* API for Windows

Component Descriptions D-2
 Card Services Callout D-2
 Card Services Callback..... D-4

Appendix E

Extensions to the Card Services Specification Provided by *CardID*

GetClientInfo Subfunctions E-1
 GetGCIExtensionsInfo (Info Subfunction 80h) E-2
 GetCardID (Info Subfunction 81h)..... E-3
 Reserved Functions..... E-3
 GetCardIDArgList (Subfunction 86h) E-4
 SetCardIDArgList (Subfunction 87h)..... E-6
 GetLastError (Subfunction 8Fh)..... E-6

Appendix F

Card Services Vendor-Specific Callbacks

Socket Configuration (83h) F-1

Preface

This document is written for software developers who need detailed information on SystemSoft's *CardSoft* software suite. The information contained within this document is intended for a technical audience. For instructions on installing and using the *CardSoft* software, refer to the *CardSoft Quick Reference Guide*.

Reference Documentation

The following are available from PCMCIA:

- PC Card Standard Overview and Glossary
- PC Card Standard Electrical Specification
- PC Card Standard Physical Specification
- PC Card Standard Metaformat Specification
- PC Card Standard Card Services Specification
- PC Card Standard Socket Services Specification
- PC Card Standard PC Card ATA Specification
- PC Card Standard Media Storage Formats Specification
- PC Card Standard XIP Specification
- PC Card Standard Guidelines
- PC Card Standard PCMCIA Specific Extensions
- PC Card Standard JEIDA Specific Extensions

Chapter 1

CardSoft PC Card Software Suite

SystemSoft's *CardSoft* PC Card solution provides OEMs with a complete software solution for integrating PCMCIA controllers and slots into their computers. The *CardSoft* software suite provides a complete "plug and play" system software solution for both DOS and Windows™ 3.1. This solution consists of the following drivers and utilities. Please be aware that your particular configuration may not include all drivers and utilities.

Socket Services

Socket Services provides a standard software interface to host controller chips and isolates the socket hardware from higher level software. Socket Services includes functions such as configuring a socket for an I/O or memory interface and controlling socket power voltages. The Socket Services driver included depends upon the host controller chip that the system supports. For information on the controllers supported, check out the SystemSoft web site at www.systemsoft.com

Card Services (CS.SYS)

The Card Services driver manages competition for system resources and manages adapter and card resources and configuration.

Card Identification (CARDID.SYS)

This client device driver detects the insertion and removal of PC cards, automatically determines the card type upon insertion, and then configures the card and slot/adaptor (if it is an I/O Card).

SRAM Card Driver (MTSRAM.EXE)

This SystemSoft device driver recognizes and supports SRAM cards.

IDE/ATA Support (ATADRV.EXE and ATAINIT.EXE)

ATADRV.EXE is a block device driver that supports ATA Type II Flash Disk or ATA Type III Hard Disk PC cards. The ATAINIT.EXE utility places a hard disk partition table onto a blank ATA drive. This utility initializes the ATA drive in the same manner as FDISK does with a standard hard drive.

Card Services Power Management Enabler (CS_APM.EXE)

CS_APM.EXE is a DOS-based background task that enables Card Services to process system power management Suspend/Resume requests. When a Suspend request is initiated by system power management software, CS_APM notifies Card Services, which then verifies that the system PCMCIA slots are idle, and can be powered down. Card Services then passes this information back to CS_APM, which then notifies the power management software that the sockets can be powered off. When a Resume request is received by CS_APM, it informs Card Services, which then powers the sockets on again.

Windows Support Drivers

SSVCD.386 (SSVCD311.386 for Windows for Workgroups), SSVRDD.386, PCCARD.386

These drivers permit hot insertion/removal of communications I/O, memory, and removable drive cards within Windows.

DOS16M.386

This DOS16M.386 driver allows Card Services (CS.SYS) to run under Windows 3.1x. CS.SYS is now a protected mode driver which requires the use of the Central Processing Unit's Global Description Table (GDT) and Interrupt Descriptor Table (IDT). After Windows loads, this driver allows CS.SYS to use the original tables for proper operation.

CardSoft Utilities and Commands

Windows Support Installation Utility (SETUP.EXE)

This installation utility automatically installs the Windows support files and DLL file, and automatically configures your Windows environment for *CardSoft* by modifying the Windows SYSTEM.INI file. Refer to the *CardSoft Quick Reference Guide* for detailed information on the Windows installation of this product.

Configuration Utility (CONFIG.EXE)

The basic Configuration Utility enables you to

- Define the general configuration settings for fax/modem, ATA, and network cards.
- Quickly view and change the system resources that are made available to Card Services.
- Select the PCMCIA events for which a system beep code will be emitted.

In addition, a Configuration Utility for OEM developers enables you to:

- Create or modify card records that define configuration and recognition parameters for specific PC cards, and save these records in a Card Library.
- Configure *CardSoft* to your specific system requirements by modifying the various parameters for the *CardSoft* executable files (making it unnecessary to configure *CardSoft* with command line/device line switches).

This version of the Configuration Utility enables OEMs to more easily define the configuration and recognition of new PC cards. Refer to Chapter 2 of this manual for detailed information on the Configuration Utility.

Resource Allocation Utility (CSALLOC.EXE)

CSALLOC is a device driver and command line utility that scans the system for available memory, I/O port, and IRQ resources, updates the file CSALLOC.INI with this information, then displays the list of available resources that can be used by Card Services. In addition, keywords can be added to CSALLOC.INI to force CSALLOC to exclude resources from being scanned, or include resources that are not detectable (due to security features or hardware design). Other keywords enable you to reserve resources that may be specifically needed by a particular PC card or other system component.

You can also reserve, include, or exclude an I/O range, IRQ, or memory address by using the *CardSoft* in Windows. See Chapter 2 of this manual for further information.

Card Information Utility (CARDINFO.EXE)

CARDINFO is a DOS utility that scans the PCMCIA slots on the system and lists information about the cards in these slots. It also lists any warnings or error messages that may have occurred during the configuration of the cards by the *CardSoft* drivers.

Card information can also be obtained by using the *CardSoft* in Windows. See Chapter 2 of this manual for further information.

Chapter 2

Using the DOS Configuration Utility

This chapter describes how to use the Configuration Utility to configure the system's PC card subsystem, and to create and manage card library records that contain the configuration settings and tuple information required for the card to work properly.

The basic Configuration Utility, *config*, enables you to:

- Define the general configuration settings for fax/modem, modem, ATA drive, and network cards.
- Quickly view and change the system resources that are made available to Card Services.
- Select the PCMCIA events for which system beep codes will be emitted.

In addition, a Configuration Utility for OEM developers, *config /all*, enables you to

- Create or modify card records that define configuration and recognition parameters for specific PC Cards, and save these records in a Card Library.
- Configure *CardSoft* to your specific system requirements by modifying the various parameters for the *CardSoft* executable files (making it unnecessary to configure *CardSoft* with command line/device line switches).

This version of the Configuration Utility enables OEMs to more easily define the configuration and recognition of new PC cards.

The Configuration Utility makes modifications to the CARDID.INI and CSALLOC.INI files (the initiator files for *CardID* and CSALLOC). When modifications are made to any of the *CardSoft* components, the time and date of the filename is changed to reflect the change (the size of the file, however, never changes).

Note *Any time modifications are made to CARDID.INI or CSALLOC.INI, you must reboot the system for the changes to take effect.*

This section describes how to use the *CardSoft* utilities and commands (CONFIG, CSALLOC, and CARDINFO) to help you configure and manage PC cards on your system.

Using CSALLOC

CSALLOC is a device driver and a DOS program that scans the system for available memory (MEM), I/O Port (IOP), and Interrupt Request Line (IRQ) resources. It is also able to write this information to the file CSALLOC.INI, which is used by *CardSoft* to determine what system resources can be used by your PC cards.

For example, other hardware or software on your system may need to use certain system resources in order to work properly. If *CardSoft* were to use these resources, then the hardware (or software) might not work. CSALLOC makes sure that *CardSoft* does not use resources that are needed for something else.

In addition, CSALLOC:

- Scans upper memory for Read Only Memory (ROM)
- Scans for the Extended Memory driver
- Checks for the presence of HIGH memory
- Checks for the availability of Upper Memory Blocks (UMBs)
- Checks for the presence of the XMS driver.
- Enables you to display the current resource status of *CardSoft*.
- Directly checks PNP ISA BIOS for I/O and IRQ settings for motherboard devices (no scan necessary).

When to Run CSALLOC

During installation, a CSALLOC.INI file is created. As a result, you need to run CSALLOC from the *CardSoft* subdirectory (c:\CardSoft\csalloc) only when you need to display which system resources are being made available to *CardSoft*. Also, if you have changed I/O Port, IRQ, or memory settings on your system (for example, if you change your COM Port settings in Windows, or you enable the XMS memory manager driver), you should run CSALLOC to scan the system for available resources and create an updated CSALLOC.INI. You should also run CSALLOC whenever you install new hardware or software that requires specific system resources, so that the resources required by the newly-installed item do not conflict with *CardSoft*.

How to Run CSALLOC

CSALLOC can only be run from the DOS command line. To run it, change to the *CardSoft* directory (cd CardSoft), then type one of the following commands (shown in bold type) and press Enter:

csalloc/h or csalloc/?

Will provide a listing of available CSALLOC switches.

csalloc /s

You should use this command only if your system contains a Plug-and-Play (PnP) BIOS and you do not want to rely on the BIOS to determine system resource usage. When you enter this command, CSALLOC forces a scan of system resources, instead of relying on the PnP BIOS to inform CSALLOC as to how system resources are assigned.

csalloc /r

When you enter this command, CSALLOC displays the current status of memory (MEM), I/O Port address (IOP), and IRQ resources on your system, as shown in the following example. Resources marked with an (R) are reserved for use by your PC cards or other system components (for example, 3F8-3FF is always reserved, since it is assigned to system COM Port 1). Resources marked with an (A) are already allocated for use by *CardSoft*. Resources marked with an (S) are shared resources (that is, they can be shared by several system components without creating a conflict).

Example:

MEM:

D000-DFFF

IOP:

108-16F, 170-177(R), 178-1EF, 1F8-2E7, 2E8-2EF(R), 2F0-2F7, 2F8-2FF (R), 300-36F, 370-377 (R),
380-3BF, 3E8-3EF(R), 508-5EF (R), 5F8-6F6 (R), 6F7-777(R), 780-7BF(R), 7E8-7EF(R), 908-9EF(R),
9F8-AF6 (R), AF7-B77(R), B80-BBF(R), BE8-BEF(R), D08-DEF(R), DF8-EF6(R), EF7-F77(R),
F80-FBF(R), FE8-FEF(R)

IRQ:

3, 5, 9, A, B, C(R), D

Notes For a description of the resource lines (MEM, IO, etc.), see Chapter 4.

You can cause resources to be included, excluded, reserved, or shared, either by modifying CSALLOC.INI, or by using the Configuration Utility. For more information, including modifying resources using the CardSoft (in Windows) see Chapter 4.

Using CARDINFO

CARDINFO is a DOS program that scans the PCMCIA slots on your computer and lists information about the cards in these slots. Also, if any of the slots contain cards when you start your computer, CARDINFO lists any warnings or error messages that may have occurred when *CardSoft* attempted to configure these cards (in Real/CardSoft mode only).

When to Run CARDINFO

CARDINFO can be run at any time. Normally, you should run CARDINFO if:

- You need to know the types of cards that are currently inserted in your PCMCIA slots.
- You need to know the I/O Ports, IRQs, and Memory areas that are being used by your PC cards. This information may be needed if you are installing other components on your system, and you need to know if there is going to be a conflict between the new component and a PC card that you are currently using. ("Conflict" means that the new component and a PC card are trying to use the same resource, such as an IRQ interrupt; if this happens, either the PC card or the new component would have to be reconfigured to use other resources that are available.)
- You want to turn off (or turn on) power to a PCMCIA slot that contains a PC card.
- You want to display manufacturer and product information about your PC cards.
- You need to know the drive letter for your ATA Hard Disk or ATA Flash Disk or memory cards.
- You need to know the latest error that occurred for an inserted card.

How to Run CARDINFO

CARDINFO can be run in several modes from the *CardSoft* subdirectory DOS command line. To run it, change to your *CardSoft* directory (`cd CardSoft`), then type one of the following commands at the DOS prompt. CARDINFO can be run from within a Microsoft Windows DOS window; you do not have to exit Windows to run CARDINFO.

`cardinfo` Runs CARDINFO in non-verbose mode. When you enter this command at the DOS prompt, information similar to the following is displayed:

Slot 0

Manufacturer = Q293-C
Product Name = DATA/FAX MODEM
Card Type = Modem (COM 3)

Slot 1

Slot 1 is empty

`cardinfo /v | more` Runs CARDINFO in verbose mode, which displays more extensive information about the PCMCIA slots in your computer. (We recommend that you use the DOS "more" switch with this command, since the information may require more than one screen to be displayed). The following section contains an example of CARDINFO information using the `/v` switch. The type of information that is displayed depends on the type of cards that your PCMCIA slots contain.

<code>cardinfo /c</code>	Provides the following additional Card Services information. <ul style="list-style-type: none"> - Card Services Release Number - Vendor Revision Number - Number of slots - Vendor copyright information <p>This command gives you the ability to view manufacturer and product information for PC card client software other than the <i>CardID</i> component of <i>CardSoft</i>.</p>
<code>cardinfo /off <:slot></code>	Turns off power to all PCMCIA slots.
<code>cardinfo /on <:slot></code>	Turns on power to all PCMCIA slots. For both of these commands, you have the option to specify a slot number. For example, <code>cardinfo /off:1</code> turns off power to Slot 1 only. This enables you to turn off power to any slot not in use, without removing the card in that slot, thereby conserving system power.
<code>cardinfo /?</code>	Displays information about currently inserted PCMCIA card(s).
	<pre> /c display Card Services information /off turn power off for all slots /on turn power on for all slots /off:s turn power off for slot s /on:s turn power on for slot s /off:s,f turn power off for slot s, function f /on:s,f turn power on for slot s, function f /v display additional information about slot configuration </pre>

Sample CARDINFO Information

This section lists an example of the type of information that is displayed when you enter the `cardinfo /v | more` command. The exact information that is listed depends on the PC cards that are currently inserted in the system, and the number of PCMCIA slots in the system.

Note *If you have CardSoft with optional Microsoft Flash File System support, information on each of the card-specific Memory Technology Drivers (MTDs) is also displayed.*

```

Client Information for handle 9B67:
Client Revision = 0.01
CS Support Level = 2.1
Revision Date = 02-07-1994
Client Name = "SRAM MTD"
Vendor Name = "SystemSoft Corporation"

Client Information for handle 9B3E:
Client Revision = 0.01
CS Support Level = 2.1
Revision Date = 03-14-1994
Client Name = "MTDDRIV"
Vendor Name = "SystemSoft Corporation"

Slot 0:
[Card Information]
Card Type = "ATA Disk" (Drive D:)
Manufacturer = <Vendor Name>
Product Name = <Vendor Product Name>

[Configuration Info]
Configuring client handle is 94D7
Memory + I/O Interface, Vcc 50, Vpp1 120, Vpp2 120
Config base 0200, Config values:
  Option value: 42
  Copy value: 00
First I/O Range 170-177, 16-bit
Second I/O Range 378-37F, 16-bit
No IRQ is assigned to this card.

```

Card Information Using Windows

The Card Wizard **Card Information** screen is the main *CardSoft* screen. It displays up-to-date information about all the PCMCIA slots in your computer. *CardSoft* displays a graphical representation of the contents of each slot. Additionally, *CardSoft* displays up to three lines of card information which it reads directly off of the card. Typically, the first line displays the card name, the second line displays the manufacturer, and the last line displays the card's function or type.

If you have a multi-function card installed, you will see up to two functions at a time and can scroll to see available functions if more than two functions are activated on your card. Functions will be labeled by slot number followed by function number. For example 0-1 would represent Slot 0, Function number 1 and 2-1 would represent Slot 2, Function number 1. Note that a single zero (0) would represent Slot 0, Function 0.

The following types of cards may appear:

- Modem
- Network
- ATA
- Flash
- SRAM
- IBM 3270 Emulator
- Other (for cards that are not configured and not recognized)
- Unknown (where *CardSoft* cannot identify either the card name or card type)

The Card Information screen displays information for four potential scenarios:

Card Configured: On the left, *CardSoft* displays a graphical representation of the card and any available card manufacturer information. On the right, *CardSoft* displays all system resources that the card is using.

Card Not Configured: On the left, *CardSoft* shows a picture of a question mark and any available card manufacturer information. On the right, *CardSoft* displays a message suggesting that you click on the "Wizard" button to correct the problem.

Card Not Recognized: On the left, *CardSoft* shows a picture of a question mark and any available card manufacturer information. On the right, *CardSoft* displays a message asking you to click on the "Wizard" button to select the type of card you have inserted.

No Card Inserted: *CardSoft* displays a picture of an empty slot.

The Card Information screen also contains the "Wizard". The Wizard can be activated at any time to diagnose and automatically correct configuration problems. *CardSoft* will go to great lengths to configure your card. Oftentimes, it will open up a dialogue with you to assure that the recommended course of action is the right choice for you.

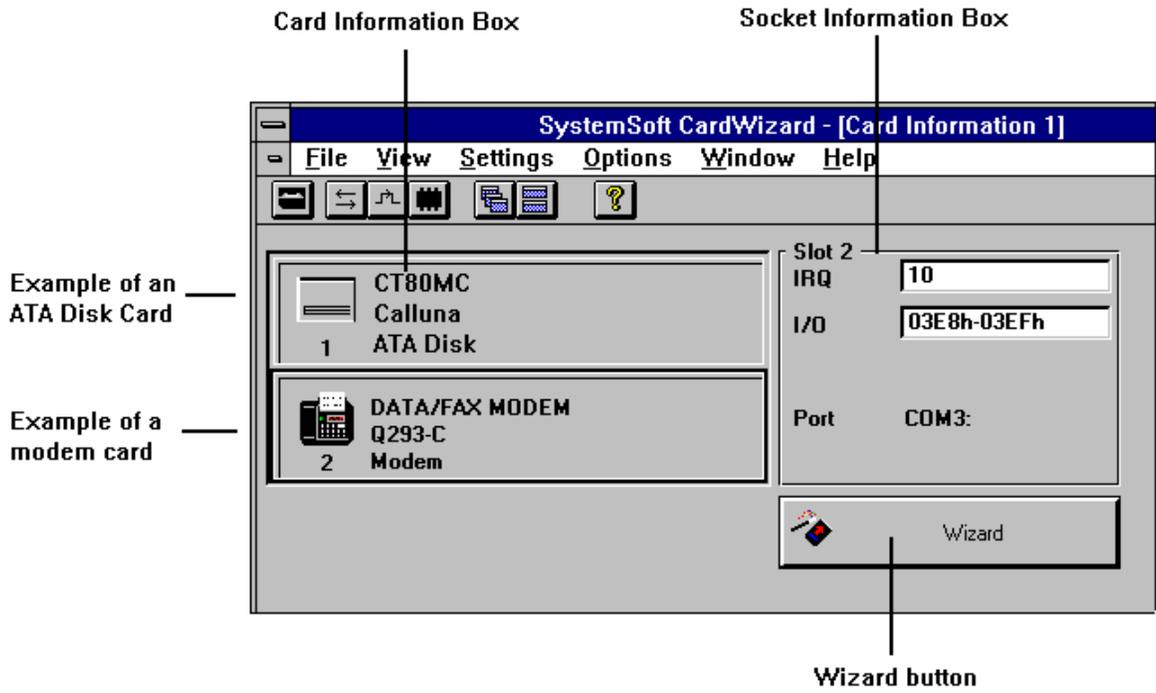
Obtaining Card Information Using CardSoft

You can also gather information about your PC Card by using CardSoft.

To obtain card information about your PC Card:

1. Enter Windows and open CardSoft.
2. Choose Card Information from the View menu.

CardSoft displays the Card Information dialog box. For example, the following illustration shows that two cards have been inserted:



The first card is an ATA hard disk card; the second a modem card. The information above the Wizard button applies to the modem card.

To display similar information about the Calluna card you would choose the ATA Disk button. To display the modem information again, choose the modem button.

Reference

The following summarizes the components of the Card information dialog box.

Field/Button/Option	Description
Card information boxes	These boxes identify the PC card in each socket giving the card's name, its manufacturer and its type. Click on one of the boxes to select the card.
Socket information box	This box shows the IRQ, I/O setting and Port currently assigned to the selected card socket.
Wizard button	Press this button to have CardSoft automatically configure the card

Using the Configuration Utility

The Configuration Utility can be used to

- Set or change the IRQs and COM Port assignment order for your fax/modem cards.
- Set or change the I/O Port address, IRQ, and memory areas that your network cards will use, as well as enable your network cards for a Fast Token Ring configuration.
- Select the address (Primary, Secondary, or Any) that your ATA cards are going to use to communicate with the system.
- Select the type of video display (Color, Monochrome, or LCD) you want to use when you run the Configuration Utility.

Note The Configuration Utility (CONFIG.EXE) modifies the CARDID.INI and CSALLOC.INI files (the initiator files for CardID and CSALLOC). As a result, whenever you are done using this utility, you have to restart your system for any of your changes to take effect.

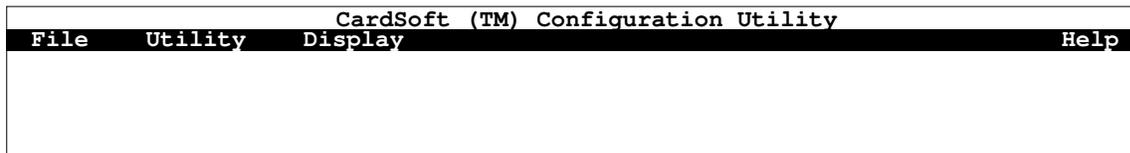
Configuration Utility Help

On-line help is available for many of the Configuration Utility fields. To access on-line help for a particular field, position the cursor in the field (or highlight the field), then press F1 (or click on the <Help> button, if available).

Running the Configuration Utility

You need to run the Configuration Utility **ONLY** if you are having trouble using a PC card, or if you want to customize your system. To run the Configuration Utility:

1. From the DOS prompt (C:\), type `cd CardSoft` to change to the *CardSoft* subdirectory.
2. Type `config` and press Enter. The following screen appears.



The menus on this screen contain the following items:

<u>File</u>	<u>Utility</u>	<u>Display</u>
Edit Configuration	Resource Allocation	Color
Save Configuration		Monochrome
Exit		LCD

Accessing Menu Items

Each menu bar item contains a pull down menu with various items to choose from. To access a pull down menu, click the left mouse button on the menu name, or press Alt + the highlighted key. For example, to access the File menu, click on the word *File* or press <Alt-F> from the keyboard. When the pull down menu appears, select a menu item by:

- Clicking on it with the left mouse button
- Pressing the down arrow key to highlight the item and then pressing <Enter>, or
- Pressing the key that corresponds to the highlighted letter.

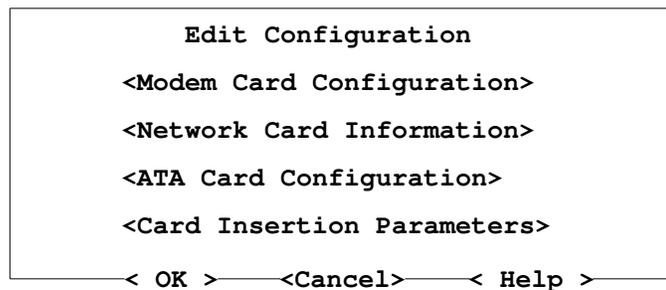
Editing PC Card Configuration Settings

The Configuration Utility enables you to edit the general configuration parameters for fax/modem, modem, network, and ATA cards. These configuration parameters include items such as COM Port, Network Port, and ATA Port information, as well as general parameters (such as having the system beep when a card is inserted into a slot).

Note *If your PC cards are working properly, you do not have to edit the PC card configuration settings. You should edit these settings only if you have a PC card that is not working properly, or if you are an advanced user and you want to custom-configure your system.*

To edit PC card configuration parameters:

1. Select *Edit Configuration* from the File pull down menu. The Edit Configuration menu is displayed.



2. If you are configuring a fax/modem or modem card, select Modem Card Configuration.

If you are configuring a network card, select Network Card Information.

If you are configuring an ATA card, select ATA Card Configuration.

3. Click on <OK>. The following sections describe the screens for each of these functions.

4. When done configuring the port information for the card, select Card Insertion Parameters. The Card Insertion Parameters screen is displayed.

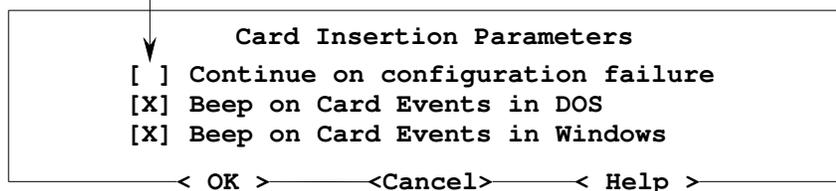
If "Continue on Configuration Failure" is enabled, *CardID* continues to search remaining card library records when a card configuration is attempted and fails.

If "Beep on Card Events in DOS" is selected, the system emits a beep while in DOS whenever a card is inserted or removed, or when there is a configuration conflict.

If "Beep on Card Events in Windows" is selected, the system emits a beep while in Windows whenever a card is inserted or removed, or when there is a configuration conflict.

Note *By default, the events that cause a beep are card insertion, card removal, and configuration or non-configuration conflict. You can, however, manually specify the events that cause a beep, as well as the environment (DOS and/or Windows) in which beeps occur, by adding a /BEEPTYPES switch, followed by a hexadecimal value, to the "devicehigh=c:\CardSoft\cs.sys" line in CONFIG.SYS (see Chapter 4 for more information).*

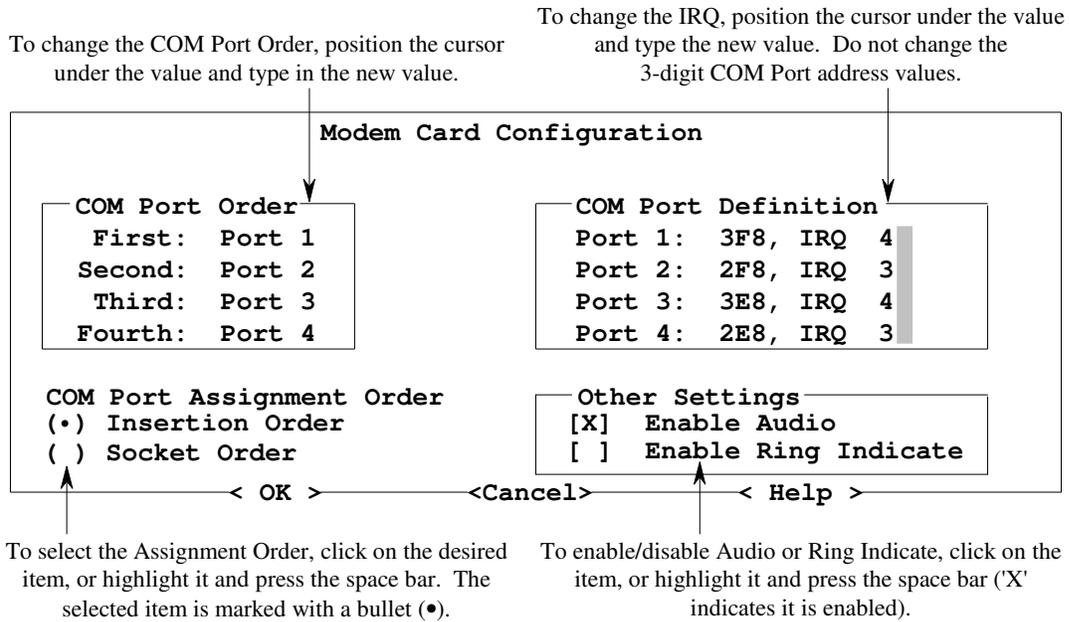
An 'X' indicates that the feature is enabled.



5. When done defining the PC card configuration, click on <OK> from the Edit Configuration screen to return to the main menu.
6. When exiting the Utility, you will be asked if you want to save any changes made.

Modem Card Configuration

The Modem Card Configuration screen is where you define the assignment of communication ports (COM Ports), COM Port addresses, and IRQ interrupts for your modem and fax/modem cards. You can define I/O ports and IRQ values for as many as four COM Ports, and you can define the COM Port Assignment Order (the order in which the ports will be selected for configuring Modem or Fax/Modem cards).

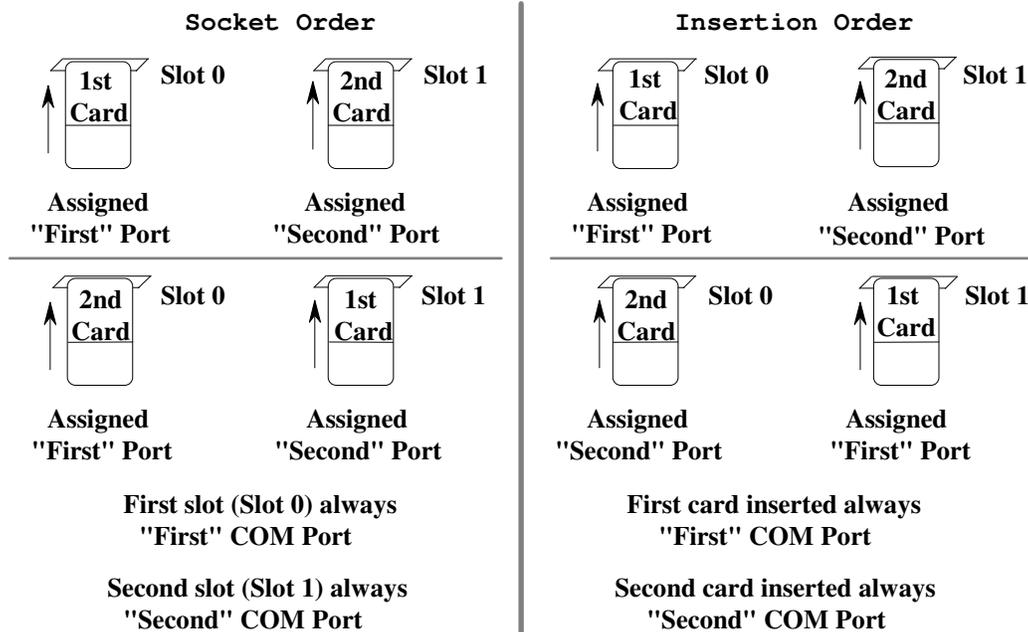


COM Port Assignment Order

The **COM Port Assignment Order** settings determine whether COM ports are assigned based on the order in which cards are inserted, or based on the PCMCIA slot into which they are inserted.

If you select Insertion order, the first modem card you insert into a PCMCIA slot would be configured as the "First" port (as listed in the COM Port Order box), the second inserted card would be configured as the "Second" port, etc. For example, if you insert a modem card in Slot 1, then you insert a card in Slot 0, the Slot 1 card would be configured as the "First" port, and the Slot 0 card would be configured as the "Second" port. In the sample screen on the previous page, the "First" port is Port 1 (COM 1). If you were to change the "First" setting to Port 4, the first card you insert would be assigned the Port 4 (COM 4) values, which are address 2E8 and IRQ 3.

If you select Socket order, the modem card you insert in your first PCMCIA slot (Slot 0) would be configured with the Port 1 values, the modem card you insert in your second PCMCIA slot (Slot 1) would be configured with the Port 2 values, etc. These assignments would be made regardless of the order in which the cards were inserted. For example, if you insert a modem card in Slot 0 after you have already inserted a modem card in Slot 1, the Slot 0 card would be configured with the "First" port values, and the Slot 1 card would be configured with the "Second" port values.



COM Port Definition

The **COM Port Definition** box determines which communication address and IRQ will be used for each COM Port. For example, in the sample screen on the previous page, Port 1 is assigned COM Port address 3F8 and interrupt IRQ 4. You can change the IRQ assigned to each port, but you should never change the COM Port addresses that are listed. If you change the IRQ setting, always change it to a value that is listed as available by CSALLOC. Also, do not change the IRQ to the same value that is assigned to the Network Card IRQ (see "Network Card Information")

COM Port Order

The **COM Port Order** settings determine which COM Ports are "First", "Second", "Third", or "Fourth". For example, the default settings are First=Port 1 (COM 1), Second=Port 2 (COM 2), Third=Port 3 (COM 3), and Fourth=Port 4 (COM 4). You can, however, change the order so that First=Port 2, Second=Port 3, etc. You would do this if you need a particular COM Port for another device (such as a mouse), and you do not want to use that COM Port for a modem card. For example, if you want to reserve COM 1 for your system mouse, enter "Port 1" for "Fourth", and "Port 2" for "First". That way, COM 2 will be the first port that your system tries to use for your modem card.

Audio

When you enable **Audio**, you can hear all the normal audio beeps that are emitted when the modem (or fax/modem) is attempting to connect to a remote modem (or fax/modem). When Audio is disabled, you do not hear these connection beeps.

Ring Indicate

Ring Indicate controls whether or not your system wakes up from a Suspend mode when a call comes in to a modem or fax card. If Ring Indicate is enabled, a system in Suspend mode will wake up from the Suspend state to receive the call. If Ring Indicate is not enabled, a system in Suspend mode will remain suspended, and the call will be ignored.

Note The Ring Indicate feature is not supported by all modem or fax/modem cards or all platforms.

Network Card Information

The Network Card Information screen is where you specify the base I/O address, IRQ, and up to two memory windows for use with a network card. The base I/O address is specified in hexadecimal. Except for rare cases, the value should be between 100h and 3FFh. The IRQ, specified in decimal, may be any value between 3 and 15 that is available on your computer. The memory windows are paragraph values between C000h and EF00h (normally, these values are in the range D000-DFFF, which is the memory range that is excluded from use by your system's memory manager, as described in Chapter 1). Since the last two digits must be zero, you only have to enter the first two digits (normally, the appropriate values to use are D000 for Memory 1 and D400 for Memory 2, or D800 for Memory 1 and DC00 for Memory 2).

Note Unless you have a specific reason for doing so, we recommend that you do not change the displayed values for Port, IRQ, Memory 1, and Memory 2. Refer to the documentation that came with the card to determine if it requires specific settings.

The Fast Token Ring field is used to specify the speed for a Token Ring card. When this field is enabled, the Token Ring speed is set to 16 Megabits per second; when disabled, the Token Ring speed is set to 4 Megabits per second. You should not enable this setting unless you are connected to a Fast Token Ring network (your network administrator can tell you whether or not you are). *Fast Token Ring is not always displayed, depending on the type of system you have.*

```

Network Card Configuration

Port 300 ■ IRQ 5 ■ Memory 1 d800 Memory 2 dc00

[ ] Fast Token Ring

< OK > <Cancel> < Help >

```

To edit the information on this screen, simply position the cursor on the value you want to change, then type in the new value. To enable/disable 'Fast Token Ring', click on the brackets (or position the cursor in the brackets and press Enter); it is enabled when an 'X' is displayed. When done, click on <OK>.

ATA Card Configuration

The ATA Card Configuration screen is where you specify how *CardID* will attempt to configure ATA cards (including Flash disks and rotating-media disks). You can specify whether *CardID* should try to use the standard primary and secondary ATA addresses, and whether it should try to use the linear-address mode available on most ATA cards. All three settings can be enabled at the same time (this causes the software to use whatever is available). It is generally best to avoid selecting only the primary ATA address, unless you are certain there is no disk installed on your computer that uses that address.

An 'X' indicates that the item is enabled.

```

ATA Parameters

[ ] Try Primary ATA Address
[X] Try Secondary ATA Address
[X] Try Any Linear ATA Address

< OK > <Cancel> < Help >

```

To enable/disable an item, click on the item, or highlight it and press the space bar, to toggle the 'X' on/off.

Saving PC Card Configuration

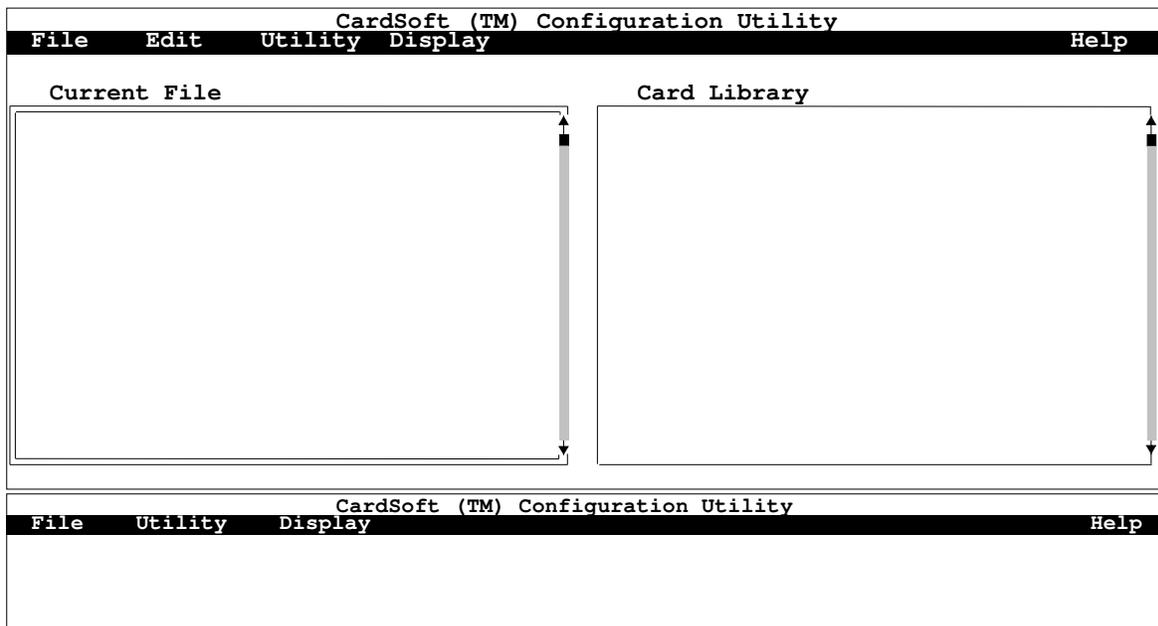
Once you have edited a PC card configuration, you must save your settings by selecting Save Configuration from the File pull down menu. The Save Configuration pick is available only if you have edited the general PC card configuration settings.

Changing the Configuration Utility Display Mode

The Configuration Utility can be displayed in three modes: color, monochrome, or LCD. To change the display mode, select the Display pull down menu, then select the type of display you want to use. The current display mode is indicated by a check mark (✓).

Using Advanced Configuration Features

To run advanced technical features available with the Configuration Utility, change to the directory containing the utility (i.e., `cd CardSoft`), then type `config /all`. If you only want to edit the card configuration settings or run the Resource Allocation utility, type `config`. Depending upon the method used to start the Configuration Utility, one of the following screens is displayed. Notice that without the `/all` switch, the Edit pull down menu is not available.



The initial Configuration Utility screen consists of a menu bar at the top and a help text section at the bottom. The middle of the screen is blank. When a Card Library file is selected, the card records in that file are displayed on the left in the Current File list box. A second Card Library can be opened in view/copy mode; the records in this file are displayed on the right in the Card Library list box.

The following is a sample list of possible Card Libraries:

<u>Card Library</u>	<u>PCMCIA Card(s)</u>
CBDAS.CLB	Computer Boards
CWLAN.CLB	Card Well Ethernet
DLINK.CLB	D-Link Ethernet Lan Card
FDCARD.CLB	Canon Floppy Disk Card
	Compaq Floppy Disk Card
GENATA.CLB	Generic ATA Disk
GENMODEM.CLB	Generic Modem
IBM3270.CLB	IBM 3270
IBMLAN.CLB	IBM Ethernet
IBMTOK.CLB	IBM Token Ring
INTELLAN.CLB	Intel Ethernet Lan Card
LINKSYS.CLB	LinkSys Ethernet Card
	LinkSys Ethernet LAN Card
LINKSYS2.CLB	LinkSys Ethernet Card (2)
PROXIM.CLB	Proxim RangeLan
SOCKETEA.CLB	Socket EA+ Ethernet Card
SUNDISK5.CLB	Sundisk 5 MB
TDKLAN2.CLB	TDK LAC-CD02x Ethernet Card
WD.CLB	WD Caviar UltraLite 40 HD
XIRCOM.CLB	XIRCOM CreditCard LAN
XIRDUAL.CLB	Excludes XIRCOM Ethernet+Modem 2 Card

Accessing Menu Items

The method for accessing menu items is the same for each Configuration Utility. Each menu bar item contains a pull down menu with various items to choose from. To access a pull down menu, click the left mouse button on the desired item or press Alt + the highlighted key. For example, to access the File menu, click on the word *File* or press <Alt-F> from the keyboard. When the pull down menu appears, select the desired item by:

- Clicking on it with the left mouse button
- Pressing the down arrow key to highlight the item and then pressing <Enter>, or
- Pressing the key that corresponds to the highlighted letter.

Card Recognition/Configuration Process

This section describes the basic process for defining card recognition and configuration settings for a specific PC card from within the Configuration Utility. To define these settings, you have to run the *config /all* version of the utility. For specific information on each step in the process, and whether or not additional steps are required for the card being defined, refer to the appropriate sections in this manual.

Note *If you only need to edit the general configuration parameters (that is, those that apply to all cards of a particular type), use the **config** command.*

1. Open an existing card library (File ⇒ Open), or create a new card library (File ⇒ New).
2. Select an existing card library record and edit it (Edit ⇒ Edit Current), or create a new card library record (Edit ⇒ Create New Record). Either method displays the Card Recognition screen.
3. Specify the Record ID, Card Type, and Recognition Algorithm.
4. Select <Edit Recognition Detail> and enter the appropriate information.
5. Select <OK> to display the Card Configuration screen.
6. Specify the Configuration Algorithm.
7. To use custom configuration or removal code, select these items, then select <Specify Custom Configuration Code>/<Specify Custom Removal Code> to enter the file names.
8. Select <Edit Recognition Detail> to display the Specify Configuration Parameters screen.
9. Specify the types of CONFIG and CFTABLE_ENTRY tuples you want to edit by selecting the appropriate <Specify ...> items, then select the matching <Edit ...> items to specify the tuple values.
10. Return to the main menu and save the new record in the card library (File ⇒ Save) or (File ⇒ Save As).

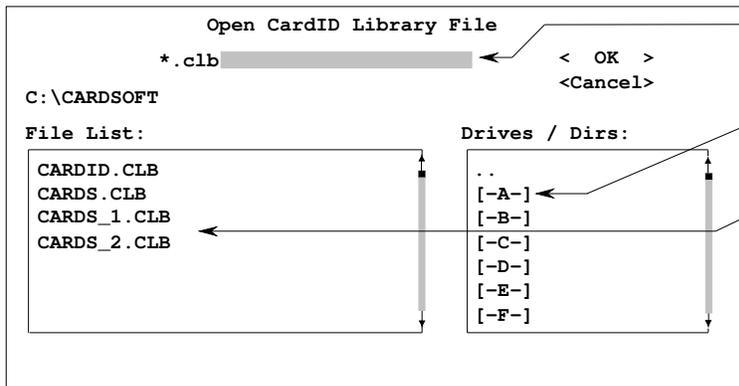
Creating a New Card Library File

New Card Library files can be created through the Configuration Utility. Once a new Card Library file is created, you can copy records into it from other Card Library files, or you can create new records and add them to the file.

To create a new Card Library file, select New from the File pull down menu. The message "Untitled (Card Library)" is displayed above the Current File box on the left side of the main menu. Refer to "Creating New Card Library Records" for information on how to create new records for the file, or refer to "Copying Records from a Card Library File" for information on how to copy records from an existing Card Library to the new Card Library.

Opening a Card Library File

To open an existing Card Library file for an editing session, select Open from the File pull down menu. The following dialog box appears.



Type the file name here, then select <OK>.

OR

Select the drive/directory.

Then select the file from the File List box, and select <OK>.

(Use the TAB key to move from area to area.)

After opening a Card Library file, the PC card records in the file are displayed in the Current File list box on the left side of the main menu. You can then select the record you want to edit by clicking on the item, or by using the ↓ key to highlight it. Once the record is selected, you can edit/view it.

Opening a Second Card Library File

You can also open a second Card Library file in order to copy records from it into the Card Library file displayed in the Current File list box. The second Card Library is opened in read-only mode; records can be copied, but they cannot be modified. To open a second Card Library file, select Open Card Library from the File pull down menu. The Open File dialog box appears, enabling you to select the file you want to open. The records in the second Card Library file are displayed in the Card Library List box on the right side of the main menu.

Viewing a Card Library File

Card Library files can also be opened in view-only mode. When you open a file in this mode, you cannot edit or copy the records in the file (all Edit commands on the Edit pull down menu are grayed out).

To open a file in view-only mode, select View from the File pull down menu. The Open File dialog box is displayed, enabling you to select the file whose records you want to view.

Creating New Card Library Records

To create new PC card configuration records in a Card Library file, open the Card Library file to which you want to add records, then select Create New Record from the Edit pull down menu. Type in the name for the record (usually the card name) in the Record ID field. Then refer to "Editing Card Recognition Parameters" and "Defining the Card Recognition Configuration" for information on how to configure the card for recognition by Card Services.

Viewing a Card Library Record

Card Library records can be opened in view-only mode. To open a record in view-only mode, highlight the record in the Current File list box, then select View Current Record. The Card Recognition screen appears (showing the Card Type and Recognition Algorithm). From the Card Recognition screen, you can click on <OK> to view the Configuration Algorithm setting on the Card Configuration screen. For more information on Card Type, Recognition Algorithms, and Configuration Algorithms, refer to "Editing Card Recognition Parameters" and "Defining the Card Recognition Configuration".

Deleting Records From a Card Library File

Records can be deleted from a Card Library file that is opened in read/write mode (but not in view-only mode). To delete a record from a Card Library file, open the file using the Open command on the File pull down menu. Then highlight the record you want to delete and select Delete From Library from the Edit pull down menu. A confirmation box is displayed. Select <Yes> to delete the file, or select <No> to cancel the delete operation.

Copying Records From a Card Library File

Records can be copied from a Card Library file that is opened in read-only mode into a Card Library file opened in read/write mode. To copy a Card Library file:

1. Open the destination Card Library file using the Open command on the File pull down menu. The records in this file are listed on the left side of the main screen.
2. Open the source Card Library file using the Open Card Library command on the File pull down menu. The records in this file are listed on the right side of the main screen.
3. In the Card Library list box, highlight the record that you want to copy. Then select Copy From Library from the Edit pull down menu. The name of the record should now appear in the Current File list box.

Saving a Card Library File

Once you have modified a Card Library file, you must save your changes. To save the file under its original name, select Save from the File pull down menu. To save the file as a new file, select Save As from the File pull down menu, then type the new file name in the File Name box that is displayed.

Closing a Card Library File

To close a Card Library file that was opened in view-only mode (that is, the Card Library file displayed on the right side of the main menu), select Close Card Library from the File pull down menu.

Exiting the Configuration Utility

To exit the Configuration Utility, select Exit from the File pull down menu. If the Current File has been modified, a dialog box is displayed that enables you to save your changes or discard them before exiting. Select <Yes> to save your changes, <No> to discard them, or <Cancel> to return to the main menu and continue your editing session.

Editing Card Recognition Parameters

To enable Card Services to recognize an inserted PC card, you must define the recognition parameters for the card. This involves:

- Specifying the Card Type
- Selecting the Recognition Algorithm
- Selecting the Configuration Algorithm

In addition, you may want to set specific configuration parameters for the card, or you may want to use custom configuration code or custom removal code to configure the card.

To define the Card Recognition parameters for a card, select the card record from the Current File list box, then select Edit Current Record from the Edit pull down menu. The Card Recognition screen is displayed.

```

      Card Recognition
      Record ID: D-Link Ethernet Lan Card
      Card Type
      ( ) Modem      (•) Ethernet      ( ) ATA Disk
      ( ) 3270 Emulator ( ) Token Ring  ( ) Other
      Card Type Name:
      Recognition Algorithm
      (•) CIS Manufacturer/Product Data
      ( ) CIS Literal Bytes
      ( ) Generic Recognition by Card Type
      ( ) Custom Code
      <Edit Recognition Detail>
      <View Recognition Detail>
      < OK > <Cancel> < Help >
    
```

Type the record name in the Record ID field. This is the text that appears in the Current File list box.

Select a card type by clicking on the appropriate type, or by highlighting it and pressing the space bar. If not listed, select "Other" and type the name in the Card Type Name field.

To edit/view the detail record for the selected Recognition Algorithm, select Edit (or View) Recognition Detail.

Select the Recognition Algorithm by clicking on the desired item, or by highlighting it and pressing the space bar.

When you are done defining the Card Recognition information, click on <OK>. The Card Configuration screen appears. Refer to "Defining the Card Recognition Configuration" to continue.

Recognition Algorithms

This section describes the available recognition algorithms.

CIS Manufacturer/Product Data

Looks up the Level 1 Version Tuple in the Card Information Structure (CIS) on the card. If this recognition algorithm is selected, the following screen appears when you select Edit/View Recognition Detail. It contains four fields. To match any or all of the strings in these fields with the CIS information on the card, a trailing star can be used as a wildcard character at the end of each string. If a string is omitted, it is not matched with the CIS when card recognition is attempted.

Level 1 Version Tuple Fields

Manufacturer: Intel

Product: Ethernet Lan Card

Add'l Info 1: iNC300

Add'l Info 2:

< OK > <Cancel> < Help >

CIS Literal Bytes

Enables you to load a file that specifies a starting offset, length, and number of bytes to match. Select this option for defective CISs (that is, when you can't find the Level 1 Version Tuple). *This function is not currently supported.*

Generic Recognition by Card Type

Recognizes modem and ATA card types generically. When you select this option, no additional card recognition information needs to be defined (that is, no Recognition Detail screen is available). This option can only be used for modems and ATA cards.

Custom Code

Use this option to select a customized code file that you have written to support a particular card type. The specified file must be in a certain format, as defined in the Library Specification in Appendix A.

Custom Recognition Code File

*.bin

C:\CARDSOFT

File List:

Drives / Dirs:

..

[-A-]

[-B-]

[-C-]

[-D-]

[-E-]

[-F-]

< OK >

<Cancel>

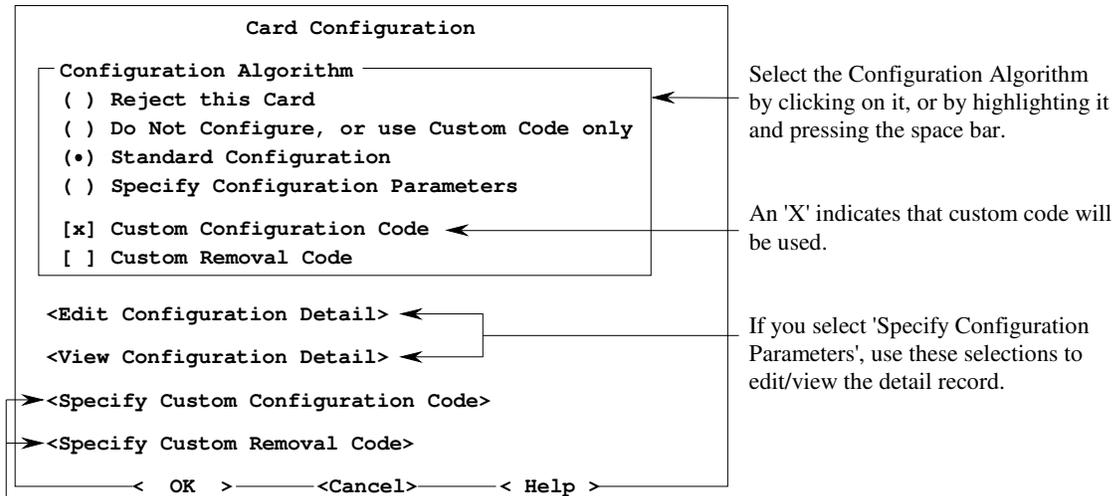
Type in a search string

OR

Select the drive/directory, then select the file name in the File List box and click on <OK>.

Defining the Card Recognition Configuration

When you select <OK> from the Card Recognition screen, the Card Configuration screen appears. This screen is where you select the Configuration Algorithm and specify whether or not you want to use Custom Configuration code or Custom Removal code.



If you selected 'Custom Configuration Code' or 'Custom Removal Code', use these selections to specify the code file you want to use.

Configuration Algorithms

Reject This Card

When the card is inserted into a slot, it will always be rejected, and an audible beep will sound.

Do Not Configure, or Use Custom Code only

When the card is inserted into a slot, it will not be configured unless 'Custom Configuration Code' is selected. If 'Custom Configuration Code' is selected, click on <Specify Custom Configuration Code> to select the custom code file you want to load. The file you specify must contain entries for the configuration parameters you want to load for the card. The format of the custom code file is defined in the Library Specification in Appendix A.

Standard Configuration

The card will use the standard configuration for card recognition.

Specify Configuration Parameters

This selection enables you to define customized configuration parameters for card recognition, including default and override CONFIG tuples, as well as default, required, and override CFTABLE_ENTRY tuples. For more information on this selection, refer to 'Defining Specific Card Recognition Parameters'.

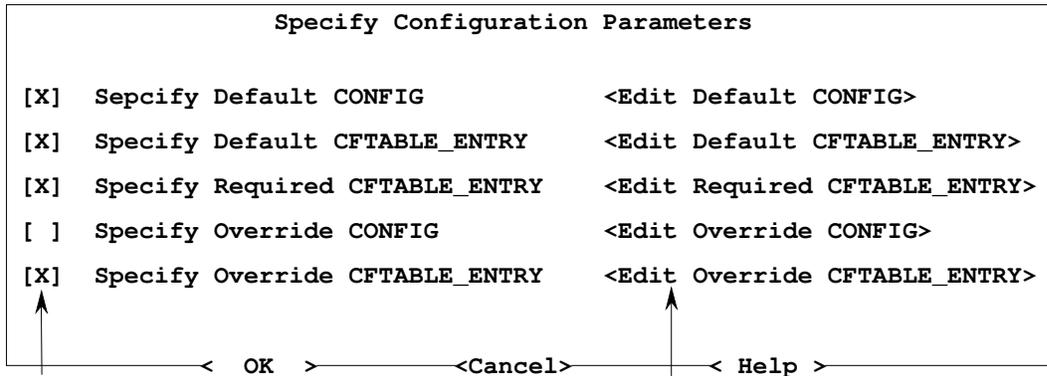
Using Custom Code

If you want to use custom configuration code to define or override one or more of the settings for the selected configuration algorithm, click between the brackets to the left of Custom Configuration Code to select it (it is selected when an 'X' is displayed between the brackets). Then click on <Specify Custom Configuration Code> to select the code file that contains the configuration information you want to use (a dialog box is displayed, from which you can select the file). For example, if you selected the Standard Configuration Algorithm, but the card is configured with an incorrect or invalid base address for Memory Window 1, you would have to create a custom configuration code file that defines the correct Memory Window 1 base address. The custom configuration code file must follow the format described in the Library Specification in Appendix A.

If a special event must occur when the card is removed, click between the brackets to the left of Custom Removal Code to select it (it is selected when an 'X' is displayed between the brackets). For example, you may have to clear a BIOS data area that was set up for a modem card when it was inserted. Then click on <Specify Custom Removal Code> (a dialog box is displayed, from which you can select the file). The custom removal code file must follow the format described in the Library Specification in Appendix A.

Defining Specific Card Recognition Parameters

If you want to define customized configuration parameters for card recognition, select 'Specify Configuration Parameters' on the Card Recognition screen. Then select <Edit Configuration Detail> to display the Specify Configuration Parameters screen.



If you want an edited tuple record to be used, click on the appropriate item in this column (or highlight it and press the space bar) to mark it with an 'X'.

To edit a particular tuple, select the appropriate item from this column to display the Edit screen for that item. The edited tuple record does not take effect unless the item is marked with an 'X' in the left column.

The parameters that can be defined are:

- CONFIG tuples (Default and Override)
 - Configuration Base Address
 - Configuration Register Mask
- CFTABLE_ENTRY tuples (Default, Override, and Required)
 - Configuration Index
 - Power Requirements (Vcc and Vpp)
 - Address Range for I/O Windows 0 and 1
 - IRQ Definition
 - Starting Segment, Length, Card Offset, and Memory Window Type for Memory Windows 1 and 2
 - Audio and Power Down support

The **Default** settings are those settings that you want to use by default for the card (usually based on PCMCIA standards). The **Override** settings override the Default settings. For example, if a card normally uses a particular Configuration Base Address, you would specify this address on the Default CONFIG tuple screen. If, however, you want to override the default setting for Configuration Base Address, you would specify the new address to use on the Override CONFIG tuple screen. In other words, if no Override value is specified for a particular setting, the Default value is used; if an Override value is specified for a particular setting, the Override value is used.

The **Required** CFTABLE_ENTRY settings are used as a search string to search for matching tuples on the card. For example, if the card requires a Memory Window with a starting segment at C000:0h and a size of 16K, you would enter the appropriate values for this configuration in the Memory Window 0 section of the Required CFTABLE_ENTRY tuple screen. When the card is inserted into a slot, the software will search for a tuple record on the card that matches these required values, and will use the matching record to identify the card.

CONFIG Tuple

The following screen appears for the <Edit CONFIG> selections.

```

          xxxxxxxx CONFIG Tuple
    _____
    | Configuration Base |
    | ( ) Unspec         |
    | ( ) Specify       |
    | Configuration Base Address  00000000 |
    |_____
    | Presence Mask     |
    | (•) Unspec        |
    | ( ) Specify       |
    | Configuration Register Mask  0 |
    |_____
    < OK > <Cancel> < Help >
  
```

This screen is where you define the default (or override) CONFIG tuple for the card. If an item does not apply to the card you are configuring, select 'Unspec' by clicking on it (or by highlighting it and pressing the space bar). Otherwise, select 'Specify' and press the TAB key to move the cursor into the field. Then type in the appropriate value. When done, click on <OK>.

Configuration Base Address — This field is used to specify the base address of the Configuration Registers. Up to eight hex digits can be entered in this field. This information is used in conjunction with the information in the Configuration Register Mask field to determine which registers are present on the card, as well as their location in the card's register space. The Configuration Base Address can be from one to four bytes, with each two characters in the field representing one byte. The length of the field depends on the number of bits required to represent the address.

Configuration Register Mask (Presence Mask) — This field is used to specify the presence mask for the Configuration Registers. This mask is used to determine the presence or absence of the corresponding Configuration Register, with 1 indicating it is present and 0 indicating it is not present. This information is used in conjunction with the Configuration Base Address field to determine which registers are present on the card, as well as their location in the card's register space.

CFTABLE_ENTRY Tuple

The following screen appears for the CFTABLE_ENTRY selections. CFTABLE_ENTRY tuples enable you to specify all of the possible configurations for a card, and enables the system to distinguish among the possible configurations.

```

xxxxxxxxxxx CFTABLE_ENTRY tuple
Configuration Index: FF
-----
Power Requirements
Vcc: ( ) Unspec   Vpp: ( ) Unspec
      ( ) 3.5V     ( ) 3.5V
      ( ) 5V       ( ) 5V
                  ( ) 12V
-----
Miscellaneous
( ) Unspec
( ) Specify
[ ] Audio is supported
[ ] Power Down is supported
-----
I/O Windows
0: ( ) Unspec      1: ( ) Unspec
   ( ) 8-bit        ( ) 8-bit
   ( ) 16-bit       ( ) 16-bit
Range: 0█ - 0█    Range: 0█ - 0█
-----
IRQ Definition
( ) Unspec
( ) Specific IRQ
( ) IRQ Mask
IRQ Level/Mask: 0█
-----
Memory Windows
0: ( ) Unspec      1: ( ) Unspec
   ( ) Common Memory   ( ) Common Memory
   ( ) Attribute Memory ( ) Attribute Memory
Start Seg: 0█00 Length (K): 0█ Start Seg: 0█00 Length (K): 0█
Card Offset (4K): 0█    Card Offset (4K): 0█
-----
< OK > <Cancel> < Help >

```

This screen is where you define the

- Configuration Index
- Power Requirements (Vcc and Vpp)
- Address Range for I/O Windows 0 and 1
- IRQ Definition
- Starting Segment, Length, Card Offset, and Memory Window Type for Memory Windows 1 and 2
- Audio and Power Down support

If an item does not apply to the card you are configuring, click on 'Unspec' (or highlight 'Unspec' and press the space bar). Otherwise, select the appropriate setting for the card. If a field is provided for entering a value, use the TAB key to position the cursor in the field, then type the value. When done, select <OK> to return to the Specify Configuration Parameters screen.

Configuration Index — This byte contains the value that must be written to the Card Configuration Register to enable the configuration described in the tuple record. It also contains a bit that indicates that the configuration described in the tuple record should be used as the default for the CFTABLE_ENTRY tuples that follow.

Power Requirements — These fields enable you to select the Vcc and Vpp power requirements for the PC card. The selected value is indicated by a bullet (•). To select a value, click on the appropriate item, or highlight it and press the space bar, to toggle the bullet on.

I/O Windows — These settings enable you to specify whether or not I/O windows are used by the card. To enable an I/O window, select the bus width for the window (8-bit or 16-bit) by clicking on the appropriate value (or highlighting it and pressing the space bar). Then use the TAB key to move into the "Range:" field, and type in the address range to be reserved for the window. The values must be in the range of 0h-3FFh.

IRQ Definition — These settings enable you to specify whether or not the card is using an IRQ. Also, you can select either a specific IRQ or an IRQ mask, and specify the IRQ or IRQ mask in the IRQ Level/Mask field. To select an IRQ, click on Specific IRQ or IRQ Mask (or highlight it and press the space bar). Then use the TAB key to move into the IRQ/Level Mask field, and type in the IRQ or IRQ Mask for the card.

Memory Windows — These two sets of fields enable you to specify the Memory Window settings for the card being defined. If the card uses memory windows, you can specify whether they are Common or Attribute memory windows. Also, you can specify the starting segment (ranging from C000-EF00, normally in the D000-DFFF range) for each window in the Starting Seg field, the window size (4, 8, ..., 64, in KB) in the Length field, and the offset (normally 0) in the Card Offset field.

Notes *If Start Seg is set to 0 for a network card, the starting segment value is extracted from the CARDID.INI file.*

If Length is set to 0 on the Override CFTABLE_ENTRY Tuple screen, it indicates that there is no memory window for the card, even if that window is defined on the Default CFTABLE_ENTRY tuple screen.

Audio Support — If this field is enabled (marked with an 'X'), it indicates that the card allows the BVD2 signal to be used as the audio waveform for the system speaker. To enable audio support, click on 'Specify' (or highlight it and press the space bar) to toggle the bullet (•) on, then click on 'Audio is supported' (or highlight it and press the space bar) to toggle the 'X' on.

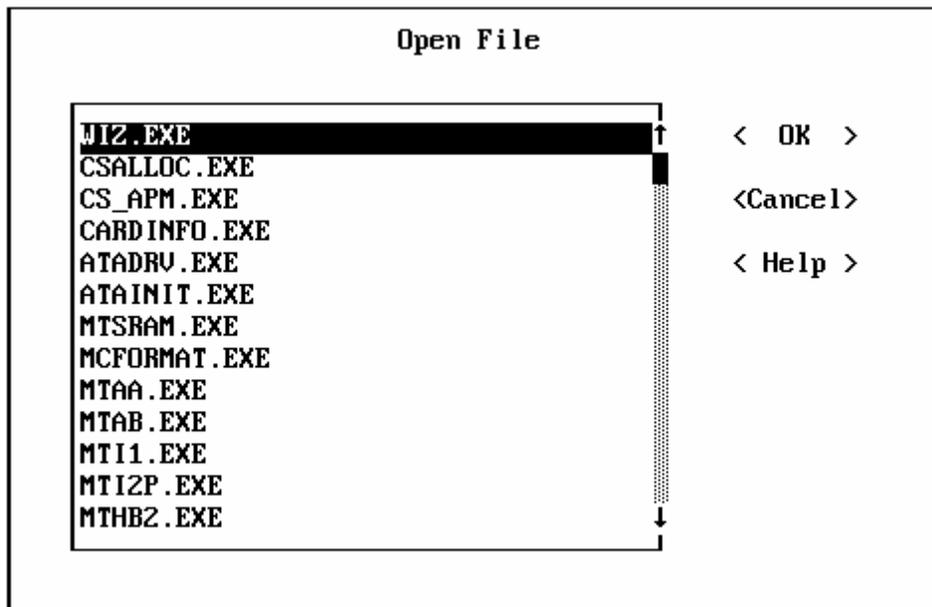
Power Down Support — If this field is enabled, it indicates that the card supports a power-down mode. To enable power-down support, click on 'Specify' (or highlight it and press the space bar) to toggle the bullet (•) on, then click on 'Power Down is supported' (or highlight it and press the space bar) to toggle the 'X' on.

Defining Parameters for Executable Files

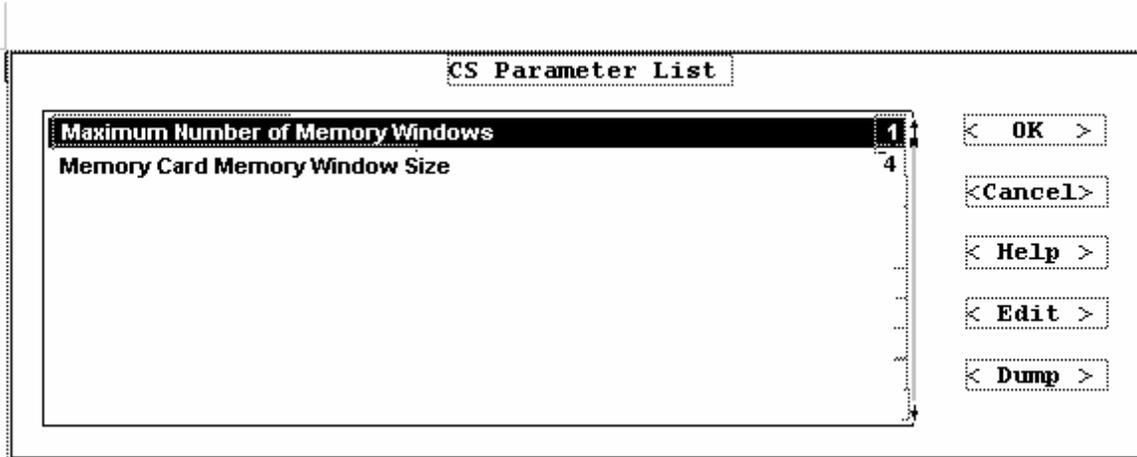
The Binary Adaptation function enables you to define the parameters for each of the executable components of the software. This section describes the process for defining the parameters. For information on each of these parameters, their default values, and the possible values that can be assigned to each, refer to Chapter 3.

To modify the parameters for the PCMCIA software's executable components:

1. Select Binary Adaptation from the Utility menu, then select Open from the dialog box that is displayed. A list of all executable files in the CARDSOFT directory is displayed (including Flash File System and *CardView* files, if these PCMCIA components are installed).

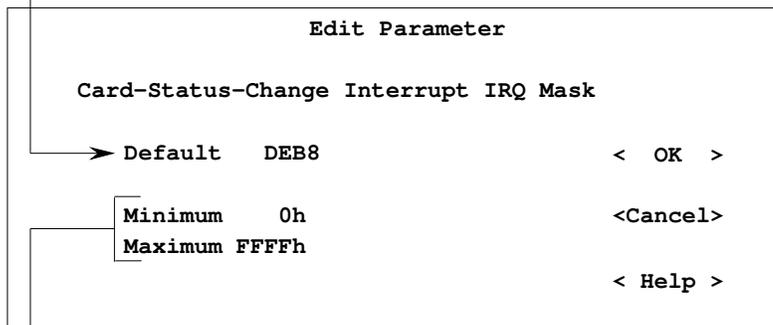


2. Select the file whose parameters you want to modify, then select <OK>. A Parameter List screen similar to the following appears. The list of parameters differs for each executable file. The sample screen shown below is the Parameter List for the Card Services executable file (MTAA.EXE).



3. To change a parameter, highlight the parameter and select <Edit>. If the parameter is a numeric value, an Edit Parameter screen similar to the following is displayed. If the parameter is a Yes/No value, the screen that is displayed enables you to select the Yes or No value.

Type the new value here.



Valid range of values is displayed here.

4. After editing the parameter, select <OK>. The new value should now be displayed on the Parameter List screen.
5. When you are done changing parameters, select <OK>. A dialog box is displayed. Select <Save> to save your changes (or select <Done>, then select <Yes>). If you want to cancel the changes you made, select <Done>, then select <No>.

Dumping the Parameter Values Into CONFIG.LST

The Binary Adaptation function allows you to save the executable file parameters into the file CONFIG.LST, which enables you to easily display or print out the list of configuration settings (using a TYPE CONFIG.LST or PRINT CONFIG.LST command). You can overwrite the existing CONFIG.LST file, or you can append the parameters to the end of the file.

To dump the parameters into CONFIG.LST, select <Dump> from the Parameter List screen. If CONFIG.LST already exists, a dialog box is displayed, giving you the option to Append to the end of the file (select <Append>) or Replace the existing file (select <Replace>).

Chapter 3

Configuring Driver Parameters

The following sections describe the parameters for each of the executable components of the PCMCIA software. The parameters are shown for both the Configuration Utility's Binary Adaptation Program (BAP) and the CONFIG.SYS DEVICEHIGH line. The Configuration Utility settings are listed on the left, while the DEVICEHIGH line settings are on the right. For information on how to configure parameters via the BAP, refer to "Defining Parameters for Executable Files" in Chapter 2.

Normally, you should not have to modify these parameters, since CSALLOC can detect the available resources and relay this information to the Card Services software. However, CSALLOC cannot detect system resources that may be required by system devices if those devices are not activated at the time CSALLOC is run (either from the command line or via the Resource Allocation menu pick in the Configuration Utility). This could result in a resource conflict when a device becomes active, since resources required for the device may already have been reserved for Card Services. If this situation occurs, you can use the Configuration Utility or DEVICEHIGH line parameters to ensure that the resources required by these devices are made unavailable to Card Services. Also, system hardware design and security features may make some available resources undetectable by CSALLOC. In this situation, you could edit CSALLOC.INI (or create a customized .INI file) to include special keywords to manually include resources that are not being detected. See Chapter 4 of this manual.

For example, if CSALLOC determines that IRQs 15, 12, 11, 10, 7, and 4 are available for use as the status-change interrupt, but you have an inactive device installed in the system that specifically needs IRQ 7 in order to operate properly, you could change the value for the Card-Status-Change Interrupt IRQ Mask parameter to exclude IRQ 7. (If however, the device is active at the time CSALLOC is run, CSALLOC would correctly determine that IRQ 7 is not available for Card Services, and would correctly set the Card-Status-Change Interrupt IRQ Mask value to exclude IRQ 7.) As another option, you could edit CSALLOC.INI, and add the line "IRQEXCLUDE=7". Both methods achieve the same result.

DEVICEHIGH Line Parameters

Modifications to the executable components can be done by way of adding or modifying DEVICEHIGH line parameters. Instead of modifying the executable file parameters via the Configuration Utility Binary Adaptation function, parameters are added to the appropriate device driver within CONFIG.SYS. The following is an *example* of how the *CardSoft* device lines might appear in CONFIG.SYS.

```
devicehigh=c:\cardwiz\ss365sl.exe /SIRM=FFFF /SKT=3
devicehigh=c:\Cardwiz\cs.sys /BEEPTYPES=CF
device=c:\Cardwiz\csalloc.exe
devicehigh=c:\Cardwiz\atadv.exe
devicehigh=c:\Cardwiz\mtaa.exe
devicehigh=c:\Cardwiz\mtab.exe
devicehigh=c:\Cardwiz\mtil.exe
devicehigh=c:\Cardwiz\mti2p.exe
devicehigh=c:\Cardwiz\mtsram.exe
devicehigh=c:\Cardwiz\mtddrv.exe /PART=2
devicehigh=c:\Cardwiz\ssmsflsh.sys
devicehigh=c:\Cardwiz\cardid.exe
install=c:\Cardwiz\cs_apm.exe
```

Note Parameters consist of an "idstring" and a "value". The "idstring" must always match exactly one of the available strings. Some "values" are entered as decimal and some are entered as hexadecimal. For example, in the example above the /SIRM (the top entry) is the required idstring and FFFF is the value.

Socket Services

Socket Services provides the standard software interface to the hardware that controls PCMCIA-compatible slots for PC cards. SystemSoft Socket Services supports the major host adapters that control the slots. The following section describes all configuration settings available for Socket Services, both from the Configuration Utility and the DEVICEHIGH line.

AMD SC300 (sselan.exe)

Parameter

Number Of Supported Sockets

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Command Line Entry

/SKT

Range: 1-4

Default: 4

/IRM

Range: 1h-FFFFh

Default: DEB8h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

Parameter**/WINBASE**

Range: C0h-D4h

Default: C8h

Undocumented entry.

/MEMWIN

Range: 2h-6h

Default: 6h

Undocumented entry.

Command Line Entry**/WINBASE**

Range: C0h-D4h

Default: C8h

/MEMWIN

Range: 2h-6h

Default: 6h

Cirrus CL-PD6710/20/22/22ZV/29/29ZV/30/30ZV (sscirrus.exe)**Parameter****/NUMADA**

Range: 1-2h

Default: 1

Undocumented entry.

Number Of Supported Slots

Range: 1-4

Default: 4

Specifies the number of slots that driver supports.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Command Line Entry**/NUMADA**

Range: 1-2h

Default: 1

/SKT

Range: 1-4

Default: 4

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

Specifies the base port address of the PCMCIA adapter(s). This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

/PORT

Range: 0h-3FFh

Default: 3E0h

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

/IRM

Range: 1h-FFFFh

Default: DEB8h

Parameter**Voltage Stabilization Delay**

Range: 0 through 65535
 Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased.

Zero Voltage Stabilization Delay

Range: 0 through 65535
 Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Low Power Dynamic Mode

Range: 0-1h
 Default: 0h

Low Power Dynamic Mode Switch. Use to set for low power operation.

Settings:

- 0 - The clock runs always (low power not set).
- 1 - Normal (low power) operation. Clock stopped when possible.

Install As Adapter Zero

Range: Yes or No
 Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Pulse Mode Interrupts

Range: 0h-1h
 Default: 0h

Allows interrupts to be generated in pulse mode, rather than level mode.

Settings:

- 0 - Interrupts are passed to IRQ pin as level-sensitive.
- 1 - When interrupt occurs, IRQ pin is driven with the pulse train and allows interrupt sharing.

5V Core Use

Range: 0h-1h
 Default: 0h

This parameter must be True (1) when the Vdd pin is connected to 5 volts (TTL compatible). This parameter must be False (0) when the Vdd pin is connected to 3.3 volts.

Card Status Change Interrupt in SMI Mode

Range: 0h-1h
 Default: 0h

When True (1), the PCMCIA adapter is configured to generate an SMI interrupt instead of an IRQ for card status change interrupts.

Bypass Frequency Synthesizer

Range: 0h-1h
 Default: 1h

When True (1), the PCMCIA adapter is configured to bypass the frequency synthesizer (internal clock = CLK_IN). When False (0), the PCMCIA adapter is configured for normal operation (internal clock = 7/4 * CLK_IN).

Command Line Entry**/PWRON**

Range: 01h-FFFFh
 Default: 1A0B

/PWROFF

Range: 01h-FFFFh
 Default: 0

/LPDM

Range 0-1h
 Default: 0h

/ADA

Range: 0 or 1
 Default: 1

/PULSE

Range: 0h-1h
 Default: 0h

/5VCORE

Range: 0h-1h
 Default: 0h

/SMI

Range: 0h-1h
 Default: 0h

/SYNTHESIZER

Range: 0h-1h
 Default: 1h

Parameter**Setup Timing Register**

Range: 0h-1h

Default: 1h

This parameter specifies the value for setup timing register.

Command Timing Register

Range: 0h-FFh

Default: 6h

This parameter specifies the value for the command timing register.

Recovery Timing Register

Range: 0h-FFh

Default: 3h

This parameter specifies the value for the recovery timing register.

DMA Mode

Range: 0-1

Default: 0

A setting of 1 permits possible use of DMA (VG-469 only). The default is 0 for Plug and Play mode. When Plug and Play is enabled, the PC Card cannot be a DMA device. When Plug and Play mode is used (VG-469 only), this parameter should be 0. The Plug and Play implementation lets the software relocate the VG-469 registers from the default at 3E0h/3E1h to 3E2h/3E3h. Plug and Play is only available in extended mode.

Clock Speed Greater than 25MHZ

Range: 0h-1h

Default: 1h

This parameter should be set to True (1) when the PCI clock speed or the external clock speed is set higher than 25MHZ.

PCI Card Status Charge Interrupt Mode

Range: 0h-1h

Default: 0h

When True (1), the PCMCIA adapter is configured to generate the card status change interrupt for the PCI bus (active low). When False (0), the PCMCIA adapter is configured to generated the card status change interrupt for the ISA bus (active high).

/DMAMASK

Range: 0h-FFh

Default: 6h

Undocumented entry.

/ZOOM_V

Range: 0h-FFh

Default: 0h

The default for Zoomed Video support will probably be 0 (which means that Zoomed Video is not enabled).

If you want to enable zoomed video, enter the number of the socket for which you want zoomed video enabled. For example, enter:

- 1 for 1st socket
- 2 for 2nd socket
- 3 for 3rd socket etc.

Command Line Entry**/SETUP_TIME**

Range: 0h-1h

Default: 1h

/COMMAND_TIME

Range: 0h-FFh

Default: 6h

/RECOVERY

Range: 0h-FFh

Default: 3h

/DMAMODE

Range: 0-1

Default: 0

/_25mhz

Range: 0h-1h

Default: 1h

/PCIIRQ

Range: 0h-1h

Default: 0h

/DMAMASK

Range: 0h-FFh

Default: 6h

/VOOM_v

Range: 0h-FFh

Default: 0h

in the default field.

/IGNORE_SKT

Range: 0h-5h

Default: 5h

Undocumented entry.

/IGNORE_SKT

Range: 0h-5h

Default: 5h

Cirrus (sscir12v.exe)**Parameter****Number Of Supported Slots**

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

`/SKT=n1,n2,n3,n4`

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Command Line Entry**/SKT**

Range: 1-4

Default: 4

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

Specifies the base port address of the PCMCIA adapter(s). This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

/PORT

Range: 0h-3FFh

Default: 3E0h

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

/IRM

Range: 1h-FFFFh

Default: DEB8h

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

/PWRON

Range: 01h-FFFFh

Default: 1A0B

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

/PWROFF

Range: 01h-FFFFh

Default: 0

Parameter**Low Power Dynamic Mode**

Range: 0-1h

Default: 0h

Low Power Dynamic Mode Switch. Use to set for low power operation.

Settings:

0 - The clock runs always (low power not set).

1 - Normal (low power) operation. Clock stopped when possible.

Command Line Entry**/LPDM**

Range 0-1h

Default: 0h

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

/ADA

Range: 0 or 1

Default: 1

Pulse Mode Interrupts

Range: 0h-1h

Default: 0h

Allows interrupts to be generated in pulse mode, rather than level mode.

Settings:

0 - Interrupts are passed to IRQ pin as level-sensitive.

1 - When interrupt occurs, IRQ pin is driven with the pulse train and allows interrupt sharing.

/PULSE

Range: 0h-1h

Default: 0h

5V Core Use

Range: 0h-1h

Default: 0h

This parameter must be True (1) when the Vdd pin is connected to 5 volts (TTL compatible). This parameter must be False (0) when the Vdd pin is connected to 3.3 volts.

/5VCORE

Range: 0h-1h

Default: 0h

Card Status Change Interrupt in SMI Mode

Range: 0h-1h

Default: 0h

When True (1), the PCMCIA adapter is configured to generate an SMI interrupt instead of an IRQ for card status change interrupts.

/SMI

Range: 0h-1h

Default: 0h

DataBook DB86072/82/82A/82B (ssdbook.exe)**Parameter****Card-Status-Change Interrupt IRQ Mask**

Range: 1h-FFFFh

Default: 4CF8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as status-change interrupt (event interrupt) IRQs. This interrupt is generated when the adapter status changes (i.e., a card is inserted or removed). Use this parameter to inform Socket Services of the IRQs that are connected to the DataBook chip for use as the status change interrupt signals. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of 4CF8h (0100-1100-1111-1000 binary) indicates that IRQs 14, 11, 10, 7, 6, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PC Card IREQ Mask

Range: 1h-FFFFh

Default: 4CF8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IREQs. PC card IREQs are those interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of 4CF8h (0100-1100-1111-1000 binary) indicates that IRQs 14, 11, 10, 7, 6, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

/IO_PORT1

Range 0h-3ffh

Default: 240

Undocumented Entry.

/IO_PORT2

Range 0h-3ffh

Default: 0

Undocumented Entry.

/IO_PORT3

Range 0h-3ffh

Default: 0

Undocumented Entry.

/IO_PORT4

Range 0h-3ffh

Default: 0

Undocumented Entry.

Voltage Stabilization Delay

Range: 1 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Command Line Entry**/SIRM**

Range: 1h-FFFFh

Default: 4CF8h

/IIRM

Range: 1h-FFFFh

Default: 4CF8h

/IO_PORT1

Range 0h-1h

Default: 240

/IO_PORT2

Range 0h-1h

Default: 0

/IO_PORT3

Range 0h-1h

Default: 0

/IO_PORT4

Range 0h-1h

Default: 0

/PWRON

Range: 01h-FFFFh

Default: 1A0B

Parameter**Install As Adapter Zero**

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. SSDBOOK installs as the next available Adapter when this parameter is not specified.

Command Line Entry**/ADA**

Range: 0 or 1

Default: 1

Number Of Supported Sockets

Range: 1-2

Default: 2

/SKT

Range: 1-2

Default: 2

Specifies the number of PCMCIA slots that SSDBOOK supports. A value of 0 causes the number of slots to be determined dynamically. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

HIO16 and HMEM16 Control Type

Range: 0 or 1

Default: 0

Specifies type of control for HIO16 and HMEM16 ISA bus signals. 0 enables open drain control. 1 enables totem-pole control.

/TOTEM

Range: 0 or 1

Default: 0

Adapter Identification Mode

Range 0h-2h

Default: 0h

Setting:

0 - auto detect adapter ID

1 - force TMI-140 type

2 - force TMB-240/250 type

IDTYPE

Range 0h-2h

Default: 0h

/Quiet

Range 0h-1h

Default: 0h

Undocumented Entry.

/Quiet

Range 0h-1h

Default: 0h

/SIZE_1K

Range 0h-1h

Default: 0h

Undocumented Entry.

/SIZE_1K

Range 0h-1h

Default: 0h

/COR_VPP

Range 0h-1h

Default: 0h

Undocumented Entry.

/COR_VPP

Range 0h-1h

Default: 0h

Hyundai HYN86C722 (sshundai.exe)

Parameter

/NUMADA

Range: 1-2h

Default: 1

Undocumented entry.

Number Of Supported Slots

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

Specifies the base port address of the PCMCIA adapter(s). This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Command Line Entry

/NUMADA

Range: 1-2h

Default: 1

/SKT

Range: 1-4

Default: 4

/PORT

Range: 0h-3FFh

Default: 3E0h

/IRM

Range: 1h-FFFFh

Default: DEB8h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

Parameter**Low Power Dynamic Mode**

Range: 0-1h

Default: 0h

Low Power Dynamic Mode Switch. Use to set for low power operation.

Settings:

0 - The clock runs always (low power not set).

1 - Normal (low power) operation. Clock stopped when possible.

Command Line Entry**/LPDM**

Range 0-1h

Default: 0h

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

/ADA

Range: 0 or 1

Default: 1

Pulse Mode Interrupts

Range: 0h-1h

Default: 0h

Allows interrupts to be generated in pulse mode, rather than level mode.

Settings:

0 - Interrupts are passed to IRQ pin as level-sensitive.

1 - When interrupt occurs, IRQ pin is driven with the pulse train and allows interrupt sharing.

/PULSE

Range: 0h-1h

Default: 0h

5V Core Use

Range: 0h-1h

Default: 0h

This parameter must be True (1) when the Vdd pin is connected to 5 volts (TTL compatible). This parameter must be False (0) when the Vdd pin is connected to 3.3 volts.

/5VCORE

Range: 0h-1h

Default: 0h

Card-Status-Change Interrupt in SMI Mode

Range: 0h-1h

Default: 0h

When True (1), the PCMCIA adapter is configured to generate an SMI interrupt instead of an IRQ for card status change interrupts.

/SMI

Range: 0h-1h

Default: 0h

Bypass Frequency Synthesizer

Range: 0h-1h

Default: 1h

When True (1), the PCMCIA adapter is configured to bypass the frequency synthesizer (internal clock = CLK_IN). When False (0), the PCMCIA adapter is configured for normal operation (internal clock = 7/4 * CLK_IN).

/SYNTHESIZER

Range: 0h-1h

Default: 1h

Setup Timing Register

Range: 0h-1h

Default: 1h

This parameter specifies the value for setup timing register.

/SETUP_TIME

Range: 0h-1h

Default: 1h

Command Timing Register

Range: 0h-FFh

Default: 6h

This parameter specifies the value for the command timing register.

/COMMAND_TIME

Range: 0h-FFh

Default: 6h

Recovery Timing Register

Range: 0h-FFh

Default: 3h

/RECOVERY

Range: 0h-FFh

Default: 3h

This parameter specifies the value for the recovery timing register.

Parameter

DMA Mode

Range: 0-1

Default: 0

Command Line Entry

/DMAMODE

Range: 0-1

Default: 0

A setting of 1 permits possible use of DMA (VG-469 only). The default is 0 for Plug and Play mode. When Plug and Play is enabled, the PC Card cannot be a DMA device. When Plug and Play mode is used (VG-469 only), this parameter should be 0. The Plug and Play implementation lets the software relocate the VG-469 registers from the default at 3E0h/3E1h to 3E2h/3E3h. Plug and Play is only available in extended mode.

Clock Speed Greater than 25MHZ

Range: 0h-1h

Default: 1h

/_25mhz

Range: 0h-1h

Default: 1h

This parameter should be set to True (1) when the PCI clock speed or the external clock speed is set higher than 25MHZ.

/DMAMASK

Range: 0h-FFh

Default: 6h

/DMAMASK

Range: 0h-FFh

Default: 6h

Undocumented entry.

/ZOOM_V

Range: 0h-FFh

Default: 0h

/VOOM_v

Range: 0h-FFh

Default: 0h

The default for Zoomed Video support will probably be 0 (which means that Zoomed Video is not enabled).

If you want to enable zoomed video, enter the number of the socket for which you want zoomed video enabled. For example, enter:

- 1 for 1st socket
- 2 for 2nd socket
- 3 for 3rd socket etc.

in the default field.

IBM ThinkPad 720 (ssibm720.exe)**Parameter****Card-Status-Change Interrupt IRQ Mask**

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as status-change interrupt (event interrupt) IRQs. The status-change interrupt is generated when the PCMCIA adapter status changes (that is, a card is removed or inserted). Use this parameter to inform Socket Services of the IRQs that are connected to the Intel PCIC or IBM Thinkpad chip for use as the status-change interrupt signals. This information should be provided by the manufacturer of the system.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PC Card IREQ Mask

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IREQs. PC card IREQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted into your system. Use this parameter to inform Socket Services which IRQs are connected to the Intel controller for use as card interrupts. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

This switch setting is dependent on the system hardware design. It specifies the base port address of the first of up to two PCMCIA adapters. This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The SS365SL installs as the next available Adapter when this parameter is not specified.

Intel 82365SL PCIC ID

Range: 0h-FFh

Default: 88h

Specifies an expected ID byte (Register 0) in addition to the standard values. The default is no additional IDs

Command Line Entry**/SIRM**

Range: 0h-FFFFh

Default: DEB8h

/IIRM

Range: 0h-FFFFh

Default: DEB8h

/PORT

Range: 0h-3FFh

Default: 3E0h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/ID

Range: 0h-FFh

Default: 88h

Parameter**Number Of Supported Sockets**

Range: 1-4

Default: 0

Specifies the number of slots that the controller supports (1-4). The default value (0) indicates that Socket Services dynamically determines the number of slots present. If a number other than zero is specified, it must be set to the exact number of slots present in the system.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Reset On CPU Resume

Range: Yes or No

Default: No

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

Command Line Entry**/RST**

Range: 0 or 1

Default: 1

Test for Warm Reboot

Range: Yes or No

Default: No

Specifies that Socket Services test for possible system resources depletion due to configured PCMCIA slots during a warm reboot. If enabled, and configured slots are discovered following a warm reboot, Socket Services clears all slot configurations and reboots the system.

Command Line Entry**/WRB**

Range: 0h or 1h

Default: 0

Language (alternate embedded Language)

Range: 0h-4h

Default: 0

Specifies the settings for alternate embedded language. The default setting is 0. This driver/utility may be configured to display messages in five different languages: 0=English (the default), 1=French, 2=German, 3=Italian, 4=Spanish.

Command Line Entry**/LANG**

Range: 0h or 4h

Default: 0

Intel 82365SL PCIC (ss365lp.exe)**Parameter****Card-Status-Change Interrupt IRQ Mask**

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as status-change interrupt (event interrupt) IRQs. The status-change interrupt is generated when the PCMCIA adapter status changes (that is, a card is removed or inserted). Use this parameter to inform Socket Services of the IRQs that are connected to the Intel PCIC or IBM Thinkpad chip for use as the status-change interrupt signals. This information should be provided by the manufacturer of the system.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PC Card IREQ Mask

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IREQs. PC card IREQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted into your system. Use this parameter to inform Socket Services which IRQs are connected to the Intel controller for use as card interrupts. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

This switch setting is dependent on the system hardware design. It specifies the base port address of the first of up to two PCMCIA adapters. This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The SS365SL installs as the next available Adapter when this parameter is not specified.

Intel 82365SL PCIC ID

Range: 0h-FFh

Default: 84h

Specifies an expected ID byte (Register 0) in addition to the standard values. The default is no additional IDs.

Command Line Entry**/SIRM**

Range: 0h-FFFFh

Default: DEB8h

/IIRM

Range: 0h-FFFFh

Default: DEB8h

/PORT

Range: 0h-3FFh

Default: 3E0h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/ID

Range: 0h-FFh

Default: 84h

Parameter**Command Line Entry**

Number Of Supported Sockets

Range: 1-4

Default: 0

/SKT

Range: 1-4

Default: 0

Specifies the number of slots that the controller supports (1-4). The default value (0) indicates that Socket Services dynamically determines the number of slots present. If a number other than zero is specified, it must be set to the exact number of slots present in the system.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Reset On CPU Resume

Range: Yes or No

Default: No

/RST

Range: 0 or 1

Default: 1

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

Test for Warm Reboot

Range: Yes or No

Default: No

/WRB

Range: 0h or 1h

Default: 0

Specifies that Socket Services test for possible system resources depletion due to configured PCMCIA slots during a warm reboot. If enabled, and configured slots are discovered following a warm reboot, Socket Services clears all slot configurations and reboots the system.

Intel 82365SL PCIC (ss365sl.exe)

Parameter

PCMCIA Adapter Base Port Address

Range: 0h-3FFh
Default: 3E0h

Specifies the base port address of the PCMCIA adapter(s). This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

PC Card IREQ Mask

Range: 0h-FFFFh
Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IREQs. PC card IREQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted into your system. Use this parameter to inform Socket Services which IRQs are connected to the Intel controller for use as card interrupts. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh
Default: 3E0h

This switch setting is dependent on the system hardware design. It specifies the base port address of the first of up to two PCMCIA adapters. This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

Voltage Stabilization Delay

Range: 0 through 65535
Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535
Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No
Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Intel 82365SL PCIC ID

Range: 0h-FFh
Default: 83h

Specifies an expected ID byte (Register 0) in addition to the standard values. The default is no additional IDs.

Command Line Entry

/PORT

Range: 0h-3FFh
Default: 3E0h

/IIRM

Range: 0h-FFFFh
Default: DEB8h

/PORT

Range: 0h-3FFh
Default: 3E0h

/PWRON

Range: 01h-FFFFh
Default: 1A0B

/PWROFF

Range: 01h-FFFFh
Default: 0

/ADA

Range: 0 or 1
Default: 1

/ID

Range: 0h-FFh
Default: 83h

Parameter**Number Of Supported Slots**

Range: 1-4

Default: 4

Command Line Entry**/SKT**

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Reset On CPU Resume

Range: Yes or No

Default: No

/RST

Range: 0 or 1

Default: 1

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

Test for Warm Reboot

Range: Yes or No

Default: No

/WRB

Range: 0h or 1h

Default: 0

Specifies that Socket Services test for possible system resources depletion due to configured PCMCIA slots during a warm reboot. If enabled, and configured slots are discovered following a warm reboot, Socket Services clears all slot configurations and reboots the system.

Intel AT2092AA (SSspec.exe)**Parameter****Number Of Supported Slots**

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

/IRQMODE

Range: 0-1h

Default: 0

Undocumented entry.

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Command Line Entry**/SKT**

Range: 1-4

Default: 4

/IRQMODE

Range: 0-1h

Default: 0

/IRM

Range: 1h-FFFFh

Default: DEB8h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

Parameter**Install As Adapter Zero**

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Command Line Entry**/ADA**

Range: 0 or 1

Default: 1

Language (alternate embedded Language)

Range: 0h-4h

Default: 0

Specifies the settings for alternate embedded language. The default setting is 0. This driver/utility may be configured to display messages in five different languages: 0=English (the default), 1=French, 2=German, 3=Italian, 4=Spanish.

/LANG

Range: 0h or 4h

Default: 0

Ricoh RF5C266/RF5C366 (ssricoh.exe)**Parameter****Number Of Supported Slots**

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Command Line Entry**/SKT**

Range: 1-4

Default: 4

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

Specifies the base port address of the PCMCIA adapter(s). This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

/PORT

Range: 0h-3FFh

Default: 3E0h

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

/IRM

Range: 1h-FFFFh

Default: DEB8h

Parameter**Voltage Stabilization Delay**

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Language (alternate embedded Language)

Range: 0h-4h

Default: 0

Specifies the settings for alternate embedded language. The default setting is 0. This driver/utility may be configured to display messages in five different languages: 0=English (the default), 1=French, 2=German, 3=Italian, 4=Spanish.

/TC_IRQ

Range: 11h-15h

Default: 11h

Undocumented entry.

Command Line Entry**/PWRON**

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/LANG

Range: 0h or 4h

Default: 0

/TC_IRQ

Range: 11h-15h

Default: 11h

SiS 5131(sssis.exe)**Parameter****Card-Status-Change Interrupt IRQ Mask**

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as status-change interrupt (event interrupt) IRQs. The status-change interrupt is generated when the PCMCIA adapter status changes (that is, a card is removed or inserted). Use this parameter to inform Socket Services of the IRQs that are connected to the Intel PCIC or IBM Thinkpad chip for use as the status-change interrupt signals. This information should be provided by the manufacturer of the system.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PC Card IREQ Mask

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IREQs. PC card IREQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted into your system. Use this parameter to inform Socket Services which IRQs are connected to the Intel controller for use as card interrupts. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

This switch setting is dependent on the system hardware design. It specifies the base port address of the first of up to two PCMCIA adapters. This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Intel 82365SL PCIC ID

Range: 0h-FFh

Default: 83h

Specifies an expected Intel 82365SL ID byte (Register 0) in addition to the standard values of 82h and 83h.

Parameter**NUMADA****Command Line Entry****/SIRM**

Range: 0h-FFFFh

Default: DEB8h

/IIRM

Range: 0h-FFFFh

Default: DEB8h

/PORT

Range: 0h-3FFh

Default: 3E0h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/ID

Range: 0h-FFh

Default: 83h

Command Line Entry**/NUMADA**

Range: 1-2h
Default: 1

Range: 1-2h
Default: 1

Undocumented entry.

Number Of Supported Slots

Range: 1-4
Default: 4

/SKT

Range: 1-4
Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

/SKT=n1,n2,n3,n4

where -

- n1 represents the number of slots in the first adapter.
- n2 represents the number of slots in the second adapter (if present).
- n3 represents the number of slots in the third adapter (if present).
- n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Reset On CPU Resume

Range: Yes or No
Default: No

/RST

Range: 0 or 1
Default: 1

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

Test for Warm Reboot

Range: Yes or No
Default: No

/WRB

Range: 0h or 1h
Default: 0

Specifies that Socket Services test for possible system resources depletion due to configured PCMCIA slots during a warm reboot. If enabled, and configured slots are discovered following a warm reboot, Socket Services clears all slot configurations and reboots the system.

/ZERO_MEM_WAIT_STATE

Range: 0-1
Default: 0

/ZERO_MEM_WAIT_STATE

Range: 0-1
Default: 0

Specifies the number of wait state cycles of 16-bit memory access to the Memory Window.

A setting of 0 indicates that System Memory access will complete in 3 CLKs (16 bit) or 6 CLKs (8 bit) with IOCHRDY high.

A setting of 1 indicates that system memory access will complete in 2 CLKs (16 bit) or 3 CLKs (8 bit) with *ZWS signal driven low.

/16BIT_MEM_WAIT_STATE

Range: 0-3
Default: 0

/16BIT_MEM_WAIT_STATE

Range: 0-3
Default: 0

These values determine the number of additional wait states for a 16-bit access to the system memory window.

Parameter**PC Card IRQ Mask**

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Language (alternate embedded Language)

Range: 0h-4h

Default: 0

Specifies the settings for alternate embedded language. The default setting is 0. This driver/utility may be configured to display messages in five different languages: 0=English (the default), 1=French, 2=German, 3=Italian, 4=Spanish.

Command Line Entry**/IRM**

Range: 1h-FFFFh

Default: DEB8h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/LANG

Range: 0h or 4h

Default: 0

Zipang (sszipang.exe)

Parameter

Card Status Change Interrupt IRQ Mask

Range: 0h-1h
Default: 0h

Specifies the 16-bit mask of available IRQ's for assignment as card Status Change Interrupt. The least significant bit represents IRQ 0 and the most significant bit represents IRQ 15 (i.e., a value of 53B8 indicates that IRQ 14, 22, 11, 10, 9, 7, 5, 4, and 3 are available for Card Status Change Interrupt).

PC Card IREQ Mask

Range: 0h-FFFFh
Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IREQs. PC card IREQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted into your system. Use this parameter to inform Socket Services which IRQs are connected to the Intel controller for use as card interrupts. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh
Default: 3E0h

Specifies the base port address of the PCMCIA adapter(s). This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

Voltage Stabilization Delay

Range: 0 through 65535
Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Install As Adapter Zero

Range: Yes or No
Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Intel 82365SL PCIC ID

Range: 0h-FFh
Default: 83h

Specifies an expected Intel 82365SL ID byte (Register 0) in addition to the standard values of 82h and 83h.

Command Line Entry

/xxxxx

Range: 0h-1h
Default: 0h

/IIRM

Range: 0h-FFFFh
Default: DEB8h

/PORT

Range: 0h-3FFh
Default: 3E0h

/PWRON

Range: 01h-FFFFh
Default: 1A0B

/ADA

Range: 0 or 1
Default: 1

/ID

Range: 0h-FFh
Default: 83h

Parameter**Number Of Supported Slots**

Range: 1-4

Default: 4

Command Line Entry**/SKT**

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Reset On CPU Resume

Range: Yes or No

Default: No

/RST

Range: 0 or 1

Default: 1

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

Test for Warm Reboot

Range: Yes or No

Default: No

/WRB

Range: 0h or 1h

Default: 0

Specifies that Socket Services test for possible system resources depletion due to configured PCMCIA slots during a warm reboot. If enabled, and configured slots are discovered following a warm reboot, Socket Services clears all slot configurations and reboots the system.

/ZERO_MEM_WAIT_STATE

Range: 0-1

Default: 0

/ZERO_MEM_WAIT_STATE

Range: 0-1

Default: 0

Specifies the number of wait state cycles of 16-bit memory access to the Memory Window.

A setting of 0 indicates that System Memory access will complete in 3 CLKs (16 bit) or 6 CLKs (8 bit) with IOCHRDY high.

A setting of 1 indicates that system memory access will complete in 2 CLKs (16 bit) or 3 CLKs (8 bit) with *ZWS signal driven low.

/16BIT_MEM_WAIT_STATE

Range: 0-3

Default: 0

/16BIT_MEM_WAIT_STATE

Range: 0-3

Default: 0

These values determine the number of additional wait states for a 16-bit access to the system memory window.

Texas Instruments (ssti.exe)**Parameter****Number Of Supported Slots**

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

/IRQMODE

Range: 0-1h

Default: 0

Undocumented entry.

/ISA IRQ

Range: 0-1h

Default: 1

Undocumented entry.

Command Line Entry**/SKT**

Range: 1-4

Default: 4

/IRQMODE

Range: 0-1h

Default: 0

/ISA IRQ

Range: 0-1h

Default: 1

Texas Instruments TI1130 (sscbti.exe)

Parameter

SystemSoft Reserved Parameter

Range: 0h -ffffh

Default: 2

This parameter is for SystemSoft diagnostic and special function purposes. Do not alter this value unless specifically directed to do so by SystemSoft Corporation.

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Reset On CPU Resume

Range: Yes or No

Default: No

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

/DEVICEID[0]

Range: 0h-ffffh

Default: AC12h

Undocumented entry.

/VENDORID

Range: 0h-ffffh

Default: 104ch

Undocumented entry.

Command Line Entry

/SYSTEMSOFT1

Range: 0h-ffffh

Default: 2

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/RST

Range: 0 or 1

Default: 1

/DEVICEID[0]

Range: 0h-ffffh

Default: AC12h

/VENDORID

Range: 0h-ffffh

Default: 104ch

Toshiba ToPIC95M (sscbto95.exe)

Parameter**SystemSoft Reserved Parameter**

Range: 0h -ffffh

Default: 2

This parameter is for SystemSoft diagnostic and special function purposes. Do not alter this value unless specifically directed to do so by SystemSoft Corporation.

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Reset On CPU Resume

Range: Yes or No

Default: No

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

/DEVICEID[0]

Range: 0h-ffffh

Default: AC12h

Undocumented entry.

/VENDORID

Range: 0h-ffffh

Default: 104ch

Undocumented entry.

Command Line Entry**/SYSTEMSOFT1**

Range: 0h-ffffh

Default: 2

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/RST

Range: 0 or 1

Default: 1

/DEVICEID[0]

Range: 0h-ffffh

Default: AC12h

/VENDORID

Range: 0h-ffffh

Default: 104ch

Trident Omega 82C722G (SSomega7.exe)

Parameter

/NUMADA

Range: 1-2h

Default: 1

Undocumented entry.

Number Of Supported Slots

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

Specifies the base port address of the PCMCIA adapter(s). This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Command Line Entry

/NUMADA

Range: 1-2h

Default: 1

/SKT

Range: 1-4

Default: 4

/PORT

Range: 0h-3FFh

Default: 3E0h

/IRM

Range: 1h-FFFFh

Default: DEB8h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Parameter**Low Power Dynamic Mode**

Range: 0-1h

Default: 0h

Low Power Dynamic Mode Switch. Use to set for low power operation.

Settings:

0 - The clock runs always (low power not set).

1 - Normal (low power) operation. Clock stopped when possible.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Pulse Mode Interrupts

Range: 0h-1h

Default: 0h

Allows interrupts to be generated in pulse mode, rather than level mode.

Settings:

0 - Interrupts are passed to IRQ pin as level-sensitive.

1 - When interrupt occurs, IRQ pin is driven with the pulse train and allows interrupt sharing.

5V Core Use

Range: 0h-1h

Default: 0h

This parameter must be True (1) when the Vdd pin is connected to 5 volts (TTL compatible). This parameter must be False (0) when the Vdd pin is connected to 3.3 volts.

Card-Status-Change Interrupt in SMI Mode

Range: 0h-1h

Default: 0h

When True (1), the PCMCIA adapter is configured to generate an SMI interrupt instead of an IRQ for card status change interrupts.

Bypass Frequency Synthesizer

Range: 0h-1h

Default: 1h

When True (1), the PCMCIA adapter is configured to bypass the frequency synthesizer (internal clock = CLK_IN). When False (0), the PCMCIA adapter is configured for normal operation (internal clock = 7/4 * CLK_IN).

Setup Timing Register

Range: 0h-1h

Default: 1h

This parameter specifies the value for setup timing register.

Command Timing Register

Range: 0h-FFh

Default: 6h

This parameter specifies the value for the command timing register.

Recovery Timing Register

Range: 0h-FFh

Default: 3h

Command Line Entry**/LPDM**

Range 0-1h

Default: 0h

/ADA

Range: 0 or 1

Default: 1

/PULSE

Range: 0h-1h

Default: 0h

/5VCORE

Range: 0h-1h

Default: 0h

/SMI

Range: 0h-1h

Default: 0h

/SYSTHESIZER

Range: 0h-1h

Default: 1h

/SETUP_TIME

Range: 0h-1h

Default: 1h

/COMMAND_TIME

Range: 0h-FFh

Default: 6h

/RECOVERY

Range: 0h-FFh

Default: 3h

This parameter specifies the value for the recovery timing register.

Parameter**DMA Mode**

Range: 0-1

Default: 0

Command Line Entry**/DMAMODE**

Range: 0-1

Default: 0

A setting of 1 permits possible use of DMA (VG-469 only). The default is 0 for Plug and Play mode. When Plug and Play is enabled, the PC Card cannot be a DMA device. When Plug and Play mode is used (VG-469 only), this parameter should be 0. The Plug and Play implementation lets the software relocate the VG-469 registers from the default at 3E0h/3E1h to 3E2h/3E3h. Plug and Play is only available in extended mode.

Clock Speed Greater than 25MHZ

Range: 0h-1h

Default: 1h

/_25MHZ

Range: 0h-1h

Default: 1h

This parameter should be set to True (1) when the PCI clock speed or the external clock speed is set higher than 25MHZ.

Trident Omega 82C094, 82C194 (SSOMG094.exe)**Parameter****Number Of Supported Slots**

Range: 1-4

Default: 4

Command Line Entry**/SKT**

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

/PORT

Range: 0h-3FFh

Default: 3E0h

Specifies the base port address of the PCMCIA adapter(s). This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

/IRM

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

Parameter**Voltage Stabilization Delay**

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Language (alternate embedded Language)

Range: 0h-4h

Default: 0

Specifies the settings for alternate embedded language. The default setting is 0. This driver/utility may be configured to display messages in five different languages: 0=English (the default), 1=French, 2=German, 3=Italian, 4=Spanish.

Command Line Entry**/PWRON**

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/LANG

Range: 0h or 4h

Default: 0

Toshiba ToPIC (sstopic.exe)

Parameter

Card-Status-Change Interrupt IRQ Mask

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as status-change interrupt (event interrupt) IRQs. The status-change interrupt is generated when the PCMCIA adapter status changes (that is, a card is removed or inserted). Use this parameter to inform Socket Services of the IRQs that are connected to the Intel PCIC or IBM Thinkpad chip for use as the status-change interrupt signals. This information should be provided by the manufacturer of the system.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PC Card IREQ Mask

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IREQs. PC card IREQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted into your system. Use this parameter to inform Socket Services which IRQs are connected to the Intel controller for use as card interrupts. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

This switch setting is dependent on the system hardware design. It specifies the base port address of the first of up to two PCMCIA adapters. This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Intel 82365SL PCIC ID

Range: 0h-FFh

Default: 83h

Specifies an expected Intel 82365SL ID byte (Register 0) in addition to the standard values of 82h and 83h.

Command Line Entry

/SIRM

Range: 0h-FFFFh

Default: DEB8h

/IIRM

Range: 0h-FFFFh

Default: DEB8h

/PORT

Range: 0h-3FFh

Default: 3E0h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/ID

Range: 0h-FFh

Default: 83h

Parameter

Command Line Entry

NUMADA

Range: 1-2h
Default: 1

Undocumented entry.

/NUMADA

Range: 1-2h
Default: 1

Number Of Supported Slots

Range: 1-4
Default: 4

/SKT

Range: 1-4
Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

- n1 represents the number of slots in the first adapter.
- n2 represents the number of slots in the second adapter (if present).
- n3 represents the number of slots in the third adapter (if present).
- n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Reset On CPU Resume

Range: Yes or No
Default: No

/RST

Range: 0 or 1
Default: 1

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

Test for Warm Reboot

Range: Yes or No
Default: No

/WRB

Range: 0h or 1h
Default: 0

Specifies that Socket Services test for possible system resources depletion due to configured PCMCIA slots during a warm reboot. If enabled, and configured slots are discovered following a warm reboot, Socket Services clears all slot configurations and reboots the system.

/ZERO_MEM_WAIT_STATE

Range: 0-1
Default: 0

/ZERO_MEM_WAIT_STATE

Range: 0-1
Default: 0

Specifies the number of wait state cycles of 16-bit memory access to the Memory Window.

A setting of 0 indicates that System Memory access will complete in 3 CLKs (16 bit) or 6 CLKs (8 bit) with IOCHRDY high.

A setting of 1 indicates that system memory access will complete in 2 CLKs (16 bit) or 3 CLKs (8 bit) with *ZWS signal driven low.

/16BIT_MEM_WAIT_STATE

Range: 0-3
Default: 0

/16BIT_MEM_WAIT_STATE

Range: 0-3
Default: 0

These values determine the number of additional wait states for a 16-bit access to the system memory window.

Toshiba ToPIC95 (sstpic95.exe)

Parameter

Card-Status-Change Interrupt IRQ Mask

Range: 0h-FFFFh
Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as status-change interrupt (event interrupt) IRQs. The status-change interrupt is generated when the PCMCIA adapter status changes (that is, a card is removed or inserted). Use this parameter to inform Socket Services of the IRQs that are connected to the Intel PCIC or IBM Thinkpad chip for use as the status-change interrupt signals. This information should be provided by the manufacturer of the system.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PC Card IREQ Mask

Range: 0h-FFFFh
Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IREQs. PC card IREQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted into your system. Use this parameter to inform Socket Services which IRQs are connected to the Intel controller for use as card interrupts. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh
Default: 3E0h

This switch setting is dependent on the system hardware design. It specifies the base port address of the first of up to two PCMCIA adapters. This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

Voltage Stabilization Delay

Range: 0 through 65535
Default: 6667

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535
Default: 0

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No
Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Intel 82365SL PCIC ID

Range: 0h-FFh
Default: 83h

Specifies an expected Intel 82365SL ID byte (Register 0) in addition to the standard values of 82h and 83h.

Command Line Entry

/SIRM

Range: 0h-FFFFh
Default: DEB8h

/IIRM

Range: 0h-FFFFh
Default: DEB8h

/PORT

Range: 0h-3FFh
Default: 3E0h

/PWRON

Range: 01h-FFFFh
Default: 1A0B

/PWROFF

Range: 01h-FFFFh
Default: 0

/ADA

Range: 0 or 1
Default: 1

/ID

Range: 0h-FFh
Default: 83h

Parameter

Command Line Entry

Number Of Supported Slots

Range: 1-4
Default: 4

/SKT

Range: 1-4
Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

`/SKT=n1,n2,n3,n4`

where -

- n1 represents the number of slots in the first adapter.
- n2 represents the number of slots in the second adapter (if present).
- n3 represents the number of slots in the third adapter (if present).
- n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Reset On CPU Resume

Range: Yes or No
Default: No

/RST

Range: 0 or 1
Default: 1

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

Test for Warm Reboot

Range: Yes or No
Default: No

/WRB

Range: 0h or 1h
Default: 0

Specifies that Socket Services test for possible system resources depletion due to configured PCMCIA slots during a warm reboot. If enabled, and configured slots are discovered following a warm reboot, Socket Services clears all slot configurations and reboots the system.

/ZERO_MEM_WAIT_STATE

Range: 0-1
Default: 0

/ZERO_MEM_WAIT_STATE

Range: 0-1
Default: 0

Specifies the number of wait state cycles of 16-bit memory access to the Memory Window.

A setting of 0 indicates that System Memory access will complete in 3 CLKs (16 bit) or 6 CLKs (8 bit) with IOCHRDY high.

A setting of 1 indicates that system memory access will complete in 2 CLKs (16 bit) or 3 CLKs (8 bit) with *ZWS signal driven low.

/16BIT_MEM_WAIT_STATE

Range: 0-3
Default: 0

/16BIT_MEM_WAIT_STATE

Range: 0-3
Default: 0

These values determine the number of additional wait states for a 16-bit access to the system memory window.

Vadem VG365/465/468/469 (ssvadem.exe)**Parameter****/NUMADA**

Range: 1-2h

Default: 1

Undocumented entry.

Number Of Supported Sockets

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

Specifies the base port address of the PCMCIA adapters. This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Command Line Entry**/NUMADA**

Range: 1-2h

Default: 1

/SKT

Range: 1-4

Default: 4

/PORT

Range: 0h-3FFh

Default: 3E0h

/IRM

Range: 1h-FFFFh

Default: DEB8h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

Parameter**Install As Adapter Zero**

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

Command Line Entry**/ADA**

Range: 0 or 1

Default: 1

Card-Status-Change Interrupt in Pulse Mode

Range: 0h-1h

Default: 0h

Allows interrupts to be generated in pulse mode, rather than level mode.

Settings:

0 - Interrupts are passed to IRQ pin as level-sensitive.

1 - When interrupt occurs, IRQ pin is driven with the pulse train and allows interrupt sharing.

/PULSE

Range: 0h-1h

Default: 0h

DMA Mode

Range: 0-1

Default: 0

A setting of 1 permits possible use of DMA (VG-469 only). The default is 0 for Plug and Play mode. When Plug and Play is enabled, the PC Card cannot be a DMA device. When Plug and Play mode is used (VG-469 only), this parameter should be 0. The Plug and Play implementation lets the software relocate the VG-469 registers from the default at 3E0h/3E1h to 3E2h/3E3h. Plug and Play is only available in extended mode.

/DMAMODE

Range: 0-1

Default: 0

/INTERNAL_V_SENSE (Change Multiple voltage sensing)

Range: 0-1

Default: 0

This switch changes the multiple voltage (3.3V/5.0V) sensing from external to internal . A zero value (the default) sets voltage sensing to the external register. A value of one sets the voltage sense to the internal register.

/INTERNAL_V_SENSE

Range: 0-1

Default: 0

Terminal Count Steering

Range: 0-3

Default: 0

This parameter must be set for the appropriate terminal count steering mechanism being used by the hardware when DMA mode is enabled.

Settings:

0 - Automatic steering (to Write Enable when DMA read, to OE when DMA write)

1 - Steered to Write Enable

2 - Steered to Output Enable

3 - Terminal Count disable

/TC-STEER

Range: 0-1

Default: 0

No DMA Wait States

Range: 0-1

Default: 1

When True (1), the PCMCIA adapter will be configured to NOT generate wait states for DMA cycles (for use with standard ISA DMA cycles). When False (0), the PCMCIA adapter will be configured to generate wait states for DMA cycles (for use with 8MHZ DMA cycles).

/DMAWSB

Range: 0-1

Default: 1

16 Bit DMA Data Bus Width

Range: 0-1

Default: 0

When True (1), the PCMCIA adapter will be configured for 16-bit DMA data bus width. When False (0), the PCMCIA adapter will be configured for an 8-bit data bus width.

/DMASIZE

Range: 0-1

Default: 0

Parameter**/16IO_WAIT_STATE (Add Wait State to 16 bit system access)**

Range: 0-1

Default: 0

If set, this switch adds an additional wait state to a 16 bit I/O system access. The default is off (0) which does not add a wait state.

Command Line Entry**/16IO_WAIT_STATE**

Range: 0-1

Default: 0

/ZERO_MEM_WAIT_STATE

Range: 0-1

Default: 0

Specifies the number of wait state cycles of 16-bit memory access to the Memory Window.

A setting of 0 indicates that System Memory access will complete in 3 CLKs (16 bit) or 6 CLKs (8 bit) with IOCHRDY high.

A setting of 1 indicates that system memory access will complete in 2 CLKs (16 bit) or 3 CLKs (8 bit) with *ZWS signal driven low.

/ZERO_MEM_WAIT_STATE

Range: 0-1

Default: 0

VIA VT83C465/469 (ssvia.exe)

Parameter**/NUMADA**

Range: 1-2h

Default: 1

Undocumented entry.

Number Of Supported Slots**Range: 1-4**

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

A value of zero causes the number of slots to be determined dynamically. On machines which have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

/SKT=n1,n2,n3,n4

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Command Line Entry**/NUMADA**

Range: 1-2h

Default: 1

/SKT**Range: 1-4**

Default: 4

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

Specifies the base port address of the PCMCIA adapter(s). This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

/PORT

Range: 0h-3FFh

Default: 3E0h

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

/IRM

Range: 1h-FFFFh

Default: DEB8h

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

/PWRON

Range: 01h-FFFFh

Default: 1A0B

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

/PWROFF

Range: 01h-FFFFh

Default: 0

Parameter**Low Power Dynamic Mode**

Range: 0-1h

Default: 0h

Low Power Dynamic Mode Switch. Use to set for low power operation.

Settings:

0 - The clock runs always (low power not set).

1- Normal (low power) operation. Clock stopped when possible.

Command Line Entry**/LPDM**

Range 0-1h

Default: 0h

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

/ADA

Range: 0 or 1

Default: 1

Card Status Change Interrupt in SMI Mode

Range: 0h-1h

Default: 0h

When True (1), the PCMCIA adapter is configured to generate an SMI interrupt instead of an IRQ for card status change interrupts.

/SMI

Range: 0h-1h

Default: 0h

DMA Mode

Range: 0-1

Default: 0

A setting of 1 permits possible use of DMA (VG-469 only). The default is 0 for Plug and Play mode. When Plug and Play is enabled, the PC Card cannot be a DMA device. When Plug and Play mode is used (VG-469 only), this parameter should be 0. The Plug and Play implementation lets the software relocate the VG-469 registers from the default at 3E0h/3E1h to 3E2h/3E3h. Plug and Play is only available in extended mode

/DMAMODE

Range: 0-1

Default: 0

16 Bit DMA Data Bus Width

Range: 0-1

Default: 0

When True (1), the PCMCIA adapter will be configured for 16-bit DMA data bus width. When False (0), the PCMCIA adapter will be configured for an 8-bit data bus width.

/DMASIZE

Range: 0-1

Default: 0

VLSI VL82C146 (ssvlsi.exe)**Parameter****Number Of Supported Sockets**

Range: 1-4

Default: 4

Specifies the number of slots that driver supports. On machines that have a PCMCIA adapter that can support more slots than are present in the machine, this value should be set to the exact number of slots present.

If there is more than 1 adapter in the system, the following command line syntax is used:

```
/SKT=n1,n2,n3,n4
```

where -

n1 represents the number of slots in the first adapter.

n2 represents the number of slots in the second adapter (if present).

n3 represents the number of slots in the third adapter (if present).

n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Command Line Entry**/SKT**

Range: 1-4

Default: 4

PC Card IRQ Mask

Range: 1h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IRQs. PC card IRQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted in the system. Use this parameter to inform Socket Services which IRQs are connected to the controller for use as card interrupts. This information should be provided by the system manufacturer.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

/IRM

Range: 1h-FFFFh

Default: DEB8h

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

/PWRON

Range: 01h-FFFFh

Default: 1A0B

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, to wait for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

/PWROFF

Range: 01h-FFFFh

Default: 0

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The controller installs as the next available Adapter when this parameter is not specified.

/ADA

Range: 0 or 1

Default: 1

Parameter**Bus Delay**

Range: 0-2

Default: 0

Command Line Entry**/BUSDLY**

Range: 0-2

Default: 0

This switch controls the enabling/disabling of the ML-BUS delay. This is a delay that controls the access time to the ELC PCMCIA controllers. For older ELC revisions, this delay needed to be disabled, or else the PCMCIA controllers could not be accessed. The newer controllers could not be accessed unless it was enabled. This parameter applies only to the SSVLSI driver. To make SSVLSI work on all systems, this switch was added.

The following settings are possible:

- 0 - Don't modify the current ML-BUS delay control (assume BIOS sets it up).
- 1 - Always disable the ML-BUS delay
- 2 - Always enable the ML-BUS delay.

3.3 Volt Mode

Range: 0-1

Default: 0

/33VOLT

Range: 0-1

Default: 0

This switch enables/disables the setting of VCC to 3.3 volts.

The following settings are possible:

- 0 - Prevents VCC from being set to 3.3 volts.
- 1 - Allows VCC to be set to 3.3 volts.

02 Microsystems 6722/29/30 (SS02M.exe)**Parameter****Card-Status-Change Interrupt IRQ Mask**

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as status-change interrupt (event interrupt) IRQs. The status-change interrupt is generated when the PCMCIA adapter status changes (that is, a card is removed or inserted). Use this parameter to inform Socket Services of the IRQs that are connected to the Intel PCIC or IBM Thinkpad chip for use as the status-change interrupt signals. This information should be provided by the manufacturer of the system.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PC Card IREQ Mask

Range: 0h-FFFFh

Default: DEB8h

This switch setting depends on the system hardware design and processor chip. It specifies the 16-bit mask of available IRQs that can be assigned as valid PC card IREQs. PC card IREQs are interrupts that are required for the operation of the device (modem card, ATA card, etc.) that is inserted into your system. Use this parameter to inform Socket Services which IRQs are connected to the Intel controller for use as card interrupts. This information should be provided by the system vendor.

The most significant bit represents IRQ 15 and the least significant bit represents IRQ 0. For example, a value of DEB8h (1101-1110-1011-1000 binary) indicates that IRQs 15, 14, 12, 11, 10, 9, 7, 5, 4, and 3 are valid for assignment as card-status-change IRQs.

PCMCIA Adapter Base Port Address

Range: 0h-3FFh

Default: 3E0h

This switch setting is dependent on the system hardware design. It specifies the base port address of the first of up to two PCMCIA adapters. This is the I/O address used by *CardSoft* to communicate with the PCMCIA Adapter(s).

Voltage Stabilization Delay

Range: 0 through 65535

Default: 6667

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after a voltage change on a slot. This occurs when a card is inserted into a slot and when a Flash memory card is written to or erased. The default is 100ms.

Zero Voltage Stabilization Delay

Range: 0 through 65535

Default: 0

Specifies the amount of time, in 15- μ s units, that the SS365SL waits for the voltage to stabilize after voltage has been set to zero for a slot. This occurs when a card is removed from a slot.

Install As Adapter Zero

Range: Yes or No

Default: No

Forces installation of Socket Services as Adapter Zero. This feature may be used to override the presence (e.g., in the BIOS) of any previously loaded non-compliant PCMCIA implementations. The SS365SL installs as the next available Adapter when this parameter is not specified.

Intel 82365SL PCIC ID

Range: 0h-FFh

Default: 84h

Specifies an expected ID byte (Register 0) in addition to the standard values. The default is no additional IDs.

Command Line Entry**/SIRM**

Range: 0h-FFFFh

Default: DEB8h

/IIRM

Range: 0h-FFFFh

Default: DEB8h

/PORT

Range: 0h-3FFh

Default: 3E0h

/PWRON

Range: 01h-FFFFh

Default: 1A0B

/PWROFF

Range: 01h-FFFFh

Default: 0

/ADA

Range: 0 or 1

Default: 1

/ID

Range: 0h-FFh

Default: 84h

Parameter**Command Line Entry**

Number Of Supported Sockets

Range: 1-4
Default: 0

/SKT

Range: 1-4
Default: 0

Specifies the number of slots that the controller supports (1-4). The default value (0) indicates that Socket Services dynamically determines the number of slots present. If a number other than zero is specified, it must be set to the exact number of slots present in the system.

If there is more than 1 adapter in the system, the following command line syntax is used:

`/SKT=n1,n2,n3,n4`

where -

- n1 represents the number of slots in the first adapter.
- n2 represents the number of slots in the second adapter (if present).
- n3 represents the number of slots in the third adapter (if present).
- n4 represents the number of slots in the fourth adapter (if present).

Up to 4 adapters are supported.

Reset On CPU Resume

Range: Yes or No
Default: No

/RST

Range: 0 or 1
Default: 1

Controls whether the controller is reset on a CPU Resume event. A reset returns the controller to its uninitialized power-up state.

Test for Warm Reboot

Range: Yes or No
Default: No

/WRB

Range: 0h or 1h
Default: 0

Specifies that Socket Services test for possible system resources depletion due to configured PCMCIA slots during a warm reboot. If enabled, and configured slots are discovered following a warm reboot, Socket Services clears all slot configurations and reboots the system.

/ZERO_MEM_WAIT_STATE

Range: 0-1
Default: 0

/ZERO_MEM_WAIT_STATE

Range: 0-1
Default: 0

Specifies the number of wait state cycles of 16-bit memory access to the Memory Window.

A setting of 0 indicates that System Memory access will complete in 3 CLKs (16 bit) or 6 CLKs (8 bit) with IOCHRDY high.

A setting of 1 indicates that system memory access will complete in 2 CLKs (16 bit) or 3 CLKs (8 bit) with *ZWS signal driven low.

/16BIT_MEM_WAIT_STATE

Range: 0-3
Default: 0

/16BIT_MEM_WAIT_STATE

Range: 0-3
Default: 0

These values determine the number of additional wait states for a 16-bit access to the system memory window.

Card Services (cs.sys)

Card Services software coordinates access to PC cards, PCMCIA slots, and system resources among multiple clients. These clients may be device drivers, system utilities, or application programs. Card Services provides access to slot hardware through Socket Services software. Card services is not configurable through this configuration utility in this release of *CardSoft*.

CSALLOC

This is a DOS utility and device driver that scans the system for available memory, I/O Port, and IRQ resources. It then writes this information to the file CSALLOC.INI, which is used by Card Services to determine what system resources are available for PC card configuration

Parameter

/O Address Aliasing Count

Range: 0-3Fh

Default: 3h

Specifies the number of times I/O Port addresses are aliased at 400h and above, and given to Card Services. For example, if this value is set to 1, and the I/O Port address is 3E8, it is given to Card Services as 3E8 and 7E8. As another example, the default value (3) with an I/O Port address of 3E8 causes 3E8, 7E8, BE8, and FE8 to be given to Card Services.

Number 1 Communications Port IRQ

Range: 1-15

Default: 4

Specifies the IRQ for COM 1. This value should be changed only if the system uses an IRQ other than IRQ 4 for COM 1.

Number 2 Communications Port IRQ

Range: 1-15

Default: 3

Specifies the IRQ for COM 2. This value should be changed only if the system uses an IRQ other than IRQ 3 for COM 2.

Number 3 Communications Port IRQ

Range: 1-15

Default: 4

Specifies the IRQ for COM 3. This value should be changed only if the system uses an IRQ other than IRQ 4 for COM 3.

Number 4 Communications Port IRQ

Range: 1-15

Default: 3

Specifies the IRQ for COM 4. This value should be changed only if the system uses an IRQ other than IRQ 3 for COM 4.

I/O Scanning Address

Range: 0-FFFFh

Default: 80h

Specifies the I/O address to be used for I/O scanning. Always specify an I/O address that is safe for performing input or output operations without interference from other components.

/IOCODE (Set I/O Lines Decoded)

Range: Ah-10h

Default: Ah

Specifies how many I/O lines are decoded by devices on this machine. The default is 10. In special cases this value can be modified to allow additional I/O addresses to be used for PCMCIA. The actual number of additional addresses is determined by this parameter's setting and can range between 10 and 16.

Command Line Entry

/IOALIASES

Range: 0-3Fh

Default: 3h

/COM1IRQ

Range: 1-15

Default: 4

/COM2IRQ

Range: 1-15

Default: 3

/COM3IRQ

Range: 1-15

Default: 4

/COM4IRQ

Range: 1-15

Default: 3

/TESTIO

Range: 0-FFFFh

Default: 80h

/IOCODE

Range: 10-16

Default: 10

Parameter
/PNPISAIOWIDTH
 Range: 4h-20h
 Default: 20h

Command Line Entry
/PNPISAIOWIDTH
 Range: 4h-20h
 Default: 20h

Undocumented entry.

SRAM Driver (MTSRAM.EXE)

MTSRAM is a driver that supports SRAM cards.

Parameter
Maximum Number of Memory Windows
 Range: 1-4h
 Default: 4h

Command Line Entry
/NUMWINS
 Range: 1-4h
 Default: 4h

Specifies the number of memory windows that will be configured and used for accessing SRAM cards. The memory windows are set up in upper memory (for example, the D000-DFFF range), in any available area that is found. A value of 2 provides better SRAM card performance than a value of 1. Optimum performance is achieved with a value of 3. The total number of memory windows configured is a multiple of this value and the number of SRAM memory cards inserted (and recognized) in the system's PCMCIA slots. This represents the number of windows per SRAM card inserted and recognized.

Memory Card Memory Window Size
 Values: 4, 8, 16, and 32
 Default: 4

/SIZ
 Values: 4, 8, 16, or 32
 Default: 4

Specifies the size of each of the system memory windows to be used to access SRAM cards. The value is stated in KB and must be 4, 8, 16, or 32. The default value of 4 represents a 4KB window.

Memory Technology Driver (MTDDR.V.EXE)

This component must be installed in order to support all Memory cards. It works in conjunction with card-specific MTDs to support a wide variety of current Flash Memory cards. It also supports SRAM cards (providing MTSRAM.EXE is also installed), and allows sharing of drive letters between the different types of memory cards (Flash, SRAM, and ATA).

Parameter
Maximum Partitions Per Card
 Range: 1-4
 Default: 1

Command Line Entry
/PART
 Range: 1-4
 Default: 1

Specifies the maximum number of partitions that a Flash Memory or SRAM card can contain (not including any area on the card that may be used for boot code). In most cases, it is best to use the default value of 1, since there are no significant advantages to having multiple partitions on a memory card (and there are some drawbacks to having multiple partitions). For more information on memory card partitions, refer to the *Flash File System User's Guide*.

IDE/ATA Device Driver (atadvr.exe)

ATADRV.EXE is a Hard Disk or Flash Disk configuration device driver for removable PCMCIA devices that emulate an IDE type hard disk. ATADRV.EXE is strictly a Hard Disk or Flash Disk driver that communicates to industry-standard hard disk controllers. ATADRV.EXE can manage up to eight PCMCIA devices.

ATADRV communicates with the *CardID* component to obtain information on card insertion and removal. Once a card is inserted and recognized by *CardID*, ATADRV communicates with the card as if it were an IDE disk. ATADRV must be installed before the *CARDID.EXE* driver. ATADRV supports the following parameters:

Configuration Parameters

Parameter

Slave Mode

Range: No or Yes

Default: No

Specifies if ATADRV is to be run in slave mode. The MTD Driver (MTDDRV) is the only master control driver currently available. Installs the ATADRV device driver as a slave (/S:x) to MTDDRV. It also specifies the number of drives (1-8) it can support. A value of 0 can also be used with /D or /S. When a value of 0 is used, only the mode that was specified (/D or /S) is implemented, not the number of drives assigned during installation or specified using the CONFIG utility. When this /S switch is used, ATADRV must be installed in CONFIG.SYS before MTDDRV and both ATADRV and MTDDRV must be installed before CARDID. See ATA Driver Modes section which follows..

Examples: devicehigh=c:\CardSoft\atadr.exe/D:1
devicehigh=c:\CardSoft\atadr.exe /S:1

Command Line Entry

/S:x

Range: 0-8

Default: 2

Number of Drive Units

Range: 0-8

Default: 2

/D:x

Range: 0-8

Default: 2

Specifies the number of drives that the system supports when installed either as a block device driver or as a slave device driver. Installs the ATADRV device driver as a block device driver (/D:x). It also specifies the number of drives (1-8) it can support. A value of 0 can also be used with /D or /S. When a value of 0 is used, only the mode that was specified (/D or /S) is implemented, not the number of drives assigned during installation or specified using the CONFIG utility. When the /D switch is used, ATADRV must be installed in CONFIG.SYS before CARDID. See ATA Driver Modes section which follows.

Examples: devicehigh=c:\CardSoft\atadr.exe /D:1
devicehigh=c:\CardSoft\atadr.exe /S:1

ATA Driver Modes

This section explains the rationale for using the /D:x and /S:x configuration modes just described.

The ATA driver can operate both as a block device drive (/D:x) and a character device driver (/S:x). When used as a character device driver, you cannot access the drive except through the master driver MTDDRV.

In most cases, the ATADRV program will be an entry in your CONFIG.SYS file configured internally in the mode in which you wish to use it. However, there may be times when you wish to change that mode, especially since CONFIG.SYS has multiple options in DOS 6.0 or later.

In one case, you may have MTDDRV and ATADRV configured for master-slave operations as they are in your CONFIG.SYS as is the case in the following example:

```
[ALLCARDS]
DEVICE=ATADRV.EXE /S:2
Device=MTDDRV.EXE
```

Here [ALLCARDS] is used if you have Flash or SRAM cards in combination with ATA cards.

This allows MTDDRV to share drive letters with ATADRV. In this case the /S switch allows the driver to be slaved to a master control driver (MTDDRV) for drive letter selection. When this parameter is used, ATADRV must be installed in CONFIG.SYS before MTDDRV.

If you have a special case, where you need to minimize RAM usage and you need the ATA drive but not the memory card drive, you might have an entry similar to the following:

```
[ONLYATACARD]
DEVICE=ATADRV.EXE /D:2
```

If your ATADRV program is originally configured for slave mode at install with 2 drives, then your CONFIG.SYS might look like the following:

```
[ALLCARDS]
DEVICE=ATADRV.EXE
DEVICE=MTTDRV.EXE

[ONLYATACARD]
DEVICE=ATADRV.EXE /D:2
```

In this case, it is not necessary to declare the driver as a slave in the first entry. On the other hand, you might have it configured originally as a block device driver, for 2 drives and , in that case, your CONFIG.SYS might look like this:

```
[ALLCARDS]
DEVICE=ATADRV.EXE /S:2
DEVICE=MTDDRV.EXE

[ONLYATACARD]
DEVICE=ATADRV.EXE
```

If you are unsure, you can specify explicitly what the mode or number of drives are, by entries similar to the following in your CONFIG.SYS file:

```
[ALLCARDS]
DEVICE=ATADRV.EXE /S:2
DEVICE=MTDDRV.EXE

[ONLYATCARD]
DEVICE=ATADRV.EXE /D:2
```

DEVICEHIGH Line Parameters

Parameter

/D or /S:

Range: 0-8

Default: 2

Examples: devicehigh=c:\CardSoft\atadriv.exe /D:1
devicehigh=c:\CardSoft\atadriv.exe /S:1

Command Line Entry

/D or /S:

Range: 0-8

Default: 2

This parameter installs the ATADRV device driver as a block device driver (/D:x) or as a slave (/S:x) to MTDDRV. It also specifies the number of drives (1-8) it can support. A value of 0 can also be used with /D or /S. When a value of 0 is used, only the mode that was specified (/D or /S) changes, not the number of drives assigned during installation or specified using the CONFIG utility. When this parameter is used, ATADRV must be installed in CONFIG.SYS before MTDDRV.

Disk Partitioning Utility (atainit.exe)

ATAINIT.EXE is a disk partitioning utility that must be used to create a partition on any drive supported by ATADRV. When a new drive is inserted into a socket, it is not recognized since there is no CMOS or other common method to find out its physical parameters. ATAINIT interrogates the drive to find the physical parameters to use.

ATAINIT Return Code

There are five return codes for ATAINIT, which are defined as follows:

Return Code	Meaning
00	No error
81	Unable to address card
82	Bad write
84	Requested quit via keyboard
88	Unable to ID the drive

Language Parameter

Parameter

Language (alternate embedded Language)

Range: 0h-4h

Default: 0

Specifies the settings for alternate embedded language. The default setting is 0. This driver/utility may be configured to display messages in five different languages: 0=English (the default), 1=French, 2=German, 3=Italian, 4=Spanish.

Command Line Entry

/LANG

Range: 0h or 4h

Default: 0

Flash File System Executables

This section lists the parameters for each of the Flash File System executables that are configurable.

Card-Specific Memory Technology Drivers

MTAA.EXE, MTAB.EXE, MTI1.EXE, MTI2P.EXE, MTHB2

The card-specific memory technology drivers work in conjunction with MTDDR to support various types of Flash Memory cards. Card types currently supported are AMD Type A, AMD Type B, AMD Type C, Intel Series 1, Intel Series 2 Plus (and compatibles) and Hitachi2 cards.

Parameter

/NUMWINS (Number of Memory Windows)

Range: 1-4h

Default: 4h (varies for different MTDs)

Specifies the number of memory windows that will be configured and used for accessing Flash Memory cards. The memory windows are set up in upper memory (for example, the D000-DFFF range), in any available area that is found. A value of 2 provides better Flash card performance than a value of 1. Optimum performance is achieved with a value of 3. The total number of memory windows configured is a multiple of this value and the number of Flash memory cards inserted (and recognized) in the system's PCMCIA slots.

Memory Card Memory Window Size

Values: 4, 8, 16, and 32

Default: 4

Specifies the size of the system memory windows to be used to access Flash Memory cards. The value is stated in KB and must be 4, 8, 16, or 32. The default value of 4 represents a 4KB window.

/EBPART

Values: 1h-8000h

Default: 0

This parameter sets the programmable erasable block size. 0 (the default) disables this setting. This parameter applies only to the MTHB2 driver.

Command Line Entry

/NUMWINS

Range: 1-4h

Default: 4h

/SIZ

Values: 4, 8, 16, or 32

Default: 4

/EBPART

Values: 1h-8000h

Default: 0

Chapter 4

Customizing Your System

This chapter describes the different ways you can customize your system:

- You can select the operating environment and situations in which system beeps are emitted relating to PC card events.
- You can modify the *CardSoft* DEVICEHIGH lines in CONFIG.SYS so that you are installing only those drivers that are needed for the PC cards you are using.
- You can modify CSALLOC.INI to:
 - Exclude resources from being scanned by CSALLOC, thereby preventing *CardSoft* from trying to use memory, IRQ, or I/O Port resources that are needed for other devices or drivers installed on your system.
 - Include resources that you know are available for use by *CardSoft*, but are not being detected by CSALLOC.
 - Reserve resources that are specifically required by a PC card that you are using.
 - Allow specific resources to be shared by *CardSoft* and other system components.

Note *CSALLOC.INI can be modified in two ways. You can (a) manually edit CSALLOC.INI to add special "keywords" that cause resources to be reserved, included, excluded, or shared, or (b) use the Resource Allocation function of the Configuration utility to change system resource assignments (the Resource Allocation function will then add the appropriate keyword entries to CSALLOC.INI).*

Using Keywords to Modify System Resource Assignments

CSALLOC.INI is a file that lists the memory, IRQ, and I/O Port resources that are available on your system for use by the *CardSoft* software. It also informs *CardSoft* which resources are reserved for use by specific PC cards that you have. CSALLOC.INI is created when *CardSoft* is installed, and can be updated at any time by running CSALLOC from the DOS command line (described in Chapter 2).

Note *Due to the fact that some system devices (such as network cards) may not be active when CSALLOC is scanning for available resources, CSALLOC may include resources that are needed for the inactive device, thereby creating a resource conflict on your system. Also, system hardware design and security features may make some resources undetectable by CSALLOC.*

To resolve this, several keywords can be manually added to CSALLOC.INI to exclude resources from being used by *CardSoft*. Other keywords also enable you to include resources that you know are available, but are not listed in CSALLOC.INI. Keywords that enable you to reserve resources are also available. These keywords can be added to the end of the CSALLOC.INI file (which is located in the *CardSoft* directory) using any text editor (such as the MS-DOS EDIT program). The keywords are described in the following sections.

Note *Reserved resources are those resources that are specifically required by a PC card in order for the card to function properly. For example, if you have a PCMCIA network card that specifically requires IRQ5 and the memory region D000-D7FF, you would have to reserve those resources to prevent *CardSoft* from assigning them to other PC cards you may be using.*

MEM, IO, and IRQ Lines in CSALLOC.INI

The MEM line in CSALLOC.INI informs *CardSoft* that the listed memory areas are not being used, and are available to *CardSoft*. *CardSoft* uses these memory areas to open "windows" to PC cards for reading from or writing to card memory. Normally, the only memory range that you should see on this line is D000-DFFF.

The IO line in CSALLOC.INI lists ranges of I/O Port addresses that are available. For example, the value 108-16F means that all I/O Port addresses from 108 through 16F are available to *CardSoft*. If a range of I/O Port addresses is not listed, it will never be used by *CardSoft*, because CSALLOC has determined that these I/O Port addresses are used by another device (or application) on your system.

The IRQ line in CSALLOC.INI lists all the IRQs that are available to *CardSoft*. If a range of numbers is listed (for example, 3-6), it means that all IRQs in that range are available. Any IRQs that are not listed are not available and will never be used by *CardSoft*, because CSALLOC has detected that they have already been assigned to something else.

Note If a resource is not listed on one of these three lines, you cannot use other keywords to make it a reserved or shared resource. You can, however, use an "include" keyword to include the resource; once it is included, you can then use other keywords to make it a reserved or shared resource.

IOEXCLUDE

IOEXCLUDE enables you to exclude I/O Port addresses from being used by *CardSoft*. For example, if you know that the I/O Port address range 1B8-1BF is required for a sound card, an ISA LAN card, a joystick, or other such device that is installed in your system, but CSALLOC is listing this address range on the IO= line, you would add the following line at the end of your CSALLOC.INI file:

```
IOEXCLUDE=1B8-1BF
```

This line prevents CSALLOC from scanning this range of I/O Port addresses and, consequently, from listing it as available. If you are having a problem using a device, refer to the documentation you received with the device to determine if it requires a specific I/O Port address.

IOINCLUDE

IOINCLUDE enables you to include I/O Port addresses for use by *CardSoft*. For example, if you know that the I/O Port address range 1F8-1FF is not being used on your system, but this range is not listed on the IO= line of CSALLOC.INI, you would add the following line at the end of your CSALLOC.INI file:

```
IOINCLUDE=1F8-1FF
```

This line causes CSALLOC to include this I/O Port address range as available (it will not, however, be listed on the IO= line when you run CSALLOC from the DOS command line to display the available resources on your system).

IRQEXCLUDE

IRQEXCLUDE enables you to exclude IRQ interrupts from being used by *CardSoft*. For example, if you have a sound card, ISA LAN card, joystick, or other such device that specifically requires IRQ 5 to function properly, but this IRQ is listed in CSALLOC.INI as an available IRQ, you would add the following line at the end of CSALLOC.INI to reserve IRQ 5 for your network card:

```
IRQEXCLUDE=5
```

This line prevents CSALLOC from scanning IRQ 5 and, consequently, from listing it as an available resource. If you are having a problem using a device, refer to the documentation you received with the device to determine if it requires a specific IRQ in order to function.

IRQINCLUDE

IRQINCLUDE enables you to include IRQ interrupts for use by *CardSoft*. For example, if you know that IRQ 11 (B in hexadecimal format) is not being used on your system, but it is not being listed in CSALLOC.INI as an available IRQ, you would add the following line at the end of CSALLOC.INI to include this resource:

```
IRQINCLUDE=B
```

This line causes CSALLOC to include IRQ 11 as an available IRQ (it will not, however, be listed on the IRQ= line when you run CSALLOC from the DOS command line to display the available resources on your system).

MEMEXCLUDE

MEMEXCLUDE enables you to exclude memory regions from being scanned by *CardSoft* (and therefore being listed as available). Since *CardSoft* normally requires only the memory region (or part of the region) between D000-DFFF, you can manually exclude any other memory regions that are listed on the MEM= line of CSALLOC.INI. For example, if the MEM= line in CSALLOC.INI is

```
MEM=C800-CFFF, D000-DFFF, E000-EFFF
```

you could add the following line at the end of CSALLOC.INI to prevent memory regions C800-CFFF and E000-EFFF from being scanned by CSALLOC, thereby making these memory regions available to other system components:

```
MEMEXCLUDE=C800-CFFF, E000-EFFF
```

After adding this line, the memory regions C800-CFFF and E000-EFFF will no longer be listed on the MEM= line when you run CSALLOC from the DOS command line to display the available resources on your system.

MEMINCLUDE

MEMINCLUDE enables you to include memory regions for use by *CardSoft*. For example, if you know that the memory region D000-D7FF is not specifically needed for other system components, but this memory region is not listed on the MEM= line in CSALLOC.INI, you would add the following line at the end of your CSALLOC.INI file to make this region available to *CardSoft*:

```
MEMINCLUDE=D000-D7FF
```

This line causes CSALLOC to include the D000-D7FF memory region as a resource that *CardSoft* can use (it will not, however, be listed on the MEM= line when you run CSALLOC from the DOS command line to display the available resources on your system).

RIO

RIO enables you to reserve I/O Port addresses that are specifically needed by a PC card that you are using, or any other system devices. For example, if you have a PCMCIA network card that can only use I/O Port addresses 300-307, and you have an I/O card that uses only I/O Port addresses 280-287, you would add the following line at the end of CSALLOC.INI to prevent *CardSoft* from assigning this address range to other PC cards that you may be using:

```
RIO=280-287, 300-307
```

Notes If the I/O Port range you want to reserve is not listed on the IO= line of CSALLOC.INI, you must add an IOINCLUDE line in CSALLOC.INI to make the resource available before you can reserve it.

The following IRQ resources are automatically reserved on most systems: the hard drive I/O addresses (170-177 and 370-377) and the COM port I/O addresses (2E8-2EF, 2F8-2FF, 3E8-3EF, and 3F8-3FF).

RIRQ

RIRQ enables you to reserve IRQ interrupts that are specifically needed by a PC card that you are using. For example, if you have a PCMCIA network card that uses only IRQ 12 (C in hexadecimal), you would add the following line at the end of CSALLOC.INI to prevent *CardSoft* from assigning this IRQ to other PC cards that you may be using:

```
RIRQ=C
```

Note If the IRQ you want to reserve is not listed on the IRQ= line of CSALLOC.INI, you must add an IRQINCLUDE line to make the resource available before you can reserve it.

RMEM

RMEM enables you to reserve memory regions that are specifically needed by a PC card that you are using. For example, if a PCMCIA network card specifically requires the memory region D800-DFFF in order to function properly, you would add the following line at the end of CSALLOC.INI to prevent *CardSoft* from assigning this memory region to other PC cards that you may be using:

```
RMEM=D800-DFFF
```

Note If the memory region you want to reserve is not listed on the MEM= line of CSALLOC.INI, you must add a MEMINCLUDE line to CSALLOC.INI to make the memory region available before you can reserve it.

SIO

SIO enables *CardSoft* to share I/O Port addresses with other system components. For example, if you have a modem application that uses COM port address 2E8-2EF, and you want your PCMCIA modem card to use that address when you are using the application, you would add the following line to CSALLOC.INI to enable *CardSoft* and the modem application to share this I/O resource:

```
SIO=2E8-2EF
```

Note If the I/O Port range you want *CardSoft* to share is not listed on the IO= line of CSALLOC.INI, you must add an IOINCLUDE line in CSALLOC.INI to make the resource available before you can make it shareable.

SIRQ

SIRQ enables *CardSoft* to share IRQ interrupts with other system components. For example, if you have a modem application that uses IRQ 4, and you want your PCMCIA modem card to also use IRQ 4 when you are using this modem application, you would add the following line to CSALLOC.INI to enable *CardSoft* and the modem application to share this IRQ:

```
SIRQ=5
```

Note If the IRQ you want to make shareable is not listed on the IRQ= line of CSALLOC.INI, you must add an IRQINCLUDE line to make the resource available before you can make it shareable.

SMEM

SMEM enables *CardSoft* to share memory regions with other system components. For example, if you have a network application that uses memory region D800-DFFF for a memory window, and you want your PCMCIA network card to use that memory region when you use the network application, you would add the following line to CSALLOC.INI to enable *CardSoft* and the network application to share this memory region:

```
SMEM=D800-DFFF
```

Note If the memory region you want to make shareable is not listed on the MEM= line of CSALLOC.INI, you must add a MEMINCLUDE line to CSALLOC.INI to make the memory region available before you can reserve it.

Using CONFIG to Modify Resource Assignments

You can use the Resource Allocation function of the CONFIG Utility to customize system resource assignments, instead of manually adding keywords to CSALLOC.INI. The Resource Allocation function automatically adds the appropriate keywords to CSALLOC.INI, based on how you assign resources.

To access the Resource Allocation function:

1. From the CARDSOFT subdirectory, type `config` and press Enter.
2. From the Utility pulldown menu, select Resource Allocation. The following screen appears.

Resource Allocation			
Memory Addresses	IO Ports Scanned Resources	IRQs	
D000-DFFF	108-1EF 1F8-377 380-3BF 3E8-3EF	3 5 9-13	< OK >
			<Cancel>
			< Help >
			< Edit >
C000-CFFF X E000-EFFF X	170-177 R 2E8-2EF R 2F8-2FF R 370-377 R		< Scan >

The top row of boxes (Scanned Resources) lists the resources that CSALLOC has determined are available for use by *CardSoft*. The information in these boxes cannot be modified; if you try to change them, three beep tones are emitted.

The bottom row of boxes (Modified Resources) list all resources that are marked as reserved, included, excluded, or shareable. Resources that were automatically excluded or reserved when *CardSoft* was installed should already be displayed in these boxes (normally, the Memory and I/O Port resources listed in the sample screen are reserved/excluded).

The following sections describe the different ways you can modify resource assignments on your system.

Editing Modified Resources List

To edit the Modified Resources list:

1. On the Resource Allocation screen, position the cursor in the appropriate resource list in the Modified Resource section (bottom section) of the screen. For example, if you want to edit the I/O Port assignments, position the cursor in the middle box in the lower row of boxes.
2. **If you want to add a range of resources to the list:**
 - a. Select <Edit> to display the Edit Options screen, then select <Add New> to display the Add New Resource screen.
 - b. Type in the value (or range of values) that you want to add to the Modified Resource list. For example, if you want to add the I/O port range 1E0-1E7, type 1E0 in the left-most field, press TAB, and type 1E7 in the right-most field.
 - c. Use the TAB key to move to the resource type list at the bottom of the screen. Then use the ↑ or ↓ arrow key to select the resource type (Include, Exclude, Reserved, or Shared). For a description of the different resource types, refer to the keyword descriptions at the beginning of this chapter. For example, if you want to know about reserved I/O ports, refer to the description of the RIO keyword.
 - d. Once the appropriate resource type is highlighted, press the Space Bar to toggle the bullet onto your selection. Then select <OK>, and proceed to Step 3.

If you want to change the assignment of a listed range of resources:

- a. Highlight the range of resources whose assignment you want to change (for example, if you want to change a range of I/O ports from Reserved to Shared, highlight the desired range of I/O ports).
- b. Select <Edit> to display the Edit Options screen, then select <Change> to display the Change Resources screen.
- c. If you want to change part of the selected range, type in the range of values whose assignment you want to change.
- d. Use the TAB key to move to the Resource Type list at the bottom of the screen.
- e. Use the ↑ or ↓ arrow key to select the resource type (Include, Exclude, Reserved, or Shared) you want to assign to the specified resource. For a description of the different resource types, refer to the keyword descriptions at the beginning of this chapter. For example, if you want to know about reserved I/O ports, refer to the description of the RIO keyword.
- f. Once the appropriate resource type is highlighted, press the Space Bar to toggle the bullet onto your selection. Then select <OK>, and proceed to Step 3.

If you want to delete a resource from the list, or part of a range of resources:

- a. Highlight the range of resources that contains the values you want to delete.
 - b. Select <Edit> to display the Edit Options screen, then select <Delete> to display the Delete Resources screen.
 - c. If you want to delete part of the selected range, type in the range of values you want to delete. Then select <OK>, and proceed to Step 3.
3. A confirmation dialog box appears. Select <OK> to complete the operation and return to the Resource Allocation screen, or select <Cancel> to return to the Resource Allocation screen without making the change. Repeat Step 2 for each resource assignment you want to modify.
 4. When you are done modifying the list, select <OK>.
 5. A dialog box appears. Select Save to save your modifications. After the save, select <Done> to return to the CONFIG main screen, or select <Open> to make more modifications to the Resource List.

Using a Customized .INI File

By default, *CardSoft* gets resource availability information from the file CSALLOC.INI. You may, however, want to create a customized version of CSALLOC.INI to enable you to automatically load special resource configurations. To do so, use any text editor to create the file, and name the file something other than CSALLOC.INI (e.g., MYFILE.INI). Then modify the CSALLOC line in CONFIG.SYS to include the name of your customized .INI file, as shown here:

```
device=c:\CardSoft\csalloc.exe c:\CardSoft\myfile.ini
```

System Beeps

When you insert a card, your system emits a beep code to let you know whether or not the *CardSoft* software was able to recognize and configure the card. These beep codes are:

Medium tone followed by high tone beep

The PC card was recognized and successfully configured when inserted.

Single low tone beep

The PC card was recognized, but not successfully configured when inserted. When you hear a single low tone, it means the card will not work, and you may have to change your system configuration or *CardSoft* configuration to get the card to work.

Also, when you remove a card, you should hear a high tone followed by a medium tone. This indicates that *CardSoft* is aware that the card has been removed.

Configuring System Beeps

By default, *CardSoft* is configured to emit system beeps in both DOS and Windows whenever you insert or remove a card, or if a configuration conflict is detected when you insert a card.

You can, however, select the situations in which you want *CardSoft* to produce a beep by adding a `/BEEPTYPES` switch, followed by a two-digit hex value, to the `devicehigh=c:\CardSoft\cs.sys` line in `CONFIG.SYS` (e.g., `devicehigh=c:\CardSoft\cs.sys /BEEPTYPES CF`). The valid hex values that you can use are listed here. The first digit controls the operating environment (DOS, Windows, or both) in which beeps are emitted. The second digit controls the types of events (card insertion, card removal, configuration conflict, and artificial events) that cause a beep to be emitted. If either digit in this value is 0, no beeps are emitted under any circumstances, regardless of the other digit.

First Digit

C = Beep in both DOS and Windows

8 = Beep in DOS only

4 = Beep in Windows only

0 = No beeps (even when event type beeps are selected)

Second Digit

F = Beep on card insertion, card removal, configuration conflict, and artificial events

E = Beep on card insertion, card removal, and configuration conflict

D = Beep on card insertion, card removal, and artificial events

C = Beep on card insertion and card removal

B = Beep on card insertion, configuration conflict, and artificial events

A = Beep on card insertion and configuration conflict

9 = Beep on card insertion and artificial events

8 = Beep on card insertion only

7 = Beep on card removal, configuration conflict, and artificial events

6 = Beep on card removal and configuration conflict

5 = Beep on card removal and artificial events

4 = Beep on card removal only

3 = Beep on configuration conflict and artificial events

2 = Beep on configuration conflict only

1 = Beep on artificial events only

0 = No beeps (even if operating environment is specified)

Note *The term "artificial events" refers to logical (as opposed to physical) events. For example, if the system enters a power management Suspend state, any cards that are inserted into PCMCIA slots are logically removed. Similarly, when the system resumes from its Suspend state, any cards that are already physically inserted into a slot are logically inserted. Both of these situations would cause a beep to be emitted if "artificial events" is selected as a beep event.*

Resolving Card-Specific Problems Using CardSoft

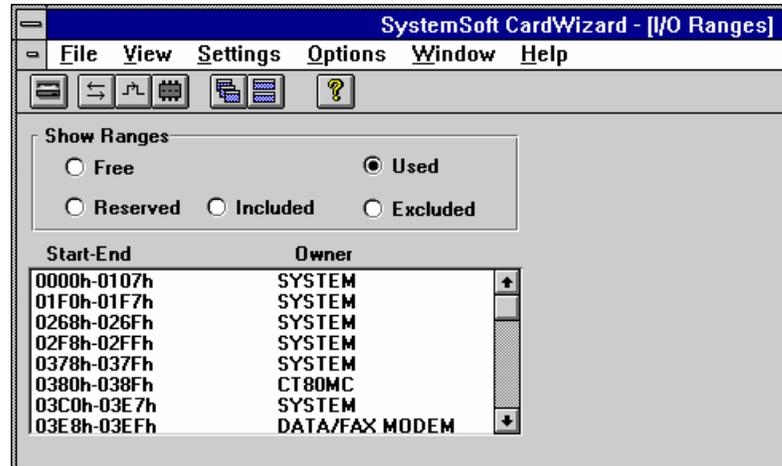
There may be occasions when you need more direct control over your system's resources within Windows. This section explains how to use CardSoft to reserve, include and exclude resources.

Procedure

Follow these steps to reserve, include or exclude an I/O range, IRQ or memory address:

1. Choose I/O Range, IRQ or Memory Range from the View menu or choose the I/O Range Icon in the toolbar.

The following is the dialog box for setting I/O ranges. CardSoft displays a similar dialog box for each of the three resources:

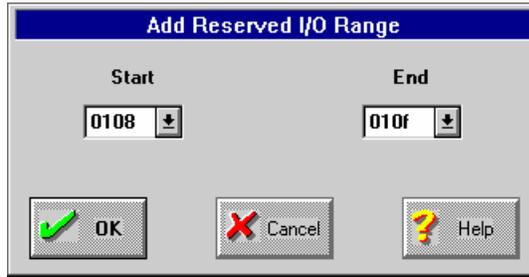


2. Select an I/O, IRQ or memory range.
3. Choose the Add button.

CardSoft displays a range dialog box. The example below is for setting the IRQ:



The range dialog boxes for memory address range is similar to the one for I/O ranges:



CardSoft displays all the resolvable/includable/excludable ranges. The I/O range and memory address ranges are in 8-bit increments.

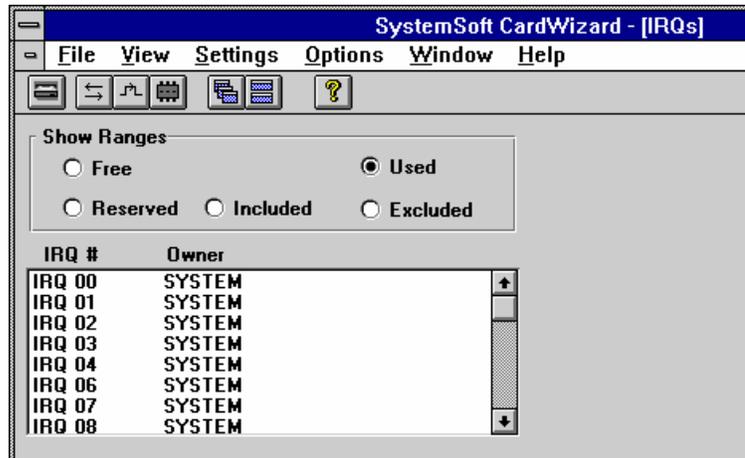
4. Use the mouse to select the range or the IRQ.

You cannot pick an illegal value. CardSoft helps you make decisions. For example, when you choose a starting address, CardSoft automatically displays the first logical ending address in the End box that is 8 bits away.

5. Choose OK.

CardSoft automatically reserves, includes or excludes the resource.

Reference



The following table summarizes the components of the I/O Range, IRQ and Memory Address Range dialog boxes.

Button/Column	Description
Free	A free resource is one that is available to CardSoft for PC card configurations.
Reserved	A reserved resource is absolutely needed by one of your PC cards. Some cards can use any resource as long as it is available. Reserving a resource prevents its use except for the specified card.
Included	An included resource is one that was originally a used resource. Some computers are not totally "standard" and currently it is not possible to create a resource scanning utility that is perfect for all computers. If you know that a system resource is available even though it was previously used, use this button to include it and make it available for PC card configurations.
Used	A used resource is one that is unavailable and will not be used to configure a newly inserted PC cards.
Excluded	An excluded resource is one that was originally free. Some computers are not totally "standard" and currently it is not possible to create a resource scanning utility that is perfect for all computers. If you know that a system resource is unavailable even though it was previously free, use this button to exclude it and make it unavailable for PC card configurations.
Start-End	This is a from-two value for the I/O range in hexadecimal.
Owner	This column displays the current users of I/O memory ranges. "SYSTEM" indicates that a non-PC card related component is using the resource. The name of the PC card next to the range indicates that the card is using the resource.

Canceling All Settings

If you decide to return all settings to their original default values, choose Revert to Defaults from the Options menu. CardSoft displays the following warning message:



Choose Yes if you are sure this is what you want to do.

CardSoft changes all settings, including beeps back to their defaults.

Customizing CONFIG.SYS

When *CardSoft* is installed, DEVICEHIGH lines are added to your CONFIG.SYS file to install the various drivers needed to support the different types of PC cards that are available. However, depending on the types of PC cards you use, you may not need to install all of these drivers in your system's memory. You can prevent these unneeded drivers from being installed by removing their devicehigh lines from CONFIG.SYS, or by adding a "rem" statement to the beginning of the devicehigh line.

Required DEVICEHIGH Lines

The following DEVICEHIGH lines must always be included in CONFIG.SYS in order for *CardSoft* to function properly. You should never delete or "rem" out these lines.

```
devicehigh=c:\CardSoft\*
devicehigh=c:\CardSoft\cs.sys
device=c:\CardSoft\csalloc.exe
<additional devicehigh lines needed to support specific card types must be inserted here>
devicehigh=c:\CardSoft\cardid.exe
install=c:\CardSoft\cs_apm.exe
```

*The socket services driver that appears here depends on the PCMCIA controller your system uses.

This minimum configuration enables you to use modem, fax/modem, network, and other types of I/O or communications cards without installing any additional *CardSoft* drivers.

DEVICEHIGH Lines Required for ATA Card Support

In order to support ATA drive cards and ATA Flash Disk cards, your CONFIG.SYS file must contain the following additional lines:

```
devicehigh=c:\CardSoft\atadv.exe
```

If you want to "slave" the ATA driver to MTDDRV, you need the following CONFIG.SYS lines. When you slave the ATA driver to MTDDRV, it allows you to share drive letters between SRAM, Flash Memory, and ATA cards. The value you use with the /S switch depends on the number of PCMCIA slots in your system. This example is for a system with one PCMCIA slot (/S:1).

```
devicehigh=c:\CardSoft\atadv.exe /S:1
devicehigh=c:\CardSoft\mtddrv.exe
```

DEVICEHIGH Lines Required for SRAM Card Support

To support SRAM cards, your CONFIG.SYS file must contain the following additional lines:

```
devicehigh=c:\CardSoft\mtsram.exe
devicehigh=c:\CardSoft\mtddrv.exe
```

DEVICEHIGH Lines Required for Flash Memory Card Support

To support Flash Memory cards, your CONFIG.SYS file must contain the following additional lines:

```
devicehigh=c:\CardSoft\mtaa.exe
devicehigh=c:\CardSoft\mtab.exe
devicehigh=c:\CardSoft\mti1.exe
devicehigh=c:\CardSoft\mti2p.exe
devicehigh=c:\CardSoft\mtddrv.exe
```

Note You will also need the optional MS FFS or FTL driver to get Flash cards to operate properly.

```
devicehigh=c:\CardSoft\ssmsflsh.sys (for MS FFS support)
```

or

```
devicehigh=c:\CardSoft\FTL.exe (for FTL support)
```

If You Have DOS 6.0 ...

If you have MS-DOS 6.0 or greater installed on your system, you can set up your CONFIG.SYS file to display a menu that will enable you to choose the driver configuration you need during a session. For example, if you only need to load network card support, you can set up a section in your CONFIG.SYS file that loads only those *CardSoft* drivers that are needed to support network cards. If you want to use an ATA card at a later time, you can have CONFIG.SYS load only those drivers that are needed to support ATA cards. By doing so, you can use your system's memory more efficiently, since you won't be loading unneeded drivers into memory when you start your system.

Following is a sample CONFIG.SYS file that displays a menu enabling you to load drivers to support all types of cards, or specific drivers to support a particular type of PC card. If you need more information on how to set up a menu in your CONFIG.SYS file, refer to your DOS manual.

Note *This example is for a system using a Cirrus PCMCIA controller and CardSoft with optional SystemSoft FTL support. Your CONFIG.SYS file may differ depending on your PCMCIA controller and whether or not you received CardSoft with optional FTL (flash) support.*

Do not include the comment statements listed in the right column; they are provided only to assist you in understanding the various sections of this example.

Example

```
device=c:\dos\setver.exe           ; This group of lines defines a standard
device=c:\dos\himem.sys           ; DOS startup. These lines contain the
files = 40                         ; commands that are normally run from
buffers = 40                       ; your CONFIG.SYS file. These lines
dos=high                           ; may differ based on your system's
shell=c:\dos\command.com /p /e:1024 ; configuration
stacks=9,256
```

```
[menu]                             ; This group of lines defines the menu
menuitem=normal, No CardSoft        ; that will be displayed when you start
menuitem=cs_all, Complete CardSoft ; your system.
menuitem=cs_io, Basic CardSoft installation
menuitem=cs_ata, CardSoft w/ATA Support
menuitem=cs_flash, CardSoft w/Flash Support
menuitem=cs_sram, CardSoft w/SRAM Support
menucolor=2, 7
```

```
[normal]                            ; No entries are needed in this section.
```

```
[cs_all]
devicehigh=c:\CardSoft\<socket services driver> ; This group of lines loads complete
devicehigh=c:\CardSoft\cs.sys                ; CardSoft support, enabling you to use
device=c:\CardSoft\csalloc.exe               ; all types of PC cards, including Flash
devicehigh=c:\CardSoft\atadriv.exe /s:2     ; memory cards.
devicehigh=c:\CardSoft\mtaa.exe
devicehigh=c:\CardSoft\mtab.exe
devicehigh=c:\CardSoft\mti1.exe
devicehigh=c:\CardSoft\mti2p.exe
devicehigh=c:\CardSoft\mtsram.exe
devicehigh=c:\CardSoft\mtddrv.exe
devicehigh=c:\CardSoft\ssmsflsh.sys
devicehigh=c:\CardSoft\cardid.exe
install=c:\CardSoft\cs_apm.exe
```

Note *CS.APM (Advanced Power Management) only works in systems that provide the appropriate power management BIOS. Power.exe is also required.*

```
[cs_io]
devicehigh=c:\CardSoft<socket services driver> ; This group of lines loads CardSoft with
devicehigh=c:\CardSoft\cs.sys ; I/O card support only (i.e., network,
device=c:\CardSoft\csalloc.exe ; fax/modem, modem, and other types of
devicehigh=c:\CardSoft\cardid.exe ; communications cards)
install=c:\CardSoft\cs_apm.exe

[cs_ata]
devicehigh=c:\CardSoft<socket services driver> ; This group of lines loads CardSoft with
devicehigh=c:\CardSoft\cs.sys ; ATA card and I/O card support only.
device=c:\CardSoft\csalloc.exe
devicehigh=c:\CardSoft\atadriv.exe
devicehigh=c:\CardSoft\cardid.exe
install=c:\CardSoft\cs_apm.exe

[cs_sram]
devicehigh=c:\CardSoft<socket services driver> ; This group of lines loads CardSoft with
devicehigh=c:\CardSoft\cs.sys ; SRAM card and I/O card support only.
device=c:\CardSoft\csalloc.exe c:\CardSoft\csalloc.ini
devicehigh=c:\CardSoft\mtsram.exe
devicehigh=c:\CardSoft\mtddrv.exe
devicehigh=c:\CardSoft\cardid.exe
install=c:\CardSoft\cs_apm.exe

[cs_flash]
devicehigh=c:\CardSoft<socket services driver> ; This group of lines loads CardSoft with
devicehigh=c:\CardSoft\cs.sys ; Flash Memory card and I/O card
device=c:\CardSoft\csalloc.exe ; support only.
devicehigh=c:\CardSoft\mtaa.exe
devicehigh=c:\CardSoft\mtab.exe
devicehigh=c:\CardSoft\mti1.exe
devicehigh=c:\CardSoft\mti2p.exe
devicehigh=c:\CardSoft\mtddrv.exe
devicehigh=c:\CardSoft\FTL.exe
devicehigh=c:\CardSoft\cardid.exe
install=c:\CardSoft\cs_apm.exe
```

Appendix A

CardID

CardID is a Card Services client that provides file-driven detection and configuration services for the different types of PCMCIA cards (fax/modems, ATA cards, network cards, communications, etc.). *CardID* utilizes an initialization file, CARDID.INI, that stores configuration information about these cards.

The *CardID* Configuration file contains two sections. The "[Common]" section defines the program parameters common to all inserted cards. The "[Libraries]" section contains references to card libraries of Card Recognition and Configuration records, which drive the process of card recognition and configuration.

The [Common] Section

The [Common] section consists of lines of the form

parameter=value

where the format of the value depends on the parameter. The following table lists all currently-defined parameters, their defaults, and their value types.

Note *The CARDID.INI file you receive may not list all of the parameters shown in this table. The parameters in your CARDID.INI file depend on your hardware platform and the PCMCIA support you are providing.*

CARDID.INI Parameters

Parameter/Default	Value Type	Description
<i>Entries for AT-style modem cards</i>		
COMPort1=3F8	Hexadecimal	COMPort1 I/O address
COMIRQ1=4	Integer, 1-4	COMPort1 Interrupt setting
COMPort2=2F8	Hexadecimal	COMPort2 I/O address
COMIRQ2=3	Integer, 1-4	COMPort2 Interrupt setting
COMPort3=3E8	Hexadecimal	COMPort3 I/O address
COMIRQ3=4	Integer, 1-4	COMPort3 Interrupt setting
COMPort4=2E8	Hexadecimal	COMPort4 I/O address
COMIRQ4=3	Integer, 1-4	COMPort4 Interrupt setting
FirstModem=1	Integer, 1-4	First Modem inserted
SecondModem=2	Integer, 1-4	Second Modem inserted
ThirdModem=3	Integer, 1-4	Third Modem inserted
FourthModem=4	Integer, 1-4	Fourth Modem inserted
COMBySocket=No	Yes/No	Assigns a COM port to a socket*
COMBDAAbsolute=Yes	Yes/No	BIOS data area*
Audio=Yes	Yes/No	Modem sounds are audible
RingIndicate=No	Yes/No	Responds to calls when in suspend mode

* See expanded description on page A-3.

Continued..

CARDID.INI Parameters Continued...

Parameter/Default	Value Type	Description
<i>Entries for LAN Cards</i>		
LAN1Port=300	Hexadecimal	Network Card1 BaseI/O Address
LAN1IRQ=5	Integer, 2-15	Network Card1 Interrupt setting
LAN1MemoryBase1=D8	Hexadecimal	Network Card1 first Memory Address
LAN1MemoryBase2=DC	Hexadecimal	Network Card1 second Memory Address
LAN2Port=310	Hexadecimal	Network Card2 BaseI/O Address
LAN2IRQ=10	Integer 2-15	Network Card2 Interrupt setting
LAN2MemoryBase1=0	Hexadecimal	Network Card2 first Memory Address
LAN2MemoryBase2=0	Hexadecimal	Network Card2 second Memory Address
FirstLAN=1	Integer, 1-2	First Network Card inserted
SecondLAN=2	Integer, 1-2	Second Network Card inserted
LANBySocket=No	Yes/No	Assigns a LAN card to a socket*
FastTokenRing=Yes	Yes/No	TR Transfer rate of 16bps
<i>Entries for "Other" Cards</i>		
Other1Port=300	Hexadecimal	Card1 Base I/O Address
Other1IRQ=5	Integer, 2-15	Card1 Interrupt setting
Other1MemoryBase1=D8	Hexadecimal	Card1 first Memory Address
Other1MemoryBase2=DC	Hexadecimal	Card1 second Memory Address
Other2Port=320	Hexadecimal	Card2 Base I/O Address
Other2IRQ=5	Integer, 2-15	Card2 Interrupt setting
Other2MemoryBase1=0	Hexadecimal	Card2 first Memory Address
Other2MemoryBase2=0	Hexadecimal	Card2 second Memory Address
FirstOther=1	Integer, 1-2	First "other" card inserted
SecondOther=2	Integer, 1-2	Second "other" card inserted
<i>Entries for ATA Cards</i>		
ATATryLinear=Yes	Yes/No	Linear Addressing attempt
ATATryMemory=No	Yes/No	Memory Addressing attempt
ATATrySecondary=Yes	Yes/No	Secondary Addressing attempt
ATATryPrimary=Yes	Yes/No	Primary Addressing attempt
ATAFirstConf=0	0=Linear	Addressing Order - 1st try
ATASecondConf=1	1=Memory	Addressing Order - 2nd try
ATAThirdConf=2	2=Secondary ATA Address	Addressing Order - 3rd try
ATAFourthConf=3	3=Primary ATA Address	Addressing Order - 4th try
ATASocketOffset=0	Integer, 0-63	ATA Socket Offset value
<i>Miscellaneous Card Configuration Parameters</i>		
DetectNonCompliant=No	Yes/No	Detect Non-compliant card
AllowIOOffsets=No	Yes/No	Allow Offsets
<i>Miscellaneous Entries</i>		
BeepinDOS=Yes	Yes/No	Emit system beeps in DOS
BeepinWindows=Yes	Yes/No	Emit system beeps in Windows
BeepOnConfigFailure=Yes	Yes/No	Emit system beep if configuration conflict
BeepOnUnrecognizedCard=No	Yes/No	Emit system beep if card not recognized
ContinueOnConfigFailure=No	Yes/No	Continue on config failure
<i>CardID Parameters</i>		
UseEMS=No	Yes/No	Use expanded memory

* See expanded description on page A-3.

COMBySocket -

When COMBySocket = No, (the default), the first modem card inserted is configured according to the setting of "FirstModem", the second card inserted according to "SecondModem", etc.

When COMBySocket = Yes, a modem card inserted in socket 0 is configured according to the setting of "FirstModem", the card inserted in socket 1 according to "SecondModem", etc. Note that , if COMBySocket = Yes, it is not possible to configure a modem card in a socket whose number is higher than 3.

COMBDAAbsolute -

If COMBDAAbsolute = No (the default), the I/O base address of a newly configured modem card is entered into the first available space in the BIOS data area.

If COMBDAAbsolute = Yes, the I/O base address of a card configures as COM Port "n" is entered into space (n-1) in the BIOS Data Area. The port which an I/O base address represents depends on the settings of COMPortn and COMIRQn.

The algorithm for COMBDAAbsolute = No corresponds to what the BIOS does when scanning for COM ports during POST.

The algorithm for COMBDAAbsolute = Yes corresponds, roughly, to the standard COM port behavior under Windows.

LANBySocket -

When LANBySocket = No, (the default), the first network card inserted is configured according to the setting of "FirstLAN", the second card inserted according to "SecondLAN", etc.

When LANBySocket = Yes, a network card inserted in socket 0 is configured according to the setting of "FirstLAN" and the card inserted in socket 1 according to "SecondLAN." Note that , if LANBySocket = Yes, it is not possible to configure a network card in a socket whose number is higher than 1.

The [Libraries Section]

The [Libraries] Section contains lines of the form

CardLib=*pathname*

where *pathname* is interpreted relative to the directory containing the initialization file if it is not a full pathname. The named file (i.e., CARDID.CLB) contains binary data records, each of which is a Comment Record, a Card Services Resource record, or a Card Recognition and Configuration Record. These record types are defined in the following sections.

Comment Record

This record allows comments (i.e., any desired text or binary data) to be inserted into a library.

Comment Record

Offset	Bytes	Description	Value
00	2	Record ID #	00E5h
02	2	Record Length	Bytes in record, including the header
04	n	Comment	Arbitrary binary data, extending to the end of the record, as defined by the Record Length.

Card Services Resource Record

This record describes the resources to be allocated to Card Services. At most, there should be one Card Services Resource Record. It is processed when *CardID* is initialized. Since resources are supposed to be allocated by CSALLOC, this record should not be necessary. It has been retained only to provide a workaround if, for some reason, CSALLOC is not functioning properly. This record consists of a header and a variable number of resource definition fields.

Record Header: Card Services Resource Record

Offset	Bytes	Description	Value
00	2	Record ID #	01E5h
02	2	Record Length	Bytes in record, including the header

Field # 00h = Memory Windows Available

Offset	Bytes	Description	Value
00	1	Field ID #	00h
01	2	Mem Win 1 Size	Size (in paragraphs) of window
03	2	Mem Win 1 Start	First paragraph of window
05	1	Mem Win 1 Flags	Bit 5 = 1 for shared
06	2	Mem Win 2 Size	Size (in paragraphs) of window
08	2	Mem Win 2 Start	First paragraph of window
0A	1	Mem Win 2 Flags	Bit 5 = 1 for shared
"	"	"	"
nn	2	End of List flag	Size = 0 marks end of list

Field # 01h = I/O Windows Available

Offset	Bytes	Description	Value
00	1	Field ID #	01h
01	1	I/O Win 1 Size	Size in bytes of first I/O window
02	2	I/O Win 1 Start	Start of I/O window 1
04	1	I/O Win 1 Flags	Bit 0 = 1 for shared
05	1	I/O Win 2 Size	Size in bytes of second I/O window
06	2	I/O Win 2 Start	Start of I/O window 2
08	1	I/O Win 2 Flags	Bit 0 = 1 for shared
"	"	"	"
nn	1	End of List Flag	Size = 0 marks end of list

Field # 02h = IRQ Available

Offset	Bytes	Description	Value
00	1	Field ID #	02h
01	2	IRQ Bit Mask	Bits 15:0 = IRQ 15-0 mask If bit = 0, IRQ is not available If bit = 1, IRQ is available
03	2	Share Bit Mask	Bits 15:0 = IRQ sharing mask If bit = 0, IRQ cannot be shared If bit = 1, IRQ can be shared
05	2	Sharing Type Bit Mask	Bits 15:0 = IRQ sharing type mask If bit = 0, IRQ must be time-multiplexed If bit = 1, IRQ can be dynamically shared <i>Note: This mask is present if some bit in the Share Bit Mask is set. Each bit in this mask is significant only if the corresponding bit in the Share Bit Mask is set.</i>

Card Recognition and Configuration Record

A Card Recognition and Configuration record consists of a variable number of recognition fields, followed by a variable number of configuration fields, which should be specified in order of field ID. If the criteria specified in the recognition fields are satisfied, the card should be configured in accordance with the configuration fields. There should be one Card Recognition and Configuration record for each type of card to be supported. The "type" in the previous sentence is a loose term; it can mean "a particular model of a particular card from a particular manufacturer", or something as loose as "any modem".

When a card is inserted, *CardID* reads Card Recognition and Configuration records in order, and uses the first one whose recognition algorithm succeeds on the card.

The Card Recognition and Configuration record begins with a record header:

Record Header: Card Recognition and Configuration Record

Offset	Bytes	Description	Value
00	2	Record ID #	03E5h
02	2	Record Length	Bytes in record, including the header
04	2	System Mask	Bitmask of systems to which this record applies. (Bit 0 = DOS, Bit 1 = Windows)
06	2	Socket Mask	Bitmask of logical slots to which this record applies. (Bit 0 = Slot 0, Bit 1 = Slot 1, etc.).
08	n	Card ID String	ASCIIZ String identifying the card, for use by programs that display and edit this file.
nn	n	Card Type String	ASCIIZ String identifying the type of card. If this string is null, the card type string is generated from the card type information specified in a Card Type field, or read from the CIS.

The Socket Mask field allows different sockets to be treated differently. For example, a modem in Slot 0 might always be configured as COM2, while a modem in Slot 1 might always be configured as COM3. This field may be set to FFFF to treat all slots identically.

Card Recognition Fields

Field # 00h = Recognition Algorithm

Offset	Bytes	Description	Value
00	1	Field ID #	00h
01	1	Recognition Algorithm	0 = Use CISTPL_VERS_1 1 = CIS literal comparison 2 = Generic type recognition 3 = Custom code

If the value is 0 (use CISTPL_VERS_1), then 1 to 4 fields follow, containing values to search for in some subset of the 4 fields of the CISTPL_VERS_1 tuple (manufacturer name, product name, and two "additional product information" fields). These fields have IDs 1, 2, 3, and 4, respectively. This algorithm presupposes a card with a CIS that is sufficiently correct that Card Services can return the CISTPL_VERS_1 tuple data to *CardID*. All fields must match for the card to be recognized using this algorithm.

If the recognition algorithm is 1 (CIS literal comparison), then a variable number of CIS comparison fields follow, each of which has ID 5. See the entry for field #05 for the format and use of these fields. At least one CIS comparison field must be present, and all CIS comparisons must succeed for the card to be recognized using this algorithm.

If the recognition algorithm is 2 (Generic type recognition), then a card type field (ID 06) must follow. *CardID* uses its internal recognition algorithm for cards of this type. If *CardID* has no internal recognition algorithm for this type, the card is not recognized. The card types recognized by *CardID* are modems (type 2) and ATA cards (type 4).

If the recognition algorithm is 3 (Custom code), then a custom recognition code record (ID 07) must follow.

Field # 01h = Manufacturer Name

Offset	Bytes	Description	Value
00	1	Field ID #	01h
01	n	Manufacturer's Name	ASCII string (null terminated)

Field # 02h = Product Name

Offset	Bytes	Description	Value
00	1	Field ID #	02h
01	n	Product Name	ASCII string (null terminated)

Field # 03h = Additional Product Information (1)

Offset	Bytes	Description	Value
00	1	Field ID #	03h
01	n	Additional Product Information String	ASCII string (null terminated)

This field describes the contents of the third string in the CISTPL_VERS_1 tuple (if one is present).

Field # 04h = Additional Product Information (2)

Offset	Bytes	Description	Value
00	1	Field ID #	04h
01	n	Additional Product Information String	ASCII string (null terminated)

This field describes the contents of the fourth string in the CISTPL_VERS_1 tuple (if one is present).

Field # 05h = CIS Comparison

Offset	Bytes	Description	Value
00	1	Field ID #	05h
01	1	Memory Type	0 = Common memory 1 = Attribute memory
02	4	Offset	Offset from start of memory (includes even and odd bytes, in both attribute and common memory)
06	1	Length	Number of bytes of comparison data
07	n	Comparison Data	Bytes to compare

This field specifies an area of the card data to compare against the specified data. If attribute memory is being compared, the Offset should be even, and odd bytes in attribute memory space are skipped. All bytes of the comparison data are used, in both common and attribute memory.

Field # 06h = Card Type

Offset	Bytes	Description	Value
00	1	Field ID #	06h
01	1	Card Type	Card Type

Field # 07h = Custom Recognition Code

		Description	Value
		Field ID #	07h
		Custom Code Length	# bytes in custom code
		Custom Recognition Code	Variable length custom recognition code

The Custom Recognition code is position-independent code called by *CardID* when this record is read during card insertion processing. The interface is as follows:

Entry point:

At the start of the record data (like a COM file). At entry, DS = CS = SS (tiny model).

Inputs:

AX Pointer to beginning of custom code (call is via CALL AX)
 BX Pointer to a common_info structure
 CX System type (0 = DOS, 1 = Windows)
 DX Client Handle for *CardID*
 SI Pointer to a SOCKET_DATA structure for this slot

Outputs:

AX Return value (0 = success, 2 = failure)

Registers that must be preserved:

BP, DI, DS

The custom recognition code may make Card Services calls using the Client Handle passed in DX, but must free any Card Services resources it allocates before returning.

Card Configuration Fields

These fields describe the configuration to be performed when a card is recognized by the Card Recognition algorithm of this Card Recognition and Configuration Record.

Field # 10h = Configuration Algorithm

Offset	Bytes	Description	Value
00	1	Field ID #	10h
01	1	Configuration Algorithm	0 = Do not configure 1 = Use internal configuration algorithm 2 = Configure as specified 3 = Reserved 4 = Custom configuration 5 = Internal + Custom 6 = Specific + Custom
02	1	Status Interrupt Mask	Mask to pass to Card Services for enabling status interrupt sources. Should be FF for cards that drive all status lines properly. The bits are as described in the Socket Services specification for SetSocket.

If the configuration algorithm is 0 (do not configure), the card should be rejected and no configuration should be attempted. This is like the "ignore" function of the current *CardID*.

If the configuration algorithm is 1 (use internal configuration algorithm), a Card Type field (ID 6) follows, to define what sort of card it is. This field may be omitted if a Card Type field was specified in the recognition section, or if a CISTPL_FUNCID tuple (possibly modified by one or more CISTPL_FUNCE tuples) is present in the CIS. *CardID* chooses I/O windows and IRQ assignments from its internal configuration code for this card type. If it has no internal configuration code for this card type, the card is rejected.

If the configuration algorithm is 2 (configure as specified), an optional CIS requirements field (ID 11), an optional CIS default field (ID 12), and an optional CIS override field (ID 13) follow. These fields, combined with the actual CIS, if present, specify the card configuration, as follows:

1. The CIS default field, if present, is read into a default CIS information structure.
2. Unspecified fields are set to their backup values, as defined by the PCMCIA PC Card Specification.
3. The CIS is scanned, with initial default values for all fields set as if the values in the default CIS information structure had been specified in a CISTPL_CFTABLE_ENTRY tuple with the "default" bit set.
4. If the CIS requirements field is present, the first CISTPL_CFTABLE_ENTRY tuple matching the requirements is selected, and read into a working CIS information structure. Otherwise, the first CISTPL_CFTABLE_ENTRY tuple found is read. If no tuple is read, the working CIS information structure is copied from the default CIS information structure.
5. If a CIS override field is present, the override fields are copied into the working CIS information structure.
6. Finally, the working CIS information structure is analyzed to determine if it contains enough information to perform a configuration. If so, the card is configured. If not, the card is not configured.

If the configuration algorithm is 4 (custom config), a custom configuration field (ID 14) follows. The code in that field is used to configure the card.

If the configuration algorithm is 5 (internal + custom), the internal configuration algorithm is run. If it succeeds, the custom configuration code is then called.

If the configuration algorithm is 6 (specific + custom) the specific configuration algorithm is run. If it succeeds, the custom configuration code is then called.

Field # 11h = CIS requirements

Offset	Bytes	Description	Value
00	1	Field ID #	11h
01	2	Field Mask	Bitmask defining which CIS fields are significant.
03	n	CE	CISTPL_CFTABLE_ENTRY image, with required values in the fields marked as significant.

The Field Mask determines which fields in CISTPL_CONF and CISTPL_CFTABLE_ENTRY images are valid. The format of a Field Mask is identical for Fields 11, 12, and 13, and is as follows:

- Bit 0: Configuration registers base address valid (CONF)
- Bit 1: Configuration register presence mask valid (CONF)
- Bit 2: Configuration index valid (CE)
- Bit 3: Vcc power description valid (CE)
- Bit 4: Vcc and Vpp power descriptions valid (CE)
- Bit 5: I/O space description valid (CE)
- Bit 6: IRQ description valid (CE)
- Bit 7: Memory space description valid (CE)
- Bit 8: Miscellaneous-fields structure valid (CE)

In this field, and in the other fields that use tuple images, the image is stored complete, with tuple code and tuple length fields, to allow the same code in *CardID* that processes CIS tuples to process tuple images. The tuple images conform to the *PCMCIA PC Card Standards*, with the following extra semantics:

- If the I/O base of an I/O range has Bit 31 set, that range is considered to require a 16-bit data path, regardless of the setting of the Bus8 and Bus16 bits of TPCE_IO.
- If the length of a memory range has Bit 31 set, that range is to be mapped to attribute memory space, rather than common memory space. It is presumed that attribute memory space requires an 8-bit data path, and common memory space requires a 16-bit data path.

Field # 12h = CIS defaults

Offset	Bytes	Description	Value
00	1	Field ID #	12h
01	2	Field Mask	Bitmask defining which CIS fields are significant.
03	n	CONF	CISTPL_CONFIG image. Present only if at least one CONF bit is set in the bitmask.
mm	n	CE	CISTPL_CFTABLE_ENTRY image, with special default values in the fields marked as significant. Present only if at least one CE bit is set in the bitmask.

Field # 13h = CIS override

Offset	Bytes	Description	Value
00	1	Field ID #	13h
01	2	Field Mask	Bitmask defining which CIS fields are significant.
03	n	CONF	CISTPL_CONFIG image. Present only if the configuration base bit is set in the bitmask.
mm	n	CE	CISTPL_CFTABLE_ENTRY image, with override values in the fields marked as significant. Present only if at least one CE bit is set in the bitmask.

Field # 14h = Custom Configuration Code

Offset	Bytes	Description	Value
00	1	Field ID #	14h
01	2	Custom Code Length	# bytes in custom code
03	n	Custom Code	Variable length custom configuration code

The Custom Configuration code is position-independent code called by *CardID* to configure a card. The interface is as follows:

Entry point:

At the start of the record data (like a COM file). At entry, CS = DS = SS (tiny model). The call is a far call.

Inputs: BX Pointer to a common_info structure
 CX System type (0 = DOS, 1 = Windows)
 DS:AX Pointer to beginning of custom code
 DX Client Handle for *CardID*
 SI Pointer to a SOCKET_DATA structure for this slot

Outputs:

AX Return value (0 = success, 4 = failure)

Registers that must be preserved:

BP, DI, DS

The custom configuration code may make Card Services calls using the Client Handle passed in DX. If it allocates Card Services resources, it should store the corresponding request structures in the appropriate fields of the SOCKET_DATA structure, so they can be freed when the card is removed.

Field # 15h = Custom Removal Code

Offset	Bytes	Description	Value
00	1	Field ID #	15h
01	2	Custom Code Length	# bytes in custom code
03	n	Custom Removal Code	Variable length custom removal code

Normally, on card removal, *CardID* releases the configuration. In addition, if any special code was performed on insertion (e.g., setting up the BIOS Data Area for a modem), it must be undone on removal. This field defines custom removal code for cards that need it. It may be present even if the card was not configured with custom code. The Custom Removal code is position-independent code called by *CardID* when a card is removed that was recognized and configured according to this record. The interface is as follows:

Entry point:

At the start of the record data (like a COM file). At entry, DS = CS = SS (tiny model).

Inputs:

AX Pointer to beginning of custom code (call is via CALL AX)
 BX Pointer to a common_info structure
 CX System type (0 = DOS, 1 = Windows)
 DX Client Handle for *CardID*
 SI Pointer to a SOCKET_DATA structure for this slot

Outputs:

None.

Registers that must be preserved:

BP, DI, DS

The custom removal code may make Card Services calls using the Client Handle passed in DX, but must free any Card Services resources it allocates before returning. Custom removal code need not free Card Services resources described in the SOCKET_DATA structure; those will be freed by *CardID* after the custom removal code returns.

Field # 16h = I/O Redirection

Offset	Bytes	Description	Value
00	1	Field ID #	16h
01	2	Field Mask	0020h
03	n	CE	CISTPL_CFTABLE_ENTRY image, with card base values in the I/O base fields. No other fields are significant.

Appendix B

Card Services Vendor-Specific API Definitions

This Appendix contains the Card Services Application Programming Interface (API) definitions for SystemSoft vendor-specific functions.

Subcomponent IDs

The following table lists the Card Services API subcomponent IDs.

Subcomponent ID	Subcomponent
0	Card Services Special API
1	Power Management
2	Ring Resume
3	Reserved
4	Reserved
5	CardBoot
6	BeepControl
7	GetWindowInfo

RegisterSpecialClient

CardServices (VendorSpecific, null/ClientHandle, ClientEntry, ArgLength, ArgPointer)

Card Services argument binding:

Inputs

[AL] = Function (VendorSpecific - 34h)
 [DX] = Handle (Null)
 [DI]:[SI] = ClientEntry, [DI] is 16-bit segment, [SI] is 16-bit offset
 [CX] = ArgLength
 [ES]:[BX] = ArgPointer, [ES] is 16-bit segment, [BX] is 16-bit offset

Outputs

[AX] = Status
 [DX] = ClientHandle
 [CF] = Reset to zero when request was successful, set to one when request failed

All entry registers are preserved unless the same register is used for a return value.

All calls and returns are FAR.

All pointers are SEGMENT16:OFFSET16.

Card Services is invoked by INT 1Ah with [AH] = AFh.

This function is used to make vendor-specific Card Services requests for registering with Card Services clients that have special requirements.

Argument Packet Definition

Offset	Field	Size	Type	Value	Description
0	InfoLen	2	I/O	N	Length of returned info in packet
2	SubComponentID	1	I	0	Special Card Services API Subcomponent ID
3	FunctionID	1	I	10h	RegisterClient Function ID
4	Argument Length	2	I	N	Length of RegisterClient Argument Packet
6	Argument Packet	*N	I	N	RegisterClient Argument Packet
*N+6	Special Attributes	4	I	N	Bit-mapped (defined in the following table)
*N+10	Callback Offset	2	I	N	Vendor-specific Client Callback Entry
*N+12	Callback Segment	2	I	N	

*N = Size of RegisterClient Argument Packet

The Special Attributes field is bit-mapped. It identifies the special handling the client requires:

Bit	Function
31:2	Reserved (reset to zero)
1	Issue vendor-specific callback for card configurations
0	Issue card removal events for powered-off sockets

Function ID is used to indicate Ring Resume Disable (1) or Enable (2).

SpecialGetStatus

CardServices (VendorSpecific, null, null, ArgLength, ArgPointer)

Card Services argument binding:

Inputs

[AL] = Function (VendorSpecific - 34h)
 [DX] = Handle (Null)
 [DI]:[SI] = ClientEntry, [DI] is 16-bit segment, [SI] is 16-bit offset
 [CX] = ArgLength
 [ES]:[BX] = ArgPointer, [ES] is 16-bit segment, [BX] is 16-bit offset

Outputs

[AX] = Status
 [CF] = Reset to zero when request was successful, set to one when request failed

All entry registers are preserved unless the same register is used for a return value.

All calls and returns are FAR.

All pointers are SEGMENT16:OFFSET16.

Card Services is invoked by INT 1Ah with [AH] = AFh.

This function is used to make vendor-specific Card Services requests for retrieving socket status information from Card Services that contains special information.

Argument Packet Definition

Offset	Field	Size	Type	Value	Description
0	InfoLen	2	I	N	Length of returned info in packet
2	SubComponentID	1	I	0	Special Card Services API Subcomponent ID
3	FunctionID	1	I	0Ch	GetStatus Function ID
4	Argument Length	2	I	N	Length of GetStatus Argument Packet
6	Argument Packet	*N	I/O	N	GetStatus Argument Packet
*N+6	Configuration State	2	O	N	Bit-mapped (defined in the following table)

*N = Size of GetStatus Argument Packet

The Configuration State field is bit-mapped. It identifies the socket configuration state:

Bit	Function
15:5	Reserved (reset to zero)
4	Memory Handle assigned to Socket (via OpenMemory)
3	MTD is registered on Socket
2	Socket is in I/O mode
1	Socket is configured
0	Socket has memory window(s)

Suspend

CardServices (VendorSpecific, null, null, ArgLength, ArgPointer)

Card Services argument binding:

Inputs

[AL] = Function (VendorSpecific - 34h)
 [DX] = Handle (Null)
 [DI]:[SI] = Pointer, [DI] is 16-bit segment, [SI] is 16-bit offset (Null)
 [CX] = ArgLength (4)
 [ES]:[BX] = ArgPointer, [ES] is 16-bit segment, [BX] is 16-bit offset

Outputs

[AX] = Status
 [CF] = Reset to zero when request was successful, set to one when request failed

All entry registers are preserved unless the same register is used for a return value.

All calls and returns are FAR.

All pointers are SEGMENT16:OFFSET16.

Card Services is invoked by INT 1Ah with [AH] = AFh.

This function is used to make vendor specific Card Services requests for Suspend/Resume (if FunctionID is 1 or 2) or Socket Suspend/Resume (if FunctionID is 3 or 4).

Argument Packet Definition

Offset	Field	Size	Type	Value	Description
0	InfoLen	2	I/O	4	Length of returned info in packet
2	SubComponentID	1	I	1	Power Management SubComponent ID
3	FunctionID	1	I	N	Suspend Function ID

Function ID is used to indicate Controller Suspend (1), Controller Resume (2), Socket Suspend (3), or Socket Resume (4). The Socket field is required only when FunctionID is set to 3 or 4.

Resume

CardServices (VendorSpecific, null, null, ArgLength, ArgPointer)

Card Services argument binding:

Inputs

[AL] = Function (VendorSpecific - 34h)
 [DX] = Handle (Null)
 [DI]:[SI] = Pointer, [DI] is 16-bit segment, [SI] is 16-bit offset (Null)
 [CX] = ArgLength (4)
 [ES]:[BX] = ArgPointer, [ES] is 16-bit segment, [BX] is 16-bit offset

Outputs

[AX] = Status
 [CF] = Reset to zero when request was successful, set to one when request failed

All entry registers are preserved unless the same register is used for a return value.

All calls and returns are FAR.

All pointers are SEGMENT16:OFFSET16.

Card Services is invoked by INT 1Ah with [AH] = AFh.

This function is used to make vendor specific Card Services requests for Suspend/Resume (if FunctionID is 1 or 2) or Socket Suspend/Resume (if FunctionID is 3 or 4).

Argument Packet Definition

Offset	Field	Size	Type	Value	Description
0	InfoLen	2	I/O	4	Length of returned info in packet
2	SubComponentID	1	I	1	Power Management SubComponent ID
3	FunctionID	1	I	N	Resume Function ID

Function ID is used to indicate Controller Suspend (1), Controller Resume (2), Socket Suspend (3), or Socket Resume (4). The Socket field is required only when FunctionID is set to 3 or 4.

Socket Suspend

CardServices (VendorSpecific, null, null, ArgLength, ArgPointer)

Card Services argument binding:

Inputs

[AL] = Function (VendorSpecific - 34h)
 [DX] = Handle (Null)
 [DI]:[SI] = Pointer, [DI] is 16-bit segment, [SI] is 16-bit offset (Null)
 [CX] = ArgLength (6)
 [ES]:[BX] = ArgPointer, [ES] is 16-bit segment, [BX] is 16-bit offset

Outputs

[AX] = Status
 [CF] = Reset to zero when request was successful, set to one when request failed

All entry registers are preserved unless the same register is used for a return value.

All calls and returns are FAR.

All pointers are SEGMENT16:OFFSET16.

Card Services is invoked by INT 1Ah with [AH] = AFh.

This function is used to make vendor specific Card Services requests for Suspend/Resume (if FunctionID is 1 or 2) or Socket Suspend/Resume (if FunctionID is 3 or 4).

Argument Packet Definition

Offset	Field	Size	Type	Value	Description
0	InfoLen	2	I/O	6	Length of returned info in packet
2	SubComponentID	1	I	1	Power Management SubComponent ID
3	FunctionID	1	I	N	Suspend Function ID
4	Socket	2	I	N	Logical Socket Number

Function ID is used to indicate Controller Suspend (1), Controller Resume (2), Socket Suspend (3), or Socket Resume (4). The Socket field is required only when FunctionID is set to 3 or 4.

Socket Resume

CardServices (VendorSpecific, null, null, ArgLength, ArgPointer)

Card Services argument binding:

Inputs

[AL] = Function (VendorSpecific - 34h)
 [DX] = Handle (Null)
 [DI]:[SI] = Pointer, [DI] is 16-bit segment, [SI] is 16-bit offset (Null)
 [CX] = ArgLength (6)
 [ES]:[BX] = ArgPointer, [ES] is 16-bit segment, [BX] is 16-bit offset

Outputs

[AX] = Status
 [CF] = Reset to zero when request was successful, set to one when request failed

All entry registers are preserved unless the same register is used for a return value.

All calls and returns are FAR.

All pointers are SEGMENT16:OFFSET16.

Card Services is invoked by INT 1Ah with [AH] = AFh.

This function is used to make vendor specific Card Services requests for Suspend/Resume (if FunctionID is 1 or 2) or Socket Suspend/Resume (if FunctionID is 3 or 4).

Argument Packet Definition

Offset	Field	Size	Type	Value	Description
0	InfoLen	2	I/O	6	Length of returned info in packet
2	SubComponentID	1	I	1	Power Management SubComponent ID
3	FunctionID	1	I	N	Resume Function ID
4	Socket	2	I	N	Logical Socket Number

Function ID is used to indicate Controller Suspend (1), Controller Resume (2), Socket Suspend (3), or Socket Resume (4). The Socket field is required only when FunctionID is set to 3 or 4.

Ring Resume

CardServices (VendorSpecific, null, null, ArgLength, ArgPointer)

Card Services argument binding:

Inputs

[AL] = Function (VendorSpecific - 34h)
 [DX] = Handle (Null)
 [DI]:[SI] = Pointer, [DI] is 16-bit segment, [SI] is 16-bit offset (Null)
 [CX] = ArgLength (6)
 [ES]:[BX] = ArgPointer, [ES] is 16-bit segment, [BX] is 16-bit offset

Outputs

[AX] = Status
 [CF] = Reset to zero when request was successful, set to one when request failed

All entry registers are preserved unless the same register is used for a return value.

All calls and returns are FAR.

All pointers are SEGMENT16:OFFSET16.

Card Services is invoked by INT 1Ah with [AH] = AFh.

This function is used to make vendor-specific Card Services requests for Ring Resume.

Argument Packet Definition

Offset	Field	Size	Type	Value	Description
0	InfoLen	2	I/O	6	Length of returned info in packet
2	SubComponentID	1	I	2	Ring Resume SubComponent ID
3	FunctionID	1	I	N	Ring Resume Function ID
4	Socket	2	I	N	Logical Socket

Function ID is used to indicate Ring Resume Disable (1) or Enable (2).

BeepControl

CardServices (VendorSpecific, null, null, ArgLength, ArgPointer)

Card Services argument binding:

Inputs

[AL] = Function (VendorSpecific - 34h)
 [DX] = Handle (Null)
 [DI]:[SI] = Pointer, [DI] is 16-bit segment, [SI] is 16-bit offset (Null)
 [CX] = ArgLength (8)
 [ES]:[BX] = ArgPointer, [ES] is 16-bit segment, [BX] is 16-bit offset

Outputs

[AX] = Status
 [CF] = Reset to zero when request was successful, set to one when request failed

All entry registers are preserved unless the same register is used for a return value.

All calls and returns are FAR.

All pointers are SEGMENT16:OFFSET16.

Card Services is invoked by INT 1Ah with [AH] = AFh.

This function is used to make vendor specific Card Services requests for BeepControl version 2.10.

Argument Packet Definition

Offset	Field	Size	Type	Value	Description
0	InfoLen	2	I/O	8	Length of returned info in packet
2	SubComponentID	1	I	6	Beep Control Subcomponent ID
3	FunctionID	1	I	0	N/A
4	Request	2	I/O	N	Set or retrieve beep settings
6	Setting	2	I/O	N	Beep settings

The Request field is defined as follows:

0=Request to retrieve current beep settings
 1=Request to set beep settings

The Setting field is a bitmap, defined as follows:

Bit	Function
15:8	Reserved for future use
7	Beep in DOS (overrides bits 3,2,1)
6	Beep in Windows (overrides bits 3,2,1)
5:4	Reserved for future use
3	Beep on card insertion
2	Beep on card removal
1	Beep on configuration conflict
0	Beep on artificial events

GetWindowInfo

CardServices (VendorSpecific, null, null, ArgLength, ArgPointer)

Card Services argument binding:

Inputs

[AL] = Function (VendorSpecific - 34h)
 [DX] = Handle (Null)
 [DI]:[SI] = Pointer, [DI] is 16-bit segment, [SI] is 16-bit offset (Null)
 [CX] = ArgLength
 [ES]:[BX] = ArgPointer, [ES] is 16-bit segment, [BX] is 16-bit offset

Outputs

[AX] = Status
 [CF] = Reset to zero when request was successful, set to one when request failed

All entry registers are preserved unless the same register is used for a return value.

All calls and returns are FAR.

All pointers are SEGMENT16:OFFSET16.

Card Services is invoked by INT 1Ah with [AH] = AFh.

This function is used to retrieve information about memory windows in use on a specific slot.

Argument Packet Definition

Offset	Field	Size	Type	Value	Description
0	InfoLen	2	I/O	N	Length of returned info in packet
2	SubComponentID	1	I	7	Get Information Subcomponent ID
3	FunctionID	1	I	0	Get Memory Window Info Function ID
4	Socket	2	I	N	Logical Socket
6	WindowEntryCnt	2	I/O	N	Number of entries in the window array
8	WindowEntryArray	N	O	N	Array of window entries

The **WindowEntryCnt** field specifies the number of entries in the window array as input, and specifies the number of possible entries returned as output (which may exceed the input).

The **WindowEntry** has the following structure:

Offset	Field	Size	Type	Value	Description
0	Base	4	O	N	System Base Address
4	Size	4	O	N	Memory Window Size
8	Card Offset	4	O	N	Card Offset Address
12	Attributes	2	O	N	(See RequestWindow Attributes)
14	AccessSpeed	1	O	N	Window Speed Field
15	Page	1	O	N	Page Number

Appendix C

Socket Services Vendor-Specific API Definitions

This Appendix contains the Socket Services Application Programming Interface (API) definitions for Vendor-specific functions.

Note that not all Socket Services support all listed subfunctions.

Subcomponent IDs

The following table lists the supported subcomponent IDs and subfunctions.

Subcomponent ID	Subfunction
2	Set/reset DMA mode
100	Get CardBoot information
8000*	Reinitialize adapter (after resume)
8001*	Set/reset Ring Indicate resume mode

* SystemSoft standard subfunctions.

Vendor Specific Functions

Inputs:

[AL] Adapter Number
All other registers are vendor-specific
[SI] Subfunction ID

Outputs:

[CF] Error
[AL] Adapter Number
[AH] Error code
01 = BAD_ADAPTER

Set/Reset DMA Mode

Inputs:

[BL] Slot Number
[BH] DMA Control Bits
[7:6] DMA Mode
00 Non-DMA mode
01 DMA (DREQ = -INPACK)
10 DMA (DREQ = WP/-IOIS16)
11 DMA (DREQ = SPKR/LED)
[5:3] Reserved (set to zero)
[2:0] 000 Use DRQ0
001 Use DRQ1
010 Use DRQ2
011 Use DRQ3
100 No DMA request steering

[SI] 2h

Re-Initialize Adapter After Resume

Inputs:

[AH] AEh (Vendor specific code)
[AL] Adapter number
[SI] 8000h

Set/Reset Ring Indicate Mode

Inputs:

[AH] AEh (Vendor specific code)
[AL] Adapter number
[SI] 8001h
[BL] Slot number
[DL] Control flags
 [7:1] Reserved
 [0] Set/reset Ring Indicate mode
 0 = Reset
 1 = Set

Outputs:

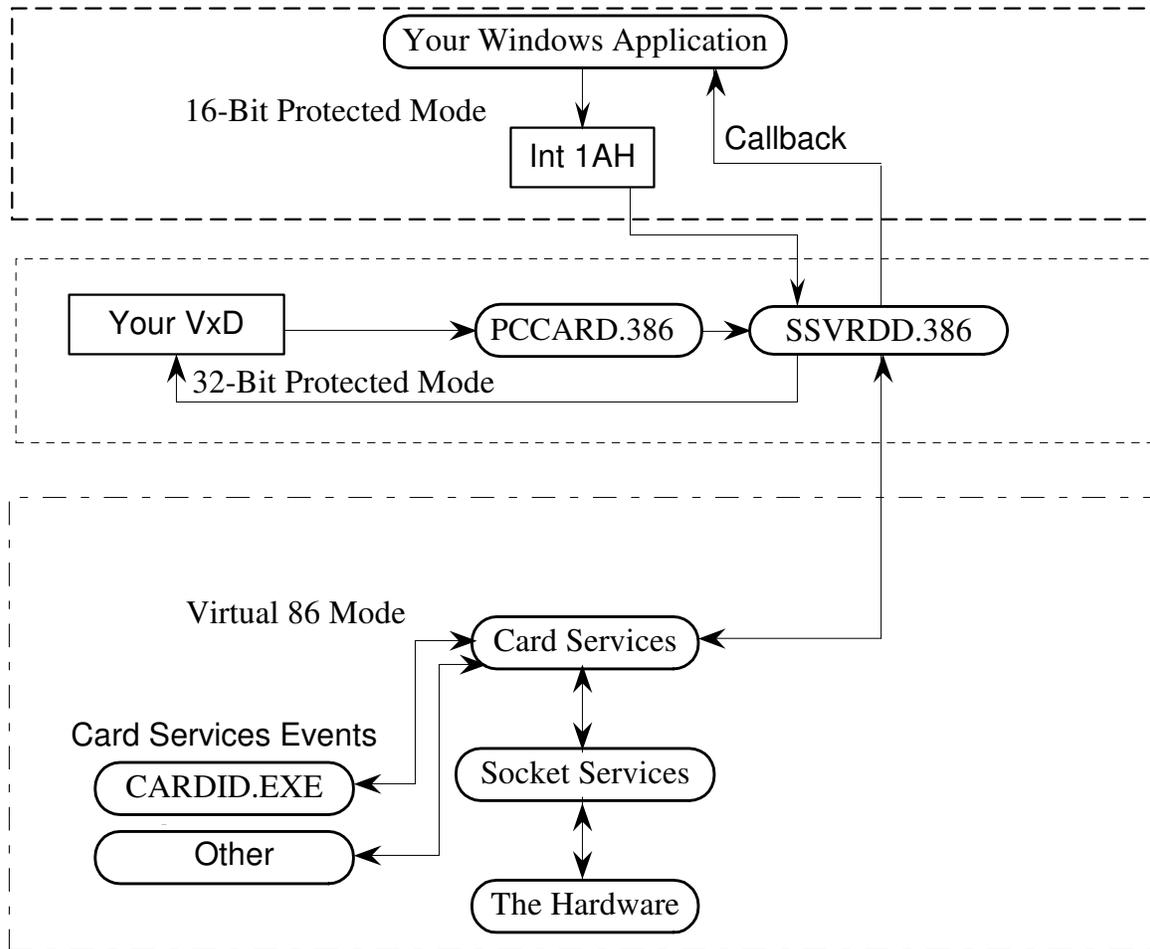
[CF] Status
 0 = Success
 1 = Error
[AH] RETCODE
 0B = BAD_SOCKET
 15 = BAD_FUNCTION

Appendix D

SystemSoft *CardSoft* API for Windows

This appendix describes the interface to Card Services that is provided for Windows 3.1 applications via the SystemSoft API for Windows. These libraries and drivers enable Windows 3.1 applications to register as Card Services clients, thereby allowing these applications to receive callbacks when PC cards are inserted into or removed from a slot.

The following illustrates the hierarchy of the libraries and drivers of the SystemSoft API for Windows.



Component Descriptions

Socket Services provides the interface to the socket adapter hardware. Card Services provides the software interface that follows the PC Card Standard. `CARDID.EXE` is the Card Client driver that provides detection and configuration services for certain PCMCIA cards. *CardID* provides much of the information that is displayed by the Windows application. All of these components comprise the SystemSoft *CardSoft* PCMCIA software suite. These drivers are loaded at boot-up time via lines in `CONFIG.SYS`. As a result, they run in Virtual 86 mode under Windows.

The most difficult problem posed by having a Windows application communicate with Card Services and Socket Services is the crossing back and forth over the line between protected mode and Virtual 86 mode. The virtual driver `SSVRDD.386` (which stands for SystemSoft Virtual Removable Device Driver) and `PCCARD.386` provide the mapping of parameters and calls across this line. The work accomplished by `SSVRDD.386` is transparent to the Windows application. The application never makes direct calls to this virtual driver.

`PCCARD.386` (see previous figure) provides the PCMCIA 32-bit interface as defined in the `PCCARD` Standard.

Card Services Callout

This first routine, `cscall.c`, is a generic card services callout using `PCCARD` and `int 1Ah`.

The Card Services API consists of one function with the following prototype:

```
DWORD CallCS ( BYTE bFunc,
              WORD wHandle,
              void far* lpPointer,
              WORD wSize,
              void far* lpArgBuf)
```

This function is really the entire API call to Card Services. All calls that adhere to the Card Service specification are called through this function.

```

/*****
* Function: CallCS
*
* Example call to Card Services from Windows Application. See
* PCCARD documentation for more information on parameters.
*
* Parameters
* bFunc - Card Services function to call
* wHandle - Card Services handle or NULL
* lpPointer - Card Services pointer
* wSize - Size of argument buffer
* lpArgBuf - Pointer to argument buffer
*
\*****/

DWORD CallCS ( BYTE bFunc,
              WORD wHandle,
              void far* lpPointer,
              WORD wSize,
              void far* lpArgBuf)
```

```
{
    DWORD  dwRet;
    union _REGS regs;
    struct _SREGS sregs;

    // save the current DS for WinInt86()
    _segread(&sregs);

    // Load calling parameters into registers for
    // Card Services call
    regs.h.ah = 0xAF;
    regs.h.al = bFunc;
    regs.x.dx = wHandle;
    regs.x.di = FP_SEG(lpPointer);
    regs.x.si = FP_OFF(lpPointer);
    regs.x.cx = wSize;
    sregs.es = FP_SEG(lpArgBuf);
    regs.x.bx = FP_OFF(lpArgBuf);

    // Execute Int 1a. This invokes Card Services
    _int86x(0x1A, &regs, &sregs);

    // returns are in registers
    dwRet = regs.x.dx;
    dwRet <<= 16;
    dwRet |= regs.x.ax;

    return dwRet;
}
```

Card Services Callback

This next routine, callback, is both an assembly wrapper and a 'C' function, to handle the callback from card services to the windows application.

WARNING: Windows calls (except for PostMessage) are unsupported in the context of a callback from 32 bit protected mode. Make sure you use post message to service callbacks quickly

```
; Assembly language subroutines shell for Card Services
; Callback to C function. Parameters are in registers
; as documented in PCCARD reference
;
; Copyright SystemSoft Corporation
; May 1995

.LIST

        TITLE  callback

        .MODEL  medium, c
        .STACK
        .DATA
        .CODE
;
@proto_0        TYPEDEF        PROTO FAR C :WORD,:WORD,:WORD,:WORD
                EXTERN  CSCallBackEvent: PROTO @proto_0
;
;void FAR CSCallBack(void)
;
PUBLIC CSCALLBACK
CSCALLBACK    PROC FAR

start:
        pushf
        push  es
        push  ds        ;save caller's data segment
        push  bp        ;save caller's frame pointer
        mov   bp,sp     ;this routine's frame pointer

        push  di
        push  si

        ; push parameters for callback in 'C'

        push  bx        ;4th parameter, buffer address
        push  es        ;3rd parameter, buffer segment address
        push  cx        ;socket is the second parameter of CardInsrtRemv
        xor   ah,ah
        push  ax        ;call back event code is the first parameter of CardInsrtRemv

        ; call 'C' routine

        call  CSCallBackEvent
;
exit:
```

```
    pop    si
    pop    di
    mov    sp,bp        ;remove parameter spaces on the stack
    pop    bp          ;restore caller's frame pointer
    pop    ds          ;restore caller's data segment
    pop    es
    popf
;
; the following are for all events:
    clc                ;successful
    mov    ax,0        ;return code, no error
    ret
```

```
CSCALLBACK ENDP
```

```
END
```

```
; ASM MODULE Ends here
```

```

/*****
* CSCallBack - C function to handle callbacks
* Parameters:
*   wEvent   - Event Code
*   wSocket  - Socket on which event occurred
*   wBufSeg  - Segment for callback buffer
*   wBufOff  - Offset for callback buffer
*
* Note: Some rarely used callbacks use more parameters. Refer
*       to PCMCIA documentation for details and usage
*
* WARNING: Windows calls (except for PostMessage) are unsupported
*          in the context of a callback from 32 bit protected mode
*          Make sure you use post message to service callbacks quickly
\*****/
void FAR CSCallBackEvent (WORD wEvent,
                          WORD wSocket,
                          WORD wBufSeg,
                          WORD wBufAddr)           //callback event buffer address
{
    LPPKTCBBUF lpBuf;

    switch ( wEvent )           //which call back event is this?
    {
        case CSCB_CARD_REMOVAL:
        case CSCB_CARD_INSERTION:
            PostMessage(hwndWiz,WM_USR_MSGUPDATE,wSocket,(LPARAM)wEvent);
            break;
        case CSCB_CLIENT_INFO:
            FP_SEG(lpBuf) = wBufSeg;
            FP_OFF(lpBuf) = wBufAddr;
            lpBuf->wInfoLen = sizeof(pktCBClientInfo);
            lpBuf->wRevision = wRevision;
            lpBuf->wCSLevel = 0x210;           //conform to PCMCIA 2.1
            lpBuf->wRevDate = wDate; //the date in DOS date format
            lpBuf->wNameOffset = 20;
            lpBuf->wNameLen = 11;
            lpBuf->wVendorOffset = 31;
            lpBuf->wVendorLen = 22;
            lstrcpy((LPSTR)lpBuf->cName,(LPSTR)"TEST CLIENT");
            lstrcpy((LPSTR)lpBuf->cVendor,(LPSTR)"SystemSoft Corporation");

            break;
        default:
            break;
    }
}
//end of CSCallBackEvent()

```

Appendix E

Extensions to the Card Services Specification Provided by *CardID*

The *CardID* driver provides extensions to the Card Services specification. These extensions are accessed through the GetClientInfo subfunction (03h) by specifying an Info Subfunction in the range 80h-FFh. The extensions are listed here by their Info Subfunction value.

GetClientInfo Subfunctions

The following table lists the Info subfunctions for GetClientInfo.

Info Subfunction	Function
80h	Returns a bit-mapped field showing which GetClientInfo extensions are supported by the specified client.
81h	Returns strings identifying which card resides in the specified logical slot.
82h	Reserved by SystemSoft
83h	Reserved by SystemSoft
84h	Reserved by SystemSoft
85h	Reserved by SystemSoft
86h	Fills a buffer with values that describe the current <i>CardID</i> parameters.
87h	Sets the current <i>CardID</i> parameters with values contained in a buffer.
88h	Reserved
89h	Reserved
8Ah	Reserved
8Bh	Reserved
8Ch	Reserved
8Dh	Reserved
8Eh	Reserved
8Fh	Returns error information for a specific slot.

GetGCIExtensionsInfo (Info Subfunction 80h)

This function returns a bit-mapped field showing which GetClientInfo extensions the specified client supports.

On Entry:

Attributes: 80XXh
Client Info Packet:

Offset	Size	Contents
6	2	Reset to zero

On Exit:
Client Info Packet:

Offset	Size	Contents
6	2	Supported functions

The Supported Functions field is bit-mapped and is defined as follows:

Bit	Description if set
15:6	Reserved (reset to zero)
5	Has system diagram bitmap
4	Supports configured range for memory cards
3	Supports size for memory cards
2	Has Device Independent Bitmap
1	Supports Device Identification
0	Supports Card Identification

GetCardID (Info Subfunction 81h)

This function returns strings identifying which card resides in the specified logical socket. Zero length strings are returned when the information is not available.

On Entry:

Attributes: 81XXh
Client Info Packet:

Offset	Size	Contents
6	2	Logical socket number

On Exit:

Client Info Packet:

Offset	Size	Description
6	2	Logical socket number
8	2	Offset in packet to Card Type string buffer
10	2	Length of Card Type string
12	2	Offset in packet to Manufacturer string buffer
14	2	Length of Manufacturer string
16	2	Offset in packet Product string buffer
18	2	Length of Product string
N	N	Card Type ASCIIZ string
N	N	Manufacturer ASCIIZ string
N	N	Product ASCIIZ string

Legal card type strings are defined as follows (case sensitive):

Card Type String	Description
"Modem"	Modems, FAX/Modems
"Ethernet"	Ethernet Adapters
"ATA Disk"	ATA Disks
"Flash Card"	Flash Memory Cards
"SRAM Card"	SRAM Memory Cards

Reserved Functions

Subfunctions 82h, 84h, and 85h are reserved by SystemSoft.

GetCardIDArgList (Subfunction 86h)

This function fills a buffer with values that describe the current *CardID* parameters. These parameters match those that are contained in the `CARDID.INI` file. The buffer filled has the following "C" structures:

```
#define MAX_COM_PORTS      4
#define MAX_9800_PORTS    2
#define MAX_LAN_PORTS     2
#define MAX_OTHER_PORTS   2
#define MAX_PORTS         4      /* the maximum of the above */

typedef struct {
    UINT16 addr;
    BYTE   irq;
    BYTE   ordinal;      /* 0 is first, 1 is second, etc. */
} io_info;

typedef struct {
    BYTE   flags;
    io_info io_info;
    port[MAX_COM_PORTS];
} com_def;

typedef struct {
    BYTE   flags;
    io_info io_info;
    port[MAX_9800_PORTS];
} com9800_def;

/*
 * flags for the com_def.flags byte
 */
#define SEL_SOCKET          (1<<0)      /* also used for lan_def */
#define BDA_ABSOLUTE       (1<<1)
#define ROLLOVER           (1<<2)
#define USE_AUDIO          (1<<3)
#define USE_RI             (1<<4)
#define USE_RI_PASS        (1<<5)
#define NO_COM_ADJUST      (1<<6)

typedef struct {
    BYTE   flags;
    io_info io_info;
    port[MAX_LAN_PORTS];
    mem_base[MAX_LAN_PORTS][2];
} lan_def;

typedef struct {
    BYTE   flags;
    io_info io_info;
    port[MAX_OTHER_PORTS];
    mem_base[MAX_OTHER_PORTS][2];
} other_def;

typedef struct {
    UINT16 flags;      /* keep the size constant */
    BYTE   conf_order[4];
    BYTE   socket_offset;
} ata_def;

/*
 * definitions of the ATA flags
 */
#define ATA_TRY_PRIMARY      (1<<0)
#define ATA_TRY_SECONDARY   (1<<1)
#define ATA_TRY_LINEAR      (1<<2)
#define ATA_TRY_MEMORY      (1<<3)
#define ATA_FORCE_CONFIG    (1<<4)

/*
 * definitions of ATA conf_order bytes, in standard order
 */
```

```
#define ATA_NOTHING          0
#define ATA_LINEAR          1
#define ATA_MEMORY          2
#define ATA_PRIMARY         3
#define ATA_SECONDARY       4
#define ATA_MAX             5

/*
 * here is where we keep any other CIS arguments that have
 * no home in the COM, LAN, or ATA definitions.
 */
typedef struct {
    UINT16      flags;
} misc_def;

/* definitions for the misc_def "flags" field */
#define F_MISC_TRSPD        (1<<1) /* 1 = fast Token Ring speed */

typedef struct {
    com_def      com_info;
    lan_def      lan_info;
    ata_def      ata_info;
    misc_def     misc_info;
    com9800_def  com9800_info;
    other_def    other_info;
} common_info;

#endif /* _COMINFO_INC */

typedef struct {
    WORD  max_length;
    WORD  info_length;
    WORD  attr;
    char  cmd_path[80];
    COMMON_INFO  dos_info;
    COMMON_INFO  win_info;
} GETSETARGS;

On Entry:

Attributes: 86XXh
Client Info Packet: Empty buffer as described above.

On exit:
Client Info Packet: Filled with current CardID parameters.
```

SetCardIDArgList (Subfunction 87h)

This function sets the current *CardID* parameters with values contained in a buffer. These parameters match those that are contained in the `CARDID.INI` file. The buffer is identical to that described in Subfunction 86h.

On Entry:

Attributes: 87XXh

Client Info Packet: Buffer filled with CardID parameters.

On exit:

Nothing.

GetLastError (Subfunction 8Fh)

This function returns error information for a specific socket. Specifically, it returns the function number of the last function to get a non-zero return code, and the error code that it received.

On Entry:

Attributes: 8FXXh

Client Info Packet:

Offset	Size	Contents
6	2	Logical socket number

On exit:

Offset	Size	Description
6	2	Logical socket number
8	2	Last error code received
10	2	Function number of the call that returned the error

Appendix F

Card Services Vendor-Specific Callbacks

Currently, Card Services implements only one vendor-specific callback, Socket Configuration (83h).

Socket Configuration (83h)

Callback (Socket Configuration, Socket, ConfigState, null, null, 0, ClientData)

The Socket Configuration event indicates that Card Insertion events for a PCMCIA slot have been completed, and notifies the client as to how the slot has been configured.

Cause	Card Insertion.
Pre-Client	Card Services issues a Card Insertion event to all clients that have used the RegisterClient function to indicate their interest in insertion events (see the description of Card Insertion in the <i>PC Card Standard Card Services Specification (Section 3.5.3)</i>).
Client	All clients that have used the RegisterSpecialClient vendor-specific function to indicate their interest in socket configuration events are notified by Card Services via the vendor-specific client callback entry. If further information about the configuration is needed, the client should call the GetConfigurationInfo function if the slot is configured, and should call the GetWindowInfo vendor-specific function if the slot has memory windows.
Post_Client	Card Services does not perform any post-client processing.

The ConfigState is bit-mapped. It identifies the socket configuration state:

Bit	Function
15:5	Reserved
4	Memory Handle assigned to slot (via Open Memory)
3	MTD is registered on slot
2	Slot is in I/O mode
1	Slot is configured
0	Slot has memory window(s)

SYSTEMSOFT®

CardSoft
Technical Reference
Part Number: 2-020-10