Linux PCMCIA HOWTO

David Hinds, dahinds@users.sourceforge.net.

v2.118, 06 December 2003

Abstract

This document describes how to install and use PCMCIA Card Servicesfor Linux, and answers some frequently asked questions. The latestversion of this document can always be found athttp://pcmcia-cs.sourceforge.net.

Table of Contents

General information and hardware requirementsIntroductionCopyright notice and

disclaimerWhat is the latest version, and where can I get it?What systems are supported?What cards are supported?When will my favorite (unsupported) card become supported?Mailing lists and other information sourcesCompilation and

installationPrerequisites and kernel setupKernel PCMCIA supportInstallationStartup

optionsSystem resource settingsNotes about specific Linux distributionsResolving installation and configuration problemsBase PCMCIA kernel modules do not loadSome client driver modules do not loadISA interrupt scan failuresIO port scan failuresMemory probe failuresFailure to detect card insertions and removalsInterrupt delivery problemsSystem resource starvationResource conflict only with two cards insertedDevice configuration does not completeUsage and featuresTools for configuring and monitoring PCMCIA

devices Overview of the PCMCIA configuration scripts PCMCIA network adapters PCMCIA serial and modem devices PCMCIA parallel port devices PCMCIA SCSI adapters PCMCIA memory cards PCMCIA ATA/IDE card drives Multifunction cards Advanced topics Resource allocation for PCMCIA devices PCI interrupt configuration problems and solutions How can I have separate device setups for home and work? Booting from a PCMCIA device Dealing with unsupported cards Configuring unrecognized cards Adding support for an

NE2000-compatible ethernet cardPCMCIA floppy interface cardsDebugging tips and programming informationSubmitting useful problem reportsInterpreting kernel trap

reportsLow level PCMCIA debugging aids/proc/bus/pccardWriting Card Services drivers for new cardsGuidelines for PCMCIA client driver authorsGuidelines for Linux distribution maintainers

General information and hardware requirements

Introduction

Card Services for Linux is a complete PCMCIA or ``PC Card" support package. It includes a set of loadablekernel modules that implement a version of the Card Servicesapplications program interface, a set of client drivers for specificcards, and a card manager daemon that can respond to card insertionand removal events, loading and unloading drivers on demand. Itsupports ``hot swapping" of most card types, so cards can be safelyinserted and ejected at any time.

This software is a work in progress. It contains bugs, and should beused with caution. I'll do my best to fix problems that are reported to me, but if you don't tell me, I may never know. If you use thiscode, I hope you will send me your experiences, good or bad!

If you have any suggestions for how this document could be improved, please let me know (dahinds@users.sourceforge.net).

Copyright notice and disclaimer

Copyright (c) 1998-2002 David A. Hinds

This document may be reproduced or distributed in any form without myprior permission. Modified versions of this document, includingtranslations into other languages, may be freely distributed, provided that they are clearly identified as such, and this copyright isincluded intact.

This document may be included in commercial distributions without myprior consent. While it is not required, I would like to be informedof such usage. If you intend to incorporate this document in apublished work, please contact me to make sure you have the latestavailable version.

This document is provided ``AS IS", with no express or impliedwarranties. Use the information in this document at your own risk.

What is the latest version, and where can I get it?

The current major release of Card Services is version 3.2, and minorupdates or bug fixes are numbered 3.2.1, 3.2.2, and so on.

Source code for the latest version is available on the web athttp://pcmcia-cs.sourceforge.net, aspcmcia-cs-3.2.?.tar.gz. You may find more than one releasenumber here. It is up to you to decide which version is moreappropriate, but the CHANGES file will summarize the mostimportant differences.

Pre-compiled drivers are included with current releases of essentiallyall major Linux distributions, including Slackware, Debian, Red Hat,Caldera, and SuSE, among others. So generally there is no need to compile the drivers from scratch.

What systems are supported?

This package should run on almost Intel-based Linux-capable laptop. It also runs on some Alpha, PowerPC, ARM, and MIPS platforms. Mostcommon socket controllers are supported. Card docks for desktopsystems should work as long as they use a supported controller, andare plugged directly into the ISA or PCI bus, as opposed to SCSI-to-PCMCIA or IDE-to-PCMCIA adapters. The following controllers recognized by the supplied socket drivers:

- * Cirrus Logic (now Basis Communications) PD6710, PD6720, PD6722, PD6729, PD6730, PD6832
- * ENE Technology CB1211, CB1225, CB1410, CB1420

- * Intel i82365sl B, C, and DF steps, 82092AA
- * O2Micro OZ6729, OZ6730, OZ6812, OZ6832, OZ6833, OZ6836, OZ6860,OZ6922, OZ6933, OZ6912
- * Omega Micro 82C365G, 82C092G
- ^{*} Ricoh RF5C296, RF5C396, RL5C465, RL5C466, RL5C475, RL5C476, RL5C477, RL5C478
- * SMC 34C90
- * Texas Instruments PCI1031, PCI1130, PCI1131, PCI1210, PCI1211, PCI1220, PCI1221, PCI1225, PCI1250A, PCI1251A, PCI1251B, PCI1410, PCI1410A, PCI1420, PCI1450, PCI1451A, PCI1510, PCI1520, PCI1620, PCI4410, PCI4410A, PCI4450, PCI4451, PCI4510, PCI4520, PCI7410, PCI7510, PCI7610
- * Toshiba ToPIC95, ToPIC97, ToPIC100 (experimental, incomplete)
- * Vadem VG465, VG468, VG469
- * VLSI Technologies 82C146, VCF94365
- * VIA VT83C469
- ^{*} Databook DB86082, DB86082A, DB86084, DB86084A, DB86072, DB86082B

Other controllers that are register compatible with the Intel i82365slwill generally work, as well.

Due to the rapid pace of technological change for laptop hardware, newcontrollers appear frequently, and there may be delays between when anew model appears on the market, and when driver support becomesavailable.

Support for Toshiba's ToPIC bridges was hindered for a long time by alack of sufficiently detailed technical documentation. While somedatasheets have been available, a few idiosyncracies of the ToPICchips were not adequately explained. Toshiba has given some directtechnical help on some of these issues, and I think the major oneshave been resolved. However, with the introduction of kernel PCMCIAsupport in 2.4.* and later kernels, some new Toshiba bugs may havecropped up in the new socket driver code.

The Motorola 6AHC05GA controller used in some Hyundai laptops is notsupported. The custom host controller in the HP Omnibook 600 isalso unsupported.

What cards are supported?

The current release includes drivers for a variety of ethernet cards, a driver for modem and serial port cards, several SCSI adapterdrivers, a driver for ATA/IDE drive cards, and memory card driversthat should support most SRAM cards and some flash cards.

TheSUPPORTED.CARDS file included with each release of Card Serviceslists all cards that are known to work in at least one actual system.

The likelihood that a card not on the supported list will work dependson the type of card. Essentially all modems should work with the supplied driver. Some network cards may work if they are OEM versionsof supported cards. Other types of IO cards (frame buffers, soundcards, etc) will not work until someone writes the appropriatedrivers.

When will my favorite (unsupported) card become supported?

Unfortunately, they usually don't pay me to write device drivers, soif you would like to have a driver for your favorite card, you areprobably going to have to do at least some of the work. Ideally, I'dlike to work towards a model like the Linux kernel, where I would beresponsible mainly for the ``core'' driver code and other authorswould contribute and maintain client drivers for specific cards. TheSUPPORTED.CARDS file mentions some cards for which driver work iscurrently in progress. I will try to help where I can, but be warnedthat debugging kernel device drivers by email is not particularlyeffective.

Mailing lists and other information sources

The Linux PCMCIA information page is athttp://pcmcia-cs.sourceforge.net, and has bug tracking,support and feature requests, and a variety of PCMCIA related messageforums. Users can request email notification of new responses toparticular questions, or notification for all new messages in a givencategory. I hope that this will become a useful repository ofinformation, for questions that go beyond the scope of the HOWTO.

The Linux Laptop Page at http://www.linux-on-laptops.comhas links to a vast number of sites that have information aboutconfiguring specific types of laptops for Linux. There is also asearchable database of system configuration information, and pointersto a variety of laptop-related mailing lists.

Compilation and installation

Prerequisites and kernel setup

Before starting, you should think about whether you really need tocompile the PCMCIA package yourself. All common Linux distributionscome with pre-compiled driver packages. Generally, you only need toinstall the drivers from scratch if you need a new feature of thecurrent drivers, or if you've updated and/or reconfigured your kernelin a way that is incompatible with the drivers included with yourLinux distribution. While compiling the package is not technicallydifficult, it does require some general Linux familiarity.

The following things should be installed on your system before youbegin:

- * A 2.0, 2.2, 2.4, or 2.6 series kernel source tree.
- * An appropriate set of module utilities.
- * (Optional) the ``XForms'' X11 user interface toolkit.

You need to have a complete linux source tree for your kernel, notjust an up-to-date kernel image. The driver modules contain somereferences to kernel source files. While you may want to build a newkernel to remove unnecessary drivers, installing PCMCIA does not require you to do so.

Current ``stable" kernel sources and patches are available fromftp://ftp.kernel.org/pub/linux/kernel/v2.4. Currentmodule utilities can be found in the same locations.

In the Linux kernel source tree, the Documentation/Changesfile describes the versions of all sorts of other system componentsthat are required for that kernel release. You may want to checkthrough this and verify that your system is up to date, especially ifyou have updated your kernel. If you are using a development kernel, be sure that you are using the right combination of shared libraries and module tools.

On x86 based systems, if you plan to use 16-bit PC Card devices, youshould also enable CONFIG_ISA, for recent kernels. These cardsbehave much like ISA devices, and the PCMCIA drivers useCONFIG_ISA to judge whether a platform supports ISA businterrupts.

When configuring your kernel, if you plan on using a PCMCIA ethernetcard, you should turn on networking support but turn off the normalLinux network card drivers, including the ``pocket and portableadapters''. The PCMCIA network card drivers are all implemented asloadable modules. Any drivers compiled into your kernel will onlywaste space.

If you want to use SLIP, PPP, or PLIP, you do need to either configureyour kernel with these enabled, or use the loadable module versions of these drivers.

In order to use a PCMCIA token ring adapter, your kernel should beconfigured with ``Token Ring driver support" (CONFIG_TR)enabled, though you should leave CONFIG_IBMTR off.

If you want to use a PCMCIA IDE adapter, your kernel should beconfigured with CONFIG_BLK_DEV_IDE_PCMCIA

enabled, for 2.0.*kernels. Newer kernels do not require a special configurationsetting.

If you will be using a PCMCIA SCSI adapter, then enableCONFIG_SCSI when configuring your kernel. Also, enable any toplevel drivers (SCSI disk, tape, cdrom, generic) that you expect touse. All low-level drivers for particular host adapters should be disabled, as they will just take up space.

This package includes an X-based card status utility calledcardinfo. This utility is based on a freely distributed userinterface toolkit called the XForms Library. This library isavailable as a separate package with most Linux distributions. If youwould like to build cardinfo, you should install XForms and allthe normal X header files and libraries before configuring the PCMCIApackage. This tool is completely optional.

Kernel PCMCIA support

PCMCIA driver support is included in the 2.4 and later linux kerneltrees. While it shares most of the same code with the standalonePCMCIA driver package, there are some important differences. Thekernel PCMCIA support is also still evolving.

The kernel PCMCIA code has the same functionality as the driver sideof the pcmcia-cs package. It does not eliminate the need to install the pcmcia-cs package, since it requires the same user tools(cardmgr, cardctl, /etc/pcmcia/* files). The drivers in pcmcia-cs can still be built for 2.4 kernels, so you have a choice of using either the in-kernel PCMCIA drivers, or the drivers included in pcmcia-cs. With 2.5 and later kernels, the standalone drivers cannot be used.

To use the kernel PCMCIA drivers, configure the kernel withCONFIG_HOTPLUG, CONFIG_PCMCIA, and usuallyCONFIG_CARDBUS enabled. On x86 based systems, CONFIG_ISAshould also be enabled. The drivers can either be built into thekernel or built as modules. PCMCIA client driver options are listed in their regular driver categories; thus, PCMCIA network drivers are a submenu of network drivers, and PCMCIA serial drivers are in asubmenu of character drivers.

In the standalone pcmcia-cs drivers, the i82365 module supportsboth ISA-to-PCMCIA, PCI-to-PCMCIA, and PCI-to-CardBus bridges. TheCardBus socket driver in the 2.4 tree is the yenta_socket driver.It is selected by the CONFIG_CARDBUS option. In your PCMCIAstartup options, this driver should be specified in place of thei82365 driver. The kernel version of the i82365 driver,selected by CONFIG_I82365, only supports ISA-to-PCMCIA bridges.PCI-to-PCMCIA bridges that are not CardBus capable, like the CirrusPD6729, are not supported at all by the kernel PCMCIA drivers.

When compiling the standalone PCMCIA package, the Configure scriptdecides whether or not to build any kernel modules by looking at thevalue of the CONFIG_PCMCIA option in your kernel configuration. If CONFIG_PCMCIA is enabled, then by default, no drivercomponents are built. If CONFIG_PCMCIA is disabled, then all themodules will be built and installed. It is safe to compile the usertools (cardmgr, cardctl, etc) in a PCMCIA package whose version numberdiffers from the PCMCIA version number in the kernel source tree. Thekernel PCMCIA header files take precedence over the ones included inthe PCMCIA package, if CONFIG_PCMCIA is enabled.

Installation

Here is a synopsis of the installation process:

- * Unpack pcmcia-cs-3.2.?.tar.gz in /usr/src.
- * Run ``make config" in the new pcmcia-cs-3.2.? directory.
- * Run ``make all", then ``make install".
- * Customize the startup script and the option files in/etc/pcmcia for your site, if needed.

If you plan to install any contributed client drivers not included inthe core PCMCIA distribution, unpack each of them in the top-level directory of the PCMCIA source tree. Then follow the normal build instructions. The extra drivers will be compiled and installed automatically.

Running ``make config'' prompts for a few configuration options, and checks out your system to verify that it satisfies all prerequisites for installing PCMCIA support. In most cases, you'll beable to just accept all the default configuration options. Be sure tocarefully check the output of this command in case there are problems. The following options are available:

Linux kernel source directory?

This is the location of the source tree for the kernel you want to usewith PCMCIA. Often this is /usr/src/linux, but the defaultlocation depends on what Linux distribution you're using (or on whereyou've chosen to place your kernel source tree).

Build 'trusting' versions of card utilities?

Some of the support utilities (cardctl and cardinfo) can becompiled either in ``safe" or "trusting" forms. The "safe" formsprevent non-root users from modifying card configurations. The ``trusting" forms permit ordinary users to issue commands to suspendand resume cards, reset cards, and change the current configurationscheme. The default is to build the safe forms.

Include 32-bit (CardBus) card support?

This option must be selected if you wish to use 32-bit CardBus cards. It is not required for CardBus bridge support, if you only plan to use16-bit PC Cards.

Include PnP BIOS resource checking?

This builds additional code into the PCMCIA core module to communicate with a system's PnP BIOS to obtain resource information for built-in``motherboard" devices (serial and parallel ports, sound, etc), tohelp avoid resource conflicts. If enabled, some extra resource fileswill be created under /proc/bus/pccard, and the lspnpand setpnp tools can be used to view and manipulate PnP BIOSdevices. However, this setting causes problems on some laptops and isnot turned on by default.

Module install directory?

The directory that new kernel modules will be installed into.Normally this should be the subdirectory of /lib/modules thatmatches your kernel version.

How to set kernel-specific options?

There are a few kernel configuration options that affect the PCMCIAtools. The configuration script can deduce these from the runningkernel (the default and most common case). Alternatively, if you arecompiling for installation on another machine, it can read theconfiguration from a kernel source tree, or each option can be setinteractively.

The Configure script can also be executed non-interactively, forautomatic builds or to quickly reconfigure after a kernel update. Some additional less-frequently-used options can be only be set from the command line. Running ``Configure --help" lists allavailable options.

Running ``make all" followed by ``make install" will buildand then install the kernel modules and utility programs. Kernelmodules are installed under

/lib/modules/>version>/pcmcia.The cardmgr and cardctl programs are installed in/sbin. If cardinfo is built, it is installed in/usr/bin/X11.

Configuration files will be installed in the /etc/pcmciadirectory. If you are installing over an older version, your oldconfig scripts will be backed up before being replaced. The savedscripts will be given an *.O extension.

If you don't know what kind of host controller your system uses, youcan use the pcic_probe utility in the cardmgr/subdirectory to determine this. There are several major types: theDatabook TCIC-2 type and the Intel i82365SL-compatible type. With thekernel PCMCIA subsystem, Intel compatible controllers are furthersubdivided into ISA-bus 16-bit bridges, and PCI-based CardBus bridges.

In a few cases, the pcic_probe command will be unable to determineyour controller type automatically. If you have a Halikan NBD 486system, it has a TCIC-2 controller at an unusual location: you'll needto edit rc.pcmcia to load the tcic module, and also set thePCIC&lowbar:OPTS parameter to ``tcic_base=0x02c0".

On some old pre-PCI systems using Cirrus controllers, including the NEC Versa M, the BIOS puts the controller in a special suspended stateat system startup time. On these systems, the pcic_probe command willfail to find any known host controller. If this happens, editrc.pcmcia and set PCIC to i82365, and PCIC&lowbar:OPTS to``wakeup=1".

Startup options

The PCMCIA startup script recognizes several groups of startupoptions, set via environment variables. Multiple options should beseparated by spaces and enclosed in quotes. Placement of startupoptions depends on the Linux distribution used. They may be placeddirectly in the startup script, or they may be kept in a separateoption file. See the Notes about specific Linux distributions

PCMCIA

This variable specifies whether PCMCIA support should be started up,or not. If it is set to anything other than ``yes", then the startupscript will be disabled.

PCIC

This identifies the PC Card Interface Controller driver module. Thereare several options: ``tcic", ``i82365", and (for the kernel PCMCIAsubsystem) ``yenta_socket".

Virtually all current controllers are inthe ``i82365" group for the standalone drivers, and ``yenta_socket"for the kernel drivers. This is the only mandatory option setting.

PCIC_OPTS

This specifies options for the PCIC module. Some host controllershave optional features that may or may not be implemented in aparticular system. In some cases, it is impossible for the socketdriver to detect if these features are implemented. See the corresponding man page for a complete description of the available options.

CORE_OPTS

This specifies options for the pcmcia_core module, whichimplements the core PC Card driver services. See ``manpcmcia_core'' for more information.

CARDMGR_OPTS

This specifies options to be passed to the cardmgr daemon. See``man cardmgr" for more information.

SCHEME

If set, then the PC Card configuration scheme will be initialized tothis at driver startup time. See the Overview of the PCMCIA configuration scripts

The low level socket drivers, tcic and i82365, have variousbus timing parameters that may need to be adjusted for certain systemswith unusual bus clocking. Symptoms of timing problems can includecard recognition problems, lock-ups under heavy loads, high errorrates, or poor device performance. Only certain host bridges haveadjustable timing parameters: check the corresponding man page to seewhat options are available for your controller. Here is a briefsummary:

- * ISA-bus Cirrus controllers have numerous configurable timingparameters. The most important seems to be the cmd_time flag,which determines the length of PCMCIA bus cycles. Fast 486 systems(i.e., DX4-100) seem to often benefit from increasing this from 6 (thedefault) to 12 or 16.
- * The Cirrus PD6729 PCI controller has the fast_pci flag, whichshould be set if the PCI bus speed is greater than 25 MHz.
- * For Vadem VG-468 controllers, the async_clock flag changes therelative clocking of PCMCIA bus and host bus cycles. Setting thisflag adds extra wait states to some operations. However, I have yetto hear of a laptop that needs this.

- * The pcmcia_core module has the cis_speed parameter forchanging the memory speed used for accessing a card's Card InformationStructure (CIS). On some systems, increasing this parameter (i.e.,slowing down card accesses) may fix card recognition problems.
- * Another pcmcia_core parameter, io_speed, can be used to slowdown accesses to IO cards. It may help in certain cases with systemsthat have out-of-spec PCMCIA bus timing.
- * This is not a timing issue, but if you have more than one ISA-to-PCMCIAcontroller in your system or extra sockets in a laptop docking station, the i82365 module should be loaded with the extra_ socketsparameter set to 1. This should not be necessary for detection of PCI-to-PCMCIA or PCI-to-CardBus bridges.

Here are some timing settings for a few old systems:

- * On the ARM Pentium-90 or Midwest Micro Soundbook Plus, use``freq_bypass=1 cmd_time=8".
- * On a Compaq Presario 1220, try ``setup_time=1".
- * On a Midwest Micro Soundbook Elite, use ``cmd_time=12".
- * On a Gateway Liberty, try ``cmd_time=16".
- * On a Samsung SENS 810, use ``fast_pci=1".

Card readers for desktop systems

While almost all PCMCIA card readers and card docks work fine underLinux, some require special startup options because they do not behaveexactly like laptop PCMCIA bridges. PCI card readers, in particular,may handle interrupts differently. Some of the following parametersettings are only available for the i82365 module in thestandalone drivers; the kernel's yenta_socket driver is notconfigurable.

- * The Linksys ProConnect PCMRDWR and Antec DataChute ISA card readersare ``ISA Plug and Play'' devices. To use these, you must firstactivate them with the Linux isapnp tools. See the man pages forpnpdump and isapnp for more information.
- * For Chase CardPORT and Altec ISA card readers using the Cirrus PD6722ISA-to-PCMCIA bridge, the i82365 driver should be loaded with a``has_ring=0" parameter to prevent irq 15 conflicts.
- * For Elan P-series PCI card readers based on the Cirrus PD6729PCI-to-PCMCIA bridge chip, the i82365 driver requires a``irq_mode=1" parameter.
- * For the Sycard PCChost1200 host adapter, the i82365 driverrequires a ``p2cclk=1" parameter.
- * For the Alex Electronics PCICBI host adapter based on the TI 1221bridge, the i82365 driver requires ``p2cclk=1 irq_mode=0''as well as PCMCIA driver release 3.1.23 or later.
- * With SCM Microsystems SBP series PCI card readers (which are alsobeing distributed with Lucent WaveLAN IEEE cards), and for theSynchrotech PCM-CR-PC2IF and PCM-CR-PC2IR, it is necessary tospecify ``irq_mode=0" for the i82365 module, to force useof PCI interrupts.

- * For the ActionTec PC 750 card reader, and for the Antec Datachute PCIcard reader, the i82365 driver requires a ``irq_list=0"parameter, to indicate that ISA interrupts are unavailable.
- * The PLX Technologies PCI9052 (also sold as the Linksys WDT11) is not ageneral purpose PCMCIA card reader at all: it is a PCI interface cardfor use with certain wireless adapters, that makes them look likeordinary PCI devices. These devices are not supported.

System resource settings

Card Services should automatically avoid allocating IO ports and interrupts already in use by other standard devices. It will alsoattempt to detect conflicts with unknown devices, but this is notcompletely reliable. In some cases, you may need to explicitly exclude resources for a device in /etc/pcmcia/config.opts.

Here are some resource settings for specific laptop types. View thislist with suspicion: it may give useful hints for solving problems, but it is inevitably out of date and certainly contains mistakes.Corrections and additions are welcome.

- * On the AMS SoundPro, exclude irq 10.
- * On some AMS TravelPro 5300 models, use memory 0xc8000-0xcffff.
- * On the BMX 486DX2-66, exclude irq 5, irq 9.
- * On the Chicony NB5, use memory 0xda000-0xdffff.
- * On the Compaq Presario 900Z, exclude port 0x3b0-0x3bb.
- * On the Compaq Presario 1020, exclude port 0x2f8-0x2ff, irq 3, irq 5.
- * On the Compaq Presario 2120EA, exclude irq 10.
- * On the Dell Inspiron 7000, exclude irq 3, irq 5.
- * On the Dell Inspiron 8000, exclude port 0x800-0x8ff.
- * On the Fujitsu C series, exclude port 0x200-0x27f.
- * On the HP Omnibook 4000C, exclude port 0x300-0x30f.
- * On the HP Omnibook 4100, exclude port 0x220-0x22f.
- * On the IBM ThinkPad 380, and maybe the 385 and 600 series, excludeport 0x230-0x233, and irq 5.
- * On IBM ThinkPad 600 and 770 models with internal modems, exclude port0x2f8-0x2ff.
- * On the IBM ThinkPad 600E and 770Z, change the high memory window to0x6000000-0x60fffff.
- * On the Micron Millenia Transport, exclude irq 5, irq 9.
- * On the NEC Versa M, exclude irq 9, port 0x2e0-2ff.
- * On the NEC Versa P/75, exclude irq 5, irq 9.

On the NEC Versa S, exclude irq 9, irq 12.

- * On the NEC Versa 6000 series, exclude port 0x2f8-0x33f, irq 9, irq 10.
- * On the NEC Versa SX, exclude port 0x300-0x31f.
- * On the ProStar 9200, Altima Virage, and Acquiline Hurricane DX4-100,exclude irq 5, port 0x330-0x35f. Maybe use memory 0xd8000-0xdfff.
- * On the Siemens Nixdorf SIMATIC PG 720C, use memory 0xc0000-0xcffff,port 0x300-0x3bf.
- * On the TI TravelMate 5000, use memory 0xd4000-0xdffff.
- * On the Toshiba Satellite 4030CDS, exclude irq 9.
- * On the Toshiba T4900 CT, exclude irq 5, port 0x2e0-0x2e8, port0x330-0x338.
- * On the Toshiba Tecra 8000, exclude irq 3, irq 5, irq 9.
- * On the Twinhead 5100, HP 4000, Sharp PC-8700 and PC-8900, excludeirq 9 (sound), irq 12.
- * On an MPC 800 Series, exclude irq 5, port 0x300-0x30f for the CD-ROM.

PowerBook specific settings

On PowerPC based PowerBook systems, the default system resources in/etc/pcmcia/config.opts file are no good at all. Replace allthe IO port and window definitions with something like:

```
include port 0x100-0x4ff, port 0x1000-0x17ff
include memory 0x80000000-0x80ffffff
```

Notes about specific Linux distributions

This section is incomplete. Corrections and additions are welcome.

Debian

Debian uses a System V boot script arrangement. The PCMCIA startupscript is installed as /etc/init.d/pcmcia. New packages use/etc/default/pcmcia for startup options; older versions used/etc/pcmcia.conf for this purpose. Debian's syslogconfiguration will place kernel messages in /var/log/messagesand cardmgr messages in /var/log/daemon.log.

Debian distributes the PCMCIA system in two packages: the``pcmcia-cs" package contains cardmgr and other tools, manpages, and configuration scripts; and the ``pcmcia-modules"package contains the kernel driver modules.

Starting with 3.1.25, a clean PCMCIA install will identify Debiansystems and create a special network.opts file that, in theabsence of other network configuration settings, uses Debian'sifup and ifdown commands to configure a network card basedon settings in /etc/network/interfaces.

Red Hat, Caldera, Mandrake

These distributions use a System V boot script organization. ThePCMCIA startup script is installed as/etc/rc.d/init.d/pcmcia, and boot options are kept in/etc/sysconfig/pcmcia. Beware that installing the Red Hatpackage may install a default boot option file that has PCMCIAdisabled. To enable PCMCIA, the ``PCMCIA'' variable should beset to ``yes''. Red Hat's default syslogd configuration willrecord all interesting messages in /var/log/messages.

Red Hat's PCMCIA package contains a replacement for the network setupscript, /etc/pcmcia/network, which meshes with the Red Hatlinuxconf configuration system. This is convenient for the casewhere just one network adapter is used, with one set of networkparameters, but does not have the full flexibility of the regularPCMCIA network script. Compiling and installing a clean PCMCIA sourcedistribution will overwrite the network script, breaking the link tothe Red Hat tools. If you prefer using the Red Hat tools, either useonly Red Hat RPM's, or replace /etc/pcmcia/network.opts with the following:

```
if [ -f /etc/sysconfig/network-scripts/ifcfg-$2 ] ; then
    start_fn () {
        /etc/sysconfig/network-scripts/ifcfg-$1
        if [ "$ONBOOT" = "yes" ] ; then /sbin/ifup $1 ; fi
        }
        stop_fn () {
        /sbin/ifdown $1
        }
fi
```

Starting with the 3.1.22 release, the PCMCIA installation script willautomatically append a variant of this to the defaultnetwork.opts file, so this problem should no longer be an issue.

If you do use linuxconf (or netconf) to configure yournetwork interface, leave the ``kernel module", ``I/O port", and``irq" parameters blank. Setting these parameters may interfere withproper operation of the PCMCIA subsystem.

At boot time, when the Red Hat network subsystem starts up, it may say``Delaying eth0 initialization" and ``[FAILED]". This is actuallynot a failure: it means that this network interface will not beinitialized until after the PCMCIA network device is configured.

Red Hat bundles their slightly modified PCMCIA source distribution with their kernel sources, rather than as a separate source package. When preparing to build a new set of PCMCIA drivers, you willgenerally want to install Red Hat's kernel-source

RPM(kernel-source-*.i386.rpm), and not the kernel SRPM(kernel-*.src.rpm). The SRPM is tailored for building theirkernel RPM files, which is not exactly what you want. With Red Hat7.0, the kernel-source RPM also includes a mis-configured PCMCIAsource tree; if you want to use it, delete their PCMCIA config.outfile and re-do "make config".

Slackware

Slackware uses a BSD boot script arrangement. The PCMCIA startupscript is installed as /etc/rc.d/rc.pcmcia, and boot optionsare specified in rc.pcmcia itself. The PCMCIA startup scriptis invoked from /etc/rc.d/rc.S.

SuSE

SuSE uses a System V init script arrangement, with init scripts storedunder /etc/init.d. The PCMCIA startup script is installed as/etc/init.d/pcmcia, and startup options are kept in/etc/rc.config. Before release 7.0, init scripts were keptunder /sbin/init.d. In early SuSE releases (pre-5.3), thePCMCIA startup script was somewhat limited and did not allow PCMCIAstartup variables to be overridden from the lilo boot prompt.

SuSE 8.0 includes both the standalone PCMCIA modules, and the 2.4kernel PCMCIA subsystem modules. A new variable, PCMCIA_SYSTEM,is available in /etc/sysconfig/pcmcia to choose betweenthese. It can be set to either ``kernel'' or ``external''.

To look up current PCMCIA issues in SuSE's support database, go tohttp://sdb.suse.de/cgi-bin/sdbsearch_en.cgi?stichwort=PCMCIA.

Resolving installation and configuration problems

This section describes some of the most common failure modes for the PCMCIA subsystem. Try to match your symptoms against the examples. This section only describes general failures that are not specific a particular client driver or type of card.

Before trying to diagnose a problem, you have to know where yoursystem log is kept (see Notes about specific Linux distributions

In 3.1.15 and later releases, the debug-tools subdirectory of the PCMCIA source tree has a few scripts to help diagnose some of the mostcommon configuration problems. The test_setup script checks yourPCMCIA installation for completeness. The test_network andtest_modem scripts will try to diagnose problems with PCMCIAnetwork and modem cards. These scripts can be particularly helpful ifyou are unfamiliar with Linux and are not sure how to approach aproblem.

Try to define your problem as narrowly as possible. If you haveseveral cards, try each card in isolation, and in different combinations. Try cold Linux boots, versus warm boots from Windows.Compare booting with cards inserted, versus inserting cards afterboot. If you normally use your laptop docked, try it undocked. And sometimes, two sockets will behave differently.

For debugging problems in the device configuration scripts, it may be seful to start cardmgr with the ``-v" option. With a3.1.23 or later PCMCIA package, this will cause most important scriptactions to be recorded in the system log.

It is nearly impossible to debug driver problems encountered whenattempting to install Linux via a PCMCIA device. Even if you canidentify the problem based on its symptoms, installation disks are difficult to modify, especially without access to a running Linuxsystem. Customization of installation disks is completely dependenton the choice of Linux distribution, and is beyond the scope of thisdocument. In general, the best course of action is to install Linuxusing some other means, obtain the latest drivers, and then debug the problem if it persists.

Base PCMCIA kernel modules do not load

Symptoms:

- * Kernel version mismatch errors are reported when the PCMCIAstartup script runs.
- * After startup, Ismod does not show any PCMCIA modules.
- * cardmgr reports ``no pcmcia driver in/proc/devices" in the system log.

Kernel modules contain version information that is checked against thecurrent kernel when a module is loaded. The type of checking dependson the setting of the

CONFIG_MODVERSIONS kernel option. If thisis false, then the kernel version number is compiled into each module, and insmod checks this for a match with the running kernel. IfCONFIG_MODVERSIONS is true, then each symbol exported by thekernel is given a sort of checksum. These codes are all compared against the corresponding codes compiled into a module. The intentwas for this to make modules less version-dependent, because thechecksums would only change if a kernel interface changed, and wouldgenerally stay the same across minor kernel updates. In practice, thechecksums have turned out to be even more restrictive, because manykernel interfaces depend on compile-time kernel option settings. Also, the checksums turned out to be an excessively pessimistic judgeof compatibility.

The practical upshot of this is that kernel modules are closely tiedto both the kernel version, and the setting of many kernelconfiguration options. Generally, a set of modules compiled for one2.2.19 kernel will not load against some other 2.2.19 kernel unlessspecial care is taken to ensure that the two were built with similarconfigurations. This makes distribution of precompiled kernel modules tricky business.

You have several options:

- * If you obtained precompiled drivers as part of a Linuxdistribution, verify that you are using an unmodified kernel assupplied with that distribution. If you intend to use precompiledmodules, you generally must stick with the corresponding kernel.
- * If you have reconfigured or upgraded your kernel, you willprobably need to compile and install the PCMCIA package from scratch. This is easily done if you already have the kernel source treeinstalled. See Compilation and installation
- * In some cases, incompatibilities in other system components can prevent correct loading of kernel modules. If you have upgraded yourown kernel, pay attention to the ``minimal requirements" for moduleutilities and binutils listed in the Documentation/Changesfile in the kernel source code tree.

Some client driver modules do not load

- * The base modules (pcmcia_core, ds, i82365) load correctly.
- * Inserting a card gives a high beep + low beep pattern.
- * cardmgr reports version mismatch errors in the system log.

Some of the driver modules require kernel services that may or may notbe present, depending on kernel configuration. For instance, the SCSIcard drivers require that the kernel be configured with SCSI support, and the network drivers require a networking kernel. If a kernellacks a necessary feature, insmod may report undefined symbols and refuse to load a particular module. Note that insmod errormessages do not distinguish between version mismatch errors andmissing symbol errors.

Specifically:

- * The serial client driver serial_cs requires the kernelserial driver to be enabled with CONFIG_SERIAL. This driver maybe built as a module.
- * Support for multiport serial cards or multifunction cards thatinclude serial or modem devices requires CONFIG_SERIAL_SHARE_IRQto be enabled.
- * The SCSI client drivers require that CONFIG_SCSI beenabled, along with the appropriate top level driver options(CONFIG_BLK_DEV_SD, CONFIG_BLK_DEV_SR, etc for 2.2 and laterkernels). These may be built as modules.
- * The network client drivers require that CONFIG_INET isenabled. Kernel networking support cannot be compiled as a module.
- * The token-ring client requires that the kernel be compiled withCONFIG_TR enabled.

There are two ways to proceed:

- * Rebuild your kernel with the necessary features enabled.
- * If the features have been compiled as modules, then modify/etc/pcmcia/config to preload these modules.

The /etc/pcmcia/config file can specify that additionalmodules need to be loaded for a particular client. For example, for the serial driver, one would use:

```
device "serial_cs"
    class "serial" module "misc/serial", "serial_cs"
```

Module paths are specified relative to the top-level module directoryfor the current kernel version; if no relative path is given, then thepath defaults to the pcmcia subdirectory.

ISA interrupt scan failures

- * The system locks up when the PCMCIA drivers are loaded, even with no cards present.
- * The system log shows a successful host controller probe justbefore the lock-up, but does not show interrupt probe results.

After identifying the host controller type, the socket driver probesfor free ISA bus interrupts. The probe involves programming thecontroller for each apparently free interrupt, then generating a``soft" interrupt, to see if the interrupt can be detected correctly. In some cases, probing a particular interrupt can interfere withanother system device.

The reason for the probe is to identify interrupts which appear to befree (i.e., are not reserved by any other Linux device driver), yetare either not physically wired to the host controller, or areconnected to another device that does not have a driver.

In the system log, a successful probe might look like:

```
Intel PCIC probe:
TI 1130 CardBus at mem 0x10211000, 2 sockets
...
ISA irqs (scanned) = 5,7,9,10 status change on irq 10
```

There are two ways to proceed:

* The ISA interrupt probe can be restricted to a list of interrupts using the irq_list parameter for the socket drivers.For example,
 ``irq_list=5,9,10" would limit the scan to threeinterrupts. All 16-bit PCMCIA devices will be restricted to using these interrupts (assuming they pass the probe). You may need to userrial and error to find out which interrupts can be safely probed.

The interrupt probe can be disabled entirely by loading thesocket driver with the ``do_scan=0" option. In this case, a defaultinterrupt list will be used, which just avoids interrupts alreadyallocated for other devices.

In either case, the probe options can be specified using thePCIC_OPTS definition in the PCMCIA startup script, for example:

```
PCIC_OPTS="irq_list=5,9,10"
```

It should be noted that /proc/interrupts is completelyuseless when it comes to diagnosing interrupt probe problems. Theprobe is sensible enough to never attempt to use an interrupt that isalready in use by another Linux driver. So, the PCMCIA drivers arealready using all the information in /proc/interrupts.Depending on system design, an inactive device can still occupy aninterrupt and cause trouble if it is probed for PCMCIA.

IO port scan failures

- * The system locks up when cardmgr is first started. For3.1.24, the lockup happens even with no cards present; for 3.1.25, acard must be inserted.
- * The system log shows a successful host controller probe, including interrupt probe results, but does not show IO proberesults.
- * In some cases, the IO probe will succeed, but report largenumbers of random exclusions.

When cardmgr processes IO port ranges listed in/etc/pcmcia/config.opts, the kernel probes these ranges todetect latent devices that occupy IO space but are not associated with a Linux driver. The probe is read-only, but in rare cases, reading from a device may interfere with an important system function, resulting in a lock-up.

Your system user's guide may include a map of system devices, showingtheir IO and memory ranges. These can be explicitly excluded inconfig.opts.

Alternatively, if the probe is unreliable on your system, it can bedisabled by setting CORE_OPTS to ``probe_io=0". In thiscase, you should be very careful to specify only genuinely availableranges of ports in config.opts, instead of using the defaultsettings.

Memory probe failures

Symptoms:

- * The core drivers load correctly when no cards are present, withno errors in the system log.
- * The system freezes and/or reboots as soon as any card isinserted, before any beeps are heard.

Or alternately:

- * All card insertions generate a high beep followed by a low beep.
- * All cards are identified as ``anonymous memory cards".
- * The system log reports that various memory ranges have been excluded.

The core modules perform a memory scan at the time of first 16-bitcard insertion. This scan can potentially interfere with other memorymapped devices. Also, pre-3.0.0 driver packages perform a moreaggressive scan than more recent drivers. The memory window isdefined in /etc/pcmcia/config.opts. The default window islarge, so it may help to restrict the scan to a narrower range.Reasonable ranges to try include 0xd0000-0xdfff, 0xc0000-0xcffff, 0xc8000-0xcffff, or 0xd8000-0xdffff.

If you have DOS or Windows PCMCIA drivers, you may be able to deducewhat memory region those drivers use. Note that DOS memory addressesare often specified in ``segment" form, which leaves off the finalhex digit (so an absolute address of 0xd0000 might be given as0xd000). Be sure to add the extra digit back when making changes toconfig.opts.

Changing BIOS settings affecting how devices are mapped can sometimesbe useful. Try changing settings for BIOS shadowing, or "Plug and Play OS support".

In unusual cases, a memory probe failure can indicate a timingregister setup problem with the host controller. See the Startup options

* cs: warning: no high memory space available!

CardBus bridges can allocate memory windows outside of the 640KB-1MB``memory hole" in the ISA bus architecture. It is generally a goodidea to configure CardBus bridges to use high memory windows, becausethese are unlikely to conflict with other devices. Also, CardBuscards may require large memory windows, which may be difficult orimpossible to fit into low memory. Card Services will preferentiallyallocate windows in high memory for CardBus bridges, if both low andhigh memory windows are defined in contigoopter of he defaultconfig.opts includes several candidate high memory windows, one of which will work in most cases.

Failure to detect card insertions and removals

Symptoms:

- * Cards are detected and configured properly if present at boottime.
- ^{*} The drivers do not respond to insertion and removal events, either by recording events in the system log, or by beeping.

In most cases, the socket driver (i82365 or tcic) willautomatically probe and select an appropriate interrupt to signal cardstatus changes. The automatic interrupt probe doesn't work on someIntel-compatible controllers, including Cirrus chips and the chipsused in some IBM ThinkPads. If a device is inactive at probe time,its interrupt may also appear to be available. In these cases, thesocket driver may pick an interrupt that is used by another device.

With the i82365 and tcic drivers, the irq_list optioncan be used to limit the interrupts that will be tested. This listlimits the set of interrupts that can be used by PCMCIA cards as wellas for monitoring card status changes. The cs_irq option canalso be used to explicitly set the interrupt to be used for monitoringcard status changes.

If you can't find an interrupt number that works, there is also apolled status mode: both i82365 and tcic will accept apoll_interval=100 option, to poll for card status changes onceper second. This option should also be used if your system has ashortage of interrupts available for use by PCMCIA cards. Especiallyfor systems with more than one host controller, there is littlepoint in dedicating interrupts for monitoring card status changes.

All these options should be set in the PCIC_OPTS= line in either/etc/rc.d/rc.pcmcia or /etc/sysconfig/pcmcia,depending on your site setup.

Interrupt delivery problems

Symptoms:

- * Cards appear to be configured successfully, but don't work.
- * Serial and modem cards may respond very sluggishly.
- * Network cards may report ``interrupt(s) dropped", and/ortransmit timeouts.

The most simple interrupt delivery problems are due to conflicts withother system devices. These can generally be resolved by excludingproblem interrupts in /etc/pcmcia/config.opts. To test, justexclude interrupts one by one until either the problem is fixed or yourun out of interrupts. If no interrupts work, then device conflictsare probably not the problem.

For CardBus bridges, a variety of other interrupt delivery issues maycome into play. For a complete discussion, see PCI interrupt delivery problems

System resource starvation

- * When a card is inserted, it is identified correctly but cannotbe configured (high/low beep pattern).
- * One of the following messages will appear in the system log:

RequestIO: Resource in use RequestIRQ: Resource in use RequestWindow: Resource in use GetNextTuple: No more items could not allocate nn IO ports for CardBus socket n could not allocate nnK memory for CardBus socket n could not allocate interrupt for CardBus socket n

Interrupt starvation often indicates a problem with the interruptprobe (see ISA interrupt scan failures

If the interrupt probe is not working properly, the socket driver mayallocate an interrupt for monitoring card insertions, even wheninterrupts are too scarce for this to be a good idea. You can switchthe controller to polled mode by setting PCIC_OPTS to``poll_interval=100'. Or, if you have a CardBus controller andan older version of the PCMCIA drivers, try ``pci_csc=1", whichselects a PCI interrupt (if available) for card status changes.

In some cases, kernel misconfiguration can also produce an apparentinterrupt shortage. On 2.4 and later kernels, if CONFIG_ISAis not enabled, then the PCMCIA drivers will assume no ISA businterrupts are available.

IO port starvation is fairly uncommon, but sometimes happens withcards that require large, contiguous, aligned regions of IO portspace, or that only recognize a few specific IO port positions. Thedefault IO port ranges in /etc/pcmcia/config.opts arenormally sufficient, but may be extended. If this is the problem,try uncommenting the ``include port 0x1000-0x17ff'' line inconfig.opts. In rare cases, starvation may indicate that the IOport probe failed (see IO port scan failures

Memory starvation is also uncommon with the default memory windowsettings in config.opts. CardBus cards may require larger memoryregions than typical 16-bit cards. Since CardBus memory windows canbe mapped anywhere in the host's PCI address space (rather than justin the 640K-1MB ``hole" in PC systems), it is helpful to specifylarge memory windows in high memory, such as 0xa0000000-0xa0fffff.

Resource conflict only with two cards inserted

Symptoms:

- * Two cards each work fine when used separately.
- * When both cards are inserted, only one works.

This usually indicates a resource conflict with a system device thatLinux does not know about. PCMCIA devices are dynamically configured, so, for example, interrupts are allocated as needed, rather thanspecifically assigned to particular cards or sockets. Given a list ofresources that appear to be available, cards are assigned resources in the order they are configured. In this case, the card configured lastis being assigned a resource that in fact is not free.

Check the system log to see what resources are used by the non-workingcard. Exclude these in /etc/pcmcia/config.opts, and restart the cardmar daemon to reload the resource database.

Device configuration does not complete

Symptoms:

- * When a card is inserted, exactly one high beep is heard.
- * Subsequent card insertions and removals may be ignored.

This indicates that the card was identified successfully, however, cardmgr has been unable to complete the configuration process forsome reason. The most likely reason is that a step in the card setupscript has blocked. A good example would be the network scriptblocking if a network card is inserted with no actual network hookuppresent.

To pinpoint the problem, you can manually run a setup script to seewhere it is blocking. The scripts are in the /etc/pcmciadirectory. They take two parameters: a device name, and an action. The cardmgr daemon records the configuration commands in thesystem log. For example, if the system log shows that the command ``./network start etho" was the last command executed bycardmgr, the following command would trace the script:

```
sh -x /etc/pcmcia/network start eth0
```

Usage and features

Tools for configuring and monitoring PCMCIA devices

If the modules are all loaded correctly, the output of the Ismodcommand should look like the following, when no cards are inserted:

Module	Size	Used	by	
ds	5640	2		
i82365	15452	2		
pcmcia_core	30012	3	[ds	i82365]

The system log should also include output from the socket driverdescribing the host controller(s) found and the number of socketsdetected.

The cardmgr configuration daemon

The >cdx>cardmgr>/cdx> daemon is responsible for monitoringPCMCIA sockets,

loading client drivers when needed, and running user-level scripts inresponse to card insertions and removals. It records its actions in the system log, but also uses beeps to signal card status changes. The tones of the beeps indicate success or failure of

particular configuration steps. Two high beeps indicate that a card wasidentified and configured successfully. A high beep followed by a lowbeep indicates that a card was identified, but could not be configured for some reason. One low beep indicates that a card could not beidentified. page 20 of 55

The cardmgr daemon configures cards based on a database of knowncard types kept in >cdx>/etc/pcmcia/config>/cdx>. This filedescribes the various client drivers, then describes how to identifyvarious cards, and which driver(s) belong with which cards. Theformat of this file is described in the pcmcia(5) man page.

The socket status file, stab

Cardmgr records device information for each socket in>cdx>/var/lib/pcmcia/stab>/cdx>. Here is a samplestab listing:

Socket 0: Adaptec APA-1460 SlimSCSI 0 scsi aha152x_cs 0 sda 8 0 0 scsi aha152x_cs 1 scd0 11 0 Socket 1: Serial or Modem Card 1 serial serial_cs 0 ttyS1 5 65

For the lines describing devices, the first field is the socket, thesecond is the device class, the third is the driver name, the fourthis used to number multiple devices associated with the same driver, the fifth is the device name, and the final two fields are the majorand minor device numbers for this device (if applicable). See thestab man page for more info.

In 2.4 and later kernels, hot plut PCI drivers for CardBus cards arenot managed by cardmgr; they are managed by the hotplugsubsystem. See http://linux-hotplug.sourceforge.net forinformation about this facility. When cardmgr sees a card that isowned by a hot plug PCI driver, it will ignore that card. There willbe one beep when these cards are inserted or ejected, but they will beidentified only as a ``CardBus hotplug device'' in the system log andstab file.

The cardctl and cardinfo utilities

The >cdx>cardctl>/cdx> command can be used to check the status of asocket, or to see how it is configured. It can also be used to alterthe configuration status of a card. Here is an example of theoutput of the ``cardctl config'' command:

```
Socket 0:
  not configured
Socket 1:
  Vcc = 5.0, Vpp1 = 0.0, Vpp2 = 0.0
  Card type is memory and I/O
  IRQ 3 is dynamic shared, level mode, enabled
  Speaker output is enabled
  Function 0:
    Config register base = 0x0800
        Option = 0x63, status = 0x08
        I/O window 1: 0x0280 to 0x02bf, auto sized
        I/O window 2: 0x02f8 to 0x02ff, 8 bit
```

Or ``cardctl ident", to get card identification information:

```
Socket 0:
  no product info available
Socket 1:
    product info: "LINKSYS", "PCMLM336", "A", "0040052D6400"
    manfid: 0x0143, 0xc0ab
    function: 0 (multifunction)
```

The ``cardctl suspend" and ``cardctl resume" commands canbe used to shut down a card without unloading its associated drivers. The ``cardctl reset" command attempts to reset and reconfigure acard. ``cardctl insert" and ``cardctl eject" mimic theactions performed when a card is physically inserted or ejected, including loading or unloading drivers, and configuring or shuttingdown devices.

If you are running X, the >cdx>cardinfo>/cdx> utility produces graphical display showing the current status of all PCMCIA sockets, similar in content to ``cardctl config". It also provides agraphical interface to most other cardctl functions.

Inserting and ejecting cards

In theory, you can insert and remove PCMCIA cards at any time. However, it is a good idea not to eject a card that is currently beingused by an application program. Kernels older than 1.1.77 would oftenlock up when serial/modem cards were ejected, but this should be fixednow.

Some card types cannot be safely hot ejected. Specifically, ATA/IDEand SCSI interface cards are not hot-swap-safe. This is unlikely tobe fixed, because a complete solution would require significantchanges to the Linux block device model. Also, it is generally notsafe to hot eject CardBus cards of any type. This is likely toimprove gradually as hot swap bugs in the CardBus drivers are foundand fixed. For these card types (IDE, SCSI, CardBus), it isrecommended that you always use ``cardctl eject" beforeejecting.

Card Services and Advanced Power Management

Card Services can be compiled with support for APM(Advanced Power Management) if you've configured yourkernel with APM support. The APM kernel driver is maintained byStephen Rothwell (Stephen.Rothwell@canb.auug.org.au). The apmddaemon is maintained by Avery Pennarun (apenwarr@worldvisions.ca), withmore information available athttp://www.worldvisions.ca/~apenwarr/apmd/. The PCMCIAmodules will automatically be configured for APM if a compatibleversion is detected on your system.

Whether or not APM is configured, you can use ``cardctl suspend"before suspending your laptop, and ``cardctl resume" afterresuming, to cleanly shut down and restart your PCMCIA cards. Thiswill not work with a modem that is in use, because the serial driverisn't able to save and restore the modem operating parameters.

APM seems to be unstable on some systems. If you experience trouble with APM and PCMCIA on your system, try to narrow down the problem toone package or the other before reporting a bug.

Some drivers, notably the PCMCIA SCSI drivers, cannot recover from asuspend/resume cycle. When using a PCMCIA SCSI card, always use``cardctl eject" prior to suspending the system.

Shutting down the PCMCIA system

To unload the entire PCMCIA package, invoke rc.pcmcia with:

/etc/rc.d/rc.pcmcia stop

This script will take several seconds to run, to give all clientdrivers time to shut down gracefully. If a device is currently inuse, the shutdown will be incomplete, and some kernel modules may notbe unloaded. To avoid this, use ``cardctl eject" to shut downall sockets before invoking rc.pcmcia. The exit status of thecardctl command will indicate if any sockets could not be shutdown.

Overview of the PCMCIA configuration scripts

The following information applies to cards that are managed bycardmgr. In 2.4 and later kernels, if the kernel PCMCIAsubsystem is active, then CardBus cards are managed by thehotplug subsystem and the PCMCIA scripts are not used.

Each PCMCIA device has an associated ``class" that describes how itshould be configured and managed. Classes are associated with devicedrivers in /etc/pcmcia/config. There are currently five IOdevice classes (network, SCSI, cdrom, fixed disk, and serial) andtwo memory device classes (memory and FTL). For each class, there are twoscripts in

>cdx>/etc/pcmcia>/cdx>: a main configuration script(i.e., /etc/pcmcia/scsi for SCSI devices), and an optionsscript (i.e., /etc/pcmcia/scsi.opts). The main script for adevice will be invoked to configure that device when a card isinserted, and to shut down the device when the card is removed. Forcards with several associated devices, the script will be invoked foreach device.

The config scripts start by extracting some information about a devicefrom the stab file. Each script constructs a ``device address", that uniquely describes the device it has been asked to configure, inthe ADDRESS shell variable. This is passed to the *.optsscript, which should return information about how a device at thisaddress should be configured. For some devices, the device address isjust the socket number. For others, it includes extra informationthat may be useful in deciding how to configure the device. Forexample, network devices pass their hardware ethernet address as partof the device address, so the network.opts script could use thisto select from several different configurations.

The first part of all device addresses is the current PCMCIA``scheme". This parameter is used to support multiple sets of deviceconfigurations based on a single external user-specified variable. One use of schemes would be to have a ``home" scheme, and a ``work"scheme, which would include different sets of network configurationparameters. The current scheme is selected using the ``cardctlscheme" command. The default if no scheme is set is ``default".

There are a few additional shell variables that can be used in*.opts files in addition to ADDRESS:

SOCKET, CLASS, DRIVER, INSTANCE, DEVICE, MAJOR, MINOR

These correspond to individual fields from one line in the stabfile. See its man page for details.

PRODID_1, PRODID_2, PRODID_3,

PRODID_4, MANFID, FUNCID

These are equivalent to the output of ``cardctl info" and givemore detailed card identification information.

As the *.opts files are just shell scripts, it is not required that they follow the form of the examples, which just return settingsbased on ADDRESS.

As a general rule, when configuring Linux for a laptop, PCMCIA devicesshould only be configured from the PCMCIA device scripts. Do not tryto configure a PCMCIA device the same way you would configure apermanently attached device. However, some Linux distributionsprovide PCMCIA packages that are hooked into those distributions' owndevice configuration tools. In that case, some of the followingsections may not apply; ideally, this will be documented by the distribution maintainers.

PCMCIA network adapters

Linux ethernet-type network interfaces normally have names likeeth0, eth1, and so on. Token-ring adapters are handledsimilarly, however they are named tr0, tr1, and so on. The ifconfig command is used toview or modify the state of a network interface. A peculiarity ofLinux is that network interfaces do not have corresponding devicefiles under /dev, so do not be surprised when you do not findthem.

When an ethernet card is detected, it will be assigned the first freeinterface name, which will normally be eth0. Cardmgr willrun the /etc/pcmcia/network script to configure theinterface, which normally reads network settings from/etc/pcmcia/network.opts. The network andnetwork.opts scripts will be executed only when your ethernetcard is actually present. If your system has an automatic networkconfiguration facility, it may or may not be PCMCIA-aware. Consultthe documentation of your Linux distribution and theNotes about specific Linux distributions

The device address passed to network.opts consists of fourcomma-separated fields: the scheme, the socket number, the deviceinstance, and the card's hardware ethernet address. The deviceinstance is used to number devices for cards that have several network interfaces, so it usually be 0. If you have several network cards used fordifferent purposes, one option would be to configure the cards basedon socket position, as in:

```
case "$ADDRESS" in
*,0,*,*)
    # definitions for network card in socket 0
;;
*,1,*,*)
    # definitions for network card in socket 1
;;
esac
```

Alternatively, they could be configured using their hardwareaddresses, as in:

Network device parameters

The following parameters can be defined in network.opts:

IF_PORT

Specifies the ethernet transceiver type, for certain 16-bit cards thatdo not autodetect. See ``man ifport" and ``man mii-tool"for more information.

BOOTP

A boolean (y/n) value: indicates if the host's IP address and routinginformation should be obtained using the BOOTP protocol, withbootpc or pump.

DHCP

A boolean (y/n) value: indicates if the host's IP address and routinginformation should be obtained from a DHCP server. The network scriptfirst looks for dhcpcd, then dhclient, then pump.

DHCP_HOSTNAME

Specifies a hostname to be passed to dhcpcd or pump, forinclusion in DHCP messages. IPADDR

The IP address for this interface.

NETMASK, BROADCAST, NETWORK

Basic network parameters: see the networking HOWTO for moreinformation.

GATEWAY

The IP address of a gateway for this host's subnet. Packets withdestinations outside this subnet will be routed to this gateway.

DOMAIN

The local network domain name for this host, to be used in creating/etc/resolv.conf. SEARCH

A search list for host name lookup, to be added to/etc/resolv.conf. DOMAIN and SEARCH are mutually exclusive: see ``man resolver'' for more information.

DNS_1, DNS_2, DNS_3

Host names or IP addresses for nameservers for this interface, to beadded to /etc/resolv.conf MOUNTS

A space-separated list of NFS mount points to be mounted for this interface. IPX_FRAME, IPX_NETNUM

For IPX networks: the frame type and network number, passed to

theipx_interface command.

NO_CHECK, NO_FUSER

Boolean (y/n) settings for card eject policy. If NO_CHECK isset, then ``cardctl eject" will shut down a device even ifthere are open connections. If NO_FUSER is set, then the scriptwill not check for busy NFS mounts or kill processes using those mounts.

For example:

```
case "$ADDRESS" in
*,*,*,*)
IF_PORT="10base2"
BOOTP="n"
IPADDR="10.0.0.1"
NETMASK="255.255.255.0"
NETWORK="10.0.0.0"
BROADCAST="10.0.0.255"
GATEWAY="10.0.0.1"
DOMAIN="domain.org"
DNS_1="dns1.domain.org"
;;
```

To automatically mount and unmount NFS filesystems, first add allthese filesystems to /etc/fstab, but include noautoin the mount options. In network.opts, list the filesystemmount points in the MOUNTS variable. It is especially important to use either cardctl or cardinfo to shut down anetwork card when NFS mounts are active. It is not possible tocleanly unmount NFS filesystems if a network card is simply ejected without warning.

In addition to the usual network configuration parameters, thenetwork.opts script can specify extra actions to be taken afteran interface is configured, or before an interface is shut down. Ifnetwork.opts defines a shell function called start_fn, it will be invoked by the network script after the interface isconfigured, and the interface name will be passed to the function as itsfirst (and only) argument. Similarly, if it is defined, stop_fnwill be invoked before shutting down an interface.

The transceiver type for some (mostly old) cards must be manually beselected using the IF_PORT setting. This can either be a numericvalue, or a keyword identifying the transceiver type. All the networkdrivers default to either autodetect the interface if possible, or10baseT otherwise. The ifport command can be used to check orset the current transceiver type. For example:

ifport eth0 10base2
#
ifport eth0
eth0 2 (10base2)

Most modern 10/100baseT cards use a ``media independent interface"(MII) transceiver that automatically selects line speed and duplexsetting. The mii-tool command can be used to monitor and controlthe behavior of the MII interface.

Comments about specific cards

- * With IBM CCAE and Socket EA cards, the transceiver type (10base2,10baseT, AUI) needs to be set when the network device is configured. Make sure that the transceiver type reported in the system log matchesyour connection.
- * The Farallon EtherWave is actually based on the 3Com 3c589, with aspecial transceiver. Though the EtherWave uses 10baseT-styleconnections, its transceiver requires that the 3c589 be configured in10base2 mode.
- * If you have trouble with an IBM CCAE, NE4100, Thomas Conrad, orKingston adapter, try increasing the memory accesstime with the mem_speed=# option to the pcnet_cs module.An example of how to do this is given in the standard config.optsfile. Try speeds of up to 1000 (in nanoseconds).
- * For the New Media Ethernet adapter, on some systems, it may benecessary to increase the IO port access time with theio_speed=# option when the pcmcia_core module is loaded.Edit CORE_OPTS in the startup script to set this option.
- * The multicast support in the New Media Ethernet driver is incomplete. The latest driver will function with multicast kernels, but will ignoremulticast packets. Promiscuous mode should work properly.

- The driver used by the IBM and 3Com token ring adaptersseems to behave very badly if the cards are not connected to a ringwhen they get initialized. Always connect these cards to the netbefore they are powered up. If if config reports the hardwareaddress as all 0's, this is likely to be due to a memory windowconfiguration problem.
- Some Linksys, D-Link, and IC-Card 10baseT/10base2 cards have a uniqueway of selecting the transceiver type that isn't handled by the Linuxdrivers. One workaround is to boot DOS and use the vendor-suppliedutility to select the transceiver, then warm boot Linux. Alternatively, a Linux utility to perform this function is availableat http://pcmcia-cs.sourceforge.net/ftp/extras/dlport.c.
- 16-bit PCMCIA cards have a maximum performance of 1.5-2 MB/sec. Thatmeans that any 16-bit 100baseT card (i.e., any card that uses thepcnet_cs, 3c574&lowbar:cs, smc91c92&lowbar:cs, or xirc2ps&lowbar:csdriver) will never achieve full 100baseT throughput. Only CardBusnetwork adapters can fully exploit 100baseT data rates.
- * For WaveLAN wireless network adapters, Jean Tourrilhes(jt@hpl.hp.com) has put together a wireless HOWTO athttp://www.hpl.hp.com/personal/Jean Tourrilhes/Linux/.

Diagnosing problems with network adapters

- * In 3.1.15 and later PCMCIA releases, the test_network script inthe debug-tools subdirectory of the PCMCIA source tree will spotsome common problems.
- * Is your card recognized as an ethernet card? Check the system log andmake sure that cardmar identifies the card correctly and startsup one of the network drivers. If it doesn't, your card might stillbe usable if it is compatible with a supported card. This will be most easily done if the card claims to be ``NE2000 compatible".
- Is the card configured properly? If you are using a supported card, and it was recognized by cardmgr, but still doesn't work, theremight be an interrupt or port conflict with another device. Find outwhat resources the card is using (from the system log), and try excluding these in /etc/pcmcia/config.opts to forcethe card to use something different.
- If your card seems to be configured properly, but sometimes locks up, particularly under high load, you may need to try changing your socketdriver timing parameters. See the Startup options
- If you get ``Network is unreachable" messages when you try toaccess the network, then the routing information specified in/etc/pcmcia/network.opts is incorrect. This exact message isan absolutely foolproof indication of a routing error. On the otherhand, mis-configured cards will usually fail silently.
- * If you are trying to use DHCP to configure your network interface, trytesting things with a static IP address instead, to rule out a DHCPconfiguration problem.
- To diagnose problems in /etc/pcmcia/network.opts, start bytrying to ping other systems on the same subnet using their IPaddresses. Then try to ping your gateway, and then machines on othersubnets. Ping machines by name only after trying these simpler tests.
- Make sure your problem is really a PCMCIA one. It may help to see seeif the card works under DOS with the vendor's drivers. Double checkyour modifications to the /etc/pcmcia/network.opts script.Make sure your drop cable, ``T" jack, terminator, etc are working. page 27 of 55

* Use real network cables. Don't even think about using that old phonecord you found in your basement. And this means Category 5 cable for100baseT. It really matters.

PCMCIA serial and modem devices

Linux serial devices are accessed via the /dev/ttyS* and/dev/cua* special device files. In pre-2.2 kernels, thettyS* devices were for incoming connections, such as directlyconnected terminals, and the cua* devices were for outgoingconnections, such as modems. Use of cua* devices is deprecated in current kernels, and ttyS* can be used for all applications. The configuration of a serial device can be examined and modified with the setserial command.

When a serial or modem card is detected, it will be assigned to thefirst available serial device slot. This will usually be/dev/ttyS1 (cua1) or /dev/ttyS2 (cua2),depending on the number of built-in serial ports. The ttyS*device is the one reported in stab. The defaultserial device option script, /etc/pcmcia/serial.opts, willlink the device file to /dev/modem as a convenience. Forpre-2.2 kernels, the link is made to the cua* device.

Do not try to use /etc/rc.d/rc.serial to configure a PCMCIAmodem. This script should only be used to configure non-removabledevices. Modify /etc/pcmcia/serial.opts if you want to doanything special to set up your modem. Also, do not try to change theIO port and interrupt settings of a serial device usingsetserial. This would tell the serial driver to look for thedevice in a different place, but would not change how the card'shardware is actually configured. The serial configuration scriptallows you to specify other setserial options, as well as whether a line should be added to /etc/inittab for this port.

The device address passed to serial.opts has threecomma-separated fields: the first is the scheme, the second is thesocket number, and the third is the device instance. The deviceinstance may take several values for cards that support multipleserial ports, but for single-port cards, it will always be 0. If youcommonly use more than one modem, you may want to specify differentsettings based on socket position, as in:

If a PCMCIA modem is already configured when Linux boots, it may beincorrectly identified as an ordinary built-in serial port. This isharmless, however, when the PCMCIA drivers take control of the modem, it will be assigned a different device slot. It is best to eitherparse stab or use /dev/modem, rather than expecting a PCMCIA modem to always have the same device assignment.

If you configure your kernel to load the basic Linux serial portdriver as a module, you must edit /etc/pcmcia/config toindicate that this module must be loaded. Edit the serial deviceentry to read:

```
device "serial_cs"
    class "serial" module "misc/serial", "serial_cs"
```

Serial device parameters

The following parameters can be defined in serial.opts:

LINK

Specifies a path for a symbolic link to be created to the ``callout"device (e.g., /dev/cua* for pre-2.2, or /dev/ttyS*for 2.2 kernels).

SERIAL_OPTS

Specifies options to be passed to the setserial command.

INITTAB

If specified, this will be used to construct an inittab entry forthe device.

NO_CHECK, NO_FUSER

Boolean (y/n) settings for card eject policy. If NO_CHECK istrue, then ``cardctl eject" will shut down a device even if it busy. If NO_FUSER is true, then the script will not try tokill processes using an ejected device.

For example:

```
case "$ADDRESS" in
*,*,*)
   LINK="/dev/modem"
   SERIAL_OPTS=""
   INITTAB="/sbin/getty"
```

Comments about specific cards

- * The Uniden Data 2000 Wireless CDPD card has some special dialingstrings for initiating SLIP and PPP mode. For SLIP, use ``ATDT2";for PPP, "ATDT0".
- * Socket IO revision H serial port cards have a faster-than-normalclock rate for the UART. The card's actual baud rate is four timesfaster than the serial driver thinks it is. To work around theproblem, specify SERIAL_OPTS="baud_base 460800" in/etc/pcmcia/serial.opts.

Diagnosing problems with serial devices

- * In 3.1.15 and later PCMCIA releases, the test_modem script in thedebug-tools subdirectory of the PCMCIA source tree will spot somecommon problems.
- * Is your card recognized as a modem? Check the system log andmake sure that cardmgr identifies the card correctly and starts up theserial_cs driver. If it doesn't, you may need to add a new entry toyour /etc/pcmcia/config file so that it will be identified properly.See the Configuring unrecognized cards

- * Is the modem configured successfully by serial_cs? Again, checkthe system log and look for messages from the serial_cs driver. Ifyou see ``register_serial() failed'', you may have an I/O port conflictwith another device. Anothertip-off of a conflict is if the device is reported to be an 8250; mostmodern modems should be identified as 16550A UART's. If youthink you're seeing a port conflict, edit /etc/pcmcia/config.optsand exclude the port range that was allocated for the modem.
- * Is there an interrupt conflict? If the system log looks good,but the modem just doesn't seem to work, try using setserial tochange the irq to 0, and see if the modem works. This causes theserial driver to use a slower polled mode instead of using interrupts. If this seems to fix the problem, it is likely that some other devicein your system is using the interrupt selected by serial_cs. Youshould add a line to /etc/pcmcia/config.opts to exclude thisinterrupt.
- * If the modem seems to work only very, very slowly, this is an almostcertain indicator of an interrupt conflict.
- * Make sure your problem is really a PCMCIA one. It may help to seeif the card works under DOS with the vendor's drivers. Also, don'ttest the card with something complex like SLIP or PPP until you aresure you can make simple connections. If simple things work but SLIPdoes not, your problem is most likely with SLIP, not with PCMCIA.
- * If you get kernel messages indicating that the serial_cs module cannotbe loaded, it means that your kernel does not have serial devicesupport. If you have compiled the serial driver as a module, you mustmodify /etc/pcmcia/config to indicate that theserial module should be loaded before serial_cs.

PCMCIA parallel port devices

The Linux parallel port driver is layered so that several high-leveldevice types can share use of the same low level port driver. Printerdevices are accessed via the /dev/lp* special device files.The configuration of a printer device can be examined and modified with the tunelp command.

The parport_cs module depends on the parport andparport_pc drivers, which may be either compiled into the kernelor compiled as modules. The layered driver structure means that anytop-level parallel drivers (such as the plip driver, the printerdriver, etc) must be compiled as modules. These drivers only recognize parallel port devices at module startup time, so they needto be loaded after any PC Card parallel devices are configured.

The device address passed to parport.opts has threecomma-separated fields: the first is the scheme, the second is thesocket number, and the third is the device instance. The deviceinstance may take several values for cards that support multipleparallel ports, but for single-port cards, it will always be 0. Ifyou commonly use more than one such card, you may want to specifydifferent settings based on socket position, as in:

```
case "$ADDRESS" in
*,0,*)
# Options for card in socket 0
LINK=/dev/printer0
;;
*,1,*)
# Options for card in socket 1
LINK=/dev/printer1
;;
```

Parallel device parameters

The following parameters can be defined in parport.opts:

LINK

Specifies a path for a symbolic link to be created to the printerport.

LP_OPTS

Specifies options to be passed to the tunelp command.

NO_CHECK, NO_FUSER

Boolean (y/n) settings for card eject policy. If NO_CHECK istrue, then ``cardctl eject" will shut down a device even if it busy. If NO_FUSER is true, then the script will not try tokill processes using an ejected device.

For example:

```
case "$ADDRESS" in
*,*,*,*)
LINK="/dev/printer"
LP_OPTS=""
```

Diagnosing problems with parallel port devices

- * Is there an interrupt conflict? If the system log looks good,but the port just doesn't seem to work, try using tunelp to change the irq to 0, and see if things improve. This switches thedriver to polling mode. If this seems to fix the problem, it islikely that some other device in your system is using the interruptselected by parport_cs. You should add a line to/etc/pcmcia/config.opts to exclude this interrupt.
- * If you get kernel messages indicating that the parport_cs modulecannot be loaded, it means that your kernel does not have paralleldevice support. If you have compiled the parallel driver as a module,you may need to modify /etc/pcmcia/config to indicate that theparport and parport_pc modules should be loaded beforeparport_cs.

PCMCIA SCSI adapters

All the currently supported PCMCIA SCSI cards are work-alikes of oneof the following ISA bus cards: the Qlogic, the Adaptec AHA-152X, orthe Future Domain TMC-16x0. The PCMCIA drivers are built by linkingsome PCMCIA-specific code (in qlogic_cs.c, aha152x_cs.c, orfdomain_cs.c) with the normal Linux SCSI driver, pulled from theLinux kernel source tree. The Adaptec APA1480 CardBus driver is basedon the kernel aic7xxx PCI driver. Due to limitations in the LinuxSCSI driver model, only one removable card per driver is supported.

When a new SCSI host adapter is detected, the SCSI drivers will probe or devices. Check the system log to make sure your devices are detected properly. New SCSI devices will be assigned to the first available SCSI device files. The first SCSI disk will be/dev/sda, the first SCSI tape will be /dev/st0, and the first CD-ROM will be /dev/scd0.

A list of SCSI devices connected to this host adapter will be shown instab, and the SCSI configuration script,/etc/pcmcia/scsi, will be called once for each attacheddevice, to either configure or shut down that device. The defaultscript does not take any actions to configure SCSI devices, but willproperly unmount filesystems on SCSI devices when a card is removed.

The device addresses passed to scsi.opts are complicated, becauseof the variety of things that can be attached to a SCSI adapter.Addresses consist of either six or seven comma-separated fields: the urrent scheme, the device type, the socket number, the SCSI channel, ID, and logical unitnumber, and optionally, the partition number. The device type will be``sd" for disks, ``st" for tapes, ``sr" for CD-ROM devices, and``sg" for generic SCSI devices. For most setups, the SCSI channeland logical unit number will be 0. For disk devices with several partitions, scsi.opts will first be called for the whole device, with a five-field address. The script should set the PARTSvariable to a list of partitions. Then, scsi.opts will be called for each partition, with the longer six-field addresses.

If your kernel does not have a top-level driver (disk, tape, etc) fora particular SCSI device, then the device will not be configured bythe PCMCIA drivers. As a side effect, the device's name instab will be something like ``sd#nnnn" where ``nnnn"is a four-digit hex number. This happens when cardmgr is unableto translate a SCSI device ID into a corresponding Linux device name.

It is possible to modularize the top-level SCSI drivers so that theyare loaded on demand. To do so, you need to edit/etc/pcmcia/config to tell cardmgr which extra modulesneed to be loaded when your adapter is configured. For example:

device "aha152x_cs"
 class "scsi" module "scsi/scsi_mod", "scsi/sd_mod", "aha152x_cs"

would say to load the core SCSI module and the top-level disk drivermodule before loading the regular PCMCIA driver module.

Always turn on SCSI devices before powering up your laptop, or beforeinserting the adapter card, so that the SCSI bus is properlyterminated when the adapter is configured. Also be very careful aboutejecting a SCSI adapter. Be sure that all associated SCSI devices areunmounted and closed before ejecting the card. The best way to ensurethis is to use either cardctl or cardinfo to request cardremoval before physically ejecting the card. For now, all SCSIdevices should be powered up before plugging in a SCSI adapter, and should stay connected until after you unplug the adapter and/or powerdown your laptop.

There is a potential complication when using these cards that does notarise with ordinary ISA bus adapters. The SCSI bus carries a``termination power" signal that is necessary for proper operation ofordinary passive SCSI terminators. PCMCIA SCSI adapters do not supplytermination power, so if it is required, an external device mustsupply it. Some external SCSI devices may be configured to supplytermination power. Others, such as the Zip Drive and the SyquestEZ-Drive, use active terminators that do not depend on it. In somecases, it may be necessary to use a special terminator block such asthe APS SCSI Sentry 2, which has an external power supply. Whenconfiguring your SCSI device chain, be aware of whether or not any ofyour devices require or can provide termination power.

SCSI device parameters

The following parameters can be defined in scsi.opts:

LINK

Specifies a path for a symbolic link to be created to this device.

DO_FSTAB

A boolean (y/n) setting: specifies if an entry should be added to/etc/fstab for this device. DO_FSCK

A boolean (y/n) setting: specifies if the filesystem should be checkedbefore being mounted, with ``fsck -Ta".

DO_MOUNT

A boolean (y/n) setting: specifies if this device should beautomatically mounted at card insertion time.

FSTYPE, OPTS, MOUNTPT

The filesystem type, mount options, and mount point to be used for thefstab entry and/or mounting the device.

NO_CHECK, NO_FUSER

Boolean (y/n) settings for card eject policy. If NO_CHECK istrue, then ``cardctl eject" will shut down a device even if it busy. If NO_FUSER is true, then the script will not try tokill processes using an ejected device.

For example, here is a script for configuring a disk device at SCSI ID3, with two partitions, and a CD-ROM at SCSI ID 6:

```
case "$ADDRESS" in
*,sd,*,0,3,0)
    # This device has two partitions...
    PARTS="1 2"
*,sd,*,0,3,0,1)
    # Options for partition 1:
    # update /etc/fstab, and mount an ext2 fs on /usr1
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="ext2"
    OPTS=" "
    MOUNTPT="/usr1"
    ;;
*,sd,*,0,3,0,2)
    # Options for partition 2:
    # update /etc/fstab, and mount an MS-DOS fs on /usr2
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="msdos"
    OPTS=""
    MOUNTPT="/usr2"
    ;;
*,sr,*,0,6,0)
    # Options for CD-ROM at SCSI ID 6
    PARTS=""
    DO_FSTAB="y" ; DO_FSCK="n" ; DO_MOUNT="y"
    FSTYPE="iso9660"
    OPTS="ro"
    MOUNTPT="/cdrom"
    ;;
esac
```

Comments about specific cards

- * The Adaptec APA-1480 CardBus card needs a large IO port window (256contiguous ports aligned on a 256-port boundary). It may be necessaryto expand the IO port regions in /etc/pcmcia/config.opts toguarantee that such a large window can be found.
- * The Adaptec APA-460 SlimSCSI adapter is not supported. This card wasoriginally sold under the Trantor name, and when Adaptec merged withTrantor, they continued to sell the Trantor card with an Adapteclabel. The APA-460 is not compatible with any existing Linux driver.
- * I have had one report of a bad interaction between the New Media BusToaster and a UMAX Astra 1200s scanner. Due to the complexity of theSCSI protocol, when diagnosing problems with SCSI devices, it is worthconsidering that incompatible combinations like this may exist and maynot be documented.

Diagnosing problems with SCSI adapters

* With the aha152x_cs driver (used by Adaptec, New Media, anda few others), it seems that SCSI disconnect/reconnect support is afrequent source of trouble with tape drives. To disable this ``feature,"add the following to /etc/pcmcia/config.opts:

module "aha152x_cs" opts "reconnect=0"

* Also with the aha152x_cs driver, certain devices seem to requirea longer startup delay, controlled via the reset_delay moduleparameter. The Yamaha 4416S CDR drive is one such device. The resultis the device is identified successfully, then hangs the system. Insuch cases, try:

module "aha152x_cs" opts "reset_delay=500"

- * Another potential source of SCSI device probe problems is probing ofmultiple LUN's. If you see successful detection of a device, followedby SCSI bus timeouts when LUN 1 for that device is probed, thendisable the kernel's CONFIG_SCSI_MULTI_LUN option.
- * If you have compiled SCSI support as modules (CONFIG_SCSI is``m"), you may need to modify /etc/pomcia/config to load theSCSI modules before the appropriate
- may need to modify /etc/pcmcia/config to load theSCSI modules before the appropriate *_cs driver is loaded.
- * If you get ``aborting command due to timeout'' messages when the SCSIbus is probed, you almost certainly have an interrupt conflict.
- * If the host driver reports ``no SCSI devices found", verify that yourkernel was compiled with the appropriate top-level SCSI drivers foryour devices (i.e., disk, tape, CD-ROM, and/or generic). If atop-level driver is missing, devices of that type will be ignored.

PCMCIA memory cards

The memory_cs driver handles all types of memory cards, as wellas providing direct access to the PCMCIA memory address space forcards that have other functions. When loaded, it creates acombination of character and block devices. See the man page for themodule for a complete description of the device naming scheme. Blockdevices are used for disk-like access (creating and mountingfilesystems, etc). The character devices are for "raw" unbufferedreads and writes at arbitrary locations.

The device address passed to memory.opts consists of two fields:the scheme, and the socket number. The options are applied to thefirst common memory partition on the corresponding memory card.

Some flash memory cards, and most simple static RAM cards, lack a ``Card Information Structure" (CIS), which is the system PCMCIA cardsuse to identify themselves. Normally, cardmgr will assume thatany card that lacks a CIS is a simple memory card, and load thememory_cs driver. Thus, a common side effect of a general cardidentification problem is that other types of cards may be misdetected as memory cards.

cardidentification problem is that other types of cards may be misdetected as memory cards.

There is another issue to consider when handling memory cards that donot have CIS information. At startup time, the PCMCIA package triesto use the first detected card to determine what memory regions areusable for PCMCIA. The memory scan can be fooled if that card is asimple memory card. If you plan to use memory cards often, it is bestto limit the memory windows in /etc/pcmcia/config.opts toknown-good regions.

The memory_cs driver uses a heuristic to guess the capacity of these cards. The heuristic does not work for write protected cards,and may make mistakes in some other cases as well. If a card ismisdetected, its size should then be explicitly specified when using commands such as dd or mkfs. The memory_cs module alsohas a parameter for overriding the size detection. See the man page.

Memory device parameters

The following parameters can be specified in memory.opts:

DO_FSTAB

A boolean (y/n) setting: specifies if an entry should be added to/etc/fstab for this device. DO_FSCK

A boolean (y/n) setting: specifies if the filesystem should be checkedbefore being mounted, with ``fsck -Ta".

DO_MOUNT

A boolean (y/n) setting: specifies if this device should beautomatically mounted at card insertion time.

FSTYPE, OPTS, MOUNTPT

The filesystem type, mount options, and mount point to be used for thefstab entry and/or mounting the device.

NO_CHECK, NO_FUSER

Boolean (y/n) settings for card eject policy. If NO_CHECK istrue, then ``cardctl eject'' will shut down a device even if it busy. If NO_FUSER is true, then the script will not try tokill processes using an ejected device.

Here is an example of a script that will automatically mount memorycards based on which socket they are inserted into:

```
case "$ADDRESS" in
*,0,0)
    # Mount filesystem, but don't update /etc/fstab
    DO_FSTAB="n" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="ext2" ; OPTS=""
    MOUNTPT="/mem0"
    ;;
*,1,0)
    # Mount filesystem, but don't update /etc/fstab
    DO_FSTAB="n" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="ext2" ; OPTS=""
    MOUNTPT="/mem1"
    ;;
esac
```

Using linear flash memory cards

The following information applies only to so-called ``linear flash''memory cards. Many flash cards, including all SmartMedia andCompactFlash cards, actually include circuitry to emulate an IDE diskdevice. Those cards are thus handled as IDE devices, not memorycards.

There are two major formats for flash memory cards: the FTLor ``flash translation layer" style, and the MicrosoftFlash File System. The FTL format is generally moreflexible because it allows any ordinary high-level filesystem (ext2,ms-dos, etc) to be used on a flash card as if it were an ordinary diskdevice. The FFS is a completely different filesystem type. Linuxcannot currently handle cards formated with FFS.

To use a flash memory card as an ordinary disk-like block device, first create an FTL partition on the device with the>cdx>ftl_ format>/cdx> command. This layer hides thedevice-specific details of flash memory programming and make the cardlook like a simple block device. For example:

ftl_format -i /dev/mem0c0c

Note that this command accesses the card through the ``raw" memorycard interface. Once formatted, the card can be accessed as anordinary block device via the ftl_cs driver. For example:

mke2fs /dev/ftl0c0 mount -t ext2 /dev/ftl0c0 /mnt

Device naming for FTL devices is tricky. Minor device numbers havethree parts: the card number, the region number on that card, andoptionally, the partition within that region. A region can either betreated as a single block device with no partition table (like afloppy), or it can be partitioned like a hard disk device. The``ftl0c0" device is card 0, common memory region 0, the entireregion. The ``ftl0c0p1" through ``ftl0c0p4" devices are primarypartitions 1 through 4 if the region has been partitioned.

Configuration options for FTL partitions can be given inftl.opts, which is similar in structure to memory.opts.The device address passed to ftl.opts consists of three or fourfields: the scheme, the socket number, the region number, andoptionally, the partition number. Most flash cards have just oneflash memory region, so the region number will generally always bezero.

Intel Series 100 flash cards use the first 128K flash block to storethe cards' configuration information. To prevent accidental erasureof this information, ftl_format will automatically detect thisand skip the first block when creating an FTL partition.

PCMCIA ATA/IDE card drives

ATA/IDE drive support is based on the regular kernel IDE driver. This includes SmartMedia and CompactFlash devices: these flash memory cardsare set up so that they emulate an IDE interface. The PCMCIA-specificpart of the driver is ide_cs. Be sure to use cardctl orcardinfo to shut down an ATA/IDE card before ejecting it, as the driver has not been made ``hot-swap-proof".

The device addresses passed to ide.opts consist of either threeor four fields: the current scheme, the socket number, the drive'sserial number, and an optional partition number. The ide_infocommand can be used to obtain an IDE device's serial number. As withSCSI devices, ide.opts is first called for the entire device. Ifide.opts returns a list of partitions in the PARTSvariable, the script will then be called for each partition.

ATA/IDE fixed-disk device parameters

The following parameters can be specified in ide.opts:

DO_FSTAB

A boolean (y/n) setting: specifies if an entry should be added to/etc/fstab for this device. DO_FSCK

A boolean (y/n) setting: specifies if the filesystem should be checkedbefore being mounted, with ``fsck -Ta".

DO_MOUNT

A boolean (y/n) setting: specifies if this device should beautomatically mounted at card insertion time.

FSTYPE, OPTS, MOUNTPT

The filesystem type, mount options, and mount point to be used for thefstab entry and/or mounting the device.

NO_CHECK, NO_FUSER

Boolean (y/n) settings for card eject policy. If NO_CHECK istrue, then ``cardctl eject" will shut down a device even if it busy. If NO_FUSER is true, then the script will not try tokill processes using an ejected device.

Here is an example ide.opts file to mount the first partitionof any ATA/IDE card on /mnt.

```
case "$ADDRESS" in
*,*,*,1)
DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
FSTYPE="msdos"
OPTS=""
MOUNTPT="/mnt"
;;
*,*,*)
PARTS="1"
;;
esac
```

Diagnosing problems with ATA/IDE adapters

- * An IO port conflict may cause the IDE driver to misdetect the drivegeometry and report ``INVALID GEOMETRY: 0 PHYSICAL HEADS?". Tofix, try excluding the selected IO port range in/etc/pcmcia/config.opts.
- * Some IDE drives violate the PCMCIA specification by requiring moretime to spin up than the maximum allowed card setup time. This mayresult in ``timed out during reset" messages at card detect time.Adjust the unreset_delay and/or unreset_limit parameters forthe pcmcia_core module to give a drive more time to spin up; seethe pcmcia_core man page for parameter details. For example:

CORE_OPTS="unreset_delay=400"

- * To use an ATA/IDE CD-ROM device, your kernel must be compiled withCONFIG_BLK_DEV_IDECD enabled. This will normally be the case forstandard kernels, however it is something to be aware of if youcompile a custom kernel.
- * A common error when using IDE drives is try to mount the wrong devicefile. Generally, you want to mount a partition of the device, not theentire device (i.e., /dev/hde1, not /dev/hde).
- * The Linux IDE driver may have trouble configuring certainremovable-media drives if no media is present at insertion time. TheIBM Portable DriveBay has this problem.
- * Some kernels will report a pair of ``drive_cmd" errors at insertiontime. These errors can be ignored: they pop up when a removable IDEdevice does not accept the IDE ``door lock" command.

Multifunction cards

A single interrupt can be shared by several drivers, such as theserial driver and an ethernet driver: in fact, the PCMCIAspecification requires all card functions to share the same interrupt.Normally, all card functions are available without having to swapdrivers. All Linux kernels support this kind of interrupt sharing.

Simultaneous use of two card functions is ``tricky" and varioushardware vendors have implemented interrupt sharing in their ownincompatible (and sometimes proprietary) ways. The drivers for somecards (Ositech Jack of Diamonds, 3Com 3c562 and related cards, Linksyscards) properly support simultaneous access, but others (olderMegahertz cards in particular) do not. If you have trouble using acard with both functions active, try using each function in isolation.That may require explicitly doing an ``ifconfig down" to shutdown a network interface and use a modem on the same card.

Advanced topics

Resource allocation for PCMCIA devices

In theory, it should not really matter which interrupt is allocated towhich device, as long as two devices are not configured to use thesame interrupt. In /etc/pcmcia/config.opts you'll finda place for excluding interrupts that are used by non-PCMCIA devices.

Similarly, there is no way to directly specify the I/O addresses for acard to use. The /etc/pcmcia/config.opts file allowsyou to specify ranges of ports available for use by any card, or toexclude ranges that conflict with other devices.

After modifying /etc/pcmcia/config.opts, you can reinitializecardmgr with ``kill -HUP". The interrupt used to monitor card status changes is chosenby the low-level socket driver module (i82365 or tcic)before cardmgr parses /etc/pcmcia/config, so it is notaffected by changes to this file. To set this interrupt, use thecs_irq= option when the socket driver is loaded, by settingthe PCIC_OPTS variable in /etc/rc.d/rc.pcmcia.

All the client card drivers have a parameter called irq_list forspecifying which interrupts they may try to allocate.These driver options should be set inyour /etc/pcmcia/config file. For example:

```
device "serial_cs"
  module "serial_cs" opts "irq_list=8,12"
   ...
```

would specify that the serial driver should only use irq 8 or irq 12.Regardless of irq_ list settings, Card Services will neverallocate an interrupt that is already in use by another device, or an interrupt that is excluded in the config file.

PCI interrupt configuration problems and solutions

Most of the following discussion applies to 2.2 and earlier kernels. With 2.4 and later kernels, the PCI subsystem has more completeresponsibility for PCI interrupt management. The following tips mayhelp diagnose a problem, though some workarounds described here maynot be available.

An overview of PCI interrupt routing issues

Each PCI slot has four PCI interrupt pins, INTA through INTD. Singlefunction devices will only use the INTA pin; multifunction devices mayuse multiple INT pins. On the processor side, on x86 single processorsystems, incoming hardware interrupts are directed to interruptrequests (irq's) numbered 0..15. The PCI interrupt router, usuallypart of the PCI-to-ISA host bridge, determines how incoming PCIInterrupts are mapped to CPU irq numbers. Most modern bridge chipshave several PCI interrupt inputs, known as PIRQ1, PIRQ2, etc, each ofwhich can be routed to any CPU irq number. So we might have somethinglike:

PCI slot 1 INTA --> router PIRQ1 --> CPU irq 9 PCI slot 1 INTB --> router PIRQ2 --> CPU irq 10 Multiple INT pins are often connected to the same PIRQ pin. Usually,the connections from INT pins to PIRQ pins are arranged to spreadinstalled devices out as much as possible, to give the OS the mostflexibility for choosing how interrupts are shared. The mapping frombridge PIRQ pins to CPU irq numbers can be obtained by readingregisters in the interrupt router. The mapping from INT pins to therouter's PIRQ pins, however, depends on how the board designer decided connect things up, and cannot be directly determined by driversoftware.

For most PCI devices, the OS does not need to understand the interruptrouter details. Each PCI device has a configuration register, the PCIInterrupt Line Register, that the BIOS initializes with theappropriate CPU irq number for that device. Unfortunately, the BIOSgenerally will not configure PCI interrupts for CardBus bridge devices.

The PCI BIOS's Interrupt Routing Table is a data structure thatcontains information about the mapping from PCI INT pins to the PIRQpins on the PCI interrupt router. The routing information in thetable is stored in a somewhat unhelpful form, however. For eachdevice's INT pins, the table specifies a ``link value". Allinterrupts with the same link value are wired to the same PIRQ pin; however, the meaning of the link values is defined by the chipsetvendor.

Several tools are available for examining PCI interrupt routinginformation:

Ispci, /proc/pci

These will show you resource information (including interruptassignments, where they are known) for all your PCI devices.

dump_pirq

This is in the debug-tools directory of the PCMCIA sourcedistribution. It dumps the contents of your PCI interrupt routingtable, if available. It also scans for known interrupt routers anddumps their current interrupt steering settings.

Several PCMCIA module parameters affect PCI interrupt routing:

pcmcia_core module: cb_pci_irq=n

This option specifies one interrupt number to be used to program thePCI interrupt router for all CardBus sockets that do not already havean interrupt assignment. It only has any effect on systems that have aPCI irq routing table, and a known interrupt router.

i82365 module: irq_mode=n

Most CardBus bridges offer several methods for delivering interrupts to the host. The i82365 module by default assumes that a bridge candeliver both PCI and ISA interrupts, since this is normal for laptops. A setting of ``irq_mode=0" can be used to force a bridge to use onlyPCI interrupts. See the man page for the i82365 module for adescription of what other values mean for different bridge types.

i82365 module: irq_list=n,n,...

This parameter lists which ISA interrupt(s) can be used for PCMCIA.If no ISA interrupts are available, specify ``irq_list=0". Notethat ``irq_mode=0" implies ``irq_list=0".

i82365 module: pci_irq_list=n,n,...

This option specifies a list of PCI interrupt numbers to use forCardBus sockets. It differs from cb_pci_irq, because it does notactually program the PCI interrupt router; it can be used when youknow the PCI interrupts are already set up a certain way, even if youdo not know how the router works.

If you are having problems that you think may be related to PClinterrupt configuration, you should first verify that you have areasonably current PCMCIA driver package. Also carefully look at thestartup messages when the PCMCIA kernel modules are loaded. Youshould see something like:

Linux PCMCIA Card Services 3.1.18 kernel build: 2.2.14-5.0 #1 Tue May 9 10:44:24 PDT 2000 options: [pci] [cardbus] [apm] [pnp] PCI routing table version 1.0 at 0xfdf30 Intel PCIC probe: TI 1125 rev 02 PCI-to-CardBus at slot 00:07, mem 0x20000000 host opts [0]: [ring] [serial pci & irq] [pci irq 11] ... host opts [0]: [ring] [serial pci & irq] [pci irq 11] ... ISA irqs (scanned) = 3,4,7 PCI status changes

The ``PCI routing table" message indicates that a valid routingtable was found. The ``host opts" lines indicate the interruptdelivery mode and whether or not a PCI interrupt could be determined for each socket. And the final line indicates the results of the scanfor available interrupts.

CardBus bridge is not detected by the PCI BIOS

Symptoms:

- * Intel PCIC probe: not found.
- * The bridge does not show up in Ispci or in /proc/pci.

The Lucent/SCM PCI-to-CardBus adapters seem to confuse the PCI BIOS onsome older systems. Lucent says that this card is only supportedon systems that have a BIOS that supports the PCI 2.2 specification,or are PC99 compliant. Some older systems will not detect the Lucentcard at all, and if the system can't detect it, the Linux driverscannot use it. The only possible resolutions are a BIOS upgrade, orusing a different motherboard or CardBus adapter.

PCI interrupt delivery problems

Symptoms:

- * Cards seem to be configured correctly, but do not work.
- * /proc/interrupts shows a count of 0 for interruptsassigned to PCMCIA drivers.

CardBus bridges usually support two types of interrupts, PCI and ISA.Partly for historical reasons, it has become conventional to use PClinterrupts for signaling card insertion and removal events, and forCardBus card interrupts; and ISA interrupts for 16-bit cards. Sinceversion 3.1.9, this is the scheme that the Linux PCMCIA system willuse by default. Most CardBus bridges support multiple methods fordelivering interrupts to the host CPU. Methods include ``parallel''interrupts, where each supported irq has a dedicated pin on thebridge; various serial interrupt protocols, where one or two pins areused to communicate with an interrupt controller; and hybrids, wherePCI interrupts might be signalled using dedicated pins, while ISAinterrupts are delivered via a serial controller.

In general, it is the responsibility of the BIOS to program a bridgefor the appropriate interrupt delivery method. However, there aresystems that do this incorrectly, and in some cases, there is no wayfor software to safely detect the correct delivery method. Thei82365 module reports the bridge mode at startup time, and has aparameter, irq_mode, that can be used to reconfigure it. Not allbridges support this parameter, and the meaning of irq_modedepends on the bridge type. See the i82365 man page for adescription of what values are supported by your bridge. In somecases, a bridge may function correctly in more than one interruptmode.

Most PCMCIA card readers that fit in a PCI bus slot only provide PCIinterrupt routing. The Linux drivers assume that all bridges have ISAinterrupt capability, since that is generally correct on laptops. With a card reader, it will generally be necessary to use theirq_mode parameter to specify a ``PCI only" interrupt deliverymode; the value of the parameter depends on the bridge type, so checkthe i82365 man page. A few PCI card readers require anirq_mode that permits ISA interrupts, but those interrupts arenot actually connected; in that case, use ``irq_list=0".

Check the system log and verify that the CardBus bridge has a PClinterrupt assignment. If it does not, then resolve that problemfirst, then return here if the symptoms persist. Next, experimentwith different values for the irq_mode parameter.

No PCI interrupt assignment; valid routing table

Symptoms:

- * The Intel PCIC probe reports ``no pci irq" for each socket.
- * There is a routing table, and the router type is supported.

When a routing table is present, the pcmcia_core module will tryto automatically configure the PCI interrupt router, but only does sowhen it has a safe and unambiguous choice for what PCI interrupt touse. If there are several valid choices, then you must use the``cb_pci_irq=...'' option to specify which interrupt to assign.Your best bet is to pick the most lightly used interrupt that isalready assigned to another PCI device.

Moving the card to another slot sometimes offers a quick solution. If that slot shares its interrupt with an already-configured device, then the PCMCIA drivers will have no trouble figuring out the assignment.

No PCI interrupt assignment; unknown interrupt router

Symptoms:

- * The Intel PCIC probe reports ``no pci irq" for each socket.
- * There is a routing table, but the router is an unknown type.

Adding support for a new interrupt router is tricky but not a bigjob. First determine, from a datasheet, how your interrupt routersteers PCI interrupts. Then, see if you can guess the meaning of thelink values from the output of dump_pirq. Usually this isreasonably obvious. Most routers have four PIRQ pins, and the linkvalues might be something like 1,2,3,4, or 0x10,0x18,0x20,0x28, or0x60,0x61,0x62,0x63. The values are usually chosen so that they canbe easily converted to the location of the appropriate interruptsteering register. Finally, add small functions tomodules/pci_fixup.c.to f55 get/set the interrupt steeringinformation for this router, using the other routers as examples.

No PCI interrupt assignment; no routing table

Symptoms:

- * The Intel PCIC probe reports ``no pci irq" for each socket.
- * No interrupt routing table is found.

Without an interrupt routing table, we cannot tell how interrupts from the CardBus bridge are directed to CPU irq numbers. All hope is notlost: you may be able to guess the PCI interrupt assignment and usethe ``pci_irq_list=..." option to pass this information to thei82365 module. Good guesses might include the interrupt(s)assigned to other PCI devices, the interrupt(s) used under Windows, orany other interrupts that are unaccounted for.

You may also want to experiment with putting the adapter in differentPCI slots, for each pci_irq_list you try. You are trying to finda slot that shares its interrupt with an already-configured device, and might need to try several slots to find one.

How can I have separate device setups for home and work?

This is fairly easy using ``scheme" support.Use two configuration schemes, called ``home" and ``work". Here isan example of a network.opts script with scheme-specificsettings:

```
case "$ADDRESS" in
work,*,*,*)
    # definitions for network card in work scheme
    ...
;;
home,*,*,*|default,*,*,*)
    # definitions for network card in home scheme
    ...
;;
esac
```

The first part of a device address is always the configurationscheme. In this example, the second ``case" clause will select forboth the ``home" and ``default" schemes. So, if the scheme is unsetfor any reason, it will default to the ``home" setup.

Now, to select between the two sets of settings, run either:

cardctl scheme home

or

```
cardctl scheme work
```

The cardctl command does the equivalent of shutting down all yourcards and restarting them. The command can be safely executed whetheror not the PCMCIA system is loaded, but the command may fail if youare using other PCMCIA devices at the time (even if theirconfigurations are not explicitly dependent on the scheme setting). To find out the current scheme setting, run:

cardctl scheme

By default, the scheme setting is persistent across boots. This canhave undesirable effects if networking is initialized for the wrongenvironment. Optionally, you can set the initial scheme value with the SCHEME startup option (see Startup options

To save even more keystrokes, schemes can be specified in lilo'sconfiguration file. For instance, you could have:

root = /dev/hda1 read-only image = /boot/vmlinuz label = home append = "SCHEME=home" image = /boot/vmlinuz label = work append = "SCHEME=work"

Typing ``home" or ``work" at the boot prompt would then boot intothe appropriate scheme.

Booting from a PCMCIA device

Having the root filesystem on a PCMCIA device is tricky because theLinux PCMCIA system is not designed to be linked into the kernel. Itscore components, the loadable kernel modules and the user mode cardmgrdaemon, depend on an already running system. The kernel's``initrd" facility works around this requirement byallowing Linux to boot using a temporary ram disk as a minimal rootimage, load drivers, and then re-mount a different root filesystem. The temporary root can configure PCMCIA devices and then re-mount aPCMCIA device as root.

The initrd image absolutely must reside on a bootable device: thisgenerally cannot be put on a PCMCIA device. This is a BIOSlimitation, not a kernel limitation. It is useful here to distinguishbetween ``boot-able'' devices (i.e., devices that can be booted), and ``root-able'' devices (i.e., devices that can be mounted as root). ``Boot-able'' devices are determined by the BIOS, and are generallylimited to internal floppy and hard disk drives.

``Root-able"devices are any block devices that the kernel supports once it hasbeen loaded. The initrd facility makes more devices ``root-able",not ``boot-able".

Some Linux distributions will allow installation to a device connected a PCMCIA SCSI adapter, as an unintended side-effect of theirsupport for installs from PCMCIA SCSI CD-ROM devices. However, atpresent, no Linux installation tools support configuring anappropriate ``initrd" to boot Linux with a PCMCIA root filesystem.Setting up a system with a PCMCIA root thus requires that you useanother Linux system to create the ``initrd" image. If another Linuxsystem is not available, another option would be to temporarilyinstall a minimal Linux setup on a non-PCMCIA drive, create an initrdimage, and then reinstall to the PCMCIA target.

The Linux Bootdisk-HOWTO has some general information about setting upboot disks but nothing specific to initrd. The main initrd documentis included with recent kernel source code distributions, inlinux/Documentation/initrd.txt. Before beginning, you shouldread this document. A familiarity with lilo is also helpful.Using initrd also requires that you have a kernel compiled withCONFIG_BLK_DEV_RAM and Page 44 of 55 CONFIG_BLK_DEV_INITRD enabled.

This is an advanced configuration technique, and requires a high levelof familiarity with Linux and the PCMCIA system. Be sure to read allthe relevant documentation before starting. The following cookbookinstructions should work, but deviations from the examples willquickly put you in uncharted and ``unsupported" territory, and youwill be on your own.

This method absolutely requires that you use a PCMCIA driver release of 2.9.5 or later. Older PCMCIA packages or individual components will not work in the initrd context. Do not mix components from different releases.

The pcinitrd helper script

The pcinitrd script creates a basic initrd image for booting witha PCMCIA root partition. The image includes a minimal directoryheirarchy, a handful of device files, a few binaries, sharedlibraries, and a set of PCMCIA driver modules. When invokingpcinitrd, you specify the driver modules that you want to beincluded in the image. The core PCMCIA components, pcmcia_coreand ds, are automatically included.

As an example, say that your laptop uses an i82365-compatiblehost controller, and you want to boot Linux with the root filesystemon a hard drive attached to an Adaptec SlimSCSI adapter. You couldcreate an appropriate initrd image with:

pcinitrd -v initrd pcmcia/i82365.o pcmcia/aha152x_cs.o

To customize the initrd startup sequence, you could mount the imageusing the ``loopback" device with a command like:

mount -o loop -t ext2 initrd /mnt

and then edit the linuxrc script. The configuration fileswill be installed under /etc in the image, and can also becustomized. See the man page for pcinitrd for more information.

Creating an initrd boot floppy

After creating an image with pcinitrd, you can create a bootfloppy by copying the kernel, the compressed initrd image, and a fewsupport files for lilo to a clean floppy. In the followingexample, we assume that the desired PCMCIA root device is/dev/sda1:

```
mke2fs /dev/fd0
mount /dev/fd0 /mnt
mkdir /mnt/etc /mnt/boot /mnt/dev
cp -a /dev/fd0 /dev/sda1 /mnt/dev
cp [kernel-image] /mnt/vmlinuz
cp /boot/boot.b /mnt/boot/boot.b
gzip > [initrd-image] > /mnt/initrd
```

Create /mnt/etc/lilo.conf with the contents:

```
boot=/dev/fd0
compact
image=/vmlinuz
label=linux
initrd=/initrd
read-only
root=/dev/sda1
```

Finally, invoke lilo with:

lilo -r /mnt

When lilo is invoked with -r, it performs all actionsrelative to the specified alternate root directory. The reason forcreating the device files under /mnt/dev was that lilowill not be able to use the files in /dev when it is runningin this alternate-root mode.

Installing an initrd image on a non-Linux drive

One common use of the initrd facility would be on systems where the internal hard drive is dedicated to another operating system. The Linux kernel and initrd image can be placed in a non-Linux partition, and lilo or LOADLIN can be set up to boot Linux from these images.

Assuming that you have a kernel has been configured for theappropriate root device, and an initrd image created on anothersystem, the easiest way to get started is to boot Linux usingLOADLIN, as:

LOADLIN >kernel> initrd=>initrd-image>

Once you can boot Linux on your target machine, you could theninstall lilo to allow booting Linux directly.For example, say that /dev/hda1 is the non-Linux targetpartition and /mnt can be used as a mount point. First,create a subdirectory on the target for the Linux files:

mount /dev/hda1 /mnt
mkdir /mnt/linux
cp [kernel-image] /mnt/linux/vmlinuz
cp [initrd-image] /mnt/linux/initrd

In this example, say that /dev/sda1 is the desired Linux rootpartition, a SCSI hard drive mounted via a PCMCIA SCSI adapter. Toinstall lilo, create a lilo.conf file with the contents:

```
boot=/dev/hda
map=/mnt/linux/map
compact
image=/mnt/linux/vmlinuz
label=linux
root=/dev/sda1
initrd=/mnt/linux/initrd
read-only
other=/dev/hda1
table=/dev/hda
label=windows
```

The boot= line says to install the boot loader in the master bootrecord of the specified device. The root= line identifies the desired root filesystem to be used after loading the initrd image, andmay be unnecessary if the kernel image is already configured this way. The other= section is used to describe the other operating systeminstalled on /dev/hda1.

To install lilo in this case, use:

lilo -C lilo.conf

Note that in this case, the lilo.conf file uses absolute pathsthat include /mnt. I did this in the example because the targetfilesystem may not support the creation of Linux device files for theboot= and root= options.

Dealing with unsupported cards

Configuring unrecognized cards

Assuming that your card is supported by an existing driver, allthat needs to be done is to add an entry to/etc/pcmcia/config to tell cardmgr how to identify the card,and which driver(s) need to be linked up to this card. Check the manpage for pcmcia for more information about the config file format.If you insert an unknown card, cardmgr will normally record someidentification information in the system log that can beused to construct the config entry. This information can also bedisplayed with the ``cardctl ident" command.

Here is an example of how cardmgr will report an unsupported card in the system log:

cardmgr[460]: unsupported card in socket 1
cardmgr[460]: product info: "MEGAHERTZ", "XJ2288", "V.34 PCMCIA MODEM"
cardmgr[460]: manfid: 0x0101, 0x1234 function: 2 (serial)

The corresponding entry in /etc/pcmcia/config would be:

card "Megahertz XJ2288 V.34 Fax Modem"
 version "MEGAHERTZ", "XJ2288", "V.34 PCMCIA MODEM"
 bind "serial_cs"

or using the more compact product ID codes:

```
card "Megahertz XJ2288 V.34 Fax Modem"
  manfid 0x0101, 0x1234
  bind "serial_cs"
```

You can use ``*" to match strings that don't need to match exactly,like version numbers. When making new config entries, be carefulto copy the strings exactly, preserving case and blank spaces. Also be sure that the config entry has the same number of strings asare reported in the log file.

Beware that you can specify just about any driver for a card, but ifyou're just shooting in the dark, there is not much reason to expect this to be productive. You may get lucky and find that your card issupported by an existing driver. However, the most likely outcome is that the driver won't work, and may have unfortunate side effects like locking up your system. Unlike most ordinary device drivers, which probe for an appropriate card, the probe for a PCMCIA device isdone by cardmgr, and the driver itself may not do much validation before attempting to communicate with the device.

After editing /etc/pcmcia/config, you can signal cardmgrto reload the file with:

kill -HUP `cat /var/run/cardmgr.pid`

If you do set up an entry for a new card, please send me a copy sothat I can include it in the standard config file.

Adding support for an NE2000-compatible ethernet card

Before you begin: this procedure will only work for simple 16-bitethernet cards. Multifunction cards (i.e., ethernet/modem combocards) have an extra layer of complexity regarding how the twofunctions are integrated, and generally cannot be supported withoutobtaining some configuration information from the card vendor. Using the following procedure for a multifunction card will not be productive.

First, see if the card is already recognized by cardmgr. Somecards not listed in SUPPORTED.CARDS are actually OEM versions ofcards that are supported. If you find a card like this, let me knowso I can add it to the list.

If your card is not recognized, follow the instructions in theConfiguring unrecognized cards

If the pcnet_cs driver says that it is unable to determine yourcard's hardware ethernet address, then edit your new config entry tobind the card to the memory card driver, memory_cs.Restart cardmgr to use the new updated config file.You will need to know your card's hardware ethernet address. Thisaddress is a series of six two-digit hex numbers, often printed on thecard itself. If it is not printed on the card, you may be able to usea DOS driver to display the address. In any case, once you know it,run:

dd if=/dev/mem0a count=20 | od -Ax -t x1

and search the output for your address. Only the even bytes aredefined, so ignore the odd bytes in the dump. Record the hex offset of thefirst byte of the address. Now, edit clients/pcnet_cs.c andfind the hw_info structure. You'll need to create a new entryfor your card. The first field is the memory offset. Thenext three fields are the first three bytes of the hardware address. The final field contains some flags for specific card features; tostart, try setting it to 0.

After editing pcnet_cs.c, compile and install the new module.Edit /etc/pcmcia/config again, and change the card bindingfrom memory_cs to pcnet_cs. Follow the instructions forreloading the config file, and you should be all set. Please send mecopies of your new hw_info and config entries.

If you can't find your card's hardware address in the hex dump, as amethod of last resort, it is possible to ``hard-wire" the address whenthe pcnet_cs module is initialized. Edit/etc/pcmcia/config.opts and add a hw_addr= option, likeso:

module "pcnet_cs" opts "hw_addr=0x00,0x80,0xc8,0x01,0x02,0x03"

Substitute your own card's hardware address in the appropriate spot,of course. Beware that if you've gotten this far, it is very unlikelythat your card is genuinely NE2000 compatible. In fact, I'm not sureif there are any cards that are not handled by one of the firsttwo methods.

PCMCIA floppy interface cards

The PCMCIA floppy interface used in the Compaq Aero and a few otherlaptops is not yet supported by this package. The snag in supporting the Aero floppy is that the Aero seems to use a customized PCMCIA controller to support DMA to the floppy. Withoutknowing exactly how this is done, there isn't any way to implement support under Linux.

If the floppy adapter card is present when an Aero is booted, the AeroBIOS will configure the card, and Linux will identify it as a normalfloppy drive. When the Linux PCMCIA drivers are loaded, they willnotice that the card is already configured and attached to a Linuxdriver, and this socket will be left alone. So, the drive can be used if it is present at boot time, but the card is not hot swappable.

Debugging tips and programming information

Submitting useful problem reports

The best way to submit reports is to use the online pcmcia-cs forumsor the bug tracker at SourceForge. That way, other people can seecurrent problems (and fixes or workarounds, if available). Here aresome things that should be included in all bug reports:

- * Your system brand and model.
- * All PCMCIA card(s) you are using.
- * Your Linux kernel version (i.e., ``uname -rv"), and PCMCIAdriver version (i.e., ``cardctl -V").
- * Output of 'Ispci -v'
- * Any changes you have made to the startup files in/etc/pcmcia, or to the PCMCIA startup script.
- * All PCMCIA-related messages in your system log file. That includes startup messages, and messages generated when yourcards are configured.

All the PCMCIA modules and the cardmgr daemon send statusmessages to the system log. These will usually end up somewhere like/var/log/messages or /var/log/daemon.log. Thesefiles should be the first place to look when tracking down a problem. When submitting a bug report, always include the relevant contents of these files. If you are having trouble finding your system messages, check /etc/syslog.conf to see how different classes of messages are handled.

Before submitting a bug report, please check to make sure that you areusing an up-to-date copy of the driver package. While it is somewhatgratifying to read bug reports for things I've already fixed, it isn'ta particularly constructive use of my time.

If you do not have web access, bug reports can be sent to me atdahinds@users.sourceforge.net. However, I prefer thatbug reports be posted to the pcmcia-cs SourceForge site, so that theycan be seen by others.

Interpreting kernel trap reports

If your problem involves a kernel fault, the register dump from thefault is only useful if you can translate the fault address, EIP, tosomething meaningful. Recent versions of klogd attempt totranslate fault addresses based on the current kernel symbol map, butthis may not work if the fault is in a module, or if the problem issevere enough that klogd cannot finish writing the faultinformation to the system log.

If a fault is in the main kernel, the fault address can be looked upin the System.map file. This may be installed in/System.map or /boot/System.map. If a fault is in amodule, the nm command gives the same information, however, thefault address has to be adjusted based on the module's load address.Let's say that you have the following kernel fault:

```
Unable to handle kernel NULL pointer dereference
current->tss.cr3 = 014c9000, %cr3 = 014c9000
*pde = 00000000
Oops: 0002
CPU: 0
EIP: 0010:[>c2026081>]
EFLAGS: 00010282
```

The fault address is 0xc2026081. Looking at System.map, wesee that this is past the end of the kernel, i.e., is in a kernelmodule. To determine which module, check the output of ``ksyms -m | sort":

Address	Symbol	Defined by
c200d000	(35k)	[pcmcia_core]
c200d10c	register_ss_entry	[pcmcia_core]
c200d230	unregister_ss_entry	[pcmcia_core]
c2026000	(9k)	[3c574_cs]
c202a000	(4k)	[serial_cs]

So, 0xc2026081 is in the 3c574_cs module, and is at an offset of 0x0081 from the start of the module. We cannot look up this offset in3c574_cs.o yet: when the kernel loads a module, it inserts aheader at the module load address, so the real start of the module isoffset from the address shown in ksyms. The size of the headervaries with kernel version: to find out the size for your kernel,check a module that exports symbols (like pcmcia_core above), andcompare a symbol address with nm output for that symbol. In thisexample, register_ss_entry is loaded at an offset of 0xc200d10c -0xc200d000 = 0x010c, while ``nm pcmcia_core.o'' shows the offsetas 0x00c0, so the header size is 0x010c - 0x00c0 = 0x004c bytes.

Back to 3c574_cs, our fault offset is 0x0081, and subtracting the0x004c header, the real module offset is 0x0035. Now looking at``nm 3c574_cs.o | sort", we see:

0000002c d if_names 0000002c t tc574_attach 00000040 d mii_preamble_required 00000041 d dev_info So, the fault is located in tc574_attach().

In this example, the fault did not cause a total system lockup, soksyms could be executed after the fault happened. In othercases, you may have to infer the module load addresses indirectly. The same sequence of events will normally load modules in the sameorder and at the same addresses. If a fault happens when a certaincard is inserted, get the ksyms output before inserting the card, or with a different card inserted. You can also manually load thecard's driver modules with insmod and run ksyms before inserting the card.

For more background, see ``man insmod", ``man ksyms", and``man klogd". In the kernel source tree,Documentation/oops-tracing.txt is also relevant. Here are afew more kernel debugging hints:

- * Depending on the fault, it may also be useful to translateaddresses in the ``Call Trace'', using the same procedure as for themain fault address.
- ^{*} To diagnose a silent lock-up, try to provoke the problem with Xdisabled, since kernel messages sent to the text console will not bevisible under X.
- * If you kill klogd, most kernel messages will be echoeddirectly on the text console, which may be helpful if the problemprevents klogd from writing to the system log.
- * To cause all kernel messages to be sent to the console, for 2.2and later kernels, if /proc/sys/kernel/printk exists, do:

echo 8 > /proc/sys/kernel/printk

* The key combination >RightAlt>>ScrLk> prints aregister dump on the text console. This may work even if the systemis otherwise completely unresponsive, and the EIP address can beinterpreted as for a kernel fault.

* For 2.2 and later kernels configured withCONFIG_MAGIC_SYSRQ enabled, various emergency functions areavailable via special >Alt>>SysRq> key combinations,documented in Documentation/sysrq.txt in the kernel sourcetree.

Low level PCMCIA debugging aids

The PCMCIA modules contain a lot of conditionally-compiled debuggingcode. Most of this code is under control of the PCMCIA_DEBUGpreprocessor define. If this is undefined, debugging code willnot be compiled. If set to 0, the code is compiled but inactive.Larger numbers specify increasing levels of verbosity. Each modulebuilt with PCMCIA_DEBUG defined will have an integer

parameter,pc_debug, that controls the verbosity of its output. Thiscan be adjusted when the module is loaded, so output can be controlledon a per-module basis without recompiling.

Your default configuration for syslogd may discard kerneldebugging messages. To ensure that they are recorded, edit/etc/syslog.conf to ensure that ``kern.debug'' messagesare recorded somewhere. See ``man syslog.conf'' for details.

There are a few register-level debugging tools in thedebug_tools/ subdirectory of the PCMCIA distribution. Thedump_tcic and dump_i365 utilities generate register dumpsfor ISA-to-PCMCIA controllers. In 3.1.15 and later releases,dump_i365 is replaced by dump_exca, which is similar butalso works for PCI-to-CardBus bridges. Also new in 3.1.15 for CardBusbridges is the dump_cardbus tool, which interprets theCardBus-specific registers. These are all most useful if you haveaccess to a datasheet for the corresponding controller chip. Thedump_cis utility (dump_tuples in pre-3.0.2 distributions)lists the contents of a card's CIS (Card Information Structure), anddecodes most of the important bits. And the dump_cisreg utilitydisplays a card's local configuration registers.

The memory_cs memory card driver is also sometimes useful fordebugging problems with 16-bit PC Cards. It can be bound to any card,and does not interfere with other drivers. It can be used to directlyaccess any card's attribute memory or common memory. Similarly forCardBus cards, the memory_cb driver can be bound to any 32-bitcard, to give direct access to that card's address spaces. See theman pages for more information.

/proc/bus/pccard

On 2.2 and later kernels, the PCMCIA package will create a treeof status information under >cdx>/proc/bus/pccard>/cdx>.Much of the information can only be interpreted using the data sheetsfor the PCMCIA host controller. Its contents may depend on how thedrivers were configured, but may include all or some of the following:

/proc/bus/pccard/{irq,ioport,memory}

If present, these files contain resource allocation information to supplement the normal kernel resource tables. Recent versions of the PCMCIA system may obtain additional resource information from the Plug and Play BIOS if configured to do so.

/proc/bus/pccard/drivers

In recent releases, this lists all currently loaded PCMCIA clientdrivers. Unlike /proc/modules, it also lists drivers thatmay be statically linked into the kernel.

/proc/bus/pccard/*/info

For each socket, describes that socket's host controller and itscapabilities.

/proc/bus/pccard/*/exca

This contains a dump of a controller's ``ExCA'' Inteli82365sl-compatible register set. /proc/bus/pccard/*/{pci,cardbus}

For CardBus bridges, a dump of the bridge's PCI configuration space, and a dump of the bridge's CardBus configuration registers.

Writing Card Services drivers for new cards

The Linux PCMCIA Programmer's Guide is the best documentation for theclient driver interface. The latest version is always available fromprojects.sourceforge.net in /pub/pcmcia-cs/doc, or on the web at http://pcmcia-cs.sourceforge.net.

For devices that are close relatives of normal ISA devices, you willprobably be able to use parts of existing Linux drivers. In somecases, the biggest stumbling block will be modifying an existingdriver so that it can handle adding and removing devices after boottime. Of the current drivers, the memory card driver is the only``self-contained'' driver that does not depend on other parts of theLinux kernel to do most of the dirty work.

In many cases, the largest barrier to supporting a new card type isobtaining technical information from the manufacturer. It may bedifficult to figure out who to ask, or to explain exactly whatinformation is needed. However, with a few exceptions, it is verydifficult if not impossible to implement a driver for a card withouttechnical information from the manufacturer.

I have written a dummy driver with lots of comments that explains alot of how a driver communicates with Card Services; you will findthis in the PCMCIA source distribution in clients/dummy_cs.c.

Guidelines for PCMCIA client driver authors

I have decided that it is not really feasible for me to distribute allPCMCIA client drivers as part of the PCMCIA package. Each new drivermakes the main package incrementally harder to maintain, and includinga driver inevitably transfers some of the maintenance work from thedriver author to me. Instead, I will decide on a case by case basiswhether or not to include contributed drivers, based on user demand aswell as maintainability. For drivers not included in the corepackage, I suggest that driver authors adopt the following scheme forpackaging their drivers for distribution.

Driver files should be arranged in the same directory scheme used inthe PCMCIA source distribution, so that the driver can be unpacked ontop of a complete PCMCIA source tree. A driver should include sourcefiles (in ./modules/), a man page (in ./man/), and configuration files (in ./etc/). The top level directory should also include a README file.

The top-level directory should include a makefile, set up sothat ``make -f ... all" and ``make -f ... install" compile thedriver and install all appropriate files. If this makefile is given an extension of .mk, then it will automatically be invoked by thetop-level Makefile for the all and install targets. Here is an example of how such a makefile could be constructed:

```
# Sample Makefile for contributed client driver
FILES = sample_cs.mk README.sample_cs \
modules/sample_cs.c modules/sample_cs.h \
etc/sample.conf etc/sample etc/sample.opts \
man/sample_cs.4
all:
  $(MAKE) -C modules MODULES=sample_cs.0
install:
  $(MAKE) -C modules install-modules MODULES=sample_cs.0
  $(MAKE) -C etc install-clients CLIENTS=sample
  $(MAKE) -C man install-man4 MAN4=sample_cs.4
dist:
  tar czvf sample_cs.tar.gz $(FILES)
```

This makefile uses install targets defined in 2.9.10 and laterversions of the PCMCIA package. This makefile also includes a``dist" target for the convenience of the driver author. You wouldprobably want to add a version number to the final package filename(for example, sample_cs-1.5.tar.gz). A complete distributioncould look like:

sample_cs.mk
README.sample_cs
modules/sample_cs.c
modules/sample_cs.h
etc/sample.conf
etc/sample
etc/sample.opts
man/sample_cs.4

With this arrangement, when the contributed driver is unpacked, itbecomes essentially part of the PCMCIA source tree. It can make useof the PCMCIA header files, as well as the machinery for checking theuser's system configuration, and automatic dependency checking, justlike a ``normal'' client driver.

In this example, etc/sample and etc/sample.optswould be the new driver's configuration scripts (if needed), andetc/sample.conf would contain any additions to the PCMCIAcard configuration file. Starting with the 3.1.6 release,cardmgr will automatically process any *.conf filesinstalled in /etc/pcmcia, so installation of contributeddrivers should no longer require hand editing configuration files.

I will accept client drivers prepared according to this specificationand place them in the /pub/pcmcia-cs/contrib directory onprojects.sourceforge.net. The README in this directory willdescribe how to unpack a contributed driver.

The client driver interface has not changed much over time, and hasalmost always preserved backwards compatibility. A client driver willnot normally need to be updated for minor revisions in the mainpackage. I will try to notify authors of contributed drivers of changes that require updates to their drivers.

Guidelines for Linux distribution maintainers

If your distribution has system configuration tools that you wouldlike to be PCMCIA-aware, please use the *.opts files in/etc/pcmcia for your ``hooks." These files will not be modified if a user compiles and installs a new release of the PCMCIApackage. If you modify the main configuration scripts, then a freshinstall will silently overwrite your custom scripts and break the connection with your configuration tools. Contact me if you are notsure how to write an appropriate option script, or if you needadditional capabilities.

It is helpful for users (and for me) if you can document how yourdistribution deviates from the PCMCIA package as described in thisdocument. In particular, please document changes to the startupscript and configuration scripts. If you send me the appropriate information, I will include it in the Notes about specific Linux distributions

When building PCMCIA for distribution, consider including contributeddrivers that are not part of the main PCMCIA package. For reasons of maintainability, I am trying to limit the core package size, by onlyadding new drivers if I think they are of particularly broad interest. Other drivers will be distributed separately, as described in the previous section. The split between integral and separate drivers issomewhat arbitrary and partly historical, and should not imply adifference in quality.

rate this article: current rating: Your rating:

image:rdf newsfeed / //static.linuxhowtos.org/data/rdf.png (null)

image:rss newsfeed / //static.linuxhowtos.org/data/rss.png (null)

image:Atom newsfeed / //static.linuxhowtos.org/data/atom.png (null)

- Powered by

image:LeopardCMS / //static.linuxhowtos.org/data/leopardcms.png (null) - Running on

image:Gentoo / //static.linuxhowtos.org/data/gentoo.png (null)

Copyright 2004-2020 Sascha Nitsch Unternehmensberatung GmbH image:Valid XHTML1.1 / //static.linuxhowtos.org/data/xhtml.png (null)

image:Valid CSS / //static.linuxhowtos.org/data/css.png (null)

image:buttonmaker / //static.linuxhowtos.org/data/buttonmaker.png (null)

- Level Triple-A Conformance to Web Content Accessibility Guidelines 1.0 -

- Copyright and legal notices -

Time to create this page: ms

<!--

:

:

image:system status display / /status/output.jpg (null)

-->