

Linmodem–Mini–HOWTO (DRAFT)

Table of Contents

<u>Linmodem–Mini–HOWTO (DRAFT)</u>	1
Sean Walbran sean(at)walbran.org and Marvin Stodolsky stodolsk(at)erols.com	1
<u>1.Introduction</u>	1
<u>2.Linmodems</u>	1
<u>3.General Setup and Kernel Module Issues</u>	1
<u>4.Specific installation instructions</u>	2
<u>5.FAQ</u>	2
<u>6.Troubleshooting</u>	2
<u>7.Appendix: Useful Scripts and Other Information</u>	2
<u>1.Introduction</u>	3
<u>1.1 Copyright</u>	3
<u>1.2 Disclaimer</u>	3
<u>1.3 Credits</u>	4
<u>1.4 Updates and Corrections</u>	4
<u>2.Linmodems</u>	4
<u>2.1 What is a Linmodem?</u>	5
<u>2.2 Which Linmodem hardware is supported?</u>	5
<u>Conexant/Rockwell HSF</u>	5
<u>Ambient Technology (formerly Cirrus Logic)</u>	5
<u>Lucent LT</u>	6
<u>PCTel</u>	6
<u>ESS</u>	6
<u>3Com Mini–PCI</u>	6
<u>2.3 How can I find out if my GeeWhiz 9.8.7 Modem / Laptop has one of these chipsets?</u>	6
<u>Information from the system</u>	6
<u>Modem names and Identification Numbers</u>	7
<u>Laptops, Internal Modems, Machines You Don't Own</u>	7
<u>3.General Setup and Kernel Module Issues</u>	8
<u>3.1 Kernel Module Support</u>	8
<u>3.2 ISA Plug–n–Play</u>	9
<u>3.3 PCI Modems</u>	9
<u>3.4 Precompiled Modules and Kernel Versions: insmod –f, Fixscripting, and ppp.o</u>	9
<u>insmod –f</u>	10
<u>Fixscripting</u>	10
<u>ppp.o and Later Kernels</u>	10
<u>4.Specific installation instructions</u>	11
<u>4.1 Read this first</u>	11
<u>4.2 ISA</u>	11
<u>4.3 Obtain the driver package</u>	11
<u>4.4 Perform a trial insertion of the module</u>	11
<u>4.5 Align the module and kernel versions</u>	12
<u>4.6 Fix the 2.2.14 ppp.o if necessary and desirable</u>	13
<u>4.7 Copy your modules to the module directory</u>	14
<u>4.8 Make the device file</u>	14
<u>4.9 Link /dev/modem to the device file</u>	15
<u>4.10 Insert your module into the kernel</u>	15
<u>4.11 Modifying your serial port configuration</u>	15

Table of Contents

Older kernels (2.0.x)	15
Newer kernels	16
4.12 Try it out with a minimal terminal program	17
4.13 Configure your ppp dialer	17
4.14 Dial up	17
It works!	17
It doesn't work!	17
4.15 Post–Installation	18
5.FAQ.....	18
5.1 I have a winmodem. Will it work under Linux?	18
5.2 I get a "device or resource busy" error	18
5.3 I get unresolved symbols when fixscripting/insmoding	18
5.4 The ppp2214.o doesn't work with my SuSE (or other) kernel	19
5.5 My PCTel modem doesn't work	19
5.6 The modem dials and connects fine, but then it drops the connection	19
5.7 I get a kernel panic on closing the connection or unloading the driver	20
5.8 Nothing seems to work. To whom can I turn for help?	20
5.9 Who wrote the driver for my winmodem, and how do I contact him/her?	20
6.Troubleshooting.....	20
7. Appendix: Useful Scripts and Other Information	23
7.1 Fixscript: Changes binary module kernel version information (Mark Spieth)	23
7.2 Coincident insertion and removal of ppp related modules (Marvin Stodolsky)	25
7.3 Example PPP scripts (Marvin Stodolsky)	26
7.4 PCTel Module Parameters: Country Code	27

Linmodem–Mini–HOWTO (DRAFT)

Sean Walbran [sean\(at\)walbran.org](mailto:sean(at)walbran.org) and **Marvin Stodolsky** [stodolsk\(at\)erols.com](mailto:stodolsk(at)erols.com)

v0.26, 28 September 2000

This document describes Linmodem (winmodem hardware) support under Linux. At this point, such support is very limited, almost exclusively in the form of manufacturer–created but unsupported binary kernel modules. This document addresses the special complications dealing with these binary modules rather than with source code, particularly focusing on the compromises of using mis–matched binaries and kernels.

1. [Introduction](#)

- [1.1 Copyright](#)
- [1.2 Disclaimer](#)
- [1.3 Credits](#)
- [1.4 Updates and Corrections](#)

2. [Linmodems](#)

- [2.1 What is a Linmodem?](#)
- [2.2 Which Linmodem hardware is supported?](#)
- [2.3 How can I find out if my GeeWhiz 9.8.7 Modem / Laptop has one of these chipsets?](#)

3. [General Setup and Kernel Module Issues](#)

- [3.1 Kernel Module Support](#)
- [3.2 ISA Plug–n–Play](#)
- [3.3 PCI Modems](#)
- [3.4 Precompiled Modules and Kernel Versions: `insmod -f`, Fixscripting, and `ppp.o`](#)

4. Specific installation instructions

- [4.1 Read this first](#)
- [4.2 ISA](#)
- [4.3 Obtain the driver package](#)
- [4.4 Perform a trial insertion of the module](#)
- [4.5 Align the module and kernel versions](#)
- [4.6 Fix the 2.2.14 ppp.o if necessary and desirable](#)
- [4.7 Copy your modules to the module directory](#)
- [4.8 Make the device file](#)
- [4.9 Link /dev/modem to the device file](#)
- [4.10 Insert your module into the kernel](#)
- [4.11 Modifying your serial port configuration.](#)
- [4.12 Try it out with a minimal terminal program](#)
- [4.13 Configure your ppp dialer](#)
- [4.14 Dial up](#)
- [4.15 Post–Installation](#)

5. FAQ

- [5.1 I have a winmodem. Will it work under Linux?](#)
- [5.2 I get a "device or resource busy" error.](#)
- [5.3 I get unresolved symbols when fixscripting/insmoding.](#)
- [5.4 The ppp2214.o doesn't work with my SuSE \(or other\) kernel.](#)
- [5.5 My PCTel modem doesn't work.](#)
- [5.6 The modem dials and connects fine, but then it drops the connection.](#)
- [5.7 I get a kernel panic on closing the connection or unloading the driver.](#)
- [5.8 Nothing seems to work. To whom can I turn for help?](#)
- [5.9 Who wrote the driver for my winmodem, and how do I contact him/her?](#)

6. Troubleshooting

7. Appendix: Useful Scripts and Other Information

- [7.1 Fixscript: Changes binary module kernel version information \(Mark Spieth\)](#)
 - [7.2 Coincident insertion and removal of ppp related modules \(Marvin Stodolsky\)](#)
 - [7.3 Example PPP scripts \(Marvin Stodolsky\)](#)
 - [7.4 PCTel Module Parameters: Country Code](#)
-

1. [Introduction](#)

This is the Linux Linmodem Mini-HOWTO document. It is intended as a quick reference to help you find out if there is a way to get your (so-called) winmodem working under Linux, and, if so, how to do it. You should understand from the outset that there is probably *no* support for your winmodem: at this time, there is limited support for such modems, often in the form of vendor-created but vendor-unsupported, binary-only kernel modules; however, a small number of open-source projects exist. For the most up-to-date information about available Linmodem drivers, visit [Rob Clark's site](#) and the [Linmodems.org](#) site. General modem issues, such as IRQ settings and dialup scripts, are dealt with much more thoroughly in the more general [Modem-HOWTO](#), [Serial-HOWTO](#), [PPP-HOWTO](#), and other related HOWTOs available at the [Linux Documentation Project](#) site.

1.1 Copyright

Copyright (c) 2000 by Sean Walbran, Marvin Stodolsky

Please freely copy and distribute (sell or give away) this document in any format. It's requested that corrections and/or comments be forwarded to the document maintainer. You may create a derivative work and distribute it provided that you:

- Send your derivative work (in the most suitable format such as shtml) to the LDP (Linux Documentation Project) or the like for posting on the Internet. If not the LDP, then let the LDP know where it is available.
- License the derivative work with this same license or use GPL. Include a copyright notice and at least a pointer to the license used.
- Give due credit to previous authors and major contributors.

If you're considering making a derived work other than a translation, it's requested that you discuss your plans with the current maintainer.

1.2 Disclaimer

Use the information in this document at your own risk. I disavow any potential liability for the contents of this document. Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

All copyrights are owned by their owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. In particular, since the term "Winmodem" is a trademark of US Robotics/3Com, we use the term "winmodem" here as does Rob Clark: to be read as "Winmodems(tm), host-based modems, HCF-modems, HSP-modems, and all similar modem-like hardware." Linux is a trademark of Linus Torvalds.

Naming of particular products or brands should not be seen as endorsements.

It are strongly recommended to make a backup of important and/or relevant files before any installation procedure.

1.3 Credits

A large amount of information contained in this document comes a variety of great sources such as [Rob Clark's site](#), [Linmodems.org](#), and the mailing lists there, and Werner Heuser's [LiLAC: Linux with LAptop Computers](#) pages.

Thanks to Willie Green (willjr(at)lcc.net) for his fine contribution of the ESS modem information.

Thanks to Mark Spieth (mark(at)digivation.com.au) for discussions, advice, and the contribution of his "fixscript" for module kernel version editing (originally to the Linmodem mailing list [here](#) , and a newer version which is included in the appendix to this document).

Thanks to Werner Heuser wehe (at) snafu . de for tips on setting serial parameters with older kernels, and a number of other good points.

Other individual credits are given in the body of the text where appropriate.

This document itself was created using the SGML HOWTO template created by Stein Gojen, as described in the [HOWTO–HOWTO](#) site.

1.4 Updates and Corrections

The most recent HTML version of this document is available at <http://walbran.org/sean/linux/linmodem–howto.html>, SGML at <http://walbran.org/sean/linux/linmodem–howto.sgml>, and some other formats in the same directory.

Do you have a Linmodem which works, but is not described here? Are you developing a driver? Do you think something in this document is incorrect or misleading? Do you think that your or someone else's work has been used here but not appropriately credited? Please don't hesitate to email me at [sean\(at\)walbran.org](mailto:sean(at)walbran.org) with corrections and suggestions.

2. [Linmodems](#)

2.1 What is a Linmodem?

A Linmodem is the Linux implementation of a "winmodem" (see disclaimer). These devices are 'less than' a modem in the sense that they depend on software to perform, to a greater or lesser extent, the functions traditionally handled by modem hardware. The rationale for this is, of course, that software is cheaper than hardware, and can be upgraded/expanded/improved without the use of screwdrivers (usually); however, for the modem to function at all, one requires software that can run on one's preferred operating system.

2.2 Which Linmodem hardware is supported?

At the time of this writing, only a few winmodems will work under Linux:

Conexant/Rockwell HSF

There exists a driver at <http://www.olitec.com/pci56kv2.html> which was only recently (Sept. 2000) "discovered" by Denis Havlik (denis(at)mandrakesoft.com). The page is in French, but the installation commands are given on the page in boldface red text. Essentially, download the package, unpack it with `tar -zxvf`, and run the installation script `ins_all`. Some things already known about the driver include:

- It was compiled against the RedHat 6.2 kernel 2.2.14-5.0.
- It appears to be tuned for the French telephone system. Mark Allen posted an `.inf` file for some other phone systems to the linmodems mailing list at <http://linmodems.org/cgi-bin/ezmlm-cgi?1:misp:1773:nlifphijcfgekncagkpa>. Also, you can try disabling dial-tone detection with your dialer.

This driver has not yet been fully explored; check out the mailing list archives at linmodems.org for further discussion and information.

Ambient Technology (formerly Cirrus Logic)

Mikhail Moreyra has written a GPL'ed driver for the CL-MD5620DT chipset which can do up to 33.6 kbps; however, this is alpha software and should be treated with due care. The driver can be obtained at <http://linmodems.org/CLModem-0.3.0.tar.gz>. Recently, Gabriel Gambetta (ggambett(at)internet.com.uy) issued a patched version of the driver to allow standard AT modem commands; you can get this version at Rob Clark's site [here](#).

In addition, [Rob Clark's site](#) reports that Ambient may release Linux drivers for their MD563X HaM modems at <http://www.ambient.com> at some undetermined point in the future.

Lucent LT

There exists a manufacturer–unsupported, binary–only kernel module compiled for the 2.2.12 Linux kernel released for Lucent LT (PCI and ISA) modems. As discussed below, this module will work with minor complaints under kernel 2.2.14, and with some additional effort under kernels up through 2.2.17, but the module does not insert into the (still experimental) 2.40–test6 kernel. The driver can be obtained at <http://linmodems.org/linux568.zip>

Some open source tools for use with Lucent modems are available at <http://www.close.u-net.com/ltmodem.html>. Pavel Machek writes that "It is not too useful, however: it is a hardware driver, and without a v.34 protocol stack, you can't connect to your ISP. It is enough to turn your Lucent winmodem into an answering machine, however."

PCTel

Binary drivers for PCI, AMR, and Zoltrix Phantom can be found at <http://www.kcdata.com/~gromitkc/winmodem.html#drivers>

ESS

Binary drivers for ES56T–PI (PCI) and ES56V–I (ISA) can be obtained at

- ISA: <ftp://ftp.esstech.com/pub/modem/isa/unsupported/56v-i/linux/kernel61/linux111.zip> or
- PCI: <ftp://ftp.esstech.com/pub/modem/pci/unsupported/56t-pi/linux/Kernel61/111.zip>. man

3Com Mini–PCI

A request for comments was posted by a 3Com official about the possible demand for a binary–only driver for their miniPCI combination NIC/winmodem on the [Linmodems.org mailing list](#). Though to my knowledge no driver has yet been released, Werner Heuser's [miniPCI page](#) has more information and links.

2.3 How can I find out if my GeeWhiz 9.8.7 Modem / Laptop has one of these chipsets?

Information from the system

If you're already running Linux, you can obtain information about installed hardware using commands like:

- PCI: `cat /proc/pci`

- ISA: `pnpdump`
- Internal PCMCIA: `cardctl ident`
- `dmesg`

The Device Manager under Windows can provide similar information, but it should be noted that a manufacturer will often simply put its brand name on a built-in modem, so this information may not be as useful as you might hope (e.g., what chipset does a "Compaq Internal 56k" modem have?).

Modem names and Identification Numbers

If you know the precise name of your modem, you can try searching the large Linux Modem Compatibility [database](#) at [Rob Clark's site](#). The color/letter code on the left side of the table will indicate if your modem is known to function or not under Linux. The code "LM" indicates a Linmodem, and the modem notes should indicate which driver you need. A "WM" means it's a winmodem, but no support is known to exist. Be careful not to assume that modems with similar names will contain the same chipsets, or will necessarily behave similarly whatsoever! Your WhizBang LX56 and your friend's WhizBang GT56 could have entirely different innards.

If you do not know the precise name of your modem, you can search based on the identification number of the modem: on every modem there must be printed a registration number, which may either be the board producer's designation, or, alternatively, an FCC registration number. An example photo of such an ID number on a modem board can be found at <http://www.kcdata.com/~gromitkc/fcc1.jpg> on [Rob Clark's site](#). You can then proceed to use your web browser to search his [table](#) of modems and FCC ID's to obtain chipset/driver information. Alternatively, you can directly search the US Federal Communications Commission (FCC) database at <http://www.fcc.gov/oet/fccid/>. Read the directions carefully, and be careful not to confuse O (the letter) with 0 (the number), and other possible mixups.

Laptops, Internal Modems, Machines You Don't Own

You may not be able to obtain the FCC ID number if you have a laptop which you prefer not to open up, or are looking to buy a particular machine and the vendor has not been polite enough to provide you with the information nor a sample box for you to take apart and play with. In these cases, you have a few options:

- Werner Heuser's [LiLAC: Linux Modems](#) and [LiLAC: Linux Mini-PCI](#) pages include lists of specifications for laptops with internal modems and NIC's, as well as useful tips for obtaining more information in case the model is not listed there.
- Kenneth Harker's [Linux on Laptops](#) site indexes a large number of user-created sites describing their experiences with Linux on particular laptop models.
- If you have physical access to, e.g., a floor demonstration model, Marvin Stodolsky writes of a way to actually test a laptop or PC:

There is a quick test that can be run from a floppy disk before you commit to buy the the just advertised WhizBang PC/Laptop. The intent is to use Linux itself to check whether the pre-installed modem is supported under Linux.

Most Linux rescue floppies boot up a RAM disk, without "touching" the hard disks unless/until

specifically commanded to. A collection of the available Linux modem drivers can be previously collected on a separate floppy. Linux will not install a hardware dependent module in the kernel unless the compatible hardware is present. Thus if `lsmod` displays a successful insertion after:

```
insmod /mnt/modem-driver1.o
```

or at worst with forcing

```
insmod -f /mnt/modem-driver1.o
```

there is a good chance that the modem will work under a full Linux installation. This insertion test can be repeated in turn with all the available `modem-driverN.o`, and hopefully one will succeed. For this test the DOS formatted floppy disk with drivers is first made accessible with:

```
mount /dev/fd0 -t msdos /mnt
```

and drivers can be listed with:

```
ls /mnt
```

There is a caveat that this test may fail merely because of mismatches between kernel sources versions of the `modem.o` and the kernel being used in the test. Kernels on the Rescue disks should thus be version matched as closely as possible with the that of the `modem.o`. The Lucent `ltmodem.o` and Archtek `esscom.o` modules were compiled under `kernel-sources-2.2.12`, so, for their tests, the kernel on the Rescue floppy should best be replaced. The Rescue disk, made from the Debian `resc1440.bin` by `rawrite.exe` under DOS, carries instructions from substituting and properly activating its kernel. A suitable kernel has been compiled and is available at <http://walbran.org/sean/linux/stodolsk/kern2212.zip> along with a copy of the Debian rescue image at <http://walbran.org/sean/linux/stodolsk/resc1440.bin>. The Linmodem driver modules are listed above.

There is another caveat, however: I (Sean) tried this out with a stock Red Hat 6.2 distribution, and found that it gave me no complaints whatsoever when I `insmod`'ed the `esscom.o` module, despite the fact that I have no such hardware. You, therefore, probably want to be sure to use the Debian rescue disk image given by MarvS above, rather than one from another distribution.

3. General Setup and Kernel Module Issues

3.1 Kernel Module Support

All of the drivers listed here are released as kernel modules; therefore, you must be sure to have a kernel which supports modules. In addition, "module version" support should be enabled to aid the use of kernels and modules which are not version matched, as described further below.

If you use a kernel from a reasonably recent Linux distribution, module support is most likely already enabled. If you're compiling the kernel yourself, then you should already be aware of how to enable modules, via the [Kernel HOWTO](#). In any case, you can check to make sure that the following settings exist in your kernel configuration file (which is usually found under `/usr/src/linux`):

```
CONFIG_MODULES=y
CONFIG_MODVERSIONS=y
```

3.2 ISA Plug-n-Play

If you have an ISA Plug-n-Play modem (such as I do on my Thinkpad i1411), you will most likely need to use `isapnptools` to allocate resources to the modem card. For this, you need to have `isapnptools` installed and have an entry in the `/etc/isapnp.conf` file for the modem. You should read the [Plug-and-Play-HOWTO](#), but if you have no other ISA devices you're concerned about, basically all you need to do is:

1. As root, run `pnpdump` to generate a prototype `isapnp.conf` file based on your system's current resource usage.
2. Look for your modem in this output, and uncomment the lines corresponding to the (otherwise unused) IRQ you wish to use for the modem. For example, my `isapnp.conf` includes:

```
(CONFIGURE ACRd119/1 (LD 0
  (INT 0 (IRQ 11 (MODE +E)))
  (IO 1 (SIZE 8) (BASE 0x0100) (CHECK))
  (NAME "ACRd119/1[0]{LT Win Modem      }")
# (ACT Y)
))
```

Strangely, for me at least, it is necessary to leave the `#(ACT Y)` commented out. If it doesn't work for you one way, try it the other.

3. Copy the file to `/etc/isapnp.conf`
4. Reboot. You should see a message along the lines of 'Initializing ISA PNP devices...OK' on booting. If it fails, you have probably selected an IRQ/DMA setting which is already in use; try another of the options given in the `pnpdump` output. (Note that it is probably not necessary to reboot, if you run `isapnp` with the right flags. However, it's easiest for the beginner to simply reboot at this point.)

3.3 PCI Modems

To access more information than `cat /proc/pci` gives for cards with a PCI interface, utilities within the software package `pciutils` are useful, such as `scanpci` and `lspci`.

3.4 Precompiled Modules and Kernel Versions: `insmod -f`, Fixscripting, and `ppp.o`

As of the writing of this document, source code is only available for the Ambient Technology driver and the Lucent LT user-space tools listed above; the remainder have been released only as precompiled binary drivers. These modules/binaries transparently function only with the kernel against which they were co-compiled. Since the Linux kernel is a dynamically changing beast, it is very unfortunate that the modem vendors have not yet chosen to release source-code versions of their drivers, which would ensure the ability to modify and recompile these drivers for use with improved Linux kernels. In the meantime, however, some luck has been had in coaxing the binary modules to function under some later kernel versions, as described below.

insmod -f

A kernel module generally may be loaded using the command `insmod module_name`. If the module was compiled under a different kernel than the current one, `insmod` will report the version mismatch and fail to load the module. One can pass a flag to force the module to load despite the mismatch as `insmod -f module_name`. If the kernel interface the module uses did not actually change with the kernel version, forcing the module to load in this way can give you a functioning module. This is the case with, for example, the Lucent LT modem module `ltmodem.o` which, while compiled under 2.2.12, can be forcibly inserted and run without problems under kernel 2.2.14; however, running it with kernel version $> 2.2.15-x$ requires the replacement of the module `ppp.o` with one from kernel 2.2.14, as described below.

Fixscripting

Mark Spieth (mark(at)digivation.com.au) has contributed a script (see appendix) which edits a binary module so that the version mismatch warning is eliminated, allowing module insertion to proceed without the forcing flag, i.e. simply `insmod module_name`. Continuing with the particular example of `ltmodem.o`, if you save this file as `fixscript`, make it executable with `chmod +x fixscript`, and, as the root user, generate an edited module with

```
./fixscript ltmodem.o ltmodem14.o
```

, you can insert the module under 2.2.14 without mismatch complaints by

```
insmod ltmodem14
```

Note, however, that this script does not update the module code to fix any actual kernel API discrepancies, but simply masquerades the module as having been compiled under a different version; i.e., this suppresses the symptoms of kernel version mismatch, but does not cure any actual diseases which may exist. The "source code" supplied with some PCTel modules (a small C file) performs similar masquerading when compiled and linked with the binary libraries in those packages, but again does not allow for a changed kernel interface.

ppp.o and Later Kernels

Initial attempts to use the Lucent LT modem driver `ltmodem.o` under kernels later than 2.2.15-x failed. However, Christoph Hebeisen (cth(at)sfu.ca) reported success by using the modem module compiled under 2.2.12 with a PPP module (`ppp.o`) compiled under 2.2.14 with a kernel version 2.2.16. Willie Green (willjr(at)lcc.net) confirms that this works also with the ESS module. Marvin Stodolsky (stodolsk(at)erols.com) points out that the *same* `fixscript` described above can be used to version-masquerade the `ppp.o` module from 2.2.14 to 2.2.16 as

```
./fixscript ppp.o pppl4to16.o
```

thus allowing "soft insertion" of both the `ltmodem` and `ppp` modules to obtain a functioning Linmodem under kernel version 2.2.16.

At some point, this chained use of outdated modules will probably reach a breaking point; it would be very surprising, for example, if some complex combination of older modules would allow these binary drivers to work under kernel 2.4 when it is released. The best solution would be, of course, having an open source driver; at the very least, vendor support with binary modules for more recent kernel versions would be useful. The reader is encouraged to do his/her best to support and encourage such developments.

4. Specific installation instructions

4.1 Read this first

Installation instructions are given for the binary drivers described above, since the accompanying documentation is scant (if existing at all) and can be very misleading. Installation of open-source drivers is not covered here, since it is usually better described by its own documentation, and varies more dramatically from package to package and from time to time.

Installation instructions for the Conexant driver are given, to the extent that they are known, above. This driver has not yet been thoroughly explored.

You may wish to use the installation script which may have accompanied your driver instead of following the instructions given here.

You should read through the instructions given below before beginning, to get a feeling for what's going on and where you might run into problems.

4.2 ISA

If your modem is an ISA PNP modem, make sure that the PNP resources have been appropriately allocated using `isapnptools`. (See "ISA Plug-n-Play" above.)

4.3 Obtain the driver package

See "Which Linmodem hardware is supported?" above for the appropriate URL for the driver for your hardware.

If necessary, unpack the driver package in an empty directory with `unzip` (if `.zip`), `tar -zxvf` (if `.tar.gz`), etc. The files present should include binaries with names like `some-driver.o` (e.g., in the example in the next step, the module is named `ltmodem.o`).

4.4 Perform a trial insertion of the module

A trial insertion of the module is a good first check to do. Though its results are not necessarily indicative of your final success, it is a fair indicator of whether the remaining steps will be worth your time.

In the directory in which the files were unzipped, perform the following steps, replacing `ltmodem.o` with the name of the appropriate module(s) for your modem:

First, try to simply insert the module with:

```
insmod ltmodem.o
```

There may be a complaint such as:

```
kernel-module version mismatch
ltmodem.o was compiled for kernel version 2.2.12-20
while this kernel is version 2.2.14.
```

If so, try forcing the insertion with:

```
insmod -f ltmodem.o
```

Having done this, check for successful insertion into the kernel with:

```
lsmod
```

which will, if insertion was successful, display a list including `ltmodem.o` such as:

```
Module                Size  Used by
ltmodem                453200  0 (unused)
nls_iso8859-1          2268   1 (autoclean)
nls_cp437              3744   1 (autoclean)
vfat                   9052   1 (autoclean)
fat                   29248   1 (autoclean) [vfat]:
...
```

If the module (`ltmodem`) was not inserted, there is an outstanding problem. The `some-driver.o` may not be matched with the modem hardware; the differences between the sources from which the kernel and module were compiled may be too great; or, your module may require another module to be loaded beforehand (some PCTel packages have two modules; try reversing the insertion order of the two?). Before abandoning the current effort entirely, a more closely version matched kernel could be installed and tested. Source files for [Linux kernels](#) can be downloaded and already compiled kernels are available with many Linux distributions.

If the insertion was successful, it is worth proceeding further.

4.5 Align the module and kernel versions

- Lucent LT and ESS:

In the following, all steps except the final one are optional, but will allow the modules to be inserted without forcing and without version-mismatch warning messages. (Below, replace "ltmodem" with "esscom" if you have an ESS modem.)

1. Make a workplace, say `mkdir /root/modem` and `cd /root/modem`
2. Save the "fixscript" (see appendix) to a file named `/root/modem/fixscript`
3. Make the script executable with: `chmod +x fixscript`
4. Rename the module file: `mv ltmodem.o ltmodem.2212.o`
5. "Fix" the module file with: `./fixscript ltmodem.2212.o ltmodem.o`
6. Test insertion with the fixed module file with: `insmod ltmodem.o` (it should insert without version warnings)
7. Copy a functional `ltmodem.o` to the standard module location with: `cp ltmodem.o /lib/modules/`uname -r`/misc/ltmodem.o` (The ``uname -r`` specifies the

version of currently running kernel.)

The last three steps will have to be repeated when a new kernel is installed, and are ONLY effective during a session under the new kernel.

If you are running Linux kernel 2.2.15-x or newer, you need to obtain the `ppp.o` module from kernel 2.2.14 (and should use the "fixscript" on it as well), as described in the next step.

- PCTel:

There are apparently two types of PCTel module package around.

1. A package (rpm or deb) which installs two module files, `pctel_hsp.o` and `pctel_pci.o`, in `/lib/modules/2.2.16`.

With such a package, if you are running a kernel more recent than 2.2.16, you should try the "fixscript" method used with the Lucent and ESS modules above, but note that this has not to my knowledge been tried out yet. If you are running a kernel older than 2.2.16, you should consider upgrading your kernel, or else try the fixscripting as well (this is also not guaranteed to work). You should not have to get the 2.2.14 `ppp.o` module. Please send me a report if you get these to work.

2. A package which, when unpacked, gives a set of libraries (`hsp.a`, etc...) and a small C source file (`ptmodule.c`), which should be in directories like `lib/` and `src/module/`. If there are no directories, create them and arrange the files with:

```
mkdir lib
mkdir src
mkdir src/module
mv *.a lib/
mv Makefile *.c src/module
```

Now go to the directory `src/module` and type `make`. This should generate the module file `pctel.o`, which will appear back up in the directory `lib`. (The driver module is *not* the file `ptmodule.o` in `src/module`!)

The version of the module generated in this way will match your current kernel version.

4.6 Fix the 2.2.14 `ppp.o` if necessary and desirable

If you have a Lucent or ESS modem, and your kernel is more recent than 2.2.15-x), use of a `ppp.o` from 2.2.14 sources will be necessary. A copy of this module is available at <http://walbran.org/sean/linux/stodolsk/ppp2214.o>.

It can be copied to the appropriate `/lib/modules/^uname -r^/net` directory as described below. To avoid forcing `insmod -f ppp.o`, the mismatched `ppp.o` can be version fixed as described here for the example of a kernel version 2.2.17 mismatch.

1. Type `su root` and give the password. Change to the working directory `cd /root/modem`
2. Align the module version with the running kernel version: `fixscript ppp2214.o pppfix.o`
3. Remove the original module from the kernel with: `rmmmod ppp`, if it's already loaded
4. `insmod pppfix.o`, which should proceed without any mismatch complaint.
5. `lsmod` should display the inserted `ppp` in the list.
6. If so, finally save the original module, found at (say) `/lib/modules/2.2.17/net/ppp.o`, by renaming it with:

```
mv /lib/modules/2.2.17/net/ppp.o /lib/modules/2.2.17/net/ppp2217.o
and place the fixed pppfix.o with
```

```
cp pppfix.o /lib/modules/2.2.17/net/ppp.o
```

4.7 Copy your modules to the module directory

If not already done copy the other module file(s) (such as `ltmodem.o`, `esscom.o`, and `pctel*.o`) into the directory

```
/lib/modules/`uname -r`/misc/
```

4.8 Make the device file

- Lucent LT:

```
mknod /dev/ttyS14 c 62 78
```

- PCTel:

```
mknod /dev/ttyS15 c 62 79
```

- Esscom:

```
mknod /dev/esscom c 127 1
```

- ◆ Esscom users may wish to link `/dev/ttyS14` to `/dev/esscom` for use with `ppp` dialers which only recognize names of the `ttyS*` form, such as `wvdial`. Note, however, that `wvdial` is reported to give a kernel panic on `ppp-off` with this module under kernel 2.2.16 (and possibly others). This may not be a general situation, however.

If you wish to allow non-root users to dial out using this device, you can set the file permissions with, for example,

```
chgrp uucp /dev/ttyS14
chmod 666 /dev/ttyS14
```

Group definitions can be modified in `/etc/group`.

4.9 Link `/dev/modem` to the device file

```
ln -s /dev/yourdevicefile /dev/modem
```

4.10 Insert your module into the kernel

Note: if you chose to skip the "Align the module and kernel versions" step above, you may need to use "insmod -f" here.

- Lucent LT:

```
insmod ltmodem
```

- Esscom:

```
insmod essmodem
```

- PCTel:

```
insmod pctel
```

- ◆ Note: The PCTel drivers can take a Country Code parameter. See the appendix for a list of codes.

4.11 Modifying your serial port configuration.

Your Linux system will in many cases need to be informed of the addition of the new "serial" device `/dev/ttyS14` or `15`.

Older kernels (2.0.x)

The drivers available tend to have been compiled for a kernel version 2.2.x, with x in the teens. However, if you are for some reason unable/unwilling to update your kernel, Werner Heuser ([wehe\(at\)snafu.de](mailto:wehe(at)snafu.de)) points out the following tips for older kernels:

With 2.0.x kernels, the serial ports are defined in the serial driver source itself, i.e. `/usr/src/linux/drivers/char/serial.c`; after 2.1.98, these moved to

`/usr/src/linux/include/asm-i386/serial.h` and require `CONFIG_SERIAL_MANY_PORTS`, `MULTIPORT` and `SHARE_IRQ` to be set during kernel configuration. You should modify the appropriate line for your device file, i.e. either of

```

        /* UART CLK   PORT IRQ       FLAGS           */
...
        { 0, BASE_BAUD, 0x000, 0, 0 },           /* ttyS14 (spare; user configurable) */
        { 0, BASE_BAUD, 0x000, 0, 0 },           /* ttyS15 (spare; user configurable) */

```

to read something like

```

        { 0, BASE_BAUD, 0x0260, 3, STD_COM_FLAGS}},

```

with the appropriate port/irq for your hardware. When you boot the new kernel with these changes, you should see a message like the following:

```

Serial driver version 4.13 with no serial options enabled
tty00 at 0x03f8 (irq = 4) is a 16550A
tty14 at 0x0260 (irq = 3) is a 16550A

```

Newer kernels

With more modern Linux kernels, script files such as `/etc/serial.conf` and the program `setserial` are generally used to govern the parameters of serial ports; your configuration will likely have to be modified to accommodate the new device, though this apparently depends on your kernel/distribution of choice. The best reference for such modifications is David S. Lawyer's excellent [Serial HOWTO](#), in particular the section on [Setserial](#), where he notes, in particular, "Don't ever use `setserial` with Laptops (PCMCIA)". The documentation with your distribution should provide you with more information on the particular defaults and initialization scripts used.

As an example, Sean's laptop with the Lucent LT modem, running Red Hat 6.2/kernel 2.2.14-5, required no modifications to the (in fact nonexistent) `/etc/serial.conf` whatsoever. With Marvin's PCI Lucent winmodem with a Debian installation, however, the following section is needed in the file `/etc/serial.conf`:

```

# These are two spare devices you can use to customize for
# some board which is not supported above....
#
#Lucent Modem driver version 4.27.5.66
# ltmodem.o was compiled for kernel version 2.2.12-20
# with MANY_PORTS MULTIPORT SHARE_IRQ enabled
# ttyS14 at 0x0260 (irq = 3) is a Lucent
/dev/ttyS14 uart 16450 port 0x0260 irq 3
#/dev/ttyS15 uart XXXXX port XXXX irq X
# These are the ports used for either the Usenet Serial II
# board, or the Boca Board 4, 8, or 16 port boards.

```

Whatever your particular configuration, conflicts for interrupt (IRQ) assignments are generally to be avoided. Information on your serial port properties can be displayed with:

```

setserial -agv /dev/ttyS*

```

which returns information like:

```

/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
    Baud_base: 115200, close_delay: 50, divisor: 0
    closing_wait: 3000
    Flags: spd_normal skip_test session_lockout

```

```
/dev/ttyS14, Line 14, UART: 16950/954, Port: 0x0260, IRQ: 3
  Baud_base: 115200, close_delay: 50, divisor: 0
  closing_wait: 3000
  Flags: spd_normal skip_test
```

4.12 Try it out with a minimal terminal program

At this point you probably want to try dialing out with a simple terminal program like `minicom`.

4.13 Configure your ppp dialer

The name and location of these scripts depend on the particular Linux distribution (among Redhat, Debian, Suse and many others) and the particular software mediating the dialup (such as `kppp` and `wvdial`). Many distributions/dial-up software have a configuration utility which will request your input and then automatically generate the dial-up scripts. The minimum information required is:

- The telephone number of your Internet Service Provider (ISP)
- Your username and password for the ISP
- The modem device file (i.e. `/dev/modem`, `/dev/ttyS14`, etc.)

If you have trouble, consult the documentation of your ppp dialer and/or the [PPP-HOWTO](#). Example scripts for the particular example of `pppconfig` with Debian are given in the appendix.

4.14 Dial up

Try to connect to your ISP with the ppp dialer.

It works!

Congratulations!

It doesn't work!

Dang!

See the Troubleshooting and FAQ sections of this howto for some ideas on how to fix it.

4.15 Post-Installation

To avoid having to insert the module(s) in the kernel every time you want to dial in, see the appendix section "Coincident insertion and removal of ppp related modules."

5.FAQ

5.1 I have a winmodem. Will it work under Linux?

Probably not. Please see the section "Which Linmodem hardware is supported?" above, and check the Linux Modem Compatibility [database](#) at [Rob Clark's site](#).

5.2 I get a "device or resource busy" error.

- If you have an ISA modem, did you use the `isapnptools` to allocate IRQ and DMA resources to the card? See "ISA Plug-n-Play", above, for more information.
- If you do `setserial -agv /dev/ttyS*`, does it return something sane for your device file? If not, see the "Modifying your serial port configuration" section above.
- Otherwise, double-check that you created the device file correctly, and try to eliminate any IRQ conflicts you might have. If all looks well, but it still doesn't work, check the [Linmodems.org](#) mailing list to see if someone else has (and has perhaps fixed) the same problem, or try to fix it yourself and inform others of your results.

5.3 I get unresolved symbols when `fixscripting/insmoding`.

Unresolved symbols are a true danger of version mismatching and are, in general, bad, but are also almost inevitable with binary modules. If the `fixscript` reports unresolved symbols, or the module does not work despite the unresolved symbols, you may be out of luck with that kernel/module combination; however, a few common cases involve symbols like:

- `slhc_XXXX`: You probably need to `insmod` the `slhc` module before inserting the modem/ppp modules.
- `printk`, `jiffies`: Your kernel may be compiled with SMP enabled. None of the binary modules are known to be SMP-safe, and will probably only work on a single-processor machine with a single-processor kernel, i.e. SMP disabled. You should try recompiling your kernel or otherwise obtaining a version with SMP disabled. (Thanks to Tom Reinertson ([treinertson\(at\)uswest.net](mailto:treinertson@uswest.net)))
- `tty_XXXX` with `esscom.o`: Earlier `fixscripts` were not able to handle the version-specific symbols in this module. A preliminary, expanded `fixscript` has been included in this version of the HOWTO

which should be able to fix this module as well. It assumes you have a fairly recent version of the depmod utility; later versions will improve the reliability and flexibility of these new parts of the fixscript code. Look for more recent versions of this HOWTO <http://walbran.org/sean/linux/linmodem-howto.html> to see if the script has been improved.

If a module works in an unstable fashion, it could be that, under some circumstances, you are avoiding those symbols, while in others, you slam up against them. Try out different ppp dialup programs (wvdial, kppp), which call a different set of functions under similar conditions. It is also possible that the fixscript, which was designed for the lucent module, is not "fixing" the symbols used in your module. If you find no combination that works, consider "downgrading" to a kernel which has a closer version match to that of the module.

5.4 The ppp2214.o doesn't work with my SuSE (or other) kernel.

The 2.2.14 ppp.o trick appears to be sensitive to kernel versions and configurations. Try to obtain one that matches your kernel as well as possible, and if that doesn't work, try to obtain one from any other source you can find. For SuSE 7.0, Christoph Hebeisen has posted a 2.2.14 ppp.o pre-fixed for the SuSE 2.2.16 kernel at his site, <http://www.physcip.uni-stuttgart.de/heby/computer/ltmodem/>, which should work.

5.5 My PCTel modem doesn't work.

- Do you need to give the module a country code parameter? See the appendix.
- Are you using the right driver module? There are a few PCTel drivers around (see the section "Which Linmodem hardware is supported?" above). You might try one of the others and see if that helps.

5.6 The modem dials and connects fine, but then it drops the connection.

This is an often-reported problem that may not have a solution.

It is possible that the module is installed correctly and is working, but that you have a problem with your ppp configuration. See "Configure your PPP dialer", above, for more information.

It has been reported that, with some kernel/module mismatches, a program like kppp will give this error, while an alternative like wvdial does not, for the same modules and hardware. You may wish to try a different ppp dialer and see if that helps.

If none of these helps, you may wish to consider trying to use a kernel version which is closer to the module. Otherwise, try the mailing list at Linmodems.org for help.

5.7 I get a kernel panic on closing the connection or unloading the driver.

There are a couple of possible solutions to this, neither of which may work:

- Try a different ppp dialer (wvdial, kppp).
- Configure the module so that it stays in the kernel, i.e. so that it is not unloaded.

5.8 Nothing seems to work. To whom can I turn for help?

- Double-check that the modem you have is actually supported by the module you have. See "Which Linmodem Hardware Is Supported?" above.
- Try to determine at which stage of the installation process things break down. Check the man pages on the commands used in that stage and see if you can determine the source of the problem.

If all seems lost, please see the section "Troubleshooting", below, and consider sending a message with the complete information described there to the mailing list at [Linmodems.org](mailto:linmodems.org).

5.9 Who wrote the driver for my winmodem, and how do I contact him/her?

Probably somebody on contract to the manufacturer, who probably does not have the authority the update/release/change the source code, and who probably doesn't have time to reply to your email in any case. See, for example, <http://lwn.net/1999/1209/a/lucent.html>

6. [Troubleshooting](#)

So you've read through this document, the [Modem-HOWTO](#), and the [PPP Howto](#), are pretty sure that your modem matches one of the drivers available, but it still doesn't work? There are a number of points in the process at which something could break down. Marvin Stodolsky writes:

Linux generally maintains records of networking connections which are very useful in troubleshooting problems. Their particular filenames vary with both the Linux distribution and Dial-in software. For both your own trouble shooting and queries for help to a list, it will be useful if you accumulate the information requested below. As Root, start a script record named say, Modem test. After this script is terminated with "exit," copy it out of your Linux partition for transmission to the List which may aid you. Change to the directory in which the modem install scripts are located. Below, # are explanatory comments.

```
# start the recording,  
script ModemTest.txt
```

Linmodem-Mini-HOWTO (DRAFT)

```
# type in as much info on your Modem card as you have
echo winmodem name, manufacturer, designation, and chip if possible
# this gives your current kernel version
uname -r
# this gives information on your serial ports
setserial -agv /dev/ttyS*
# this information on your interrupts (irq)
cat /proc/interrupts
# show the contents of your module installation script (insert script name):
cat ScriptName
# Check if your script is executable:
ls -l ScriptName
# a response is OK if it has "x" such as below:
# -rwxrw-rw- 1 root      root   654 Jan  6  2000 ltinst
# otherwise make it executable with:
chmod o+x ScriptName
# verify with
ls -l ScriptName
# if ScriptName has not been successfully run before under this kernel
# run it with:
./ScriptName
# what is the symbolic link /dev/modem set to:
ls -l /dev/modem
# What is the DeviceName specified in the ScriptName (/dev/ttyS14 or ...?)
echo DeviceName
# what is your modem driver name? Something like DriverName.o
# with the ".o" indicating it is a compiled binary
echo This is my DriverName.o
# if should have been inserted in the Modules Path
# Try to display it there with:
find /lib/modules | grep DriverName
# Is DriverName among the modules installed in the running kernel?
lsmod
# if not try a simple insertion:
insmod ./DriverName.o
# or if it was in the Modules Path, the following will suffice:
insmod DriverName
# check for insertion:
lsmod
# if not inserted, try forcing:
insmod -f ./DriverName
# list your inserted modules again.
lsmod
# If DriverName is NOT listed,
# there is an incompatibility between modem hardware, driver and kernel.
# Further effort will be of No use.
# If DriverName is listed, let's do a bit more information.
# You may first wish to rerun the configuration utility
# used to setup dial-in connections for your Linux installation.
# Remember to edit your PassWord from this record later.
# You will probably be queried for the following information
# which you should have ready:
#Port to be used (/dev/modem or /dev/ttySn),Dial-inNumber, UserName, PassWord.
# Run your configuration utility.
YourSetUpConf
# To stop recording
exit
```

If dialin was not successful, append to this a record from your log file. As an example, a section of a `/var/log/syslog` from a Debian Linux system is below.

Linmodem-Mini-HOWTO (DRAFT)

```
Aug 21 08:35:41 koala kernel: CSLIP: code copyright 1989 Regents of the University of California
Aug 21 08:35:41 koala kernel: PPP: version 2.3.7 (demand dialling)
Aug 21 08:35:41 koala kernel: PPP line discipline registered.
Aug 21 08:35:42 koala kernel: registered device ppp0
Aug 21 08:35:42 koala pppd[1539]: pppd 2.3.11 started by root, uid 0
Aug 21 08:35:43 koala chat[1545]: abort on (BUSY)
Aug 21 08:35:43 koala chat[1545]: abort on (NO CARRIER)
Aug 21 08:35:43 koala chat[1545]: abort on (VOICE)
Aug 21 08:35:43 koala chat[1545]: abort on (NO DIALTONE)
Aug 21 08:35:43 koala chat[1545]: abort on (NO DIAL TONE)
Aug 21 08:35:43 koala chat[1545]: abort on (NO ANSWER)
Aug 21 08:35:43 koala chat[1545]: send (ATZ^M)
Aug 21 08:35:43 koala chat[1545]: expect (OK)
Aug 21 08:35:43 koala chat[1545]: ATZ^M^M
Aug 21 08:35:43 koala chat[1545]: OK
Aug 21 08:35:43 koala chat[1545]: -- got it
Aug 21 08:35:43 koala chat[1545]: send (ATQ0V1E1S0=0&C1&D2S11=55+FCLASS=0^M)
Aug 21 08:35:44 koala chat[1545]: expect (OK)
Aug 21 08:35:44 koala chat[1545]: ^M
Aug 21 08:35:44 koala chat[1545]: ATQ0V1E1S0=0&C1&D2S11=55+FCLASS=0^M^M
Aug 21 08:35:44 koala chat[1545]: OK
Aug 21 08:35:44 koala chat[1545]: -- got it
Aug 21 08:35:44 koala chat[1545]: send (ATDT17574238738^M)
Aug 21 08:35:44 koala chat[1545]: expect (CONNECT)
Aug 21 08:35:44 koala chat[1545]: ^M
Aug 21 08:36:16 koala chat[1545]: ATDT17574238738^M^M
Aug 21 08:36:16 koala chat[1545]: CONNECT
Aug 21 08:36:16 koala chat[1545]: -- got it
Aug 21 08:36:16 koala chat[1545]: send (\d)
Aug 21 08:36:17 koala pppd[1539]: Serial connection established.
Aug 21 08:36:17 koala pppd[1539]: Using interface ppp0
Aug 21 08:36:17 koala pppd[1539]: Connect: ppp0 <--> /dev/ttyS14
Aug 21 08:36:18 koala pppd[1539]: sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x64acd5df> <proto 0x0>]
Aug 21 08:36:18 koala pppd[1539]: rcvd [LCP ConfReq id=0x1 < 00 04 00 00> <mru 1524> <asyncmap 0x000000>]
Aug 21 08:36:18 koala pppd[1539]: sent [LCP ConfRej id=0x1 < 00 04 00 00> < 11 04 05 f4> < 13 09 00 00>]
Aug 21 08:36:18 koala pppd[1539]: rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x64acd5df> <proto 0x0>]
Aug 21 08:36:18 koala pppd[1539]: rcvd [LCP ConfReq id=0x2 <mru 1524> <asyncmap 0xa0000> <auth protocol 0x0>]
Aug 21 08:36:18 koala pppd[1539]: sent [LCP ConfAck id=0x2 <mru 1524> <asyncmap 0xa0000> <auth protocol 0x0>]
Aug 21 08:36:18 koala pppd[1539]: sent [LCP EchoReq id=0x0 magic=0x64acd5df]
Aug 21 08:36:18 koala pppd[1539]: sent [PAP AuthReq id=0x1 user="stodolsk" password=<hidden>]
Aug 21 08:36:19 koala pppd[1539]: rcvd [LCP EchoRep id=0x0 magic=0x0]
Aug 21 08:36:19 koala pppd[1539]: rcvd [PAP AuthAck id=0x1 ""]
Aug 21 08:36:19 koala pppd[1539]: sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <compress VJ 0f 01>]
Aug 21 08:36:19 koala kernel: PPP BSD Compression module registered
Aug 21 08:36:19 koala kernel: PPP Deflate Compression module registered
Aug 21 08:36:19 koala pppd[1539]: sent [CCP ConfReq id=0x1 <deflate 15> <deflate(old#) 15> <bsd version 0x0>]
Aug 21 08:36:19 koala pppd[1539]: rcvd [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 10.65.9.14>]
Aug 21 08:36:19 koala pppd[1539]: sent [IPCP ConfAck id=0x1 <compress VJ 0f 01> <addr 10.65.9.14>]
Aug 21 08:36:19 koala pppd[1539]: rcvd [IPCP ConfNak id=0x1 <addr 207.172.212.104>]
Aug 21 08:36:19 koala pppd[1539]: sent [IPCP ConfReq id=0x2 <addr 207.172.212.104> <compress VJ 0f 01>]
Aug 21 08:36:19 koala pppd[1539]: rcvd [LCP ProtRej id=0x3 80 fd 01 01 00 0f 1a 04 78 00 18 04 78]
Aug 21 08:36:19 koala pppd[1539]: rcvd [IPCP ConfAck id=0x2 <addr 207.172.212.104> <compress VJ 0f 01>]
Aug 21 08:36:19 koala pppd[1539]: Cannot determine ethernet address for proxy ARP
Aug 21 08:36:19 koala pppd[1539]: local IP address 207.172.212.104
Aug 21 08:36:19 koala pppd[1539]: remote IP address 10.65.9.14
Aug 21 08:36:19 koala pppd[1539]: Script /etc/ppp/ip-up started (pid 1548)
Aug 21 08:36:20 koala pppd[1539]: Script /etc/ppp/ip-up finished (pid 1548), status = 0x0
```

7. [Appendix: Useful Scripts and Other Information](#)

7.1 Fixscript: Changes binary module kernel version information (Mark Spieth)

The following script edits the kernel version information of a module to match the running kernel. Note that this does not update the module code to fix any actual kernel API discrepancies, but simply masquerades the module as having been compiled under a different version. This allows the module's insertion without warnings using a plain `insmod` call, but by no means guarantees the module's functioning.

The script is given below, but is also available for download at <http://walbran.org/sean/linux/stodolsk/fixscript>

Note that the script must be made executable by something like `chmod +x fixscript`.

The script:

```
#!/bin/bash
# To make this file executable:  chmod +x fixscript
# This is a very slight edit of the ltfixscript contributed to
# the Linmodems Newslst
# From - Sun Jul 23 04:27:38 2000
# From: "Mark Spieth" <mark at digivation.com.au>
# To: <discuss@linmodems.org>
# Subject: ltmodem symbols and version fixed
# Date: Sun, 23 Jul 2000 12:39:44 +1000
# Organization: Digivation Pty Ltd
```

```
echo "Fixscript V1.21"
```

```
if [ -z "$1" -o -z "$2" ]; then
cat <<END
This script changes version number tags of binary
kernel modules to match the version of the currently
running kernel. It also renames any symbol that the
current kernel cannot resolve into their equivalent
resolvable symbols.
```

```
For inserting binary modules into kernels, the fixed
```

```
module can be inserted with:
```

```
insmod module
```

```
which is used in automated kernel module management,
rather than forcing module loading with:
```

```
insmod -f module
```

```
which is necessary when kernel and module versions
are not matched.
```

```
WARNING! This change is purely cosmetic, and the use
of version matched binaries whenever possible is
strongly advised. It may crash your kernel due to
```

inconsistencies in data structures between the kernel as it stands and the headers used to originally compile the module being fixed. No guarantees are given or implied under any circumstances.

GNU objcopy version 2.9.5 or later is required; this is provided as part of the binary utilities packages such as the Debian binutils.deb

```
USAGE: fixscript input-file output-file
END
exit 1
fi
```

```
MI=/tmp/modinfo
DI=/tmp/depinfo
```

```
#new kernel version modinfo section
echo -ne "kernel_version="\`uname -r`\`" > $MI
```

```
#build the objcopy command
```

```
CMD="objcopy"
```

```
#for i in `depmod -e $1 | grep -vE "^$1:"` ; do
```

```
depmod -e $1 2>$DI
```

```
AWKSTR=\$2
```

```
for i in `awk "{print $AWKSTR}" $DI | grep _R` ; do
```

```
echo doing $i
```

```
il=`echo $i | awk '{
  gsub(/_R[0-9a-fA-F]+/, "");
  printf("%s", $1);
}'`
```

```
echo "il=" $il
```

```
echo " trunc="\`$il\`"
```

```
searchstr=$il_R
```

```
echo "searchstr=" $searchstr
```

```
new=`cat /proc/ksyms | grep $searchstr | awk "{print $AWKSTR}"`
```

```
echo " new=$new"
```

```
CMD="$CMD --redefine-sym=$i=$new"
```

```
done
```

```
#replace the modinfo section with the new one
```

```
CMD="$CMD --remove-section=.modinfo --add-section=.modinfo=$MI"
```

```
CMD="$CMD $*"
```

```
#run the command
```

```
echo "CMD:" $CMD
```

```
$CMD
```

```
#remove the section file
```

```
rm -f $MI
```

```
rm -f $DI
```

7.2 Coincident insertion and removal of ppp related modules (Marvin Stodolsky)

The Lucent driver installation script defaults to having the module loaded upon boot, by appending the line

```
insmod -f ltmodem
```

to the end of the initialization script `/etc/rc.d/rc.local`. If you have, i.e., an ESS modem, you could replace "ltmodem" with "esscom" above to have your module automatically loaded on boot. Note, however, that the initialization scripts differ with different Linux distributions, so you will need to find, modify, or create the appropriate script(s) for your particular setup.

However, many users prefer to run "lean kernels" which only have the auxiliary modules inserted when necessary. Below are examples of scripts for starting and stopping an online session using a Lucent winmodem with the ltmodem.o driver. It can be applied to many other drivers by simply replacing "ltmodem" with "YourModemDriver".

Modemup:

```
#!/bin/sh
# This script inserts the kernel modules supporting
# WinModems using the Lucent ltmodem.o module.
# Save as /usr/local/bin/Modemup, then
#   chmod a+x /usr/local/bin/Modemup
# to make it executable.
# Since insmod & rmmmod require root permission, permission for an
# ordinary User must be given under secure-su or sudo.
#
# the ltmodem.o driver must be within /lib/modules/kernel-version/misc
#
# When kernel-source-2.?.?? code becomes available for ltmodem.o ,
#   forced "-f" insertion should be removed from the following line:
/sbin/insmod -f ltmodem
# and the complaint about version mis-match will also then disappear.
insmod slhc
# is needed to support ppp
insmod ppp
# if you are using a ppp.o which is not version matched with the kernel
#   insmod -f ppp
# may be necessary instead
echo Loaded kernel modules are:
lsmod
# An automatic start of the ppp connection is specified
# by entering the command that starts your online session such as:
#   wvdial, pon, ppp-on, kppp or
# Whatever
## End Modemup
```

Modown:

```
#!/bin/sh
# /usr/local/bin/Modown ends a pppd session and does cleanup.
# Save as /usr/local/bin/Modown, then
#   chmod a+x /usr/local/bin/Modown
# Starting pppd session related modules are:
#   ppp_deflate          39108   1 (autoclean)
#   bsd_comp             3664   0 (autoclean)
#   ppp                  19916   2 (autoclean) [ppp_deflate bsd_comp]
```

```

# slhc                4200  1  (autoclean) [ppp]
# ltmodem             452936 1
# NOTE THAT ltmodem did NOT acquire autoclean status
echo " "
echo Terminating ppp0 with poff
poff
sleep 1
# is a pause to let the poff process to terminate, after which
/sbin/rmmod ltmodem
# removes ltmodem.o module
echo " "
echo Removed module    ltmodem
# /sbin/lsmmod
# echo " "
# echo A pause before removing modules ppp_deflate bsd_comp, then ppp slhc
sleep 1
/sbin/rmmod ppp_deflate bsd_comp
/sbin/rmmod ppp slhc
/sbin/lsmmod
# but doesn't remove the sound related modules called up. Thus
echo " "
echo Removing sound related module called by LTmodem, soundcore
echo " " if not otherwise in use.
/sbin/rmmod soundcore
echo Remaining modules in kernel-`uname -r` are:
/sbin/lsmmod
# displays remaining modules
echo " "
## End Modown

```

7.3 Example PPP scripts (Marvin Stodolsky)

Example PPP scripts configured for a Debian installation by pppconfig are:

`/etc/ppp/peers/provider:`

```

# This optionfile was generated by pppconfig 2.0.5.
hide-password
noauth
connect "/usr/sbin/chat -v -f /etc/chatscripts/provider"
debug
/dev/ttyS14 # port used by the Lucent Winmodem
115200
defaultroute
noipdefault
user NameAtIP
remotename provider
ipparam provider

```

`/etc/chatscripts/provider:`

```

# This chatfile was generated by pppconfig 2.0.5.
# Please do not delete any of the comments. Pppconfig needs them.
#
# ispaath PAP
# abortstring
ABORT BUSY ABORT 'NO CARRIER' ABORT VOICE ABORT 'NO DIALTONE' ABORT 'NO DIAL TONE' ABORT 'NO ANSWER'
# modeminit

```

```

'' ATZ
OK ATQOV1E1S0=0&C1&D2S11=55+FCLASS=0
# ispnnumber
OK-AT-OK ATDT3019178111
# ispconnect
CONNECT \d\c
# prelogin
# ispname
# isppassword
# postlogin
# end of pppconfig stuff

```

7.4 PCTel Module Parameters: Country Code

The following is quoted from one of the PCTel readme files. My interpretation of this is that you should do something like

```
insmod pctel.o country_sel=7
```

(depending on your country code, below). If this works for you (or doesn't), please let me know at [sean\(at\)walbran.org](mailto:sean(at)walbran.org)

Set and report country code.

This driver takes a module parameter to setup the correct country code setting for various country's telephone networks and it also can report back the country code been set.

Here are the two versions for country_code selection and reporting:

VERSION #1:

To set country code:

```
"country_sel_rep sel 7" will sets the country code to 7.
```

To query the driver for the currently set country code:

```
"country_sel_rep rep" returns the current country code as the exit code.
```

VERSION #2:

To set country code:

```
"country_sel 7" to set the country code to 7.
```

To query the driver for the currently set country code:

```
"country_rep" return the current country code as the exit code.
```

country_code	country_name
1	USA
2	FRANCE
3	GERMANY
4	ITALY
5	SWEDEN
6	UK
7	JAPAN

Linmodem-Mini-HOWTO (DRAFT)

8	AUSTRALIA
9	SPAIN
10	TAIWAN
11	SINGAPORE
12	KOREA
13	SWITZERLAND
14	NORWAY
15	NETHERLANDS
16	BELGIUM
17	CANADA
18	IRELAND
19	PORTUGAL
20	POLAND
21	HUNGARY
22	FINLAND
23	DENMARK
24	AUSTRIA
25	S.AFRICA
26	CTR21 COUNTRIES
27	CHINA
28	MALAYSIA
29	LUXUMBURG
30	GREECE
31	ICELAND
32	NEW ZEALAND
33	BRAZIL
