The Linux iPAQ HOWTO

Instructions for Familiar v0.5.3

Michel Stempin

michel.stempin@wanadoo.fr

The Linux iPAQ HOWTO: Instructions for Familiar v0.5.3

by Michel Stempin

Revision History

Revision v1.1 2002-08-30 Revised by: ms
Corrections as suggested by Jamey Hicks
Revision v1.0 2002-08-11 Revised by: ms
Initial version

Table of Contents

Preface

Today, Personal Digital Assistants (PDAs) are getting more and more popular, allowing the power of a 5-year old PC to fit in your hand, for the high-end models. Most of these PDAs come with an operating system and a set of softwares such as a contact management, a diary, a mail reader, a task manager, a calculator, a terminal emulator, a file explorer, a multimedia reader, a spreadsheet, a word processor, and of course, games.

Unfortunately, all these softwares are proprietary, and only licensed to the final user, who have no way to modify or improve them. Even worse, software development is difficult too, since proprietary development tools are required, which mean in turn that a programmer has to learn this tool and the provided Application Programming interface (API).

Hopefully, an alternative to this sad situation exists. Free Software, best represented by the GNU/Linux operating system, has proven its capability to provide quality tools through knowledge sharing, community support and good will. Moreover, the power of creativity is given back to the developers, allowing them to write programs using whatever API they prefer best, or even write their own if they want to.

This document proposes you to free your iPAQ from commercial software licenses by replacing the proprietary softwares by a GNU/Linux operating system and a set of application softwares that will provide the same (or even better...) tools than the original ones. Not only will it give you a taste of freedom, but it will also bring you back the joy of creativity, knowledge, and fun.

I hope you'll enjoy reading this document as much as I have enjoyed writing it.

1. Introduction

As an electronic engineer, I always enjoyed minimalistic systems. They always provide a kind of challenge to your accepted rules, as your dogmas tend to collapse when facing the crude reality. On minimal systems, this phenomenon tends to happen even quicker than on complex ones. One other interesting feature of this minimalistic approach is that it tends to keep things humanly manageable, which is comfortable, as today's computer science requires you to master a broad set of concepts, tools and computer languages...

Back in 1980, my first computer was a Goupil 2, quickly followed by an Apple II+, where I started programming. Having undergone dozens of computers, computer languages, and proprietary operating systems, I discovered Linux in 1994, when my company at that time (Lectra Systèmes), decided to take a Unix-like operating system on a PC platform as its base for application development. After careful evaluation and pushed by Pierre Ficheux and Daniel Roche, Linux was preferred to other commercial products, first because of its good performance but also because of its price. At that time, emails were not on everybody's computer, and the computer community was just emerging from the universities. We thus followed the growth of Free Software, and as an engineer, I found working like that was fun.

Since then, I tried to keep on working with GNU/Linux as much as possible, although I should not be considered as a Free Software activist. I am just pragmatic, and I still use commercial software whenever I don't find the corresponding free tools. I know, I should write my own, but sometimes, I am just too lazy, or focused on the more important goal that required this tool. I don't feel too guilty for that, as I try to participate to the Free Software movement whenever I can.

I got a first Uniden® PDA, but I should say that it belongs now to one of my brothers, since I was very disappointed by the fact that there was no (at least documented) way to change the software on it, or even to write programs for it without having to learn tons of things about DLL's major enhancement (I already played with shared libraries on my

Amiga 500 in 1987...), and the virtues of Hungarian notation (I still find this is one of the silliest things ever invented, especially since polymorphisms in Object Oriented Programming).

I was a kind of shocked when I first saw my actual boss, Jean-Marc Lange, play with its iPAQ PDA. But what really stunt me is when I discovered by browsing on the Web that some crazy persons *did it*. Yes, thanks to what was formerly named Compaq® (now Hewlett Packard®), and a lot of good-willed persons, it was possible to put Free Software on this machine.

I took this as a challenge, and I think that writing this document on converting an iPAQ to GNU/Linux is my best effort to help the Free Software Community.

This document was also a good exercise to practice DocBook. If you read these lines, it means that I was able to master it!

2. Audience of this document

This text should be considered as a step-by-step approach of converting an iPAQ to a GNU/Linux operating system, onto which the Opie application suite will be installed. Of course, you can choose to install some different softwares (like the X11 system), but I chose this alternative to provide a strict replacement to the original softwares, and I considered Opie as one of the most advanced projects.

Although a lot of things on this can be found on the Internet (especially at http://www.handhelds.org), I think there is a need for a self-contained source of information that can help everybody achieve this transformation successfully.

3. Organization of the material

The document introduces its topics in a logical order, corresponding to the steps required to put GNU/Linux on an iPAQ.

Chapter 1 begins with the proper setup of connection of an iPAQ running PocketPC to a GNU/Linux host.

Chapter 2 covers the backup of the original system and user data.

Chapter 3 quickly shows how to setup a terminal connection to the iPAQ, that will be used later to enter commands for the bootloader and the first stage of the GNU/Linux installation.

Chapter 4 shows the bootloader's installation procedure.

Chapter 5 continues with the installation of a minimum GNU/Linux bootstrap

The GNU/Linux bootstrap will be used to install the desired packages. At that point, we should consider that we have a running system, though not very attractive from a user point of view.

Chapter 6 provides an in-depth view on how to setup an USB network between the GNU/Linux host and the iPAQ.

Chapter 7 finishes the installation of the Familiar GNU/Linux distribution.

Chapter 8 covers the installation of the basic Opie packages, providing us with a more comfortable graphical interface, then completes the Opie installation by describing the installation of the remaining Opie packages.

Chapter 9 covers the optional step of restoring the original software back on the iPAQ, just in case where your boss want its original system back...

4. Prerequisite

There is no prerequisite background to use this document, beside being able to launch a shell command under GNU/Linux either in a text display or in an Xterm, to be able to edit a text file using whatever text editor you are familiar with, and know how to use the iPAQ user interface.

For this guide, I used a Compaq (now HP) iPAQ 3850 iPAQ running PocketPC 2002, but all the given examples should work without problem on other iPAQs too, as long as they have a serial connection capability. I will be glad to add new targets if you succeeded in doing it on other platforms.

Of course, you will need a computer running the Linux operating system. For this guide, I used the Red Hat 7.2 distribution, but it should work on others, with minor changes due to distribution-specific glitches.

This computer will need to be connected to the Internet, as the transformation requires downloading some files for the iPAQ. It is also possible to pre-download them and work locally.

In order to download files to the iPAQ, the computer will also have to run either an FTP and/or an HTTP (like apache) server.

5. Versions

This document describes the installation of the following versions:

- Backup and flashing utility - BootBlaster_1.18
- Bootloader - bootldr-2.18.01
- Linux distribution - familiar-0.5.3
- Graphical interface - opie_1.5.0-20020515

6. Sources of further information

There is much useful information available on the Internet; the following is a sampling. Internet sites, of course, tend to be highly volatile while printed books are hard to update. Thus, this list should be regarded as being somewhat out of date.

http://www.handhelds.org

This site is the home of GNU/Linux on iPAQs. You'll find the distributions, tools, packages and related information.

7. Legal information

7.1. Copyright

7.2. NO WARRANTY

7.3. Trademarks

8. Online version

The original DocBook SGML version of this document, as well as some various other formats (DVI, multiple-files HTML, single-file HTML, PDF, PostScript, RTF, and TeX) can be found at:

http://mstempin.free.fr/linux-ipaq

9. Conventions used in this document

The following is a list of the typographical conventions used in this document:

New Term	Used for new terms
`Filename`	Used for file and directory names
Command	Used for program and command names

`Constant Width`	Used in the text to indicate words that appear in code or other literal strings
`Listing`	Used in examples to show the contents of files
`Output`	Used in examples to show the output from commands
`bash#`	Used in examples to show a "root" shell prompt
`bash$`	Used in examples to show a normal user shell prompt
`Replaceable`	Used to indicate variable options, keywords, or text that the user is to replace with an actual value
`User Input`	Used in examples to show commands or other text that should be typed literally by the user

Pay special attention to notes set apart from the text with the following icons:

Note: This is a tip. It contains useful supplementary information about the topic at hand.

Warning

This is a warning. It helps you solve and avoid annoying problems.

10. Feedback

I have tested and verified the information in this document to the best of my ability, but you may find that features have changed (or even that I have made mistakes!). Please let me know about any errors you find, as well as your suggestions for future editions, by writing to:

`<michel.stempin@wanadoo.fr>`

Feel free to send comments or ask technical questions about this document to the same address.

I also have a web page for the document, where I list errata, examples, or any additional information. You can access this page at:

http://mstempin.free.fr/linux-ipaq

11. Acknowledgments

I would like to thank Sophie Calot for understanding the long hours I spent on writing this document, having confidence in me that I was doing something *important*, even if she is not the kind of girl that will use it...

I would like to thank also everybody at HandHelds.org (http://www.handhelds.org). Without their first work, nothing would have been possible. This document is only another way of explaining what they discovered, I simply hope that I added some valuable information and a usefull approach of the subject.

Chapter 1. Connecting a PocketPC iPAQ to a GNU/Linux host

1.1. Creating a connection on the GNU/Linux host

Although the iPAQ's cradle has an USB cable, we will not be able to use it while the iPAQ is running PocketPC, as its use is limited to the proprietary ActiveSync program.

The only possibility will be to use the slower serial cable attached to the cradle and PPP (Point to Point protocol), which provides a way to use TCP/IP over a serial line. In Chapter 6, as GNU/Linux is installed, we will setup a faster USB network.

For now, we will have to setup PPP both on the PocketPC iPAQ and on the GNU/Linux host.

1.1.1. pppd configuration

On the GNU/Linux host, PPP is implemented by the **pppd** command, which uses either options specified after the name **pppd** on the command line, separated by space characters, or a configuration file. This method will be preferred, as it is tedious to type the same things over and over.

The configuration file can have any name, but is best if placed in the directory `/etc/ppp/peers`, where **pppd** will look for its configuration files. In this document, I will use a file named `/etc/ppp/peers/PocketPC`.

The content of this file is as follows:

```
# /etc/ppp/peers/PocketPC -*- sh -*- pppd options for direct iPAQ connection
# created 08-May-2002 Michel Stempin <michel.stempin@wanadoo.fr>
# autodate: 08-May-2002
# autotime: 16:38

# The serial line to use. Use either "/dev/ttyS0" for "COM1" or "/dev/ttyS1"
# for "COM2" PC ports.
/dev/ttyS1

# Set the speed to 115200 bps.
115200
# As PocketPC considers us as a modem ;-), we have to pretend by sending
# the expected answers.
# For this, we use the "chat" program and "expect/send" pairs. You can specify
# the "-v" option to /usr/sbin/chat if you want to debug the "modem" initial dialog.
# Basically, the chat is (\r means a <CR>, i.e. a Carriage Return character):
#     - expect ATZ<CR> (reset modem command)
#     - send OK<CR>
#     - expect AT<CR> (are you here? command)
#     - send OK<CR>
#     - expect ATE0V1&C1&D2<CR> (disable echo,
#                                 send ASCII result codes,
#                                 Carrier Detect pin is used,
#                                 Data Terminal Ready pins reflects online state),
#     - send OK<CR>
```

```
#     - expect AS7=120<CR> (timeout 120s waiting for Carrier)
#     - send OK<CR>
#     - expect ATX3<CR> (ignore dial tone, report BUSY)
#     - send OK<CR>
#     - expect ATDT<CR> (dial using tone modulation)
#     - send CONNECT<CR>
connect "/usr/sbin/chat 'ATZ\r' OK 'AT\r' OK 'ATE0V1&C1&D2\r' \
OK 'ATS7=120\r' OK 'ATX3\r' OK 'ATDT' CONNECT"

# async character map -- 32-bit hex; each bit is a character
# that needs to be escaped for pppd to receive it.  0x00000001
# represents '\x01', and 0x80000000 represents '\x1f'.
asyncmap 0

# Do not require PocketPC to authenticate itself before allowing network
# packets to be sent or received.
# NOTE: This option is only available to the "root" user.
noauth

# No hardware flow control (i.e. RTS/CTS) to control the flow of data
# on the serial port, as the iPAQ seems not to handle it.
nocrtscts

# Don't use the modem control lines
local

# Specifies that pppd should use a UUCP-style lock on the serial device
# to ensure exclusive access to the device.
lock
# Don't fork to become a background process (otherwise pppd will do so
# if a serial device is specified).
-detach

# Increase debugging level (same as -d).  If this option is given, pppd
# will log the contents of all control packets sent or received in a
# readable form.  The packets are logged through syslog with facility
# daemon and level debug. This information can be directed to a file by
# setting up /etc/syslog.conf appropriately (see syslog.conf(5)).  (If
# pppd is compiled with extra debugging enabled, it will log messages
# using facility local2 instead of daemon).
#debug

# Enable debugging code in the kernel-level PPP driver.  The argument n
# is a number which is the sum of the following values: 1 to enable
# general debug messages, 2 to request that the contents of received
# packets be printed, and 4 to request that the contents of transmitted
# packets be printed.
#kdebug 7

# Sets the IP addresses on both side of the link.
192.168.1.100:192.168.1.101
```

1.1.2. Naming the iPAQ

As a convenience, we will add the following line to the `/etc/hosts` file:

```
192.168.1.101 ipaq ipaq.localdomain
```

This will allow us to connect to the iPAQ using its name, instead of its IP address. This can be achieved using you favorite text editor, or with the following simple command:

```
bash# echo "192.168.1.101 ipaq ipaq.localdomain" >> /etc/hosts
```

1.2. Creating a connection on the iPAQ

1.2.1. Disabling ActiveSync

The ActiveSync program is a nice tool if you want to synchronize your PocketPC to a Windows host. Alas, as it uses a proprietary protocol, it is of no use to us. Worse, it will prevent us from achieving a connection to a GNU/Linux host by capturing some characters on the serial line.

We thus have to disable it, following these simple instructions:

1. In the "Start" menu, tap on the "ActiveSync" entry. The ActiveSync main window will show up
2. Tap on the "Tools" menu, then on the "Options..." entry
3. Uncheck the "Synchronize automatically..." box and tap "ok". We are back to the ActiveSync main screen
4. Tap on "X" to close the program.

In order for this modification to take effect, I recommend you switch the iPAQ off, then back on (I spent hours figuring out why it was not working ;-)).

1.2.2. Creating a new "modem" connection

For PocketPC, a serial connection has to be a modem... This is how to do it, using the standard PocketPC parameters settings:

1. In the "Start" menu, tap on the "Parameters" entry
2. Tap on the "Connections" tab
3. Tap on the "Connections" icon
4. Select "Internet parameters" in the first pulldown menu, then tap on "Modify..."
5. Tap on "New..."
6. Enter a name for the connection ("Linux" for example), then select "115200" in the "Baud rate" pulldown menu, then tap on "Advanced..."
7. Select 8 data bits, parity "none", 1 stop bit, "None" flow control, and uncheck all the boxes. Tap on the "TCP/IP" tab

8. Check "IP address allocated by server", uncheck "Slip", and check both "compression" boxes. Tap on the "Servers" tab
9. Check "Address allocated by server", then tap "ok". We are back to the new connection setup screen
10. Tap on "Next..."
11. Enter "0" as the phone number. Tap on "Next"
12. Check "Cancel call..." and uncheck "Wait for tone before dialing", then tap on "Finish"
13. Our new connection is created. Tap on "ok". We are back to the main screen
14. Tap on "ok" to exit "connections"
15. Tap on "X" to exit "parameters"

1.3. Connecting the iPAQ to the PC

1.3.1. On the GNU/Linux host

1.3.1.1. Manual connection

On the GNU/Linux host, you can launch the connection manually by typing:

```
bash# /usr/sbin/pppd call PocketPC
```

This launches **pppd**, calling the iPAQ with the option file we created in Section 1.1.1.

1.3.1.2. Automatic connection

Alternatively, you can launch it automatically, by adding the following lines to your `/etc/inittab` file:

```
# Run a connection to the iPAQ
ppp:2345:respawn:/usr/sbin/pppd call PocketPC
```

This last line tells the **init** process (father of all) to launch the same command whenever it enters into run levels 2,3,4 or 5, and to respawn (relaunch) it when it dies.

You can enter these two lines either using your favorite text editor, or by issuing the following commands:

```
bash# echo "# Run a connection to the iPAQ" >> /etc/inittab
bash# echo "ppp:2345:respawn:/usr/sbin/pppd call PocketPC" >> /etc/inittab
```

For this automatic feature to take effect, you have to tell the running **init** process to re-read its configuration by typing:

```
bash# init q
```

You can observe if **pppd** is running by checking the `/var/log/messages` file.

1.3.2. On the iPAQ

You just have to open your "modem" connection:

1. From the "Start" menu, tap on the "Parameters" entry
2. Tap on the "Connections" tab
3. Tap on the "Connections" icon
4. Tap on "Connect"
5. If a "Login/Password/Domain" dialog appears, just check "Remember password" and tap on "OK"
6. A "Connecting..." window will pop up, soon replaced by a "Connected" window, that will close automatically after 5 sec., unless you tap on "Hide" before...
7. Tap on "ok" to close the "connections" screen
8. Tap on "X" to close the "parameters" screen
9. The connection status is visible in the top bar as two small arrows. By tapping on these, a popup window will appear, that gives you the ability to close the connection by tapping on "Finish".

1.4. Transferring files between the iPAQ and the GNU/Linux host

As is, PocketPC has only one tool that uses the standard TCP/IP protocols able to transfer files: it is the Internet Explorer Web browser. Although its usage is straightforward, it has two drawbacks:

1. it requires a running Web server on the GNU/Linux host
2. it is only able to transfer file in one direction (from the GNU/Linux host to the iPAQ)

That is, it does not fulfill our requirements: to be able to exchange files in both directions.

The workaround I have found is to use it initially to download a minimum FTP server on the PocketPC iPAQ, which will be used for all subsequent transfers.

1.4.1. Downloading and running the mini FTP server

As we will use a Web browser on the iPAQ, we need first to set up a Web server on the GNU/Linux host, place the mini FTP server so it can be accessed from the iPAQ, then download it and install it.

1.4.1.1. Setting up a Web server on the GNU/Linux host

Giving all the details required to set up a Web server on a GNU/Linux host is beyond the scope of this guide. Maybe you already have done it for another task, or your GNU/Linux distribution has already done it for you.

So, I will only describe what is required to set up the Apache Web server. I will assume that you already have installed the required package for it, using your distribution's packaging tool.

We first have to locate the Apache configuration file, by running the following command:

```
bash# find etc -name httpd.conf -print
/etc/httpd/conf/httpd.conf
```

This tells us that the file is located in the directory `/etc/httpd/conf`. In this directory, we will find where the *root* of the Web server is; that is, in which directory we will have to place files so they can be seen from the iPAQ. I search on the whole directory, as the information is not necessarily in the `httpd.conf` file in old Apache versions:

```
bash# grep DocumentRoot /etc/httpd/conf/*
/etc/httpd/conf/httpd.conf:# DocumentRoot: The directory out of which you will serve your
/etc/httpd/conf/httpd.conf:DocumentRoot "/var/www/html"
/etc/httpd/conf/httpd.conf:# This should be changed to whatever you set DocumentRoot to.
/etc/httpd/conf/httpd.conf:#    DocumentRoot /www/docs/dummy-host.example.com
/etc/httpd/conf/httpd.conf:#DocumentRoot "/etc/httpd/htdocs"
```

Some other lines that may list errors can be safely ignored. The only interesting thing here is the second line, saying that the *DocumentRoot* is located in the `/var/www/html` directory. This may be different for your system.

Using this information, we will create a directory where we will store the files for the iPAQ: let's call it `iPAQ`:

```
bash# cd /var/www/html
bash# mkdir iPAQ; cd iPAQ
```

We then have to launch the Apache Web server. On my Red Hat 7.2 distribution, I can issue either:

```
bash# service httpd start
```

or

```
bash# /etc/rc.d/init.d/httpd start
```

Depending on your GNU/Linux distribution, you may have to find the right way to do it...

1.4.1.2. Getting the PocketPC mini FTP server

Copy in this directory the `FTPS006AP.ZIP` file (aka, the ftpserv zipped file), downloaded from the following location:

http://www.oohito.com/wince/arm_j.htm

Unzip the archive using the following command:

```
bash# unzip FPS006AP.ZIP
Archive:  FTPS006AP.ZIP
  inflating: ftpsvr.exe
  inflating: ftpsvr.txt
```

We are interested in the `ftpsvr.exe` file.

1.4.1.3. Downloading the mini FTP server on the PocketPC iPAQ

If everything is OK, you should be able to download the `ftpsvr.exe` file by fetching the following URL in the Internet Explorer Web browser on the PocketPC iPAQ:

http://192.168.1.100/iPAQ/ftpsvr.exe

> **Note:** You may have to go into the "Display" menu, and enable the "URL bar" option to be able to enter the URL.

A "Download" popup menu should open, asking you if you want to download the `ftpsvr.exe` file, giving you the option to open it after downloading. I suggest you to check this last option.

On the PocketPC iPAQ, you can now close Internet Explorer and go back to the main "Today" screen. In the lower right corner, you should have a small icon. When you tap on it, a window "FtpSvr for Windows® CE Ver 0.06" should open, that is a little too large on my iPAQ, so you may have to move it around to see it completely.

This window tells you that the FTP server is running, listening on port 21. You have the option to stop it, uninstall it (don't!), or exit.

Everything is ready now to be able to transfer files in both directions.

Whenever you want to restart the mini FTP server, you just have to click on the ftpsvr.exe icon where you saved it after downloading it (normally in the "`My Documents`" folder).

1.4.2. Transferring files using FTP

On the GNU/Linux host, you can either use the command line **ftp**, or your favorite graphical FTP client. I will show here the basic, command line option:

```
bash$ ftp ipaq
Connected to ipaq (192.168.1.101).
220 FtpSvr (Version 0.06).
Name (ipaq:root):
331 Password required for root.
Password:
230 User root logged in.
ftp> passive
Passive mode off.
ftp>
```

> **Note:** Do not enter any name or password, as the mini FTP server on the PocketPC iPAQ is not able to handle them. The **passive** command is because the mini FTP server does not handle the "passive" FTP protocol.

> **Warning**
>
> The mini FTP server should not be used permanently, as it lacks some important security features. You should only turn it on whenever necessary.

You can now transfer files back and forth:

```
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for \467456.FTP(782 bytes).
drwxrwxrwx   1 owner    group      0          Jan  1  1998 iPAQ File Store
-rw-rw-rw-   1 owner    group      76         Sep  4  7:44 467456.FTP
-rw-rw-rw-   1 owner    group      282        Sep  4  7:16 mdmlog0.txt
```

```
-rw-rw-rw-   1 owner    group      72         Sep  4  7:14 mdmlog5.txt
-rw-rw-rw-   1 owner    group      10500      Sep  4  7:15 GCounterFile.mmf
-rw-rw-rw-   1 owner    group      56         Sep  3 20:00 CMMapP
-rw-rw-rw-   1 owner    group      60         Sep  3 20:00 CMMapG
drwxrwxrwx   1 owner    group      0          Sep  3 12:00 Program Files
drwxrwxrwx   1 owner    group      0          Sep  3 12:00 My Documents
drwxrwxrwx   1 owner    group      0          Sep  3 12:00 Temp
drwxrwxrwx   1 owner    group      0          Sep  3 12:00 Windows
226 Transfer complete.
ftp> put whatever
local: whatever remote: whatever
200 PORT command successful.
150 Opening BINARY mode data connection for \whatever.
226 Transfer complete.
6 bytes sent in 8.4e-05 secs (70 Kbytes/sec)
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for \467456.FTP(851 bytes).
drwxrwxrwx   1 owner    group      0          Jan  1  1998 iPAQ File Store
-rw-rw-rw-   1 owner    group      76         Sep  4  7:45 467456.FTP
-rw-rw-rw-   1 owner    group      6          Sep  4  7:45 whatever
-rw-rw-rw-   1 owner    group      282        Sep  4  7:16 mdmlog0.txt
-rw-rw-rw-   1 owner    group      72         Sep  4  7:14 mdmlog5.txt
-rw-rw-rw-   1 owner    group      10500      Sep  4  7:15 GCounterFile.mmf
-rw-rw-rw-   1 owner    group      56         Sep  3 20:00 CMMapP
-rw-rw-rw-   1 owner    group      60         Sep  3 20:00 CMMapG
drwxrwxrwx   1 owner    group      0          Sep  3 12:00 Program Files
drwxrwxrwx   1 owner    group      0          Sep  3 12:00 My Documents
drwxrwxrwx   1 owner    group      0          Sep  3 12:00 Temp
drwxrwxrwx   1 owner    group      0          Sep  3 12:00 Windows
226 Transfer complete.
ftp> get whatever
local: whatever remote: whatever
200 PORT command successful.
150 Opening BINARY mode data connection for \whatever(6 bytes).
226 Transfer complete.
6 bytes received in 1.01 secs (0.0058 Kbytes/sec)
ftp> bye
221 Goodbye
```

Chapter 2. Backing up the PocketPC iPAQ

There are several reasons why you should consider backing up the PocketPC iPAQ before proceeding to the GNU/Linux replacement:

- the operating system replacement is a *destructive* operation, that will erase everything on the machine
- the software that comes pre-installed on the machine is licensed to you (remember you paid for it!), so even if you don't want it, you should keep it as valuable
- if you are not satisfied with GNU/Linux, this will be the only way to go back to the original operating system
- concerning your own data, remember that you are the only one able to re-create it...
- if you experiment on someone else's iPAQ (you probably shouldn't, anyway), you will have to restore everything to its original state

This is why I *urge* you to backup your iPAQ. When it comes to safety procedure, it is better to have suspenders, a belt, and a parachute than nothing...

2.1. Downloading the backup software

The software that we will use to backup everything on the PocketPC iPAQ is included in the software that will be used in chapter Chapter 4 to replace the original bootloader. This software can be downloaded from:

ftp://handhelds.org/feeds/BootBlaster/BootBlaster_1.18.exe

I recommend you also download the corresponding *md5sum* file, so you can check BootBlaster's integrity. The *md5sum* can be downloaded from:

ftp://handhelds.org/feeds/BootBlaster/BootBlaster_1.18.exe.md5sum

Transfer the BootBlaster_1.18.exe file onto the iPAQ, by following the steps in Section 1.3 and Section 1.4.2.

You can check the integrity by running the following command:

```
bash$ md5sum -c BootBlaster_1.18.exe.md5sum
BootBlaster_1.18.exe: OK
```

2.2. Backing up the Bootloader

Once BootBlaster is copied onto the iPAQ:

1. In the "Start" menu, tap on the "Programs" entry.
2. Tap on the "File Explorer" icon to launch File Explorer.
3. Locate the BootBlaster icon (probably somewhere in "`My Documents`") and tap on it to launch it.
4. Tap on the "Flash" menu.
5. Tap on the "Save Bootldr" entry. A popup window will open, asking you to confirm.
6. Tap on "Yes" to confirm.

7. Shortly, a new popup window will appear, stating that the Bootldr segment was successfully saved, giving you the location of the "`saved_bootldr.bin`" file (normally "`\My Documents\saved_bootldr.bin`". Tap on "ok"
8. Tap on the "X" button to close BootBlaster.
9. Tap on the "X" button to close File Explorer.
10. Transfer the `saved_bootldr.bin` file onto the GNU/Linux host, by following the steps in Section 1.3 and Section 1.4.2.

2.3. Backing up PocketPC

This operation is very similar to the previous one:

1. In the "Start" menu, tap on the "Programs" entry.
2. Tap on the "File Explorer" icon to launch File Explorer.
3. Locate the BootBlaster icon (probably somewhere in "`My Documents`") and tap on it to launch it.
4. Tap on the "Flash" menu.
5. Tap on the "Save Wince .gz Format" entry. A popup window will open, asking you to confirm.
6. Tap on "Yes" to confirm.
7. A progress bar will display, and after 5 minutes (on my iPAQ 3850), a new popup window will appear, stating that the Wince segment and Assets were successfully saved, giving you the location of the "`wince_image.gz`" file (normally "`\My Documents\wince_image.gz`", and of the "`asset_image.gz`" file (normally "`\My Documents\asset_image.gz`". Tap on "ok"
8. Tap on the "X" button to close BootBlaster.
9. Tap on the "X" button to close File Explorer.
10. Transfer the `wince_image.gz` and `asset_image.gz` files onto the GNU/Linux host, by following the steps in Section 1.3 and Section 1.4.2.

Chapter 3. Setting up a serial terminal connection to the iPAQ

In this chapter, we will set up a serial terminal connection to the iPAQ that will be used later for:

- taking control of the bootloader to send the GNU/Linux bootstrap image to the iPAQ and flash it
- taking control of the GNU/Linux on the iPAQ while there is no graphical interface

On the GNU/Linux host, we will use minicom as a terminal emulator.

3.1. Shutting down ppp

If you followed the chapters in this book in a logical order, you should have an automatic ppp on the GNU/Linux host, waiting for connections from the iPAQ (see Section 1.3.1.2). You must disable it before proceeding. Comment out the following line in the `/etc/inittab` file by putting a "#" at the beginning of the line, using you favorite text editor:

```
#ppp:2345:respawn:/usr/sbin/pppd call PocketPC
```

You must tell the running **init** process to re-read its configuration by typing:

```
bash# init q
```

The serial line is now free, so we can use the terminal emulator.

3.2. Setting up minicom

Launch minicom in setup mode by typing:

```
bash# minicom -s
```

The "configuration" window will popup. Using the **arrow down** key, move to the "Serial port setup" entry and press **Enter**.

A new popup window will appear, giving you the ability to change the serial port settings. Press the **A** key and enter the serial line to suit your hardware configuration (`/dev/ttyS0` for BIOS port "COM1", `/dev/ttyS1` for BIOS port "COM2"). Press **Enter**

Press **E** to change the serial parameters. A new window "comm parameters" will show up.

In this new window, press **I** to set the speed to 115200bps, and **Q** to use 8 data bits, no parity, and 1 stop bit. Press **Enter** to go back to the previous window.

Press **F** and **G** to set both the hardware and software flow control to "none". Press **Enter** to go back to the main "configuration" window.

Using the **arrow down** key, move down to either "Save setup as dfl" or "Save setup as...", depending if you want to save this setup as the default setup, or save it into an alternate configuration file.

Using the **arrow down key**, move down to "Exit" to leave the configuration mode and go into terminal emulation mode.

Chapter 4. Installing the bootloader

This chapter covers the only *dangerous* step in the whole conversion process. By *dangerous*, I mean that this can really turn your iPAQ into a useless high-tech paperweight! Fortunately, this only happen very seldom, and the risk is close to zero if you follow the instructions carefully.

As a service, Compaq (now HP) Research ensures that it will fix your unit in case it was rendered nonbooting by a failed installation. Of course, this may take a while before you get it back, and this should only be seen as the last resort.

> **Warning**
>
> As this operation may erase everything on your unit, be sure to have a backup as described in Chapter 2 and md5sum-verified `BootBlaster_1.18.exe` and `bootldr-2.18.01.bin` files as described in Chapter 2 and Section 4.1.
>
> Remember, THIS OPERATION IS PERFORMED AT YOUR OWN RISKS.

4.1. Downloading the bootloader

The default bootloader (code name Parrot) is quite sophisticated. It can be accessed by pushing the central "arrow pad" on the iPAQ and resetting by putting the stylus into the narrow hole at the bottom of the case (don't do it!). Then, using a terminal emulator as set up in the previous chapter, you can obtain a command line that allow you to do various things.

Unfortunately, this bootloader is only able to boot the pre-installed operating system, so we have to replace it in order to boot another OS, like GNU/Linux.

Compaq's (now HP's) Cambridge Research Laboratories developed such a bootloader. You can download it from:

ftp://handhelds.org/feeds/bootldr/bootldr-2.18.01.bin

You *must* download the corresponding md5sum file from:

ftp://handhelds.org/feeds/bootldr/bootldr-2.18.01.bin.md5sum

You *must* check the file integrity by running the following command:

```
bash$ md5sum -c bootldr-2.18.01.bin.md5sum
bootldr-2.18.01.bin: OK
```

> **Warning**
>
> Not checking the bootldr's integrity may be catastrophic, as if the file is damaged, you will not be able to boot your iPAQ anymore, transforming it into a *brick*.

Transfer the `bootldr-2.18.01.bin` file onto the iPAQ, by following the steps in Section 1.3 and Section 1.4.2.

> **Note:** A good way to check bootldr's integrity on the iPAQ is to transfer it back onto the GNU/Linux host and check it again with md5sum.

4.2. Flashing the bootloader

If you followed the steps sequentially, you should have `BootBlaster_1.18.exe` and `bootldr-2.18.01.bin` on your PocketPC iPAQ. We will flash the new bootloader by doing the following:

1. In the "Start" menu, tap on the "Programs" entry.
2. Tap on the "File Explorer" icon to launch File Explorer.
3. Locate the BootBlaster icon (probably somewhere in "`My Documents`") and tap on it to launch it.
4. Tap on the "Flash" menu.
5. Tap on the "Program" entry. A warning popup window will open, asking you to confirm.
6. Tap on "Yes" to confirm.
7. A File dialog will open, enabling you to choose the bootloader file you want to use. Select the one you downloaded in Section 4.1
8. A dialog will ask you to confirm the file you want to flash. Tap on "Yes" to continue.
9. *Wait patiently.* The process of flashing the new bootloader takes about 15 seconds. During this time, BootBlaster will display what it is doing:

   a. `Erasing flash...`

   b. `Protecting Wince Partitions...`

      BootBlaster protects the PocketPC partitions before proceeding to bootloader flashing. It looks like PocketPC is still able to boot the machine, even if the bootloader is erased, as long as it PocketPC is not touched.

      Warning

      It is not safe to erase the bootloader if GNU/Linux is installed in flash -- GNU/Linux depends on the bootloader to configure the machine before jumping into GNU/Linux.

      Up to and including this point, it is perfectly safe to reboot your iPAQ. If your machine takes more than 15 seconds to perform this operation, reboot the iPAQ by pushing the recessed reset button at the bottom right and repeat this step.

   c. `Programming Flash...`

      This is the sensitive point, as during this time, your bootloader is not in a consistent state. This operation is very short, so the chance that you get a power loss while doing it is small.

      Warning

      If your iPAQ hangs while programming, send e-mail to <`bootldr@handhelds.org`> and/or get on the IRC and ask for help. Leave the iPAQ plugged in and do *not* reset it.

10. Once flashing has been performed, a dialog will open, stating that you have a valid OHH (Open HandHeld) Boot loader 2.18.1 in flash, and how to access the serial console. Tap on "ok" to close this dialog.

4.3. Verifying the bootloader

Before reseting your iPAQ, you *must* check that you have a valid bootloader.This is how to verify it:

1. In BootBlaster, tap on the "Flash" menu.
2. Tap on the "Verify" entry. A popup window will open, stating that you have a valid OHH (Open HandHeld) Boot loader 2.18.1 in flash, and how to access the serial console. Tap on "ok" to close this dialog.

Note: It looks like this step is performed automatically after flashing the bootloader in recent versions of BootBlaster.

Warning

If it does not say that you have a valid bootloader, do *not* reset your iPAQ. Instead, try programming the flash again. If that doesn't work, program your flash with your saved bootloader. If that doesn't work, send e-mail to `<bootldr@handhelds.org>` and/or get on the IRC and ask for help. Leave the iPAQ plugged in and do *not* reset it.

4.4. Rebooting the iPAQ

If you reset your iPAQ by pushing the recessed reset button at the lower right of the unit, the bootloader will boot PocketPC if present, just as usual.

What is different, is that you can get a serial console onto the iPAQ and that you get the following splash screen:

To get this serial console, configure a terminal emulator as described in Chapter 3.

Then, reboot your iPAQ by pushing the recessed reset button at the lower right of the unit while holding the center of the joypad (you will have to remove the iPAQ from its cradle to access the reset button). Put it back into the cradle and press the **space** bar into the terminal emulator or press the Calendar button on the iPAQ. You should get bootldr's boot prompt "`boot>`".

It is now safe to unprotect PocketPC, by typing the following in the terminal emulator:

```
boot> pflash 0x40000 0xffff 0
  addr=00040000
  len=0000FFFF
  protect=00000000
Protect=00000000
boot>
```

Bootloader's installation is now complete.

Chapter 5. Bootstrapping Familiar GNU/Linux

In this chapter, we will install the Familiar GNU/Linux bootstrap onto the iPAQ. It will be used in next chapter to install the full-blown Familiar GNU/Linux onto the iPAQ.

5.1. Downloading the root image

The root image is a verbatim copy of the GNU/Linux root partition that provides a minimal bootable GNU/Linux. You can download it from:

ftp://handhelds.org/familiar/releases/v0.5.3/install/H3800/task-bootstrap.jffs2

In this chapter, I suppose that you have set up a terminal connection, as described in Chapter 3.

5.2. Performing bootloader configuration

In order to boot GNU/Linux with the correct kernel arguments, the flash partitions need to be setup correctly, by typing:

```
boot> partition reset
argv[1]=reset
defining partition: bootldr
defining partition: root
defining partition: asset
boot>
```

This command defines three partitions:

- the bootloader partition
- the GNU/Linux partition
- the asset partition, containing unit specific information

5.3. Flashing the root filesystem

Issue the following commands:

```
boot> set ymodem 1
  setting param <ymodem> to value <1>
boot> load root
loading flash region root
using ymodem
ready for YMODEM transfer...
```

This command tells the iPAQ to be ready to receive the root image using the xmodem protocol. You just have to feed it from the running minicom window, by pressing **Ctrl-A** and **S**. A small popup window will appear, allowing you to choose the transfer protocol. Using the **arrow down** key, move down to the "ymodem" entry and press **Enter**.

A file selector window will open, where you need to specify the file to download. Using the **arrow right** key and pressing **Enter**, you can go to the directory containing the root image file transfered in Section 5.1. Using the **arrow down** key, move down to the correct file, and hit **space** to select the file, then press **Enter**.

A window will open, displaying the current transfer status. The transfer should take about 15 minutes, so be patient. When completed, a beep will awake you, so press **Esc** to close the window. The following lines should appear:

```
94208F6013CD23402361CC8905022946  task-bootstrap.jffs2
00600000 bytes loaded to C0000400

Erasing sector 00040000
Erasing sector 00080000
Erasing sector 000C0000
Erasing sector 00100000
Erasing sector 00140000
...
Erasing sector 01F40000
Erasing sector 01F80000
writing flash..
addr: 00040000 data: E0021985
addr: 00050000 data: 12512501
addr: 00060000 data: 8CF607FF
...
addr: 00610000 data: C8E314FE
addr: 00620000 data: 9B95ACF1
addr: 00630000 data: FFFFFFFF
verifying ...
formatting ... done.
boot>
```

This mean that the bootloader first erases the flash memory, then writes to it the downloaded image, verifies it and finally formats it correctly.

5.4. Booting GNU/Linux for the first time

Simply issue the following command:

```
boot>boot
```

The familiar penguin should appear on the iPAQ's screen, as well as the following output in the terminal emulator:

```
booting jffs2...
booting boot/zImage from partition >root<
Scanning all of flash for your JFFS2 convenience. Est Time: 6-12 seconds!
build_list: max = 01F7FFBC

...

Generating SSH DSA host key (this is slow, please be patient)...
Generating DSA parameter and key.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
```

```
The key fingerprint is:
0f:45:15:0e:dd:b3:cf:10:a3:5e:89:27:d6:c7:f6:04 root@familiar
Done.
Starting OpenBSD Secure Shell server: sshd.

familiar login:
```

> **Note:** Generating the DSA keys takes quite a long time, but don't be afraid, as this is done only when you boot for the first time.

You can login as the "`root`" user by providing the default "`rootme`" password.

Chapter 6. Setting up an USB network

Setting up an USB network will allow us to speed up the installation by providing a faster, as well as an easier way to transfer files than the serial method used up to now.

There are two steps involved in setting up the network:

1. Establish the iPAQ connection to the LAN (Local Area Network), if it exists. Anyway, if it does not exist, this connection will be used to access the Internet. We will use the mechanism known as *proxy_arp*, that will assign two IP addresses to the host, one for itself and one for the iPAQ.
2. Establish the iPAQ connection to the Internet. If the GNU/Linux host is part of a LAN and already has access to it, then no further work is required, and the iPAQ should be able to access it too. However, if the host is directly connected to the Internet through a modem, then we will have to set a mechanism called *NAT* (Network Address Translation), in order to map iPAQ's address to the same IP address the GNU/Linux host uses to access the Internet. This is required, as your ISP (Internet Service Provider) probably supplied only a single address to you. Another consideration will be to declare the iPAQ to the firewall, so it gets proper authorizations to access the Internet.

6.1. Connecting to the LAN

For this step, work is needed both on the GNU/Linux host and on the iPAQ.

6.1.1. USB networking on the GNU/Linux host

You need to have a supported USB hub in your PC, and have the kernel module installed for it. You should also obtain the kernel module named `usbnet`, which is bundled in Linux kernel since version 2.4.10, but working version is only bundled since 2.4.13. However, I found the experimental module in my stock Red Hat 7.2 distribution (kernel 2.4.7), and it seems not to behave too badly...

It is normally found as `/lib/modules/`*`kernel_version`*`/kernel/drivers/usb`.

If you cannot find the module, but have a supported kernel, you need to enable the `CONFIG_USB_USBNET` option in the kernel config (from the main config menu, go into "USB Support", you may need the "Code maturity level option" set to "EXPERIMENTAL"), and compile it with a command line like:

```
bash$ gcc -O2 -I /usr/src/linux/include -DMODULE -D__KERNEL__ -c usbnet.c
```

> **Note:** You must have gcc and correct kernel headers installed in order for this to work.

Otherwise, consider upgrading your kernel. As an alternative, you can continue to download all the required files using the serial method, it will only be slower.

Once the module is compiled, copy `usbnet.o` into the module directories and update the module dependencies with:

```
bash# cp build_dir/usbnet.o /lib/modules/kernel_version/kernel/drivers/usb
bash# depmod -a
```

you can then insert it with:

```
bash# modprobe usbnet
```

6.1.2. USB networking on the iPAQ

On the iPAQ side, the USB driver is included in the bootstrap image. However, we need to create all the files needed to start/stop/suspend/resume the USB network.

> **Note:** At the time of this writing, there is an existing unstable package named `usbnet_1.1_arm.ipk` that is supposed to do what we describe here. Unfortunately, it is relying on ifup/ifdown configuration that is not described, so we will not use it.

Download the following file onto your GNU/Linux host, and name it `usbnet`:

```
#!/bin/sh
# chkconfig: 2 45 96
# processname: usbnet
# pidfile: /var/run/usbnet.pid
# lockfile: /var/lock/subsys/usbnet

usage()
{
    echo "Usage: $0 {start|stop|status}"
}

PC_ADDR=192.168.1.100
IPAQ_ADDR=192.168.1.101
GW_ADDR=192.168.1.100
DRIVER=usb-eth
INTERFACE=usbf

if [ $# -lt 1 ]; then
    usage
    break
fi
case "$1" in
    start)
 modprobe $DRIVER
 ifconfig $INTERFACE inet $IPAQ_ADDR up
 route add default gw $GW_ADDR
 ;;
    stop)
 ifconfig $INTERFACE down
 modprobe -r $DRIVER
 ;;
    status)
 ifconfig $INTERFACE
 ;;
    *)
 usage
 ;;
```

```
esac
```

Replace the IP addresses by the ones corresponding to your network, then transfer it using the zmodem protocol. When you have set up a terminal connection using minicom, as described in Chapter 3, issue the following commands on the iPAQ:

```
# cd /etc/init.d
# rz
.. waiting to receive.**B0100000023be50
```

This command tells the iPAQ to be ready to receive the package using the zmodem protocol. You just have to feed it from the running minicom window, by pressing **Ctrl**-**A** and **S**. A small popup window will appear, allowing you to choose the transfer protocol. Using the **arrow down** key, move down to the "zmodem" entry and press **Enter**.

A file selector window will open, where you need to specify the file to download. Using the **arrow right** key and pressing **Enter**, you can go to the directory containing the `usbnet` file. Using the **arrow down** key, move down to the correct file, and hit **space** to select the file, then press **Enter**.

A window will open, and minicom will beep. Press **Esc** to close the window. Press **Enter** to get back the shell prompt.

Make this script executable by issuing:

```
# chmod +x /etc/init.d/usbnet
```

Based on this file, we will create the necessary setup needed to launch USB networking automatically when you boot, so we have to create a symbolic link to this script into the `/etc/rc2.d` directory, which is the directory containing the scripts to launch when you go into runlevel 2 (the normal runlevel):

```
# cd /etc/rc2.d
# ln -s ../init.d/usbnet S45usbnet.sh
```

For this first time, we can just launch it manually by typing:

```
# /etc/init.d/usbnet start
SA1100 USB Controller Core Initialized
usbctl: Opened for usb-eth
usbctl: Started for usb-eth
```

We also need to create the files required to stop/start the USB network when going into suspend/resume mode. Using the previous script, they are easy to create:

```
# cat >/etc/suspend-scripts/S33usbnet
#!/bin/sh
/etc/init.d/usbnet stop
Ctrl-D
# cat >/etc/resume-scripts/R33usbnet
#!/bin/sh
/etc/init.d/usbnet start
Ctrl-D
# chmod +x /etc/*scripts/*usbnet
```

6.1.3. Bringing up the USB network

If you have not already done it, insert the `usbnet` module on the GNU/Linux host:

```
bash# modprobe usbnet
```

> **Note:** Depending on your GNU/Linux USB configuration, you may not have to issue this command: it will be done automatically for you by the USB subsystem.

Make sure the cradle's USB and serial cables are connected to the host PC. Remove the iPAQ from the cradle and put it back. If you run minicom, you should see the following message in the terminal emulator window:

```
usbctl: zombie suspended --> [resume] --> zombie. Device in powered state.
usbctl: zombie --> [reset] --> default. Device in default state.
usbctl: Resetting
usbctl: default --> [address assigned] --> address. Device in address state.
usbep0: I have been assigned address: 14
usbctl: address --> [configure] --> configured. Device in configured state.
usbep0: Configured (OMP=00000040 IMP=00000040)
usbep0: fixme: set interface not supported
```

This means that our host's USB subsystem has correctly detected and configured the iPAQ. On the iPAQ, you can check that everything is working by issuing the following command:

```
# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:728 (728.0 b)  TX bytes:728 (728.0 b)

usbf      Link encap:Ethernet  HWaddr 9A:4D:0E:33:E7:01
          inet addr:192.168.1.101  Mask:255.255.255.0
          UP RUNNING  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:224 (224.0 b)  TX bytes:280 (280.0 b)
```

This last paragraph provides the necessary information about the USB network interface.

On the GNU/Linux host, a new network interface `usbN` (where *N* is a number starting from 0) will appear when the iPAQ is connected to the USB bus. You can then configure it like any other ethernet interface.

To save a lot of typing if you are going to use it on a regular basis, I suggest you to create the following `ipaqnet` script on your GNU/Linux host (please replace the addresses and UPLINK_IF with your appropriate values):

```
#!/bin/bash
#
# ipaqnet Control script for iPAQ USBNet connection
```

```
#
# Author: Michel Stempin
# Creation: 11/08/2002

PC_ADDR=192.168.1.100
IPAQ_ADDR=192.168.1.101/32
IPAQ_NET=192.168.1.0/24
UPLINK_IF=ppp0

start() {
    modprobe usbnet
    ifconfig usb0 inet $PC_ADDR up
    if [ $? -ne 0 ]; then
        exit 1
    fi
    echo "1" >/proc/sys/net/ipv4/conf/usb0/proxy_arp
    UPLINK=`ifconfig $UPLINK_IF >/dev/null 2>&1`
    if [ $? -ne 0 ]; then
        exit 1
    fi
    echo "1" >/proc/sys/net/ipv4/conf/$UPLINK_IF/proxy_arp
    echo "1" >/proc/sys/net/ipv4/ip_forward
    ip route del $IPAQ_NET dev usb0
    ip route add $IPAQ_ADDR dev usb0
}

stop() {
    ifconfig usb0 down
    rmmod -r usbnet
}
case "$1" in
    start|add)
        start
        ;;
    stop|remove)
        stop
        ;;
    *)
        echo $"Usage: $0 {start|stop|add|remove}"
        exit 1
esac
```

Make this script executable by typing:

```
bash# chmod +x path_to_ipaqnet/ipaqnet
```

We should now be able to launch the connection by typing:

```
bash# path_to_ipaqnet/ipaqnet start
```

If some sort of error message appears at this point, it means that the iPAQ insertions was not correctly handled, try re-inserting it.

This script uses the *proxy_arp* mechanism introduced above to forward data to the iPAQ.

On the GNU/Linux host, you can check that everything is working by issuing the following command:

```
# ifconfig
lo        Link encap:Local loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:748 (748.0 b)  TX bytes:748 (748.0 b)
...
usb0      Link encap:Ethernet  HWaddr C0:E3:56:1A:B8:CD
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Again, this last paragraph provides the necessary information about the USB network interface.

6.2. Connecting to the Internet

If your GNU/Linux host is part of a local network, then no further work is required. If it is connected to the Internet directly through a modem, then we will have to configure the firewall to do both address translation and to accept the iPAQ connection.

Configuring a firewall is very configuration-dependent. As a guideline, I provide the following `/etc/sysconfig/iptables` script, which work for the RedHat distribution (it can be adapted to other distributions too):

```
# iptables config file used by /etc/init.d/iptables
# Authors: Benoit PAPILLAULT & Philippe LUCAS
# Creation: 11/07/2002
# Modified by Michel STEMPIN for USB network

# default policy are not reset before executing lines in this file
# by default a packet goes either to INPUT chain or FORWARD chain
#
*filter
# INPUT chain
-P INPUT DROP
-A INPUT -i lo -j ACCEPT
-A INPUT -i usb0 -j ACCEPT
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#
# list of port that are allowed by default
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
#
# send all other packets to syslog
```

```
-A INPUT -j LOG --log-prefix "FIREWALL "
#
# FORWARD chain
-P FORWARD DROP
-A FORWARD -i usb0 -j ACCEPT
-A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
#
# OUTPUT chain
-P OUTPUT ACCEPT
COMMIT
#
*nat
# PREROUTING chain
-P PREROUTING ACCEPT
#
# POSTROUTING chain
-P POSTROUTING ACCEPT
-A POSTROUTING -o lo -j ACCEPT
-A POSTROUTING -o usb0 -j ACCEPT
# the SNAT line will be added by the right pppd
# (the one connected to the Internet)
-A POSTROUTING -s 192.168.1.0/24 -d ! 192.168.1.0/24 -j MASQUERADE
#
# OUTPUT chain
-P OUTPUT ACCEPT
COMMIT
```

You may have to adapt this simple script to your needs, as a firewall configuration is based on rule chains that are processed in order. The general idea is to allow traffic between the iPAQ and the GNU/Linux host, to forward traffic from the iPAQ to the external world, translating its address to appear to be the host's address. For more information, I suggest you read the Firewall and Proxy Server HOWTO (http://www.tldp.org/HOWTO/Firewall-HOWTO.html) and the Linux IP Masquerade HOWTO (http://www.tldp.org/HOWTO/IP-Masquerade-HOWTO).

> **Warning**
>
> This firewall configuration is very basic and does not provide a bulletproof (although reasonable) setup for your machine. You should *really* consider writing your own, based on your particular needs.

On the RedHat distribution, You will need to restart the firewall by typing:

```
# service iptables restart
```

For other distributions, this is probably something close to:

```
# /etc/rc.d/init.d/iptables restart
```

When everything is fine, you should be able to "ping" the iPAQ from the GNU/Linux host, and the other way around. On the GNU/Linux host (press **Ctrl**-**C** to stop):

```
bash# ping 192.168.1.101
PING 192.168.1.101 (192.168.1.101) from 192.168.1.100 : 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=0 ttl=255 time=2.187 msec
```

```
64 bytes from 192.168.1.101: icmp_seq=1 ttl=255 time=1.422 msec
64 bytes from 192.168.1.101: icmp_seq=2 ttl=255 time=1.236 msec
64 bytes from 192.168.1.101: icmp_seq=3 ttl=255 time=1.050 msec

--- 192.168.1.101 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/mdev = 1.050/1.473/2.187/0.434 ms
```

On the iPAQ:

```
# ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100): 56 data bytes
64 bytes from 192.168.1.100: icmp_seq=0 ttl=255 time=1.6 ms
64 bytes from 192.168.1.100: icmp_seq=1 ttl=255 time=0.8 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=255 time=1.5 ms

--- 192.168.1.100 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.8/1.3/1.6 ms
```

One last thing to do is to declare your DNS (Domain Name Servers) on the iPAQ. The best way is probably to copy or mimic the `/etc/resolv.conf` file from your GNU/Linux host machine. I suggest to enter it on the iPAQ with:

```
# echo "nameserver first_name_server" >> /etc/resolv.conf
# echo "nameserver second_name_server" >> /etc/resolv.conf
```

You should now be able to "ping" by its name any machine on the Internet that you know is responding to such requests.

If you are able to "ping" the GNU/Linux host, but not the rest of the world, this may be due to the *ip_forward* mechanism not being enabled. Try re-inserting the iPAQ or re-launch the `ipaqnet` script.

6.3. Shutting down USB networking

Whenever you want to stop the USB network, on the iPAQ issue:

```
# /etc/init.d/usbnet stop
```

On the GNU/Linux host, issue:

```
# path_to_ipaqnet/ipaqnet stop
```

6.4. Automating the USB connection

Unfortunately, automating the USB connection on the GNU/Linux host heavily depends on the GNU/Linux distribution you are using. As a guideline, here is what happens when an USB device is plugged or unplugged. The kernel first checks for a `/proc/sys/kernel/hotplug` pseudo-file that should contain the name of a user program to launch with the event type (usb, pci, isapnp, net, ieee1394, printer, disk, parport, ...) and a device descriptor as arguments. This program is usually a shell script, that will delegate processing to some other scripts, using text configuration files. Not two different GNU/Linux distributions are using the same schema, which makes it very difficult to write a general way of automating. So, the only general way to handle this will be to edit the `hotplug` script.

On my RedHat 7.2, I need to add the following lines at the beginning of the `/sbin/hotplug` file to use the `/etc/rc.d/init.d/ipaq` script introduced above:

```
if [ "$*" = "usb" ]; then
    if [ "$PRODUCT" = "49f/505a/0" ]; then
        path_to_ipaqnet/ipaqnet $ACTION
        exit
    fi
fi
```

This way, the USB network will be correctly brought up/shut down when you plug/unplug the iPAQ.

Chapter 7. Installing Familiar GNU/Linux

In this chapter, we will install the Familiar GNU/Linux distribution onto the iPAQ.

Provided that the iPAQ is able to access the Internet as described in Chapter 6, it is quite easy, as the Familiar GNU/Linux distribution is made up of packages that can be downloaded and installed directly.

The first thing will be to update the package definitions to match the latest packages available.

We will then run SSH to definitively get rid of the serial terminal and work at full speed using USB.

7.1. Upgrading packages definition

I assume you still have the terminal emulation running, as described in Chapter 3, and that USB networking is working, as described in Chapter 6. In order to update the packages definitions, just issue the following command on the iPAQ:

```
# ipkg update
Downloading http://familiar.handhelds.org/familiar/feeds/unstable/packages/armv
4--14:14:56--  http://familiar.handhelds.org:80/familiar/feeds/unstable/package
s/           => `//tmp/ipkg/Packages'
Connecting to familiar.handhelds.org:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 212,734 [text/plain]

    0K -> .......... .......... .......... .......... .......... [ 24%]
   50K -> .......... .......... .......... .......... .......... [ 48%]
  100K -> .......... .......... .......... .......... .......... [ 72%]
  150K -> .......... .......... .......... .......... .......... [ 96%]
  200K -> .......                                                [100%]

14:15:00 (59.24 KB/s) - `//tmp/ipkg/Packages' saved [212734/212734]

Done.
Updated list of available packages in /usr/lib/ipkg/lists/familiar-unstable
```

Then, upgrade the installed packages to their latest version:

```
# ipkg upgrade
...
    N or O  : keep your currently-installed version
      D     : show the differences between the versions (if diff is installed)
 The default action is to keep your current version.
*** wlan.opts (Y/I/N/O/D) [default=N] ? y
Configuring pcmcia-cs...Done
Package mount (2.11h-1-fam1) installed in root is up to date
Package shellutils (2.0.11-5) installed in root is up to date
Package wvlan-modules-2.4.18-rmk3 (hh12) installed in root is up to date
Package libncurses5 (5.0-6.0potato1) installed in root is up to date
Package tar (1.13.19-1) installed in root is up to date
Package setserial (2.17-20) installed in root is up to date
Package orinoco-modules-2.4.18-rmk3 (hh12) installed in root is up to date
```

You may have to enter *Y* a couple of time to accept modifying some existing files.

One of the first useful package to install is the `familiar-postint` package, that will provide clock synchronization using the **ntpdate** command:

```
# ipkg install familiar-postinst
...
Unpacking familiar-postinst...Done.
Configuring familiar-postinst...Done.
# cd
# ./postinst
...
Familiar post-installation completed successfully.
```

7.2. Running SSH

You can now forget minicom and the serial line, as you can take control of the iPAQ using the SSHD daemon running on it. The only remaining usage for minicom would be to get a serial bootloader console. To access the iPAQ from your GNU/Linux host, issue:

```
bash# ssh 192.168.1.101
```

or

```
bash# ssh ipaq
```

if you still have the name definition set up in Section 1.1.2. The first time you use it, ssh will ask you if you want to save the new key: answer yes, then provide the default "`rootme`" password.

When you want to transfer files, you can issue an **scp** command from the iPAQ to the GNU/Linux host (or the other way around). One useful file to download will be the `/etc/localtime` from your GNU/Linux host:

```
# scp 192.168.1.100:/etc/localtime /etc/localtime
```

This should update the iPAQ's time to the correct timezone for you.

7.3. Installing additional Familiar packages

At your convenience, you can install more Familiar packages. You can get the list of packages available by issuing the following command:

```
# ipkg list
```

I recommend you to install at least a decent shell (`bash`), a tool to list the running processes (`psmisc`), and a text display utility (`less`).

If you want to be at the leading (bleeding?) edge of technology, you can uncomment the *familiar-unstable* line in the `/etc/ipkg.conf` file and perform an **ipkg update** followed by an **ipkg upgrade** to get the state of the art packages. Just remember what *unstable* means...

From this point, you have the choice to install a full blown GNU/Linux system, with or without X11, or to continue with the Opie installation. As said before, this last option was chosen in this document to demonstrate that an Open Source system can compare to a commercial system, and even do better.

Chapter 8. Installing Opie

In this chapter, we will install the Opie environment onto the iPAQ.

Provided that the iPAQ is able to access the Internet as described in Chapter 6, it is quite easy, as Opie is made up of packages that can be downloaded and installed directly.

8.1. Bootstrapping Opie

We need first to declare the Opie *feed* to the `ipkg` installer. This is done by typing:

```
# echo "src opie http://131.152.105.154/feeds/ipaq/unstable" >> /etc/ipkg.conf
```

Update the package definitions by typing:

```
# ipkg update
Downloading http://familiar.handhelds.org/familiar/releases/v0.5.3/packages/armv4l/Packages
...
Updated list of available packages in /usr/lib/ipkg/lists/opie
```

Install the `task-opie` meta-package by typing:

```
# ipkg -force-depends install task-opie
```

> **Note:** At the time of this writing, the `e2fsprogs` package from the Familiar GNU/Linux v0.5.3 does not declare the `libuuid` library that it provides, resulting in a broken dependency problem. This is why we have to add the *-force-depends* flag to the previous command.

For the iPAQ 38xx, we have to install a package fixing the screen rotation problem:

```
# ipkg install qt-embedded-rotation
```

The only thing left to do is to launch Opie:

```
# /etc/init.d/opie start
```

Opie should now be running. I can’t resist to show you how it looks:

8.2. Finishing Opie installation

At this point, you have the choice either to continue to install the Opie applications from the command line, as we did before, or to use the graphical package manager. For this last one, you can launch it by tapping on the Settings tab, then tapping on the Software icon.

However, from my experience, it still suffers from minor problems, so my recommendation would be to continue to use the **ipkg** command line tool.

Use the `task-opie-xxx` meta-package to install a group of application at once.

You may choose to install the Konqueror/Embedded Web browser, which is definitively one of the best browser with such a small footprint. To install it, add a new *feed* to the `ipkg` installer by typing:

```
# echo "src konqueror http://bolug.uni-bonn.de/files/konq-e" >> /etc/ipkg.conf
```

Update the package definitions by typing:

```
# ipkg update
Downloading http://familiar.handhelds.org/familiar/releases/v0.5.3/packages/armv4l/Packages
...
Updated list of available packages in /usr/lib/ipkg/lists/konqueror
```

Install the `konqueror` package by typing:

```
# ipkg install konqueror
```

At this point, you now have a fully functional iPAQ running GNU/Linux, as a perfect replacement for PocketPC, having almost the same capabilities (except maybe for the synchronization facility) but with all the advantages of an open source solution.

Enjoy and have fun! I just hope you will not follow the last chapter of this document, dealing with PocketPC re-installation!

Chapter 9. Reinstalling PocketPC

This chapter is provided in case you are not satisfied with GNU/Linux on your iPAQ, or are forced to switch back to the original operating system. The steps of restoring the original PocketPC and bootloader images saved in Chapter 2 are covered here.

9.1. Restoring PocketPC

First, make sure you have set up a terminal connection, as described in Chapter 3.

You need to reboot the iPAQ. If you have GNU/Linux running on it, this can be achieved by typing:

```
# shutdown -r now
```

Another alternative is to push the recessed button at the lower right of the unit with a sharp object.

> **Warning**
>
> Make sure you have bootldr 2.18.01 or higher installed on the iPAQ before proceeding with the following instructions.
>
> Compaq/HP Cambridge Research Labs will fix any units that fail to restore properly, but this should be seen as a last resort solution.

When the bootldr screen appears, press on the "Calendar" button at the lower left corner of the unit to get a serial bootloader console. At the boot prompt, type:

```
boot> set ymodem 1
boot> load root
loading flash region root
using ymodem
ready for YMODEM transfer...
```

This command tells the iPAQ to be ready to receive the root image using the xmodem protocol. You just have to feed it from the running minicom window, by pressing **Ctrl-A** and **S**. A small popup window will appear, allowing you to choose the transfer protocol. Using the **arrow down** key, move down to the "ymodem" entry and press **Enter**.

A file selector window will open, where you need to specify the file to download. Using the **arrow right** key and pressing **Enter**, you can go to the directory containing the `wince_image.gz` you created in Chapter 2. Using the **arrow down** key, move down to the correct file, and hit **space** to select the file, then press **Enter**.

A window will open, displaying the current transfer status. The transfer should take about 20 minutes (yes, it is much larger than GNU/Linux), so be patient. When completed, a beep will awake you, so press **Esc** to close the window. The following lines should appear:

```
Looks like a gzipped image, let's verify it...
Verifying gzipped image
......................................;
programming flash...erasing ...
Erasing sector 00040000
Erasing sector 00080000
```

```
Erasing sector 000C0000
...
Erasing sector 01F80000
writing flash..
addr: 00040000 data: EA0003FE
addr: 00050000 data: 0F6A0F6A
addr: 00060000 data: 11A0F003
...
addr: 01FA0000 data: FFFFFFFF
addr: 01FB0000 data: FFFFFFFF
verifying ...
calculated crc32 = 0x282F31FE
desired    crc32 = 0x282F31FE
formatting ... done.
boot>
```

You can now boot PocketPC by typing:

```
boot> boot wince
booting wince...
Disabling LCD controller
```

The iPAQ will boot PocketPC automatically, unless you hold the center of the joypad while pushing the recessed reset button.

9.2. Checking PocketPC restoration

To be safe, you can check that PocketPC restoration worked successfully. You must launch the iTask manager by pushing the iTask button at the front lower right corner of the unit. Tap on the "iTask" icon at the lower left corner of the window that will appear. In the menu that will open, tap on the "Self Test" menu entry. Check at least the "ROM checksum" box and tap on the "Next" button.

In the new screen that will open, check the "Perform automatic Reset" checkbox and tap on the "Start" button.

Answer all the questions and verify that the "ROM checksum test Passed" line appears. If not, try reflashing PocketPC using the procedure above.

Tap on the "Finish" button and save a log of the test if you wish so, then tap on the "Ok" at the top right of the screen.

9.3. Restoring the original bootloader

> **Note:** This step is optional, as the ARM bootldr is able to boot PocketPC automatically. It is only needed in case you want to restore the unit to its original version.

First, make sure that PocketPC boots correctly by pushing the recessed button at the lower right of the unit with a sharp object.

> **Note:** If it is not booting, try reloading the PocketPC image, as described above.

You will need to use the same terminal emulation setup described in Chapter 3, then reboot the iPAQ as in Section 9.1. At the boot prompt, type:

```
boot> load bootldr
```

Use the same procedure as in Section 9.1 to transfer the `saved_bootldr.bin` you saved in Chapter 2.

Appendix A. GNU Free Documentation License

Version 1.1, March 2000

A.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

A.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

A.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

A.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.