Burning a RedHat CD HOWTO

Table of Contents

Burning a RedHat CD HOWTO

Morten Kjeldgaard, mok@imsb.au.dk and Peter von der Ahé, pahe+rhcd@daimi.au.dk

$Date: 2000/03/02 16:28:37 $ $Revision: 1.34 $

This document describes how to make your own CDs from the Red Hat Linux distribution equivalent to the ones commercially available from Red Hat. The structure of the distribution is described, as well as the procedure needed to include updated RPMS into the distribution. Prerequisites are a good network connection, and a CD−writer.

1.Introduction

There may be several reasons for making your own CD. Perhaps you're a cheapskate and want to save the $50 cost of the Red Hat distribution. Or, perhaps you want a distribution CD containing the latest distribution with all current updates. This is highly relevant, because after each major release of the Red Hat distribution, there have been loads of updates, several of which are security related. Just take a look at the errata page.

2.Anatomy of the Red Hat FTP site

In the spirit of the Linux community, Red Hat Software has made available their Linux distributions for several platforms on their FTP site. These are all available from the top distribution directory.

2.1 The top level directory

The toplevel directory for RedHat Linux release 6.1 (pub/redhat/redhat−6.1) contains distributions for the different platforms, and a directory containing updates and corrections to program packages that have been published since the release.

```
SRPMS/     alpha/     i386/      sparc/      iso/
```

In this document, we use the i386 distribution as an example. The procedure given in this document is likely to work on all platforms supported by Red Hat (Alpha, SPARC, ppc, etc.), but we have only tested it on the i386 platform (the authors would be most interested in additional information). The root of the i386 directory looks like this:

```
-rw-r--r--   1 root     root    18391  Mar 19  1999   COPYING
-rw-r--r--   1 root     root     3409  Sep 26 01:12   README
-rw-r--r--   1 root     root     1908  Sep 25 18:20   RPM−GPG−KEY
drwxr-xr-x   5 root     root     4096  Oct 26 09:01   RedHat/
-rwxr-xr-x   1 root     root      538  Sep 26 04:16   autorun
-rwxr--r--   1 root     root     2048  Sep 30 20:00   boot.cat
drwxr-xr-x   7 root     root     4096  Oct 26 09:01   doc/
drwxr-xr-x   6 root     root     4096  Oct 26 09:01   dosutils/
drwxr-xr-x   2 root     root     4096  Oct 26 09:01   images/
-rw-r--r--   1 root     root   391123  Jan  5 09:05   ls−lR
-rw-r--r--   1 root     root    51786  Jan  5 09:05   ls−lR.gz
drwxr-xr-x   4 root     root     4096  Oct 26 09:01   misc/
```

The `doc` directory contains an abundance of information. Most importantly, the RedHat installation manual can be found in HTML format in the directory doc/rhmanual/manual/. Next, there is a number of FAQs, and finally, the entire collection of HOWTOs and mini−HOWTOs.

The `images` directory contains boot floppy images that must be copied to a diskette. In the most recent distribution (6.1), there are two disk images available. The boot image is called `boot.img`, which is required when installation is performed directly from a CD−ROM. If installation from a local hard disk, NFS mounted disk or FTP is required, the supplementary disk image `supp.img` might be needed. See section Installing from the CD and references therein for details.

The `misc` directory contains source and executables of a number of programs needed for the installation. Finally, the `iso` directory contains iso9660 images just like the one you we about to make.

2.2 The "RedHat" directory −− the core of the distribution

The most important part of the directory tree is rooted in the RedHat directory:

```
drwxr-xr-x   2 root     root    28672   Oct 26 09:01   RPMS
drwxr-xr-x   2 root     root     4096   Oct 26 09:01   base
-rw-r--r--   1 root     root        0   Jan 19  1999   i386
drwxr-xr-x   6 root     root     4096   Oct 26 09:01   instimage
```

The RPMS directory contains the major part of the Red Hat distribution consisting of a set of RPM (Redhat Package Manager) files. An RPM package typically contains binary executables, along with relevant configuration files and documentation. See the section RPM packages for more information.

The base directory holds different book−keeping files needed during the installation process, e.g. the comps file, which defines the *components* (groups of packages) used during the "Choose packages to install" phase. See section The comps file for more information on this file, and how to use it. Another important file in the base directory is the hdlist file containing most of the header fields from all the RPMs in the RPMS directory.

This means that all the interdependencies among RPM packages can be determined just by reading hdlist without having to read all the RPM packages which is quite convenient especially during FTP installs.

Another use of hdlist is mapping package names to file name, eg. perl to perl−5.004−6.i386.rpm. This means that if you want to incorporate updates from RedHat (see section Incorporating the updates) or add your own packages to the RPMS directory, you need to update hdlist. This is descriped later in Generating a new hdlist file.

The instimage directory contains a bare−bones live file system with a number of programs and shared libraries needed during the installation procedure.

2.3 The "updates" directory

The /pub/redhat/updates directory has updates for all releases of RedHat's distribution since version 3.0.3. This is the place to find software packages that have been uppdated for some reason or other. You should especially be aware of security updates. These are publicised on RedHat's errata page whenever a fix is available. The most important files found in the updates directory are:

```
drwxrwxr-x   8 ftpuser  ftpusers      8192 Aug 18 08:05 5.1/
drwxrwxr-x   9 ftpuser  ftpusers      8192 Sep 18 08:07 5.2/
drwxrwxr-x   9 ftpuser  ftpusers      8192 Jan  7 23:09 6.0/
```

```
drwxrwxr-x   9 ftpuser  ftpusers      8192 Jan  7 23:09 6.1/
lrwxrwxrwx   1 ftpuser  ftpusers      3 Aug 25 13:14 current -> 6.1/
```

The structure of each of these directories is similar to that described in section The top level directory

3.RPM packages

The major part of the Red Hat distribution consists of a set of RPM (Redhat Package Manager) files. An RPM package typically contains binary executables, along with relevant configuration files and documentation. The `rpm` program is a powerful package manager, which can be used to install, query, verify, update, erase and build software packages in the RPM format. `Rpm` convieniently maintains a database of all the software packages it has installed, so information on the installed software is available at any time.

The binary RPM files in the distribution have been built on a system running the distribution itself. This is important, because most of the programs in the packages rely on shared libraries. From RedHat version 5.0, the new version 2 of the GNU standard C library (which is 64−bit clean) has been used. This version of the library is commonly referred to as `glibc` or in Linux: `libc 6`. All executables in the distribution have been linked against this library. If you attempt to install binary files from a different distribution, chances are that they will not work, unless you install the libc5 package for backwards compability.

The names of the RPM packages contain the suffix *.arch*.`rpm`, where *arch* is the architechture, having the value `i386` for Intel platform binaries. The packages you install must match the versions of the shared libraries available on the machine. The `rpm` program is usually quite good at ensuring that this is indeed the case, however, there are ways around this check, and you should be sure that you know what you are doing if you force installation of packages this way. However, using the RedHat installation boot disk, it is ensured that the correct set of RPM packages are installed on the machine.

If you discover an RPM package that was not installed on your system during the installation process, don't despair. At any time, you may (as root) install RPM packages, for example:

```
rpm --install  WindowMaker−0.18−1b.i386.rpm
```

You can even install directly from the Internet, if you know the URL of an RPM package:

```
rpm --install ftp://rufus.w3.org/redhat−contrib/noarch/mirror−2.9−2.noarch.rpm
```

Another version of the RPM packages contain the original sources used to build the binaries. These packages have the suffix `.src.rpm` and are situated in the `SRPMS` directory. These packages are not needed on the installation CD, and in fact, there is not even enough disk space on an 74 minute burnable CD to accomodate them. Of course, you can make a separate CD with the SRPMS.

4.Obtaining your local copy of the distribution

You need a copy of the distribution on a writable disk which is accessible from the computer having the CD writer (duh!). If you want to incorporate the latest updates, this directory should (also) be accessible from from a Linux machine, either from a local disk, an NFS mounted disk on a different computer, or a JAZ disk.

You could copy the distribution from a RedHat CD, or you could get it via FTP. If you choose to use FTP, the best way to get a correct copy of the distribution is to use the `mirror` package.

Mirror is a sophisticated perl script that compares the content of a directory on a remote site with a local directory. It will use FTP to fetch the files that are on the remote site but not the local site, and delete files on the local site that are not on the remote site. The mirror program is configured with a configuration file. The mirror package is available as an RPM from rufus.w3.org.

Make your local copy `mirror.redhat` of the mirror configuration file, and edit the relevant fields at the top of the file. After the default section, define these packages:

```
package=updates
        site=ftp.sunsite.auc.dk
        exclude_patt=(alpha/|sparc/|SRPMS/)
        remote_dir=/disk1/ftp.redhat.com/pub/redhat/updates/6.1
        local_dir=/jaz/updates/6.1

package=dist
        site=ftp.sunsite.auc.dk
        exclude_patt=(alpha/|sparc/|SRPMS/|iso/)
        remote_dir=/disk1/ftp.redhat.com/pub/redhat/redhat-6.1
        local_dir=/jaz/redhat-6.1
```

The following command will download a copy of the entire RedHat tree on your local disk. **Think* before you do this, you are about to transfer approximately 500Mb of data!

```
mirror -pdist mirror.redhat
```

This will mirror the Red Hat FTP site on your local disk. The content of a Red Hat distribution does not change between releases, so you only need to download this package *ONCE*. All changes to the distribution are in the `updates` directory. Thus, if you want to keep an up−to−date mirror of the Red Hat distribution, you only need to keep the `updates` directory current. This is done using the command

```
        mirror -pupdates mirror.redhat
```

You can run this regularly, say, once a week, through a cron script. The RedHat distribution is available on a great number of FTP servers around the world, which are updated daily from the master site at ftp://ftp.redhat.com/pub. You should choose an FTP site close to you, see the RedHat list of mirror sites.

5.Incorporating the updates

To incorporate the updates, you need write access to the distribution directory from a Linux machine, with a working version of rpm installed. There are three steps involved:

1. Correct the file protection modes.
2. Replace updated RPMs.
3. Generate the hdlist file

If you maintain a mirror of the `updates` directory, you can at any time produce a CD including the current updates by repeating these steps.

5.1 Correcting the file protection modes

During the installation process, some programs are run directly off the CD. Unfortunately, the FTP program does not always preserve the protection modes of the files and directories that are copied. Therefore, it is necessary to make sure that execute permission is given to programs, shell scripts and shared libraries, before the directory is burned on the CD. This is done by running the `updatePerm` script on your local copy of the distribution:

```
#!/bin/bash

RHVERSION=6.1

LIST=/tmp/er3hd3w25
RHROOT=/jaz/redhat-${RHVERSION}

# Find all directories, and make sure they have +x permission
find $RHROOT -type d -exec chmod -c 755 {} \;

# Find all files that are executables, shell, python or perl scripts
find $RHROOT -type f | file -f - | grep -v RPM \
   | egrep -i 'executable|perl|bourne|shell|python' | cut -f1 -d: > $LIST
#   ====>     NEEDED FOR RedHat 6.1        ^^^^^^^

# Find shared libraries
find $RHROOT -name \*.so >> $LIST
```

```
# Make them executable
while read file
do
   if [ ! -x $file ] ; then
      chmod -c 755 $file
   fi
done < $LIST

/bin/rm $LIST

exit 0
```

5.2 Replacing the updated RPMS

The following script called `updateCD` copies all files from the update directory to the RPMS directory. The script uses some nifty rpm tricks to determine what packages in the updates directory are more recent. Older packages are moved to the `${OLD}` directory.

```
#! /bin/bash
# This script updates rpms in a RedHat distribution found in $RPMDIR.
# The old rpms will be placed in $OLDDIR.
# The new rpms should be located in $UPDDIR.
# The new images are in $IMGDIR
# The images to be updated are in $OMGDIR
# The architechture is $ARCH.

RHVERSION=6.1
ARCH=i386

RHROOT=/jaz/redhat-${RHVERSION}
RPMDIR=${RHROOT}/${ARCH}/RedHat/RPMS
UPDDIR=${RHROOT}/updates/${ARCH}
IMGDIR=${RHROOT}/updates/images/${ARCH}
OMGDIR=${RHROOT}/${ARCH}/images
OLDDIR=${RHROOT}/old

if [ ! -d $OLDDIR ] ; then
   echo making directory $OLDDIR
   mkdir $OLDDIR
fi

allow_null_glob_expansion=1

for rpm in ${UPDDIR}/*.rpm ; do
  NAME=`rpm --queryformat "%{NAME}" -qp $rpm`
  unset OLDNAME
  for oldrpm in ${RPMDIR}/${NAME}*.rpm ; do
    if [ `rpm --queryformat "%{NAME}" -qp $oldrpm` = "$NAME" ]; then
      OLDNAME=$oldrpm;
      break
    fi
  done
  if [ -z "$OLDNAME" ]; then
    echo $NAME is new
    cp -pv $rpm $RPMDIR
  else
    if [ `basename $rpm` != `basename $OLDNAME` ]; then
      mv $OLDNAME $OLDDIR
      cp −pv $rpm $RPMDIR
    fi
  fi
done


# Copy new boot image files to the right place...
for newfile in ${IMGDIR}/* ; do
  file=${OMGDIR}/$(basename ${newfile})
  if [ $newfile −nt $file ] ; then
     cp −pv $newfile $file
  fi
done

exit 0
```

Important note for RedHat 6.0, 6.1

Certain RPMs, specifically the kernel and kernel−smp packages, include the platform in the filename but not in the package name. For example, the "kernel" package comes in several flavors:

```
       kernel−2.2.5−22.i386.rpm
       kernel−2.2.5−22.i586.rpm
       kernel−2.2.5−22.i686.rpm
```

but for all three rpm −qp returns just "kernel" for the package name.

As you can see, this "outsmarts" the updateCD script. The result is that only the last one gets properly copied. The first two get copied, but are then moved to the $OLD directory! Preferably, RedHat should name these differently. But for now, the easy solution is to move the packages by hand, after running updateCD. (Thanks to Kyle B. Ferrio <kyle@U.Arizona.EDU>)

Joshua Sarro <mthed@shore.net> has contributed a perl script called updateMirror.pl which can deal with the situation. You can fetch it here: http://imsb.au.dk/~mok/linux/doc/updateMirror.pl.

5.3 Generating a new hdlist file

When installing from the CD, the installation program on the CD relies on the file `RedHat/base/hdlist` describing what RPM packages are available on the CD. The `hdlist` file can be generated by the program `misc/src/install/genhdlist`. This program must be run with the absolute path to the root of the distribution as the only argument. Here is the `updateHdlist` script which calls that program:

```
#!/bin/bash

RHVERSION=6.1
ARCH=i386

echo generating hdlist...
RHROOT=/jaz/redhat−${RHVERSION}
GENHDDIR=${RHROOT}/${ARCH}/misc/src/install

chmod u+x ${GENHDDIR}/genhdlist
chmod 644 ${RHROOT}/${ARCH}/RedHat/base/hdlist
${GENHDDIR}/genhdlist ${RHROOT}/${ARCH} || echo "*** GENHDLIST FAILED ***"

exit 0
```

NOTE: After having incorporated the updates in the main `RedHat/RPMS` directory, your copy of the distribution is no longer a mirror of the Red Hat distribution site. Actually, it is more up−to−date! Therefore, if you attempt to mirror the distribution, older versions of the RPM's that have been updated will be downloaded once more, and the updates deleted.

Important note for RedHat 6.1

The installation in RedHat 6.1 is completely changed from earlier versions, and RedHat have introduced a system called ``anaconda''. The genhdlist program is now found in a different place, so in the script above, use

```
GENHDDIR=${RHROOT}/${ARCH}/misc/src/anaconda/utils
```

The updatePerm script must be changed to include ``python'' in line 13. This *has* been done in the listing above, but if you've cut the script from earlier versions of this document (before 1.30), you need to make the change!

In some cases, `genhdlist` fails to run, because the executable is not statically linked. In such a case, you can add a new line `${RHROOT}/${ARCH}/RedHat/instimage/usr/lib` in `/etc/ld.so.conf` and run `ldconfig −v`.

Another solution is to recompile `genhdlist`. The following modification to the `updateHdlist` script worked under RedHat 5.2:

```
#!/bin/bash

RHVERSION=6.1
ARCH=i386

RHROOT=/misc/redhat/redhat−${RHVERSION}
```

```
GENHDDIR=${RHROOT}/${ARCH}/misc/src/anaconda/utils

echo Compiling genhdlist...
sed −e 's/FD_t/int/' \
    −e 's/fdOpen/open/' \
    −e 's/fdClose/close/' \
    −e 's/fdFileno//' < ${GENHDDIR}/genhdlist.c > /tmp/genhdlist.c
cc −o /tmp/genhdlist −I/usr/include/rpm /tmp/genhdlist.c −lrpm −lz

echo generating hdlist...
chmod 644 ${RHROOT}/${ARCH}/RedHat/base/hdlist
/tmp/genhdlist ${RHROOT}/${ARCH} || echo "*** GENHDLIST FAILED ***"

exit 0
```

In this version of the script, a copy of the C source of `genhdlist.c` is piped through `sed` to create a copy in `/tmp` that will compile under RedHat 5.2. This version of `genhdlist` is then used to create the `hdlist` file

Important note for RedHat 5.2

As distributed with RedHat version 5.2 and earlier, `genhdlist` CRASHES if there are files in the `RedHat/RPMS` directory which are *not* RPM files! This causes problems, because in the 5.2 distribution, there are a couple of non−RPM files named `ls−lR` and `ls−lR.gz` in `RedHat/RPMS`. Therefore, you must remove all non−RPM files from the directory. Alternatively, you can apply the following patch to `misc/src/install/genhdlist.c` and do a make. The patch will cause `genhdlist` to ignore any non−RPM files.

```
*** genhdlist.c.orig    Fri Nov 27 12:08:13 1998
−−− genhdlist.c Fri Nov 27 12:08:20 1998
***************
*** 12,23 ****
−−− 12,26 −−−−

  #define FILENAME_TAG 1000000

+ /* Not used apparently...
+
  int tags[] =  { RPMTAG_NAME, RPMTAG_VERSION, RPMTAG_RELEASE, RPMTAG_SERIAL,
                RPMTAG_FILENAMES, RPMTAG_FILESIZES, RPMTAG_GROUP,
                RPMTAG_REQUIREFLAGS, RPMTAG_REQUIRENAME, RPMTAG_REQUIREVERSION,
                RPMTAG_DESCRIPTION, RPMTAG_SUMMARY, RPMTAG_PROVIDES,
                RPMTAG_SIZE, RPMTAG_OBSOLETES };
  int numTags = sizeof(tags) / sizeof(int);
+ */

  int main(int argc, char ** argv) {
      char buf[300];
***************
*** 26,34 ****
−−− 29,39 −−−−
      struct dirent * ent;
      int fd, rc, isSource;
      Header h;
```

```
+     /* not used
      int count, type;
      int i;
      void * ptr;
+     */

      if (argc != 2) {
        fprintf(stderr, "usage: genhdlist <dir>\n");
***************
*** 74,79 ****
--- 79,85 ----

            rc = rpmReadPackageHeader(fd, &h, &isSource, NULL, NULL);

+           if (!rc) {
            headerRemoveEntry(h, RPMTAG_POSTIN);
            headerRemoveEntry(h, RPMTAG_POSTUN);
            headerRemoveEntry(h, RPMTAG_PREIN);
***************
*** 110,115 ****
--- 116,122 ----
            headerWrite(outfd, h, HEADER_MAGIC_YES);
            headerFree(h);
            close(fd);
+           }
        }

        errno = 0;
```

5.4 The comps file

The comps file defines how the packages are bundled during the installation. In the Red Hat distribution, this is done according to the functionality they provide, for example:

- Printer Support
- X Window System
- GNOME
- KDE
- Mail/WWW/News Tools
- ...
- Kernel Development
- Extra Documentation

Sometime during the installation process, the user is presented with a dialog called "Components to install". Some of the components have been preselected, and others not. The last item on the components list is called "Everything". According to the Red Hat documentation, selecting every package will require close to 1 Gb of free disk space.

On the dialog box, there also is an option that enables the user to customize exactly what packages will be installed. Customizing the installation by hand, or selecting "Everything" in the components list is the only way to have your own packages installed unless you modify the `RedHat/base/comps` file.

Format of comps file in RedHat versions < 6.1

The format of the `comps` file currently starts with a header describing the version of the comps format, followed by an empty line.

```
0.1
<empty line>
```

After this, the components are listed, separated by empty lines:

```
<component 1>
<empty line>
<component 2>
<empty line>
.
.
<component n>
<empty line>
EOF
```

Each component has the following definition:

```
(0|1) (−−hide)? <name>
<RPM 1>
<RPM 2>
...
<RPM n>
end
```

Before the name of each component, 0 or 1 is given. A value of 1 here means that the component is chosen by default, and 0 means it's not. The option "−−hide" means that you will not see the entry, unless you choose "expert" installation. The first component is called "Base", and that is special, in the sense that it *must* be present and it does not show up in the dialog (you can't deselect the base installation, which makes sense...)

Next follows a list of rpm packages belonging to that component. Note that this is the package name stored *in the rpm file*, and *not* any part of the file name of the package (although it is often the same).

By adding your packages to the `comps` file, you can customize your own distribution, and make sure that your packages will be installed by default. One thing to be careful about is interdependence among your packages, but here, you are on your own :−) A word of warning: be careful not to add or remove extra whitespace in the file. Examine the existing `comps` file (make a copy of the original) to see how it's done (or check `i386/misc/src/install/pkgs.c` if you want to see how the file is parsed).

Format of comps file in RedHat version 6.1

With RedHat version 6.1, the format of the `comps` file has changed. We have not yet disected it to discover the file format. With the above information, however, it should be easy enough to do it yourself. The decoding takes place in `${RHROOT}/${ARCH}/misc/src/anaconda/comps.py`.

6.At last: burning the CD

As we assume that you have a working CD−writer on your system, and you know how to use it, we wont go into much detail about burning the CD. If you are burning your CD on a Linux system, you can install the excellent XCDroast package. In XCDRoast, switch on the Rock Ridge extensions, and the creation of a TRANS.TBL file.

Make sure the *top* directory of the CD contains at least the following files and directories:

```
        COPYING        RPM−PGP−KEY README        RedHat/
```

The following directories might come in handy:

```
        doc/           gnome/      misc/         dosutils/    images/
```

6.1 Creating a bootable CD

Since XCDroast doesn't support creation of bootable disks you'll need to use other tools, for example `mkisofs` (RPMs). and `cdrecord`. (RPMs). You'll need to create an image file which will be written to the CD. This file will be 500Mb or more so find a partition with enough free space and change the path for `redhat.img` in the following commands if necessary. You may need to be root to use `mount` and `cdrecord`.

Create iso9660 disk image

Change directory to the place in your mirror that will be the root directory of the cd. For instance, redhat−6.1/i386.

```
        mkisofs −v −r −T −J −V "Red Hat 6.1" −b images/boot.img −c boot.catalog −o /tmp/redhat.img
```

The file `${RHROOT}/boot.catalog` will be created by `mkisofs`.

Some versions of `mkisofs` do not have a `−J` option. It can be omitted since the Joliet extensions it specifies are not used by the installer although they make the CD more readable in Windows.

Test the image

If you're paranoid you can test your new disk image by mounting it. If you forgot to fix the file permissions or set the rock ridge extensions then the error will be obvious here since the file names and directory structure will be wrong.

```
        mount −t iso9660 −o ro,loop=/dev/loop0 /tmp/redhat.img /mnt/cdrom
```

When you're done, don't forget to unmount it.

```
        umount /mnt/cdrom
```

Burn the disk

Be sure to set the correct speed for your device. This command is for a 4X CDR. You may have a 1X or 2X drive.

```
cdrecord −v speed=4 dev=0,3 /tmp/redhat.img
```

In the above command, it is assumed that the CD writer is on SCSI bus 0, with ID number 3.

7.Installing from the CD

When installing from the new CD, you may first need to create a bootable installation diskette. IMPORTANT: use a NEW, freshly MS−DOS formatted diskette! Using an old, worn−out, faulty diskette can result in strange problems during the installation!

On a Linux system, you can create the diskette using the `dd` command:

```
dd if=/mnt/cdrom/images/boot.img of=/dev/fd0 bs=1440k
```

On a system running DOS or Windows−9x, you need to use the `RAWRITE.EXE` program, which is found on the CD in the `dosutils` directory.

Shut down the machine you want to install (or do a system upgrade) on, insert the boot diskette and your freshly burned CD, and let the machine boot from the diskette. For more information on the installation process, se the documents and the Installation−HOWTO or the Bootdisk−HOWTO which are on the CD in the `doc/HOWTO` directory.

7.1 Booting from a bootable CD

Most modern machines are able to boot directly from a CD, provided it is made bootable with the procedure outlined in section Creating a bootable CD. Often, however, you need to change the setting of the BIOS to make the CD drive bootable. See the documentation for your mother board to see how it's done.

8.Other Linux distributions

The information in this document also applies to other Linux distributions that are RedHat clones, such as Mandrake, although we have not actually tried them ourself. The Mandrake distributions uses a simpler `hdgenlist` program (it's a shell script), and other details may be different.

The LinuxPPC distribution for Apple PowerMacs and other PowerPPC machines is also somewhat similar to the RedHat distribution. When making a distribution for the PowerMac platform, you need to use mkhybrid) instead of `mkisofs`. We have not tried it, but we'd like to hear from people who have.

9.This document...

The SGML source of the most recent version of this document can be retrieved from http://imsb.au.dk/~mok/linux/doc/RedHat−CD.sgml (HTML version).

You can link to the RedHat−CD mini HOWTO by making the following reference on your home page:

```
<a href="http://imsb.au.dk/~mok/linux/doc/RedHat-CD.html">
<img src="http://imsb.au.dk/~mok/logos/RedHat-CD.png" alt="RedHat CD mini-howto" height=30 nosave
</a>
```

9.1 Related documentation

Ed Schlunder <zilym@asu.edu> has written a utility called `fix−rhcd` to let you check your Red Hat Linux distribution mirror for matching file sizes, names, permissions, and symlinks against an "ls −lNR" listing from the offical Red Hat ftp site. Any permissions that are wrong are changed to match the `ls` listing. See the fix−rhcd homepage.

Rod Smith <smithrod@bellatlantic.net> has written a Do−It−Yourself Red Hat Installation guide, which also includes information on creating RedHat install CD's. Especially aimed at burning a CD from a non−UNIX system. Find it at http://members.bellatlantic.net/~smithrod/rhjol.html.

A document in french ``*Comment graver un CD de la RedHat 5.x a partir de fichiers telecharges sur Internet...''* by <skooter@hol.fr> is available from http://linuxfr.org/docs/article/gravure−CD−RH51.html.

With the sense of the good things in life Jussi Torhonen from Finland <jussi.torhonen@tietosavo.fi> tells us Howto make a homebrew bootable RedHat Linux 5.2 CD−ROM.

>From the LDP project, see the CD−writing HOWTO.

9.2 Acknowledgements

Apart from those mentioned above, thanks are given to the following people for valuable input, feedback and discussions:

- Lars Christensen <larsch@cs.auc.dk>

- Thomas Duffy <tbd@cs.brown.edu>
- Dawn Endico <dawn@math.wayne.edu>
- Seva <seva@null.cc.uic.edu>
- Michael Thomas Cope <mcope@orion.ac.hmc.edu>
- Charles J. Fisher <charles_fisher@bigfoot.com>
- Eric Thomas <eric.thomas@ericsson.com>
- Gordon Yuen <gdccyuen@yahoo.com>

10.DISCLAIMER

While the given information in this document is believed to be correct, the authors assume no responsibility whatsoever for any damage to hardware and/or software, or any loss of data resulting from the procedures outlined in this document.