RPM−for−Unix HOW−TO

Table of Contents

Table of Contents

RPM−for−Unix HOW−TO

Al Dev (Alavoor Vasudevan) alavoor[AT]yahoo.com

v1.7, 12 Feb 2002

This document will help you to setup Redhat RPM program on various Unix operating systems like Solaris, AIX, Irix, HPUX and many others. The information in this document applies to all flavors of Unix like − Solaris, HPUX, AIX, SCO, Sinix, BSD, NetBSD, Apple Macintosh (which is BSD), UnixWare, etc.. and may be useful to other operating systems (MS Windows 2000) which support "C" compiler.

---

1. Introduction

(The latest version of this document is at http://www.milkywaygalaxy.freeservers.com. You may want to check there for changes).

Redhat RPM program is the key program to install Redhat Linux Source RPMs. There are more than 8000 Redhat application programs available on the internet in rpm source−code format. If you just install **rpm** program on other flavors of Unix like Solaris, AIX, Irix or HPUX, then you can very easily compile, build and install any of those 8000 application source RPMs.

RPM is the "gateway" to thousands of applications and programs, which are in source code rpm format.

Maintainence of application programs is extremely easy with RPM program. You can very easily remove, add and upgrade the application package from the system. Using RPM program you can query the system to find the list of packages installed and you can even list the files installed by a particular package. RPM program can also give the exact path locations of the files installed by a particular package.

The upgrades and updates can be automated using the **rpm** program and the Redhat Network support.

*The redhat **rpm** program is extremely powerful, extremely easy to use and is the most popular packaging program in the world. Packaging programs on other Unix platforms like pkgadd, smitt are very primitive and* *are not as powerful as* ***rpm*** *program. And* ***rpm*** *program is very portable and works on most Unix operating systems.*

The guidelines given here can be useful for setting up Debian package program **deb** on Solaris and other Unix flavors. GNU Debian Linux is another popular distribution and uses **deb** program which is similar to **rpm** program . You can use either **rpm** or **deb** program.

---

2. Install RPM

To install RPM program you need gcc and BerkeleyDB. First you should decide the version of Redhat Linux which you want to use. The versions of Redhat Linux are 7.1 (latest), 6.2, 5.x (old versions).

To build **rpm** package you need atleast BerkeleyDB and gcc compiler. It is very important that you MUST select proper versions of gcc, BerkeleyDB and RPM source based on whether you want to represent Redhat 7.1, Redhat 6.2 or 5.x. If you happen to have a Redhat Linux box then do −

```
bash$ rpm −qa | grep −i gcc
bash$ rpm −qi gcc−2.96−81

bash$ rpm −qa | grep −i db3
bash$ rpm −qi db3−3.1.17−7

bash$ rpm −qa | grep −i rpm
bash$ rpm −qi rpm−4.0.2−8
```

From above, you can get the proper version numbers.

WARNING: ***If you do not use proper version numbers of gcc, db3 and rpm combinations then RPM program may not compile or work properly.*.**

For example − to target Redhat Linux 7.1 on Solaris, you should use gcc version 2.96, Sleepycat Berkeley DB version 3.1.17 and RPM source version 4.0.2. Any other combination of versions **WILL NOT** work.

You can see the list of files and programs needed to build rpm. Login on linux box and:

```
bash$ rpm −qR rpm−4.0.2−8
gawk
fileutils
textutils
mktemp
popt
glibc >= 2.1.92
db1 = 1.85
/bin/sh
/bin/sh
/sbin/ldconfig
ld−linux.so.2
libbz2.so.1
libc.so.6
libdb−3.1.so
```

```
libdb.so.2
libpopt.so.0
librpm.so.0
librpmbuild.so.0
librpmio.so.0
libz.so.1
/bin/sh
libc.so.6(GLIBC_2.0)
libc.so.6(GLIBC_2.1)
libc.so.6(GLIBC_2.1.3)
libc.so.6(GLIBC_2.2)
libdb.so.2(GLIBC_2.0)
rpmlib(CompressedFileNames) <= 3.0.4-1
bash$

To see to which package gawk belongs to do:

bash$ rpm -qif /usr/bin/gawk
Name        : gawk                         Relocations: (not relocateable)
Version     : 3.0.6                             Vendor: Red Hat, Inc.
Group       : Applications/Text             Source RPM: gawk-3.0.6-1.src.rpm
```

2.1 Get Gnu "gcc"

First, you need to install the Gnu C compiler gcc. You can use one of the following three sources:

- Unix gcc package from Solaris package , AIX package , IRIX package , HPUX package , UnixWare package , Mac OS package .
- Redhat source rpm from cdrom or from http://www.redhat.com or from http://www.rpmfind.net
- Download binaries from http://www.gnu.org or gcc−home where you can find gcc binary for Unix.

If you downloaded Solaris/Unix package:

```
# su - root
# man pkgadd
# pkgadd -d gcc-sparc*.pkg
```

If you downloaded rpm package:

```
bash$ mkdir my_temp_build
bash$ cd my_temp_build
bash$ rpm2cpio gcc-*.src.rpm | cpio -dimv
bash$ tar zxvf gcc*.tar.gz
bash$ cd gcc-2.96
bash$ ./configure
bash$ make
```

2.2 Get BerkeleyDB

Second, you need to install the BerkeleyDB embedded database from http://www.sleepycat.com. RPM internally uses the embedded database BerkeleyDB to store the package information. You can use one of the following three sources:

- Redhat source rpm from cdrom or from http://www.redhat.com or from http://www.rpmfind.net
- Download binaries from http://www.sleepycat.com go here and see Berkeley DB Patches and ChangeLogs scroll down and select proper version like 3.1.17 (one which matches with Redhat Linux 7.1).
- Unix gcc package from Solaris package , AIX package , IRIX package , HPUX package , UnixWare package , Mac OS package .

If you downloaded source rpm or source tar.gz file:

```
bash$ mkdir my_temp_build
bash$ cd my_temp_build
bash$ rpm2cpio db3-3.1.17-7.src.rpm | cpio -dimv
bash$ tar zxvf db3*.tar.gz
bash$ cd db3-3.1.17
bash$ cd build_unix
bash$ ../dist/configure --enable-compat185
bash$ make
bash$ make install
```

If you downloaded Unix package:

```
# su - root
# man pkgadd
# pkgadd -d db-3*-sparc*.pkg
```

3. BerkeleyDB and gcc

Download BerkeleyDB and gcc for Unix as given below:

3.1 Solaris − BerkeleyDB and gcc

You can get gcc and BerkeleyDB for Solaris from:

- Solaris gcc package from http://www.sunfreeware.com
- Sparc, Sun Solaris, Ultra Sparc http://www.ultralinux.org

3.2 AIX − BerkeleyDB and gcc

You can get gcc and BerkeleyDB for AIX from:

- IBM main site http://www−1.ibm.com/servers/aix/library and download−site.
- Univ of California AIX site http://aixpdslib.seas.ucla.edu
- Norcal org site http://www.aiche−norcal.org/AIXtip/freeware.htm
- Bull site http://www.bull.de/pub
- The PowerPC arch rpms are at http://penguinppc.org.

3.3 IRIX Silicon Graphics − BerkeleyDB and gcc

You can get gcc and BerkeleyDB for SGI IRIX from:

- SGI main site http://freeware.sgi.com
- SGI freware alphabetical listing http://freeware.sgi.com/index−by−alpha.html

3.4 HPUX − BerkeleyDB and gcc

You can get gcc and BerkeleyDB for HPUX from:

- Gnu gcc for HPUX http://devresource.hp.com/OpenSource/Tools/gcc/gcc.html
- HPUX opensource tools http://devresource.hp.com/OpenSource/Tools and tool−list
- HPUX freeware
- HPUX Porting and Archive Center http://hpux.cs.utah.edu and alphabetical listing and Gnu

3.5 UnixWare − BerkeleyDB and gcc

You can get gcc and BerkeleyDB for UnixWare from:

- FreeBird site http://www.freebird.org
- Skunkware site http://www.sco.com/skunkware
- Gnu gcc for gcc−unixware and faq−site

3.6 Apple Macintosh OS X (Unix) − BerkeleyDB and gcc

You can get gcc and BerkeleyDB for Apple Macintosh OS X (which is BSD Unix) from:

- Gnu gcc for Apple Mac OS shipped with OS called "ProjectBuilder IDE" http://developer.apple.com/tools/projectbuilder and faq
- List of Opensource packages for Mac Darwin−project

4. Compiling RPM

After building Berkeley DB3 and gcc, compile rpm program now. Get rpm source code from one of the following two sources:

- Redhat source rpm from cdrom or from http://www.redhat.com or from http://www.rpmfind.net and look for proper version like 7.1
- Download source code of rpm from http://www.rpm.org go here and select proper version like 4.0.2 (one which matches with Redhat Linux 7.1). Download from ftp−site.

If you downloaded rpm*.tar.gz file:

```
bash$ mkdir my_temp_build
bash$ cd my_temp_build
bash$ tar zxvf rpm*.tar.gz
bash$ cd rpm-4.0.2
bash$ export LIBS='-L/usr/local/BerkeleyDB.3.1/lib'
bash$ export CPPFLAGS='-I/usr/local/BerkeleyDB.3.1/include'
bash$ ./configure
bash$ make
```

If you downloaded rpm source code file follow the steps below. But you may need rpm2cpio program, which is packaged in rpm package. Get rpm package (may be old version like 3.0) from Unix packages .

```
bash$ mkdir my_temp_build
bash$ cd my_temp_build
bash$ rpm2cpio rpm-4.0.2-8.src.rpm | cpio -dimv
bash$ tar zxvf rpm*.tar.gz
bash$ cd rpm-4.0.2
bash$ export LIBS='-L/usr/local/BerkeleyDB.3.1/lib'
bash$ export CPPFLAGS='-I/usr/local/BerkeleyDB.3.1/include'
bash$ ./configure
bash$ make
```

4.1 Troubleshooting RPM Compile

If you encounter any problem in compiling rpm program, you may have to do some minor corrections.

In case of Solaris 8 and rpm v4.0.2, I got these problems:

- Error db3/db.h not found in lib/db3.c. Solution is − create a soft link:

```
bash# ln -s /usr/local/BerkeleyDB.3.1.17/include  /usr/include/db3

Or edit the file lib/db3.c and change
#include <db3/db.h>
to
#include <db.h>
```

- Edit Makefile and put −L before libmisc.a at line 186
- Compile error in file rpmsort.c: Function errx() not found. Solution is − edit the file rpmsort.c and copy and paste the function errx() from misc/err*.c. Or comment out line having errx() and put fprintf(stderr, "Error encountered − blah, blah"); exit(0);

5. Using RPM

See also RPM−HOWTO document.

5.1 Basic RPM Commands

In its simplest form, RPM can be used to install, erase or upgrade packages:

```

bash# rpm -i foobar-1.0-1.i386.rpm   ...(to install packages)
bash# rpm -i ftp://ftp.redhat.com/pub/redhat/RPMS/foo-1.0-1.i386.rpm ...(to install package via ftp)
bash# rpm -e foobar  ...(to uninstall package)
bash# rpm -U foobar-1.0-1.i386.rpm  ...(to upgrade package)
bash# rpm --help  ...(to see help on rpm options)
bash# man rpm  ...(to read online manual page)
```

5.2 RPM Build Commands

To build rpm packages

```

bash# rpm -i foobar*.src.rpm
bash# cd /usr/src/redhat/SPECS
bash# rpm -ba foobar-1.0-1.spec

To build in incremental steps, do:
bash# rpm -bp foobar-1.0-1.spec    ...( to do prep stage)
bash# rpm --short-circuit -bc foobar-1.0-1.spec    ...( to do build stage)
bash# rpm --short-circuit -bi foobar-1.0-1.spec    ...( to do install stage)
bash# rpm -ba foobar-1.0-1.spec  ... (to do binary and source package)
```

5.3 RPM Query Commands

You can query the rpm database with these powerful commands:

```
bash$ rpm -qpl foobar-1.0-1.i386.rpm  ....(list of files in a rpm package)
bash$ rpm -ql foobar-1.0-1   ....(list of files from already installed package)

bash$ rpm -qpR foobar-1.0-1.i386.rpm  ....(list packages on which this package depends)
bash$ rpm -qR foobar-1.0-1  ....(list packages on which this installed package depends)

bash$ # rpm -q foo  ...(print package name, version, and release number of package foo)
foo-2.0-1

bash$ rpm -qa | less   ....(list all the installed package)
bash$ rpm -qa | grep -i kde   ....(list all the installed package matching kde)
bash$ rpm -qif /bin/ls ....(list the package which installed the file /bin/ls)

To show the values RPM will use for all of the options that may be set in rpmrc files
(/usr/lib/rpm/rpmrc, /etc/rpmrc, ~/.rpmrc ), type:
bash$ rpm --showrc | less
```

5.4 RPM Verify Commands

Verifying a package compares information about files installed from a package with the same information from the original package. Among other things, verifying compares the size, MD5 sum, permissions, type, owner and group of each file.

The command **rpm −V** verifies a package. You can use any of the Package Selection Options listed for querying to specify the packages you wish to verify. A simple use is **rpm −V** foo which verifies that all the files in the foo package are as they were when they were originally installed. For example:

```
bash$ rpm -Vf /bin/vi ...(To verify a package containing particular file)
bash$ rpm -Va ...(To verify ALL installed packages)
bash$ rpm -Vp foo-1.0-1.i386.rpm ...(To verify an installed package against an RPM package file)
```

This can be useful if you suspect that your RPM databases are corrupt.

If everything verified properly there will be no output. If there are any discrepancies they will be displayed. The format of the output is a string of 8 characters, a possible "c" denoting a configuration file, and then the file name. Each of the 8 characters denotes the result of a comparison of one attribute of the file to the value of that attribute recorded in the RPM database. A single "." (period) means the test passed. The following characters denote failure of certain tests:

```
     5 -- MD5 checksum
     S -- File size
     L -- Symbolic link
     T -- File modification time
     D -- Device
```

```
U -- User
G -- Group
M -- Mode (includes permissions and file type)
? -- Unreadable file
```

If you see any output, use your best judgment to determine if you should remove or reinstall the package, or otherwise fix the problem.

6. Installing 8000 RPMs

Once RPM program is installed, it is a gateway to thousands of RPM packages which can be easily installed on the system.

Prepare the rpm directories

```
bash# su - root
bash# mkdir -p /usr/local/src/redhat/BUILD
bash# mkdir -p /usr/local/src/redhat/RPMS/sparc64
bash# mkdir -p /usr/local/src/redhat/RPMS/noarch
bash# mkdir -p /usr/local/src/redhat/SOURCES
bash# mkdir -p /usr/local/src/redhat/SPECS
bash# mkdir -p /usr/local/src/redhat/SRPMS
```

You may want to edit the **rpmrc** file. In case of solaris 8, I had to remove the **−m64** option for gcc, since it was giving compile errors. To show the values, RPM will use for all of the options that may be set in rpmrc files (/usr/local/lib/rpm/rpmrc, /usr/lib/rpm/rpmrc, /etc/rpmrc, /.rpmrc), type:

```
bash$ rpm --showrc | less
```

6.1 Bootstrap Programs − The Rocket Soars Up!!

Before building rpms you need the following basic programs:

- install program from fileutils*.tar.gz
- patch*.tar.gz
- autoconf*.tar.gz
- automake*.tar.gz
- libtool*.tar.gz
- gcc*.tar.gz

You should download the tar.gz files or Unix packages . I downloaded solaris packages for fileutils.pkg, patch.pkg, libtool.pkg, gcc and others. You should install them in a temporary location like $HOME/localtmp

```
bash# mkdir $HOME/localtmp
bash# cd $HOME/localtmp
bash# gzip -d libtool*.tar.gz
```

```
bash# tar -xvf libtool*.tar
bash# ./configure --with-prefix=$HOME/localtmp
bash# make; make install
```

You should include the temporary location of autoconf by:

```
bash# PATH=$HOME/localtmp/bin:$PATH
bash# export PATH
```

6.2 Install Foundation RPMs

There are few basic rpms which must be installed before any other rpm is installed. In this section, foundation rpms are listed which are found by using:

```
bash# rpm -qR <packagename>
bash# rpm -qR textinfo
bash# rpm -qR fileutils
bash# rpm -qR setup
```

The foundation rpms in the order of dependency are as follows:

- fileutils*.rpm
- grep*.rpm (You may have to edit grep.spec and commentout −−without−included−regex)
- gawk*.rpm
- sed*.rpm
- texinfo*.rpm
- zlib*.rpm and zlib−devel
- patch*.rpm
- setup*.rpm
- filesystem*.rpm (You may not want install this if it effects /proc directory)
- textutils*.rpm
- glibc−common*.rpm
- basesystem*.rpm
- mktemp*.rpm
- bash*.rpm
- m4*.rpm (autoconf needs this)
- autoconf
- bison
- binutils >= 2.9.1.0.25
- gas, as, ld which are in binutils
- shutils – for 'id' command

Second stage foundation rpms are as follows. After installing the foundation rpms, next important rpm is gcc, the order of rpms you need is:

- glibc*.rpm

- binutils*.rpm
- kernel−headers*.rpm
- glibc−devel*.rpm
- gcc*.rpm

Third stage rpms are as follows:

- popt*.rpm
- rpm*.rpm
- perl*.rpm
- And many others....

As from previous section you should have exported temp location of autoconf, gcc and other programs by:

```
bash# PATH=$HOME/localtmp/bin:$PATH
bash# export PATH

bash# rpm -i fileutils*.src.rpm
bash# rpm -i zlib*.src.rpm
bash# rpm -i texinfo*.src.rpm

bash# cd /usr/local/src/redhat/SPECS
bash# rpm -ba fileutils.spec
```

After the build is successful, install it with "nodeps and excludedocs" to minimize the failures. After substantial installations of many foundation rpms you can "freshen" the rpms with rpm command.

```
bash# cd /usr/local/src/redhat/RPMS/sparc64
bash# rpm -i --nodeps --excludedocs fileutils*.rpm
```

Move on to build and install the next rpm in the list texinfo, zlib, patch,....

6.3 Troubleshoot Building Foundation RPMs

On Solaris 2.8, I encountered following problems which were fixed by correcting the spec file:

- **texinfo*.rpm:** Edit the file texinfo.spec and change __spec_install_post, %build and %install sections:

```
%define Rpmpath /usr/local
%define __spec_install_post %{Rpmpath}/lib/rpm/brp-strip \; %{Rpmpath}/lib/rpm/brp-strip-comment-note \; rm -f

%build
%define _mandir %{_prefix}/share/man
%define _infodir %{_prefix}/share/info
%configure

%install
#mkdir -p ${RPM_BUILD_ROOT}/{etc,sbin}        .....this line is causing problems
```

```
mkdir -p ${RPM_BUILD_ROOT}/etc
mkdir -p ${RPM_BUILD_ROOT}/sbin
```

- **textutils*.rpm:** Create a new unix group called 'other'. Type 'groupadd other' and see 'man groupadd'. Second problem − edit textutils.spec file and in %files section replace hard−coded filenames/pathnames with %{_prefix}.

```
%files
# replace below line
#/usr/bin/*
# with the line given below
%{_prefix}/bin/*
```

- **patch*.rpm:** Edit patch.spec file and in %files section replace hard−coded filenames/pathnames with %{_prefix}.

```
%files
# replace below line
#/usr/bin/*
# with the line given below
%{_prefix}/bin/*
```

- **gawk*.rpm:** Edit gawk.spec file and in %files section and %install section replace hard−coded filenames/pathnames like /usr/something with %{_prefix}/something.
- **m4*.rpm:** Edit m4.spec file and in %build section, the configure takes only −−prefix and −−exec−prefix. Also comment out autoconf.

7. Create gcc RPM Package

The GNU gcc rpm is the most important rpm package which you should create after successfully creating rpm of RPM package. In order to create GNU gcc rpm package, you need to install rpms, unix pkgs or manual configure and install the following (given in order of dependency) :

- texinfo
- patch
- zlib and zlib−devel
- fileutils >= 4.0.41
- m4 >= 1.1 (autoconf needs this)
- autoconf
- bison
- binutils >= 2.9.1.0.25
- gas, as, ld which are in binutils
- cpp − manual install gcc
- gawk
- shutils − for 'id' command
- pthreads − Posix threads library

You should download the tar.gz files or Unix packages . I downloaded solaris packages for texinfo.pkd, fileutils.pkg, patch.pkg, libtool.pkg, gcc and others. If you download tar.gz source file then you should install them in a temporary location like $HOME/localtmp

```
bash# mkdir $HOME/localtmp
bash# cd $HOME/localtmp
bash# gzip −d libtool*.tar.gz
bash# tar −xvf libtool*.tar
bash# ./configure −−with−prefix=$HOME/localtmp
bash# make; make install
```

You should include the temporary location of autoconf by:

```
bash# PATH=$HOME/localtmp/bin:$PATH
bash# export PATH
```

Now you are ready to build gcc rpm:

```
bash# rpm −i gcc*.src.rpm
bash# cd /usr/local/src/redhat/SPECS
bash# rpm −ba gcc*.spec
```

On Solaris 2.8 I had to make few changes to gcc*.spec file:

1. In %build section : Insert ***−−enable−hacker−mode*** option for configure command. This is to correct error − "This configuration not supported in solaris 2.8". If ***−−enable−hacker−mode*** option does not work then you may have to comment out gcc_targetplatform within the code %ifarch sparc .... %endif, just before the configure command.
2. In %build section : For the configure command comment out %{_target_platform}, if configure fails.
3. In %install section: Comment out numprocs = $(getconf _NPROCESSORS) and put numprocs=1. Since my box had 2 cpus I put numprocs=2. The solaris has /usr/bin/getconf which is different from GNU getconf in glibc.

Setup shared lib in solaris:

```
solaris# cd /var/ld
solaris# man ld.so.1
solaris# man crle
solaris# crle  < hit return to see config info>
solaris# crle /usr/lib:/usr/local/lib:/usr/local/pgsql/lib:/usr/local/BerkeleyDB.3.1/lib
```

7.1 Solaris Sparc gcc, glibc, kernel−headers and binutils

You can download the source code rpm for sparc from sparc−gcc or go to http://www.rpmfind.net and type **sparc** in the search field. You can also download sparc−glibc, sparc−kernel−headers and sparc−binutils source rpms.

```
solaris# rpm -i sparc-gcc*.src.rpm
solaris# cd /usr/local/src/redhat/SPECS
solaris# rpm -ba sparc-gcc.spec
```

You may want to edit sparc−binutils.spec and in %install section: Comment out numprocs = $(getconf _NPROCESSORS) and put numprocs=1. Since my box had 2 cpus I put numprocs=2. The solaris has /usr/bin/getconf which is different from GNU getconf in glibc. Also change %files section change %doc /usr/man/man1/sparc−linux−ld.1.gz to %doc /usr/man/man1/sparc−linux−ld.1.

7.2 PowerPC gcc, glibc, kernel−headers and binutils

The PowerPC arch rpms are at http://penguinppc.org.

7.3 Create glibc RPM

If configure complains that your platform is not supported, then you may want to edit glibc.spec file and in %configure section put *−−enable−hacker−mode*,

For Solaris there is ready−made sparc−glibc*.src.rpm from sparc−gcc or go to http://www.rpmfind.net and type **sparc** in the search field.

The glibc needs the sparc−kernel−headers*.rpm which you can get from sparc−gcc.

8. Use Debian Packages − Convert deb to rpm

Debian Linux is another popular distribution. You can use the debian packages for Sparc, PowerPC, m68k, MIPS, Alpha, Arm, Intel X86 or Itanium.

8.1 Debian Packing Program

To build real debs that contain software, you'll need: **binutils, cpp, cpio, dpkg−dev, file, gcc, libc6−dev, make, patch, perl dh−make, debhelper, devscripts, fakeroot** and **lintian** . If your package needs to prompt the user for configuration information when it installs, get **debconf** .

The main debian programs you need to get are:

- dpkg−dev : This package contains the tools needed to create Debian archives. It also contains the Debian Programmers and Policy manuals.
- debmake : This is a higher level set of tools for creating Debian packages. It makes the whole process tremendously easier.

* debhelper : A collection of programs that can be used in a debian/rules file to automate common tasks related to building debian packages. Programs are included to install various files into your package, compress files, fix file permissions, integrate your package with the debian menu system, suidmanager, doc−base, etc. Most debian packages use debhelper as part of their build process.
* debconf : Debconf is a configuration management system for debian packages. Packages use Debconf to ask questions when they are installed.
* apt−howto : A Guide to APT. This manual tries to be a quick but complete source of information about the APT system and its features. It documents the main uses of APT with many examples. (You may want to edit apt−howto*.sgml and change debiandoc to linuxdoc and run sgml2html).

Get these from http://www.debian.org/distrib/packages, go here and type in the search box **dpkg−dev** or **apt−howto** and select Stable, Testing or Unstable in the distribution. Download and build on your Unix system (Solaris, AIX, HPUX).

See Debian Package Management , Debian Policy Manual, Debian New Maintainer's Guide and apt+rpm howto.

8.2 Debian to Redhat RPM

To install Debian packages on Sun Solaris 2.8 (Sparc/Ultra−sparc) I downloaded all the debian packages for sparc from http://www.debian.org. Go here and click on distribution−>sparc. Better still, purchase the Debian binary and source cdrom for Sparc (or for PowerPC, Alpha). Use **alien** program to convert the debian packages to redhat **rpm**.

Get **alien** program from http://www.rpmfind.net.

```
solaris# rpm −i alien*.src.rpm
solaris# rpm −ba alien.spec
solaris# rpm −i alien.sparc.rpm
solaris# alien −r glibc*.deb     (..... this will create the rpm file)
solaris# rpm −i glibc−sparc*.rpm
```

Convert all debian binary packages to rpm and install on Solaris 2.8 OS.

Note **alien** program works only for binary debian packages.

---

9. Convert RPM to Solaris Package rpm2pkg

After building and installing all the RPMs on Solaris (or any Unix), you can convert those programs to Solaris (or any Unix) packages.

To convert Redhat rpm to Solaris package do:

```
bash# mkdir −p  $HOME/tmp/myprogram−1.4/usr/local
```

```
bash# cd $HOME/tmp/myprogram-1.4/usr/local
bash# rpm2cpio rpm-4.0.2-8.sparc64.rpm | cpio -dimv
bash# tar -xvf rpm*.tar.gz
bash# make_package
```

And run the perl script **make_package** given at

10. Related URLs

- Main site of RPM http://www.rpm.org
- Maximum RPM book http://www.rpmdp.org/rpmbook/node1.html
- RPM HOWTO http://linuxdoc.org/HOWTO/RPM−HOWTO/index.html
- RH manual − rpm
- RPM Finder http://www.rpmfind.net
- RPM related sofware http://rpm.org/software.html
- RPM mailing lists http://rpm.org/mailing_list.html
- Sparc, Sun Solaris, Ultra Sparc http://www.ultralinux.org
- The PowerPC arch rpms are at http://penguinppc.org.

Linux goodies main site is at http://www.milkywaygalaxy.freeservers.com Mirror sites are at − http://aldev0.webjump.com, angelfire, geocities, virtualave, 50megs, theglobe, NBCi, Terrashare, Fortunecity, Freewebsites, Tripod, Spree, Escalix, Httpcity, Freeservers.

11. Other Formats of this Document

This document is published in 14 different formats namely: DVI, Postscript, Latex, Adobe Acrobat PDF, LyX, GNU−info, HTML, RTF(Rich Text Format), Plain−text, Unix man pages, single HTML file, SGML (Linuxdoc format), SGML (Docbook format), and MS WinHelp format.

This howto document is located at:

- http://www.linuxdoc.org and click on HOWTOs and search for the howto document name using CTRL+f or ALT+f within the web−browser.

You can also find this document at the following mirrors sites:

- http://www.caldera.com/LDP/HOWTO
- http://www.linux.ucla.edu/LDP
- http://www.cc.gatech.edu/linux/LDP

- http://www.redhat.com/mirrors/LDP
- Other mirror sites near you (network−address−wise) can be found at http://www.linuxdoc.org/mirrors.html select a site and go to directory /LDP/HOWTO/xxxxx−HOWTO.html

- You can get this HOWTO document as a single file tar ball in HTML, DVI, Postscript or SGML formats from − ftp://www.linuxdoc.org/pub/Linux/docs/HOWTO/other−formats/ and http://www.linuxdoc.org/docs.html#howto

- Plain text format is in: ftp://www.linuxdoc.org/pub/Linux/docs/HOWTO and http://www.linuxdoc.org/docs.html#howto

- Single HTML file format is in: http://www.linuxdoc.org/docs.html#howto

  A single HTML file can be created with the command (see man sgml2html) − sgml2html −split 0 xxxxhowto.sgml

- Translations to other languages like French, German, Spanish, Chinese, and Japanese are in ftp://www.linuxdoc.org/pub/Linux/docs/HOWTO and http://www.linuxdoc.org/docs.html#howto Any help from you to translate to other languages is welcome.

The document is written using a tool called "SGML−Tools" which can be got from: http://www.sgmltools.org Compiling the source you will get the following commands like:

- sgml2html xxxxhowto.sgml (to generate html file)
- sgml2html −split 0 xxxxhowto.sgml (to generate a single page html file)
- sgml2rtf xxxxhowto.sgml (to generate RTF file)
- sgml2latex xxxxhowto.sgml (to generate latex file)

11.1 Acrobat PDF format

A PDF file can be generated from postscript file using either acrobat **distill** or **Ghostscript**. And a postscript file is generated from DVI which in turn is generated from a LaTex file. You can download distill software from http://www.adobe.com and from unix and from windows . Given below is a sample session:

```
bash$ man sgml2latex
bash$ sgml2latex filename.sgml
bash$ man dvips
bash$ dvips -o filename.ps filename.dvi
bash$ distill filename.ps
bash$ man ghostscript
bash$ man ps2pdf
bash$ ps2pdf input.ps output.pdf
bash$ acroread output.pdf &
```

Or you can use the Ghostscript command **ps2pdf**. ps2pdf is a work−alike for nearly all the functionality of Adobe's Acrobat Distiller product: it converts PostScript files to Portable Document Format (PDF) files. **ps2pdf** is implemented as a very small command script (batch file) that invokes Ghostscript, selecting a special "output device" called **pdfwrite**. In order to use ps2pdf, the pdfwrite device must be included in the makefile when Ghostscript was compiled; see the documentation on building Ghostscript for details.

11.2 Convert Linuxdoc to Docbook format

This document is written in linuxdoc SGML format. The Docbook SGML format supercedes the linuxdoc format and has a lot more features than linuxdoc. The linuxdoc is very simple and easy to use. To convert linuxdoc SGML file to Docbook SGML use the program **ld2db.sh** and some Perl scripts. The ld2db output is not 100% clean and you need to use the **clean_ld2db.pl** Perl script. You may need to manually correct a few lines in the document.

- Download the ld2db program from http://www.dcs.gla.ac.uk/~rrt/docbook.html or from Milkyway Galaxy site
- Download the cleanup_ld2db.pl perl script from from Milkyway Galaxy site

The ld2db.sh is not 100% clean, so you will get some errors when you run it.

```
        bash$ ld2db.sh file-linuxdoc.sgml db.sgml
        bash$ cleanup.pl db.sgml > db_clean.sgml
        bash$ gvim db_clean.sgml
        bash$ docbook2html db.sgml
```

And you may have to manually edit some of the minor errors after running the Perl script. For example you may need to put closing tag < /Para> for each < Listitem>

11.3 Convert to MS WinHelp format

You can convert the SGML howto document to a Microsoft Windows Help file, First convert the sgml to html using:

```
        bash$ sgml2html xxxxhowto.sgml     (to generate html file)
        bash$ sgml2html -split 0   xxxxhowto.sgml (to generate a single page html file)
```

Then use the tool HtmlToHlp. You can also use sgml2rtf and then use the RTF files for generating winhelp files.

11.4 Reading various formats

In order to view the document in dvi format, use the xdvi program. The xdvi program is located in tetex−xdvi*.rpm package in Redhat Linux which can be located through ControlPanel | Applications | Publishing | TeX menu buttons. To read a dvi document give the command:

```
                xdvi -geometry 80x90 howto.dvi
                man xdvi
```

And resize the window with the mouse. To navigate use Arrow keys, Page Up, Page Down keys, also you can use 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' letter keys to move up, down, center, next page, previous page etc. To turn off expert menu press 'x'.

You can read a postscript file using the program 'gv' (ghostview) or 'ghostscript'. The ghostscript program is in the ghostscript*.rpm package and the gv program is in the gv*.rpm package in Redhat Linux which can be located through ControlPanel | Applications | Graphics menu buttons. The gv program is much more user friendly than ghostscript. Also ghostscript and gv are available on other platforms like OS/2, Windows 95 and NT. You can view this document even on those platforms.

- Get ghostscript for Windows 95, OS/2, and for all OSes from http://www.cs.wisc.edu/~ghost

To read a postscript document give the command:

```
gv howto.ps
ghostscript howto.ps
```

You can read an HTML format document using Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web browser or any of the 10 other web browsers.

You can read the latex, LyX output using LyX an X Window front end to LaTex.

11.5 Convert HTML to SGML html2sgml

To convert any HTML document to Linuxdoc SGML use the tool 'html2sgml' from http://www.abc.se/~m9339/prog/html2sgml.html.

---

12. Copyright
