

Minimum dPKG-buildpackage

Building *.deb* Packages from Source

By Gilbert Detillieux, 7 Nov 2023 MUUG Meeting



University
of Manitoba

About me...

- First exposed to UNIX at U of M in 1979, worked on it starting in 1980
- Began current job as a SysAdmin in October 1989 (32+ years ago)
- Introduced to Linux around 1992 (Slackware, Red Hat, and later Debian/Ubuntu & Raspberry Pi OS)
- Familiar with ***rpmbuild*** processes, but new to Debian build processes
- Need to find Debian equivalent to *Maximum RPM*

Why Build Packages from Source?

- No existing binary packages found
- Existing packages are out of date
- Existing packages not configured/made with necessary features enabled (or deps installed)
- Bugs need fixing, or security patches
- Want benefits of package management (easy install/upgrade/removal, in standard system directories)

Case in Point: *libapache2-mod-ruuid2*

- Needed alternative to (obsolete) *mod_suphp*
- A suggested alternative is *mod_ruid2*, but...
 - Dropped from Debian; Ubuntu package is stale
 - Github source (*mind04/mod-ruuid2*) *master* has since been updated (though version number unchanged?)
 - Fork “*userdir*” has a useful addition
 - I wanted my own “tweak” to config
- Want plug-in replacement binary package
- Demo at end (if time permits)

Assumptions

- **dpkg-buildpackage** is the right tool if...
 - you're starting with installed package source
 - you have the correct source directory hierarchy set up
 - you have the **debian/** subdirectory with prototype files
- Other tools may be better if...
 - you're starting with just a tarball as source
 - you're getting your source via **git**
 - you want to work at either a higher or lower level



Preliminary Steps

- **apt install build-essential debhelper quilt fakeroot devscripts dh-make**
- Add to your startup (e.g. **.bashrc**):
 - **export DEBMAIL="<e-mail_address>@<domain>.<tld>"**
 - **export DEBFULLNAME="<first_name> <last_name>"**
- Add/uncomment in **/etc/apt/sources.list.d/***:
 - **deb-src http://ca.archive.ubuntu.com/ubuntu jammy universe**
 - **deb-src http://ca.archive.ubuntu.com/ubuntu jammy-updates universe**
 - **deb-src http://ca.archive.ubuntu.com/ubuntu jammy-security universe**
- Install any other source dependencies
 - Compilers, build tools, *-dev packages for libraries

Installing Source

- **apt source** *pkgname*
- E.g.: apt source libapache2-mod-ruid2
- Run as non-root user
- **cd** to your top-level source directory first
- Installs package source tree under current dir



Source Directory Structure

- *(top level source directory)*
 - *.dsc
 - *.deb
 - *.orig.tar.*
 - *.debian.tar.*
 - <pkgname>_<maj>.<min>[.<rel>]/
 - debian/
 - changelog
 - control
 - docs
 - files
 - rules
 - patches/
 - series
 - *.patch

https://en.wikipedia.org/wiki/Debian_build_toolchain

<https://wiki.opf-labs.org/display/SP/The+Hello+World+Debian+Package>

Sample *changelog* File

libapache2-mod-ruid2 (0.9.8-3+nmu) unstable; urgency=high

Name & version number
will be used in pkg name

- * Patches to make source from tarball agree with Github master
- * Patches to add experimental userdir support (from Github userdir branch) and update README accordingly
- * Patches to allow <Files*> directives (while still forbidding .htaccess configuration), better handling of null secondary group lists, and logging of setuid/gid changes in "RMode stat|user" configs

-- Gilbert Detillieux <gedetil@cs.umanitoba.ca> **Thu, 10 May 2023 14:17:00 -0500**

libapache2-mod-ruid2 (0.9.8-3) unstable; urgency=high

Date & time will be used
as current "fakeroot" date

- * Port to Apache 2.4 (closes: #709465), many thanks to Arno Töll

-- Alessandro De Zorzi <lota@nonlontano.it> Sat, 09 Nov 2013 08:44:13 +0100

<https://manpages.debian.org/testing/dpkg-dev/deb-changelog.5.en.html>

<https://wiki.debian.org/NonMaintainerUpload>

<https://reproducible-builds.org/specs/source-date-epoch/>

Sample *control* File

Source: libapache2-mod-ruid2

Section: web

Priority: extra

Maintainer: Alessandro De Zorzi <lota@nonlontano.it>

Build-Depends: debhelper (>= 8.0.0), quilt, dh-apache2, apache2-dev, libcap-dev

Standards-Version: 3.9.4

Homepage: <http://sourceforge.net/projects/mod-ruid/>

Package: libapache2-mod-ruid2

Architecture: any

Conflicts: libapache2-mod-suphp

Depends: \${shlibs:Depends}, \${misc:Depends}

Description: suexec module for Apache 2

ruid2 is based on mod_ruid and mod_suid2 but it has better performance than mod_suid2 because it doesn't need to kill httpd children after one request.

.

It makes use of kernel capabilities and after receiving a new request suids again. There are some security issues, for instance if attacker successfully exploits the httpd process, he can set effective capabilities and setuid to root. For this reason, it is recommended to be used in conjunction with a kernel security patch (eg. grsec).

Build-time dependencies
can be auto-installed with
mk-build-deps or apt build-dep

Runtime dependencies
will be auto-installed by
apt install

<https://www.debian.org/doc/debian-policy/ch-controlfields.html>

<https://manpages.debian.org/testing/dpkg-dev/deb-control.5.en.html>

<https://serverfault.com/questions/127625/given-a-debian-source-package-how-do-i-install-the-build-deps>

Sample *rules* File

```
#!/usr/bin/make -f
# *- makefile *-
# Sample debian/rules that uses debhelper.
# This file was originally written by Joey Hess and Craig Small.
# As a special exception, when this file is copied by dh-make into a
# dh-make output file, you may use that output file without restriction.
# This special exception was added by Craig Small in version 0.37 of dh-make.
```

```
# Uncomment this to turn on verbose mode.
export DH_VERBOSE=1
```

```
include /usr/share/quilt/quilt.make
```

```
package=libapache2-mod-ruid2
```

```
DESTDIR=$(CURDIR)/debian/libapache2-mod-ruid2
```

```
%:
```

```
dh $@ --with apache2
```

```
build: $(QUILT_STAMPFN) build-stamp
```

```
build-stamp:
```

```
dh_testdir
```

```
apxs2 -l cap -c mod_ruid2.c
```

```
install: build
```

```
dh_testdir
```

Might need this if
applying patches

Modify as needed for
your package

<https://www.debian.org/doc/manuals/maint-guide/dreq.en.html>

<https://www.debian.org/doc/debian-policy/ch-source.html#main-building-script-debian-rules>

<https://manpages.debian.org/testing/dpkg-dev/deb-src-rules.5.en.html>

<https://manpages.debian.org/bookworm/debhelper/debhelper.7.en.html>

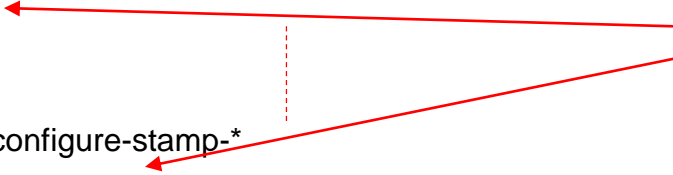


University
of Manitoba

Sample *rules* File (part 2)

```
clean: unpatch
      dh_testdir
      dh_testroot
      rm -rf .libs
      rm -f mod_ruid2.la
      rm -f mod_ruid2.lo
      rm -f mod_ruid2.slo
      rm -f build-stamp-* configure-stamp-*
      rm -rf build-tree
      dh_clean
```

Modify as needed for
your package



```
binary: binary-indep binary-arch
```

```
binary-arch: build install
            dh_testdir
            dh_testroot
            dh_installchangelogs
            dh_installdocs
            dh_installexamples
            dh_installman
            dh_apache2
            dh_link
            dh_strip
            dh_compress
            dh_installdeb
            dh_shlibdeps
            dh_gencontrol
            dh_md5sums
            dh_builddeb
```

<https://www.debian.org/doc/manuals/maint-guide/dreq.en.html>

<https://www.debian.org/doc/debian-policy/ch-source.html#main-building-script-debian-rules>

<https://manpages.debian.org/testing/dpkg-dev/deb-src-rules.5.en.html>

<https://manpages.debian.org/bookworm/debhelper/debhelper.7.en.html>



University
of Manitoba

Building the Package

- **dpkg-buildpackage** [**-b**] [**-rfakeroot**] **-uc -us**
- E.g.: `dpkg-buildpackage -rfakeroot -uc -us 2>&1 | tee ../libapache2-mod-ruid2_0.9.8-3+nmu.log`
- **-b** to build binary package(s) only
- **-rfakeroot** to get correct file uid/gid as non-root (apparently now the default?)
- **-uc** to not sign the **.buildinfo** and **.changes** files
- **-us** to not sign the source package
- Run it from inside the package source tree (top level)

Dealing with Patches

- **dpkg-buildpackage** relies on **quilt** to apply series of patches
- The **series** file lists patches in order to apply
- Unfortunately, ...
 - Each patch must exactly match the source at that stage.
 - No fuzz or even offsets allowed!
 - Makes applying patches to any newer version tricky.

Sample *patches/series* File

mod_ruid2.c

mod_ruid2-none.patch

mod_ruid2-readme.patch

mod_ruid2-license.patch

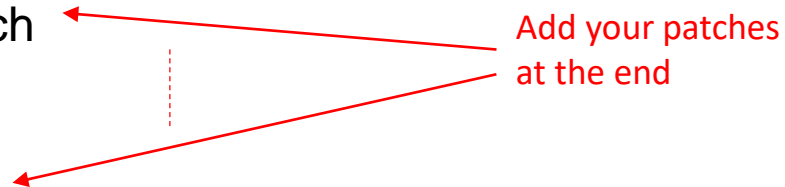
mod_ruid2-userdir.patch

mod_ruid2-readme-userdir.patch

mod_ruid2-nohtaccess.patch

mod_ruid2-nogroups.patch

mod_ruid2-statlog.patch



<https://packaging.ubuntu.com/html/patches-to-packages.html>

<https://www.debian.org/doc/manuals/maint-guide/dother.en.html>

<https://unix.stackexchange.com/questions/344071/applying-patches-in-debian-packages-part-2>



University
of Manitoba

Sample *.patch* File

Index: libapache2-mod-ruuid2-0.9.8/mod_ruid2.c

```
=====
--- libapache2-mod-ruuid2-0.9.8.orig/mod_ruid2.c ← Pathname doesn't
+++ libapache2-mod-ruuid2-0.9.8/mod_ruid2.c ← seem to matter
@@ -584,6 +584,9 @@ static int ruid_set_perm (request_rec *r
    }
}

+   if (dconf->ruid_mode >= RUID_MODE_STAT)
+       ap_log_error(APLOG_MARK, APLOG_NOTICE, 0, NULL, "%s %s %s
+s>%s:setuid(%d), setgid(%d), setgroups(%d,...)", MODULE_NAME,
+       ap_get_server_name(r), r->the_request, from_func, __func__, uid, gid, groupsnr);
+
+       /* set httpd process dumpable after setuid */
+       if (coredump) {
+           prctl(PR_SET_DUMPABLE, 1);
```

<https://packaging.ubuntu.com/html/patches-to-packages.html>

<https://www.debian.org/doc/manuals/maint-guide/dother.en.html>

<https://unix.stackexchange.com/questions/344071/applying-patches-in-debian-packages-part-2>



University
of Manitoba

Adding a New Patch to Quilt

- The “right” way...
 - `quilt push -a`
 - `quilt new mod_ruid2-statlog.patch`
 - `quilt add mod_ruid2.c`
 - `vi mod_ruid2.c`
 - `quilt refresh`
- An easier “hack” for existing, but old patches...
 - `patch -p1 -b < ../mod_ruid2-statlog.patch`
 - `diff -u mod_ruid2.c.orig mod_ruid2.c > debian/patches/mod_ruid2-statlog.patch`
 - `mv mod_ruid2.c.orig mod_ruid2.c`
 - `echo mod_ruid2-readme-userdir.patch >> debian/patches/series`
- See also **quilt push --fuzz** and **--force** options

Push all existing patches

Repeat for each file

Repeat for each patch

<https://raphaelhertzog.com/2011/07/04/how-to-prepare-patches-for-debian-packages/>

<https://wiki.debian.org/UsingQuilt>

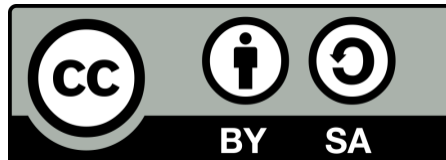
<https://www.debian.org/doc/manuals/maint-guide/modify.en.html>

<https://anarc.at/software/debian-development/#applying-patches>



University
of Manitoba

Questions?



Except where otherwise noted, all content in this presentation is licensed under a Creative Commons “Attribution-ShareAlike 2.5 Canada” License.

http://creativecommons.org/licenses/by-sa/2.5/ca/deed.en_CA

UNIX is a registered trademark of The Open Group