Packaging Linux kernel patches for Debian

Authors:

Yann Dirson, Debian <dirson@debian.org>

This document describes a debhelper add-on mechanism that allows consistent packaging of various kinds of Linux kernel patches.

Revision History

Version	Date	Revision description
1.1	2000-06-10	Initial plain-text revision
1.4	2001-05-14	New "Depends" field
1.8	2002-02-08	New syntax "Kernel-version: all"
	2002-02-10	Moved to DocBook, made it more of a real manual
1.9	2002-02-14	Kernel-version ranges
1.10	2002-02-18	First description of "revision 1" format
1.12	2002-07-16	Integrated feedback from maintainers of kpatch-packages. Default fields. Renamed operations. Miscellaneous updates and additions.
1.13	2003-04-04	New field "Debian-patch-file"

Table of Contents

1 Features

Current features

- robustness of patch application
- support for variable parameters on patch(1) command line (eg. `-p`)
- allows to apply a version of a patch that was originally targetted at another version
- handles (un-)application of patch dependencies
- records in kernel packages build with this patch applied the name and version of the package which shipped the patch.

Features to come with version 1.0

- application of multi-part patches
- support for file copying and archive unpacking as patch operation, in addition to diff application
- support for patch variants for different flavours of a kernel, both to support heavy patches, and to help dealing with otherwise overlaping patches

2 The dh_installkpatches debhelper add-on

dh_installkpatches

Name

`dh_installkpatches` — install kernel patch into package build directories

Synopsis

dh_installkpatches [*`debhelper options`*]

Description

dh_installkpatches is a debhelper script that reads `debian/`*`package`*`.kpatches` or `debian/`*`package`*`.kpatches.`*`something`* files describing one or several revisions of a single kernel patch, and installs them into the package build directory with customized `apply` and `unpatch` scripts.

It also sets the `kpatch:Depends` substitution variable, that you should use in your control file to ensure that generated material in your package get all their dependencies. The use of `kpatch:Depends` requires a build-dependency on version 0.99.3 or later of the dh-kpatches package. Using this variable is now required, since 0.99.16 (mostly because too many people forgot to set it).

Options

This program does not take any particular option in addition to the standard debhelper(1) ones.

See also

debhelper(1), make-kpkg(1).

`/usr/share/doc/dh-kpatches/dh-kpatches.html` or `/usr/share/doc/dh-kpatches/dh-kpatche`

Author

This tool and manual page were written by Yann Dirson <`dirson@debian.org`>.

3 Description of the `kpatches` file format

The syntax of `kpatches` files is inspired by the syntax of Debian control-files. A `kpatches` file contains sections, which are separated by one or more empty lines. Each section is made of `Key: value` lines.

3.1 Revision 0 kpatches files

This is the currently supported format for kpatches files. It is to be superceded shortly with a new revision, more general, described later in this document.

The file starts with a section for generic patch information, and then has one for each patchfile, allowing versions of the patch for many kernel versions to be specified in one file.

3.1.1 The generic part

3.1.1.1 Mandatory fields

Patch-name

a short descriptive name for the patch

NotePatch-name’s cannot contain "pipe" (|) and "double quote" (") characters, for purely implementation reasons. Patch-name’s that contain "slash" (/) characters would not be allowed before dh-kpatches 0.99.3.

Example 1. Examples

```
Patch-name: Kernel debugger
Patch-name: Tracing toolkit
Patch-name: Kernel crash dump
```

Patch-id

a short identifier for the patch

Example 2. Examples

```
Patch-id: kdb
Patch-id: ltt
Patch-id: lkcd
```

3.1.1.2 Optional fields

Path-strip-level

default value for the per-patchfile field of the same name (see below).

Architecture

default value for the per-patchfile field of the same name (see below).

Depends

default value for the per-patchfile field of the same name (see below).

3.1.2 The patchfile-specific parts

3.1.2.1 Mandatory fields

Patch-file

The name of the patch file. It may optionally be compressed with gzip. This should be the first field of each patchfile-specific section.

Kernel-version

The version (or range of versions) of the kernel this patch is designed for. Other optional fields may be introduced in the future to document other versions which are known to work, and versions known not to work.

Version ranges are specified as *`ver1 - ver2`*, where both versions must belong to the same "kernel branch" (eg. 2.2, or 2.4), and must be mentionned in increasing order.

NoteKernel versions used in ranges should show an empty EXTRAVERSION field. This is because no special comparision scheme has been implemented to handle the special meaning associated with EXTRAVERSION values like `pre2` or `test5`.

As a special case, the value "all" is allowed, for patches that don't depend on a specific version (eg. if they just add a set of files).

3.1.2.2 Optional fields

The optional fields, when not provided, all take hopefully reasonable default values

Architecture

a comma-separated list of the debian identifiers for archs supported by this patch.

Default is "all".

Path-strip-level

the value to pass as argument to the patch(1) `-p` option (aka `--strip`).

Default is 1.

Depends

a comma-separated list of Patch-Id's of patches this one depends on.

Default is no dependency.

Debian-patch-file

A special version of Patch-file, specifying that a version of the patch file that only applies to a Debian-patched kernel from a `kernel-source-`*`version`* package, in case the vanilla patch does not fit.

3.1.3 Examples of revision 0 `kpatches` files

3.1.3.1 kdb

These are the 2 `kpatches` files for the kernel-debugger patch. This patch was monolithic for 2.2 kernels, and has been split for recent 2.4 kernels into an architecture-independant part and one architecture-dependant part per supported architecture.

Example 3. The architecture-independant kdb patch

```
Patch-name: Kernel debugger architecture-independant core
Patch-id: kdbcore
Path-strip-level: 1

Patch-file: kdb-v2.1-2.4.17-common-1
Architecture: all
```

```
Kernel-version: 2.4.17
```

Example 4. The architecture-dependant kdb patches

```
Patch-name: Kernel debugger
Patch-id: kdb
Path-strip-level: 1

Patch-file: kdb-v1.5-2.2.18
Architecture: i386
Kernel-version: 2.2.18


Patch-file: kdb-v2.1-2.4.17-i386-1
Architecture: i386
Kernel-version: 2.4.17
Depends: kdbcore

Patch-file: kdb-v2.1-2.4.17-ia64-011226-1
Architecture: ia64
Kernel-version: 2.4.17
Depends: kdbcore
```

3.2 Revision 1 kpatches files

This will be the next supported format for kpatches files. It has not been implemented yet. It is described here for public review.

There are 3 levels of information in this file. A kernel-patch (1st level) is a set of patch alternatives (2nd level), each of which is made of a sequence of patch operations (3rd level).

The file starts with a section for generic patch information, and then, for each alternative, has an alternative-specific section followed by one section for each patch operation.

An alternative section is syntactically distinguished from an operation section by their mandatory constituent fields.

3.2.1 The generic part

3.2.1.1 Mandatory fields.

Kpatch-format-version

the version of the kpatches spec this files complies with. For this version, it should be 1.

Patch-name

a short descriptive name for the patch

NotePatch-name's cannot contain "pipe" (|) and "double quote" (") characters, for purely implementation reasons. Patch-name's that contain "slash" (/) characters would not be allowed before dh-kpatches 0.99.3.

Example 5. Examples

```
Patch-name: Kernel debugger
Patch-name: Tracing toolkit
Patch-name: Kernel crash dump
```

OS-kernel

The identifier of the kernel the patch is meant for. Current possible values include `linux`, `hurd`, `netbsd`, and there is a default value of `linux`.

Patch-id

a short identifier for the patch

Example 6. Examples

```
Patch-id: kdb
Patch-id: ltt
Patch-id: lkcd
```

3.2.1.2 Optional fields

For various fields in the alternative-specific and operation-specific parts it can be useful to provide default values. These are defined using fields named by prepending `Default-` to those fields names.

Examples include `Default-Path-strip-level` and `Default-Architecture`.

3.2.2 The alternative-specific parts

A number of alternative-specific fields are used as *conditions* for this alternative to be considered. This allows not only to specify in the same file versions of the patch for many kernel versions, but also to specify versions of the patch for different *kernel flavours*. For example, the LTT patch comes for 2 kernel flavours: plain kernel, and RTAI-patched kernel.

The order of alternative sections is significant: the 1st alternative for which all conditions are fullfilled is selected, and all further alternatives are ignored.

3.2.2.1 Mandatory fields

Kernel-version

This field is a condition.

The version (or range of versions) of the kernel this patch is designed for. Other optional fields may be introduced in the future to document other versions which are known to work, and versions known not to work.

Version ranges are specified as `ver1 - ver2`, where both versions must belong to the same "kernel branch" (eg. 2.2, or 2.4), and must be mentionned in increasing order.

As a special case, the value "all" is allowed, for patches that don't depend on a specific version (eg. if they just add a set of files).

3.2.2.2 Optional fields

The optional fields, when not provided, all take hopefully reasonable default values

Kernel-flavour

This field is a condition for this alternative to be considered - a test to check whether a patch is applied. It is not to be confused with `Depends`, which will pull others patches first if they are not applied yet.

a comma-separated list of Patch-Id's of patches that need to be applied for this one to be considered for application.

NoteIt may be necessary to add support for flavour versionning (if a patch is disruptive enough to warrant being qualified as a flavour, it is likely that some of its evolution will be disruptive as well).

NoteIt may be useful to add flavour negations, this has to be investigated at some point.

Default is "" (plain/vanilla kernel).

Architecture

This field is a condition.

a comma-separated list of the debian identifiers for archs supported by this patch.

Default is "all".

Depends

This field is NOT a condition.

a comma-separated list of Patch-Id's of patches this one depends on. Patches in this list, as opposed to those in the Kernel-flavour list, are such that this patch cannot be applied without them.

Default is "" (no dependency).

3.2.3 The operation-specific parts

3.2.3.1 Mandatory fields

NoteI have not firmly decided yet how to formalise these parts, and I'm looking for opinions here.
I had originally elected to put the emphasis on *operations*; this is fine with "copy" and "unpack", but "diff" usually refer to the inverse operation (building the data), and "patch", despite being used for this purpose for quite some time now, is already being used here with a broader meaning. Another problem with "unpack" is that it is broad enough so that the data (ie. archive) may potentially be provided in several formats (eg. tar.gz, tar.bz2, zip, etc.) - that will require some (unnecessary) intelligence within the tool, and will impair modularity (ie. adding support for more archive formats using plugin modules).
Another alternative would be to put the emphasis on the *data*. That would mean replacing `Operation` fields with something like `Format` fields, which would have values like "tree", "file", "diff", "tar.gz", "tar", etc. A careful object-oriented implementation would take care of the similarities between those.
Another one was suggested by Baruch Even - see below for an example and some discussion.

Operation

The identifier for a type of patch operation. Valid operations may include "diff", "copy", "unpack", possibly depending on installed plugins. Other fields in the section will depend on the value of this field.

3.2.3.2 Types of operation

diff

The application of a diff file (aka. patch file) to the kernel tree.

copy

The copy of a file, or of a directory tree into the kernel tree.

unpack

The unpacking of a tarball (or other archive ?) into the kernel tree.

3.2.3.3 Per-operation-type fields

3.2.3.3.1 Fields for the diff operation

3.2.3.3.1.1 Mandatory fields

File

The name of the diff file.

3.2.3.3.1.2 Optional fields

Path-strip-level

the value to pass as argument to the patch(1) `-p` option (aka `--strip`).

Default is 1.

3.2.3.3.2 Fields for the unpack operation

3.2.3.3.2.1 Mandatory fields

File

The name of the archive file.

3.2.3.3.2.2 Optional fields

Path-strip-level

Similar to the option of the same name for the `diff` operation: strips the *`N`* first path components of each member the archive.

Default is 0.

For tarballs, this option will probably require pax(1) to be applicable.

3.2.3.3.3 Fields for the copy operation

3.2.3.3.3.1 Mandatory fields

From

The path of the file or directory to copy.

To

The directory in which to copy it.

3.2.4 Examples of revision 1 `kpatches` files

3.2.4.1 ltt

The Linux trace toolkit provides kernel patches both for vanilla kernels and for RTAI kernels. Although there's not yet a packaged version of the RTAI patch, here is what it would look like.

Example 7. The LTT patch

```
Patch-name: Linux Trace Toolkit
OS-kernel: linux
Patch-id: ltt


Kernel-version: 2.4.16 - 2.4.17
Architecture: all

Operation: diff
File: Patches/patch-ltt-linux-2.4.16-vanilla-020415-1.14


Kernel-version: 2.5.7
Architecture: all

Operation: diff
File: Patches/patch-ltt-linux-2.5.7-vanilla-020415-1.14


Kernel-version: 2.4.16
Kernel-flavour: rtai
Architecture: all

Operation: diff
File: Patches/patch-ltt-rtai-24.1.8-020317-1.14
```

3.2.4.2 lkcd

The kernel crash dump patch requires that a script be copied into the `debian` part of the kernel tree. Currently this is done by a creating a new patch (named `kerntypes`) with hand-written `apply` script, not taking advantage of the dh-kpatches mechanism. Here is how it would look like in revision 1:

Example 8. The LKCD patch

```
Patch-name: Linux Kernel Crash Dump
Patch-id: lkcd
Default-Path-strip-level: 1
Default-Architecture: all


Kernel-version: 2.4.17

Operation: copy
From: debian/image.d/kerntypes
To: debian/image.d/

Operation: diff
File: lkcd-4.1-1.patch


Kernel-version: 2.4.18

Operation: copy
From: debian/image.d/kerntypes
To: debian/image.d/

Operation: diff
File: lkcd-4.1-1-2.4.18.patch
```

Baruch Even <`baruch@debian.org`> suggested a syntax that may be simpler. Here is the same example rewritten to use such a syntax, with some interpolations from myself.

Example 9. The LKCD patch (alternate version)

```
Patch-name: Linux Kernel Crash Dump
Patch-id: lkcd
Default-Path-strip-level: 1
Default-Architecture: all


Kernel-version: 2.4.17

cp: debian/image.d/kerntypes debian/image.d/

patch: lkcd-4.1-1.patch


Kernel-version: 2.4.18

cp: debian/image.d/kerntypes debian/image.d/

patch: lkcd-4.1-1-2.4.18.patch
```

The description is more concise, but I fear it would be too losely structured (much more of the information gets stored in left-hand parts of statement), possibly leading to strange-looking stanzas, and possibly impairing extensibility. Here is what the expanded version of the last sample (without default fields) would look like:

Example 10. The LKCD patch (second alternate version)

```
Patch-name: Linux Kernel Crash Dump
Patch-id: lkcd


Kernel-version: 2.4.17
Architecture: all

cp: debian/image.d/kerntypes debian/image.d/

patch: lkcd-4.1-1.patch
Path-strip-level: 1


Kernel-version: 2.4.18
Architecture: all

cp: debian/image.d/kerntypes debian/image.d/

patch: lkcd-4.1-1-2.4.18.patch
Path-strip-level: 1
```

In the same spirit, but datatype-driven instead of operation-driven. This one lies somewhere in between Baruch's proposal and my original one.

Example 11. The LKCD patch (third alternate version)

```
Patch-name: Linux Kernel Crash Dump
Patch-id: lkcd
Default-Path-strip-level: 1
Default-Architecture: all


Kernel-version: 2.4.17

File: debian/image.d/kerntypes
Target-dir: debian/image.d/

Diff: lkcd-4.1-1.patch


Kernel-version: 2.4.18

File: debian/image.d/kerntypes
Target-dir: debian/image.d/
```

```
Diff: lkcd-4.1-1-2.4.18.patch
```