




Working with patches

OpenWrt Buildroot integrates *quilt* [[http://en.wikipedia.org/wiki/Quilt_\(software\)](http://en.wikipedia.org/wiki/Quilt_(software))] for easy patch management. This document outlines some common patching tasks like adding a new patch or editing existing ones.

Prepare quilt configuration

In order to let *quilt* create patches in OpenWrt's preferred format, a configuration file `.quiltrc` containing common *diff* and *patch* options must be created in the local home directory.

```
cat > ~/.quiltrc <<EOF
QUILT_DIFF_ARGS="--no-timestamps --no-index -p ab --color=auto"
QUILT_REFRESH_ARGS="--no-timestamps --no-index -p ab"
QUILT_PATCH_OPTS="--unified"
QUILT_DIFF_OPTS="-p"
EDITOR="nano"
EOF
```

- EDITOR specifies the preferred editor for interactive patch editing
- The other variables control the patch format property like a/, b/ directory names and no timestamps
- FreeBSD does not support the `--color=auto` option and `-pab` must be written as `-p ab`
-  `"-p ab"` is working with quilt 0.63 on Linux and is documented in man page.

Adding a new patch

To add a completely new patch to an existing package *example* start with preparing the source directory:

```
make package/example/{clean,prepare} V=s QUILT=1
```

For host-side packages, you may want to detail the make target:

```
make package/example/host/{clean,prepare} V=s QUILT=1
```

This unpacks the source tarball and prepares existing patches as *quilt patch series* (if any). The verbose output will show where the source got extracted.

Change to the prepared source directory.

```
cd build_dir/target-*/example-*
```

Note : It can happen that you need to go one level lower as the source is extracted in `build_dir/target-*/BUILD_VARIANT/example-*`. This happens when multiple build variants of a package are defined in the Makefile.

Apply all existing patches using *quilt push*.

```
quilt push -a
```

Create a new, empty patch file with the *quilt new* command:

```
quilt new 010-main_code_fix.patch
```

- The name should start with a number, followed by a hyphen and a very short description of what is changed
- The chosen number should be higher than any existing patch - use *quilt series* to see the list of patches
- The patch file name should be short but descriptive

After creating the empty patch, files to edit must be associated with it. The *quilt add* command can be used for that - once the file got added it can be edited as usual.

A shortcut for both adding a file and open it in an editor is the *quilt edit* command:

```
quilt edit src/main.c
```

- `src/main.c` gets added to `010-main_code_fix.patch`
- The file is opened in the editor specified with `EDITOR` in `.quilttrc`

Repeat that for any file that needs to be edited.

After the changes are finished, they can be reviewed with the *quilt diff* command.

```
quilt diff
```

If the diff looks okay, proceed with *quilt refresh* to update the `010-main_code_fix.patch` file with the changes made.

```
quilt refresh
```

Change back to the toplevel directory of the buildroot.

```
cd ../../../../
```

To move the new patch file over to the buildroot, run *update* on the package:

```
make package/example/update V=s
```

Finally rebuild the package to test the changes:

```
make package/example/{clean,compile} package/index V=s
```

If problems occur, the patch needs to be edited again to solve the issues. Refer to the section below to learn how to edit existing patches.

Edit an existing patch

Start with preparing the source directory:

```
make package/example/{clean,prepare} V=s QUILT=1
```

Change to the prepared source directory.

```
cd build_dir/target-*/example-*
```

List the patches available:

```
quilt series
```

Advance to the patch that needs to be edited:

```
quilt push 010-main_code_fix.patch
```

- When passing a valid patch filename to *push*, *quilt* will only apply the series until it reaches the specified patch
- If unsure, use *quilt series* to see existing patches and *quilt top* to see the current position
- If the current position is beyond the desired patch, use *quilt pop* to remove patches in the reverse order

Edit the patched files using the *quilt edit* command, repeat for every file that needs changes.

```
quilt edit src/main.c
```

Check which files are to be included in the patch:

```
quilt files
```

Review the changes with *quilt diff*.

```
quilt diff
```

If the diff looks okay, proceed with *quilt refresh* to update the current patch with the changes made.

```
quilt refresh
```

Change back to the toplevel directory of the buildroot.

```
cd ../../../../
```

To move the updated patch file over to the buildroot, run *update* on the package:


```
make package/example/update V=s
```

Finally rebuild the package to test the changes:

```
make package/example/{clean,compile} package/index V=s
```

Adding or editing kernel patches

The process for modifying kernel patches is the same as for packages, only the make targets and directories differ.

 For the kernel, an additional subdirectory for patches is used, `generic/` contains patches common to all architectures and `platform/` contains patches specific to the current target.

To prepare the kernel tree, use:

```
make target/linux/{clean,prepare} V=s QUILT=1
```

For Attitude Adjustment, the source tree is in the *linux-architecture* subdirectory:

```
cd build_dir/linux-*/linux-3.*
```

For Barrier Breaker (trunk), the source tree is in the *target-architecture* subdirectory (potentially with a subarch):

```
cd build_dir/target-*/linux-*/linux-3.*
```

Moving the changes back over to the buildroot tree from the build tree is done with (you need to go back to trunk to do this):

```
make target/linux/update package/index V=s
```

( Patches should be named with the correct prefix, *platform/000-abc.patch* or *generic/000-abc.patch*. If not the update may not work correctly.)

Afterwards, if we want to verify whether our patch is applied or not, we can go to the top level directory with

```
cd ../../../../
```

and preparing again the *linux* folder for some modification with

```
make target/linux/{clean,prepare} V=s QUILT=1
```

During this process all the applied patches will be shown, ours being among them, preceded by *generic/* or *platform/* depending on what directory we placed the patch. Another way of retrieving the applied patches is through

```
quilt series
```

as explained on the previous sections, after having made *make target/linux/{clean,prepare} ...*

Adding or editing toolchain patches

For example, gcc:

To prepare the tool tree, use:

```
make toolchain/gcc/{clean,prepare} V=99 QUILT=1
```

The source tree depends on chosen lib and gcc :

```
cd build_dir/toolchain-mips_r2_gcc-4.3.3+cs_uClibc-0.9.30.1/gcc-4.3.3
```

Refreshing the patches is done with:

```
make toolchain/gcc/update V=99
```

Refreshing patches

When a patched package (or kernel) is updated to a newer version, existing patches might not apply cleanly anymore and *patch* will report *fuzz* when applying them. To rebase the whole patch series the *refresh* make target can be used:

```
make package/example/refresh V=s
```

For kernels, use:

```
make target/linux/refresh V=s
```

Iteratively modify patches without cleaning the source tree

When implementing new changes, it is often required to edit patches multiple times. To speed up the process, it is possible to retain the prepared source tree between edit operations.

1. Initially prepare the source tree as documented above
2. Change to the prepared source directory
3. Advance to the patch needing changes
4. Edit the files and refresh the patch
5. Fully apply the remaining patches using *quilt push -a* (if any)
6. From the toplevel directory, run `make package/example/{compile,install}` or `make target/linux/{compile,install}` for kernels
7. Test the binary
8. If further changes are needed, repeat from step 2.
9. Finally run `make package/example/update` or `make target/linux/update` for kernels to copy the changes back to buildroot

Further information

- [Official quilt man page \[http://linux.die.net/man/1/quilt\]](http://linux.die.net/man/1/quilt)
- [How To Survive With Many Patches - Introduction to Quilt \[http://git.savannah.gnu.org/cgi/quilt.git/plain/doc/quilt.pdf\]](http://git.savannah.gnu.org/cgi/quilt.git/plain/doc/quilt.pdf) (PDF) (read online here <https://docs.google.com/viewer?url=http%3A%2F%2Fgit.savannah.gnu.org%2Fgit%2Fquilt.git%2Fplain%2Fdoc%2Fquilt.pdf>)
- [Applying patches newbie doubt \[https://forum.openwrt.org/viewtopic.php?id=43039\]](https://forum.openwrt.org/viewtopic.php?id=43039)

doc/devel/patches.txt · Last modified: 2015/07/10 19:20 by kevindb