

Submitting patches

1. Separate your changes.

Each patch should contain only one fix or only one new feature.

Each patch should be send separately or put in separate folders.

2. Describe your changes when checking-in commits.

Create meaningful description of changes.

Describe your changes in imperative mood, e.g. “make xyzzy do frotz” instead of

“[This patch] makes xyzzy do frotz” or “[I] changed xyzzy to do frotz”, as if you are giving orders to the codebase to change its behaviour.

3. Confirm that all dependencies are committed.

Each patch should be created after all dependent repositories and sub-repositories are submitted and project compilation produce file with GIT hash number.

4. If dependent repository was change, create patch for dependent repository too according to guidelines in this document.

Usually, you can check if dependent repository was change by looking into change of .config file:

If you see that SHA1 hash number of some repository changed, then it's a sign that patch for this repository should be created too.

5. Create patch by running following command:

```
git format-patch original_commit^..your_last_commit -k --stdout > foo.patch
```

for example:

```
git format-patch
```

```
a001a6b392eddba2b43f37f8b35461f50178e397^..e8194f5af11622a487da0c5005be2ff3d82a4742 -k --stdout > foo.patch
```

6. When sending a patch, add hash number of original commit. It will help the reviewer to roll back (if needed) and apply patch to original commit.

For example, if sending patch from previous example add to email or readme file following line:

original commit: a001a6b392eddba2b43f37f8b35461f50178e397