Applying Patches

In a perfect world, applying the patches is as simply as doing:

```
$ git am 0002-This-is-the-second-commit.patch
```

assuming you are in a branch that has not yet had the changes incorporated in the patches. This command is very strong; it not only applies the patches to the working copies in your project directories, it also does a commit.

It is more than possible that one or more of the patches will fail due to conflicting lines of development between the patch sender and receiver. Such problems will have to be resolved one by one. For example, you may get a message like:

```
error: patch failed: file2:1
error: file2: patch does not apply
Patch failed at 0001.
When you have resolved this problem do: git am --resolved.
If you would prefer to skip this patch, instead do: git am --skip.
```

You can also back off the patches and restore the original branch with:

```
$ git am --abort
```

You may prefer to do things by hand more cautiously. You can apply individual patches directly to your working copy. For example, you can try:

```
$ patch --dry-run < 0002-This-is-the-second-commit.patch
```

If no problems are encountered, run again without the **--dry-run** option. Then, you would have to run **git add** on the affected files and eventually do a **git commit**.

There is another, lower-level command, **git apply**, which is the basis of **git am**. If you do:

```
$ git apply --check 0002-This-is-the-second-commit.patch
```

it functions like **patch --dry-run**; it will not actually do the patch. If there are no problems, you can run again without the **--check** option. This patches the working files in your directories and updates the Index. You will still have to invoke **git add** and **git commit** eventually.

Note that **git apply** does not modify the index; you can also use **--cached** to apply the changes only to the index.

Changes induced by **git** patch applying commands:

Command	Source Files	Index	Commit Chain	References
git am	Modified by patch	Updated to reflect patch	New commit object created and added to top of commit chain	**HEAD**; points to new commit object
git apply	Modified by patch (unless **--check** option specified)	Unchanged (unless **--index** option specified)	Unchanged	Unchanged