Git: Merging

SaltStack Seth House <shouse@saltstack.com>

2014-07-17

Not fast-forward merges

- Moves the branch pointer.
- No changes to the DAG.

Regular commit object; two parents

- Plain ol' commit object.
- Two parents.
- Easy to revert an entire branch merge.

Commit message

- For humans!
- Contains default text.
- Add, edit, or augment.

Merge conflicts

- Stages all successful automatic merges
- Surrounds conflicts with conflict markers

Demo: merge conflicts

- Make a branch
- Make an edit
- Make a new branch
- Make a conflicting edit
- Merge the other branch
- Resolve the conflict
- Stage the change
- Commit (use/modify the default message)

mergetools

Three-way merge.

- LOCAL
- BASE
- REMOTE
- MERGED

Demo: mergetools

- View the previous demo with a mergetool.

Where we've been

Cherry picking has served us well.

Where we've been

Cherry picking has served us well.

- Community-centric.
- Near-zero barrier to commit.

Where we've been

Cherry picking has served us well.

- Community-centric.
- Near-zero barrier to commit.

Cherry picking doesn't scale to multiple LTS branches.

- Book-keeping nightmare.
- Manual back-port to *each* LTS branch.

Where we're going

- Don't back-port; merge forward!

Where we're going

- Don't back-port; merge forward!
- (Except for community fixes to LTS branches.)

Where we're going

- Don't back-port; merge forward!
- (Except for community fixes to LTS branches.)
- Git traverses the history when merging.
- SHAs do not change; commits are grouped; merges are preserved.
- Git `log` and `diff` can *intelligently* compare branches.

Requirements

- Feature addition or bug?

Requirements

- Feature addition or bug?
- One "story" per pull request.

Feature additions

- Branch off `develop`:

```
git fetch upstream
git checkout -b myfeature upstream/develop
git pull --rebase
```

- Open pull request against `develop`.

Bug fixes

1. Identify the origin. Reproduce the bug on each LTS branch until the original source is found.

Bug fixes

1. Identify the origin. Reproduce the bug on each LTS branch until the original source is found.
2. Branch off the oldest affected LTS branch.

Bug fixes

1. Identify the origin. Reproduce the bug on each LTS branch until the original source is found.
2. Branch off the oldest affected LTS branch.
3. Open pull request containing the fix against that branch.

Bug fixes

1. Identify the origin. Reproduce the bug on each LTS branch until the original source is found.
2. Branch off the oldest affected LTS branch.
3. Open pull request containing the fix against that branch.
4. Merge into oldest affected LTS branch.

Bug fixes

1. Identify the origin. Reproduce the bug on each LTS branch until the original source is found.
2. Branch off the oldest affected LTS branch.
3. Open pull request containing the fix against that branch.
4. Merge into oldest affected LTS branch.
5. Merge forward.

Bug fixes

1. Identify the origin. Reproduce the bug on each LTS branch until the original source is found.
2. Branch off the oldest affected LTS branch.
3. Open pull request containing the fix against that branch.
4. Merge into oldest affected LTS branch.
5. Merge forward.
   - Merge the LTS containing the fix into the next-oldest LTS branch.

Bug fixes

1. Identify the origin. Reproduce the bug on each LTS branch until the original source is found.
2. Branch off the oldest affected LTS branch.
3. Open pull request containing the fix against that branch.
4. Merge into oldest affected LTS branch.
5. Merge forward.
   - Merge the LTS containing the fix into the next-oldest LTS branch.
   - E.g., `2014.7` is merged into `2014.11` is merged into `2015.4`.

Bug fixes

1. Identify the origin. Reproduce the bug on each LTS branch until the original source is found.
2. Branch off the oldest affected LTS branch.
3. Open pull request containing the fix against that branch.
4. Merge into oldest affected LTS branch.
5. Merge forward.
   - Merge the LTS containing the fix into the next-oldest LTS branch.
   - E.g., `2014.7` is merged into `2014.11` is merged into `2015.4`.
   - Finally merge into `develop`.

Community-sent pull requests

- One "story" per pull request.
- Features opened against `develop`.
- Bug fixes opened against current release branch.

Community-sent pull requests

- One "story" per pull request.
- Features opened against `develop`.
- Bug fixes opened against current release branch.
  - Help identify that branch.
  - Ask pull requests be split into single "stories".
  - Ask to re-open a pull request against another branch.

Back-porting community-sent fixes

We must still back-port fixes to LTS branches.

Back-porting community-sent fixes

We must still back-port fixes to LTS branches.

1. Merge pull request against current release branch.

Back-porting community-sent fixes

We must still back-port fixes to LTS branches.

1. Merge pull request against current release branch.
2. Rebase the entire pull request onto oldest affected LTS:

```
git checkout -b bp-1234 upstream/pr/1234
git log --topo-order upstream/develop...HEAD
git rebase --onto 2014.7 <orig-base> bp-1234
```

Back-porting community-sent fixes

We must still back-port fixes to LTS branches.

1. Merge pull request against current release branch.
2. Rebase the entire pull request onto oldest affected LTS:

```
git checkout -b bp-1234 upstream/pr/1234
git log --topo-order upstream/develop...HEAD
git rebase --onto 2014.7 <orig-base> bp-1234
```

3. Merge rebased branch into oldest affected LTS:

```
git checkout 2014.7
git merge -e --no-ff bp-1234
```

Back-porting community-sent fixes

We must still back-port fixes to LTS branches.

1. Merge pull request against current release branch.
2. Rebase the entire pull request onto oldest affected LTS:

```
git checkout -b bp-1234 upstream/pr/1234
git log --topo-order upstream/develop...HEAD
git rebase --onto 2014.7 <orig-base> bp-1234
```

3. Merge rebased branch into oldest affected LTS:

```
git checkout 2014.7
git merge -e --no-ff bp-1234
```

4. Annotate the commit message with relevant details. Pull request number, original commit SHAs, etc.

Back-porting community-sent fixes

We must still back-port fixes to LTS branches.

1. Merge pull request against current release branch.
2. Rebase the entire pull request onto oldest affected LTS:

```
git checkout -b bp-1234 upstream/pr/1234
git log --topo-order upstream/develop...HEAD
git rebase --onto 2014.7 <orig-base> bp-1234
```

3. Merge rebased branch into oldest affected LTS:

```
git checkout 2014.7
git merge -e --no-ff bp-1234
```

4. Annotate the commit message with relevant details. Pull request number, original commit SHAs, etc.
5. Open pull request against oldest affected LTS.
6. Merge PR. Merge forward; Git will ignore the duplicate content.