



Custom merge drivers in git

Michal Koutný <mkoutny@suse.com>
LinuxDays 2026



Outline

- Merge conflict
- Level 0: Manual resolution
- Level 1: Mergetool
- Level 2: Merge driver
- Driver challenges
- Non-standard merges

Playground

```
$ cat cities
Bangkok
Beijing
Cairo
Delhi
Dhaka
Guangzhou
Ho Chi Minh City
Istanbul
Jakarta
Lahore
Manila
Mexico City
Mumbai
São Paulo
Shanghai
Tokyo
```

Merge conflict

```
$ git merge topic-k2
Auto-merging cities
CONFLICT (content): Merge conflict in cities
Automatic merge failed; fix conflicts and then commit the
result.
```

```
$ git status

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   cities
```

Merge conflict

```
--- a/cities
+++ b/cities
@@ -7,6 +7,7 @@ Guangzhou
    Ho Chi Minh City
    Istanbul
    Jakarta
+Kolkata
    Lahore
    Manila
    Mexico City
```

```
--- a/cities
+++ b/cities
@@ -7,6 +7,7 @@ Guangzhou
    Ho Chi Minh City
    Istanbul
    Jakarta
+Karachi
    Lahore
    Manila
    Mexico City
```

Merge conflict

```
*    0f58a92 (merge-kk) Merge branch 'topic-k2' into merge-kk
|\
| * 8dd0a33 (topic-k2) Add Karachi
* | d02c1d0 (topic-k1) Add Kolkata
|/
* c1f02b1 (main) Initial commit
```

Not a merge conflict

```
* 57c7b59 (merge-bs) Merge branch 'topic-b' into merge-bs
|\
| * 6c2a600 (topic-b) Add Buenos Aires
* | 3a1dcff (topic-s) Add Seoul
|/
* c1f02b1 (main) Initial commit
```

```
$ git merge topic-b
Auto-merging cities
Merge made by the 'ort' strategy.
 cities | 1 +
 1 file changed, 1 insertion(+)
```

Not a merge conflict

```
--- a/cities
+++ b/cities
@@ -7,6 +7,7 @@ Guangzhou
   Ho Chi Minh City
   Istanbul
   Jakarta
+Kolkata
   Lahore
   Manila
   Mexico City
```

```
--- a/cities
+++ b/cities
@@ -8,6 +8,7 @@ Ho Chi Minh
   Istanbul
   Jakarta
   Lahore
+Lima
   Manila
   Mexico City
   Mumbai
```

Merge conflict

```
*    be1edb9 (merge-k11) Merge branch 'topic-1' into merge-k11
|\
| * a9b6841 (topic-1) Add Lima
* | a4a70ee (topic-k3) Update Lahore
* | d02c1d0 (topic-k1) Add Kolkata
|/
* c1f02b1 (main) Initial commit
```

```
$ git merge topic-1
Auto-merging cities
CONFLICT (content): Merge conflict in cities
Automatic merge failed; fix conflicts and then commit the
result.
```

Merge conflict

```
@@ -7,7 +7,8 @@ Guangzhou
   Ho Chi Minh City
   Istanbul
   Jakarta
  -Lahore
  +Kolkata
  +Lāhanūr
   Manila
   Mexico City
   Mumbai
```

```
@@ -8,6 +8,7 @@ Ho Chi Minh
City
   Istanbul
   Jakarta
   Lahore
  +Lima
   Manila
   Mexico City
   Mumbai
```



Resolutions



Level 0: Manual resolution

```
Ho Chi Minh City  
Istanbul  
Jakarta  
<<<<<< HEAD  
Kolkata  
=====  
Karachi  
>>>>>> topic-k2  
Lahore  
Manila  
Mexico City
```

Level 0: Manual resolution

```
$ git merge topic-k2
```

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: cities

```
$ vi cities && ...
```

```
$ git add cities
```

Changes to be committed:

modified: cities

```
$ git commit
```

Level 0: Manual resolution

```
Merge branch 'topic-k2' into merge-kk
```

```
# Conflicts:  
#       cities
```



Level 1: Mergetool

```
$ git merge topic-k2
```

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: cities

```
$ git mergetool --tool=cities-mergetool [cities]
```

Merging:
cities

Normal merge conflict for 'cities':
{local}: modified file
{remote}: modified file

Changes to be committed:

modified: cities

```
$ git commit
```

Level 1: Mergetool

```
# .git/config

[mergetool "cities-mergetool"]
    cmd = ../2026-custom-merge.tools/mergetool \
$LOCAL $BASE $REMOTE $MERGED
    trustExitCode = true
    #keepBackup = true
```

Level 1: Mergetool

```
import argparse
import locale
locale.setlocale(locale.LC_COLLATE, "en_US.UTF-8")

...
# $LOCAL $BASE $REMOTE $MERGED
args = parser.parse_args()
result_path = args.merged

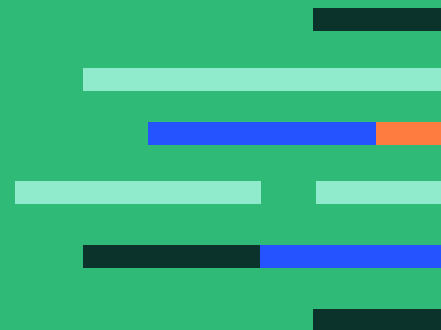
local, base, remote = [
    set(open(s_path).readlines())
    for s_path in (args.local, args.base, args.remote,)
]

added = remote - base
removed = base - remote
merged = (local | added) - removed

with open(result_path, "w") as f:
    f.writelines(sorted(merged, key=locale.strxfrm))
```



gitattributes(5)



gitattributes(5)

- Choice
 - \$GIT_COMMON_DIR/info/attributes
 - .gitattributes
- `GIT_COMMON_DIR=$(git rev-parse --git-common-dir)`

```
# attributes
cities      merge=cities-driver
<pattern>  merge=union
<pattern>  diff=...
```

- `git check-attr -a -- <file>`
- gotcha: `GIT_DIR=$(git rev-parse --git-dir)`

Level 2: Merge driver

```
# .git/config

[merge "cities-driver"]
    driver = ../2026-custom-merge.tools/mergetool \
--driver %A %O %B %P

[mergetool "cities-mergetool"]
    cmd = ../2026-custom-merge.tools/mergetool \
$LOCAL $BASE $REMOTE $MERGED
    trustExitCode = true
    #keepBackup = true
```

Level 2: Merge driver

```
@@ -5,13 +5,14 @@ import locale
    locale.setlocale(locale.LC_COLLATE, "en_US.UTF-8")

    parser = argparse.ArgumentParser(description="Merge sorted file")
+parser.add_argument("--driver", action="store_true", help="Behave like merge driver")
    parser.add_argument("local")
    parser.add_argument("base")
    parser.add_argument("remote")
    parser.add_argument("merged")

    args = parser.parse_args()
-    result_path = args.merged
+result_path = args.local if args.driver else args.merged

    local, base, remote = [
        set(open(s_path).readlines())
```

Level 2: Merge driver

```
$ git merge topic-k2
```

Auto-merging cities

Merge made by the 'ort' strategy.

cities | 1 +

1 file changed, 1 insertion(+)

Level 2: Merge driver

Merge branch 'topic-k2' into merge-kk

```
# Please enter a commit message to explain why this merge is necessary,  
# especially if it merges an updated upstream into a topic branch.  
#  
# Lines starting with '#' will be ignored, and an empty message aborts  
# the commit.
```



Driver challenges



Logging

- Avoid evil merges
- `prepare-commit-msg` hook
- `$GIT_DIR/MERGE_DRIVER_LOG`

Configuration

- `$GIT_DIR/info/attributes` vs `.gitattributes`
- Distribution of `.git/config`
 - `include.path`
- Merge driver tool in-repo
- setup and un-setup script

Availability of branch data

- Mergetool != merge driver
- Environment variable(s) `GITHEAD_*`
- Conflict resolution labels

```
<<<<<< HEAD
Kolkata
=====
Karachi
>>>>>> topic-k2
```

```
# .git/config
```

```
[merge "cities-driver"]
    driver = ../2026-custom-merge.tools/mergetool \
--driver --labels %X %S %Y %A %O %B %P
```

Performance

- Driver is called always, even for non-conflicts
- rcsmerge vs git-merge-file

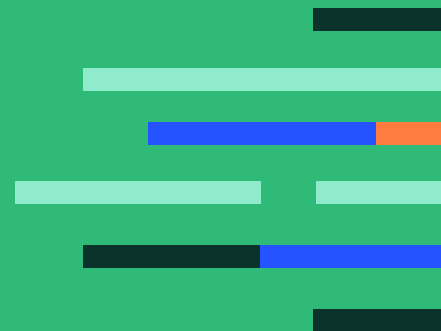
```
+def merge(local, base, remote, labels=None):
+    cmd = ["git", "merge-file"]
+    if labels:
+        for label in labels:
+            cmd += [ "-L", label ]
+    return subprocess.call(cmd + [local, base, remote]) == 0
+
+ parser.add_argument("--driver", action="store_true", help="Behave like merge driver")
+
+...
+# fast path: try regular 3-way merge
+with NamedTemporaryFile() as tmp_local:
+    shutil.copy(args.local, tmp_local.name)
+    if merge(tmp_local.name, args.base, args.remote, args.labels):
+        shutil.copy(tmp_local.name, result_path)
+    sys.exit(0)
```

Miscellaneous gotchas

- Fallback to human resolution
- Collateral changes
- Quirks of pygit2
- Cleanup of temporary files



Non-standard merges



Cherry-picks (and rebases)

```
* 8c94dda (cp-k11) Add Lima
* a4a70ee (topic-k3) Update Lahore      # local
* d02c1d0 (topic-k1) Add Kolkata
| * a9b6841 (topic-l) Add Lima          # remote
|/
* c1f02b1 (main) Initial commit        # base
```

```
$ git cherry-pick main..topic-l
Auto-merging cities
[cp-k11 8c94dda] Add Lima
Date: Wed Sep 30 14:40:09 2026 +0200
1 file changed, 1 insertion(+)
```

Merge strategies

- `git merge -s<strategy>`
- Strategies
 - ort
 - ours
 - `-s ours`
 - `-s ort -Xours`
 - octopus
 - `ref^N`
 - subtree



Summary

- What are routine conflicts
- Where to put tools (in-repo, 3rd place)
- Where to put config (in-repo, clone-local)
- Start with mergetool, driver later

References & further reading

- <https://github.com/Werkov/2026-custom-merge-tools> (setup demo)
- `git-merge(1)`
- `git-mergetool(1)`
- `gitattributes(5)`
- Benjamin Poirier (made the first [mergetool in our repo](#))
- <https://octodex.github.com/original/>
- <https://github.com/libgit2/pygit2/pull/1491>
- <https://lore.kernel.org/git/20260929051200.GA1100000@coredump.intra.peff.net/>
- <https://jvns.ca/blog/2023/11/10/how-cherry-pick-and-revert-work/>
- <https://softwareswirl.blogspot.com/2013/05/git-imerge-practical-introduction.html>



Thank you & QA

