

NAME

git-merge-file - Run a three-way file merge

SYNOPSIS

```
git merge-file [-L <current-name> [-L <base-name> [-L <other-name>]]]
[--ours|--theirs|--union] [-p|--stdout] [-q|--quiet] [--marker-size=<n>]
[--no-diff3] <current-file> <base-file> <other-file>
```

DESCRIPTION

git merge-file incorporates all changes that lead from the <base-file> to <other-file> into <current-file>. The result ordinarily goes into <current-file>. *git merge-file* is useful for combining separate changes to an original. Suppose <base-file> is the original, and both <current-file> and <other-file> are modifications of <base-file>, then *git merge-file* combines both changes.

A conflict occurs if both <current-file> and <other-file> have changes in a common segment of lines. If a conflict is found, *git merge-file* normally outputs a warning and brackets the conflict with lines containing <<<<<<< and >>>>>>> markers. A typical conflict will look like this:

```
<<<<<<< A
lines in file A
=====
lines in file B
>>>>>>> B
```

If there are conflicts, the user should edit the result and delete one of the alternatives. When --ours, --theirs, or --union option is in effect, however, these conflicts are resolved favouring lines from <current-file>, lines from <other-file>, or lines from both respectively. The length of the conflict markers can be given with the --marker-size option.

The exit value of this program is negative on error, and the number of conflicts otherwise. If the merge was clean, the exit value is 0.

git merge-file is designed to be a minimal clone of RCS *merge*; that is, it implements all of RCS *merges* functionality which is needed by [git\(1\)](#).

OPTIONS

-L <label>

This option may be given up to three times, and specifies labels to be used in place of the corresponding file names in conflict reports. That is, `git merge-file -L x -L y -L z a b c` generates output that looks like it came from files x, y and z instead of from files a, b and c.

-p

Send results to standard output instead of overwriting <current-file>.

-q

Quiet; do not warn about conflicts.

--diff3

Show conflicts in diff3 style.

--ours, --theirs, --union

Instead of leaving conflicts in the file, resolve conflicts favouring our (or their or both) side of the lines.

EXAMPLES

```
git merge-file README.my README README.upstream
  combines the changes of README.my and README.upstream since README, tries to
  merge them and writes the result into README.my.
```

```
git merge-file -L a -L b -L c tmp/a123 tmp/b234 tmp/c345
  merges tmp/a123 and tmp/c345 with the base tmp/b234, but uses labels a and c instead of
  tmp/a123 and tmp/c345.
```

GIT

Part of the [git\(1\)](#) suite