



--local-branching-on-the-cheap

Search entire site...

- [About](#)
 - [Branching and Merging](#)
 - [Small and Fast](#)
 - [Distributed](#)
 - [Data Assurance](#)
 - [Staging Area](#)
 - [Free and Open Source](#)
 - [Trademark](#)
- [Documentation](#)
 - [Reference](#)
 - [Book](#)
 - [Videos](#)
 - [External Links](#)
- [Downloads](#)
 - [GUI Clients](#)
 - [Logos](#)
- [Community](#)

[Français](#) ▼

Localized versions of **git-mergetool** manual

1. [English](#)
2. [Français](#)
3. [Português \(Brasil\)](#)

Want to read in your language or fix typos?

[You can help translate this page.](#)

[Topics](#) ▼

Setup and Config

- [git](#)
- [config](#)
- [help](#)
- [bugreport](#)

Getting and Creating Projects

- [init](#)
- [clone](#)

Basic Snapshotting

- [add](#)
- [status](#)
- [diff](#)
- [commit](#)
- [notes](#)

- [restore](#)
- [reset](#)
- [rm](#)
- [mv](#)

Branching and Merging

- [branch](#)
- [checkout](#)
- [switch](#)
- [merge](#)
- [mergetool](#)
- [log](#)
- [stash](#)
- [tag](#)
- [worktree](#)

Sharing and Updating Projects

- [fetch](#)
- [pull](#)
- [push](#)
- [remote](#)
- [submodule](#)

Inspection and Comparison

- [show](#)
- [log](#)
- [diff](#)
- [difftool](#)
- [range-diff](#)
- [shortlog](#)
- [describe](#)

Patching

- [apply](#)
- [cherry-pick](#)
- [diff](#)
- [rebase](#)
- [revert](#)

Debugging

- [bisect](#)
- [blame](#)
- [grep](#)

Email

- [am](#)

- [apply](#)
- [format-patch](#)
- [send-email](#)
- [request-pull](#)

External Systems

- [svn](#)
- [fast-import](#)

Server Admin

- [daemon](#)
- [update-server-info](#)

Guides

- [gitattributes](#)
- [Command-line interface conventions](#)
- [Everyday Git](#)
- [Frequently Asked Questions \(FAQ\)](#)
- [Glossary](#)
- [Hooks](#)
- [gitignore](#)
- [gitmodules](#)
- [Revisions](#)
- [Submodules](#)
- [Tutorial](#)
- [Workflows](#)
- [All guides...](#)

Administration

- [clean](#)
- [gc](#)
- [fck](#)
- [reflog](#)
- [filter-branch](#)
- [instaweb](#)
- [archive](#)
- [bundle](#)






Plumbing Commands

- [cat-file](#)
- [check-ignore](#)
- [checkout-index](#)
- [commit-tree](#)
- [count-objects](#)
- [diff-index](#)
- [for-each-ref](#)
- [hash-object](#)

- [ls-files](#)
- [ls-tree](#)
- [merge-base](#)
- [read-tree](#)
- [rev-list](#)
- [rev-parse](#)
- [show-ref](#)
- [symbolic-ref](#)
- [update-index](#)
- [update-ref](#)
- [verify-pack](#)
- [write-tree](#)

[Latest version](#) ▼ git-mergetool last updated in 2.33.0

Changes in the **git-mergetool** manual

1. 2.31.1 → 2.33.0 no changes
2. [2.31.0](#)  [03/15/21](#)
3. 2.22.1 → 2.30.2 no changes
4. [2.22.0](#)  [06/07/19](#)
5. 2.20.1 → 2.21.4 no changes
6. [2.20.0](#)  [12/09/18](#)
7. 2.12.5 → 2.19.6 no changes
8. [2.11.4](#)  [09/22/17](#)
9. 2.2.3 → 2.10.5 no changes
10. [2.1.4](#)  [12/17/14](#)
- 11.

Check your version of git by running

```
git --version
```

NOM

git-mergetool - Lancer les outils de résolution de conflit de fusion pour résoudre les conflits de fusion

SYNOPSIS

```
git mergetool [--tool=<outil>] [-y | --[no-]prompt] [<fichier>...]
```

DESCRIPTION

Utilisez `git mergetool` pour lancer l'un des nombreux utilitaires de fusion afin de résoudre les conflits de fusion. Il est généralement lancé après `git merge`.

Si un ou plusieurs paramètres `<fichier>` sont donnés, le programme outil de fusion sera exécuté pour résoudre les différences sur chaque fichier (en sautant ceux qui n'ont pas de conflit). La spécification d'un répertoire inclura tous les fichiers non résolus dans ce chemin. Si aucun `<fichier>` nom n'est spécifié, `git mergetool` lancera le programme de fusion sur chaque fichier ayant des conflits de fusion.

OPTIONS

-t<outil>
--tool=<outil>

Utiliser le programme de résolution de fusion spécifié par <outil>. Les valeurs valides comprennent `emerge`, `gvimdiff`, `kdifff3`, `meld`, `vimdiff` et `tortoisemerge`. Lancez `git mergetool --tool-help` pour obtenir la liste des paramètres <outil> valides.

Si un programme de résolution de fusion n'est pas spécifié, *git mergetool* utilisera la variable de configuration `merge.tool`. Si la variable de configuration `merge.tool` n'est pas définie, *git mergetool* choisira une valeur par défaut appropriée.

Vous pouvez explicitement fournir un chemin complet vers l'outil en définissant la variable de configuration `mergetool.<outil>.path`. Par exemple, vous pouvez configurer le chemin absolu vers `kdifff3` en définissant `mergetool.kdifff3.path`. Sinon, *git mergetool* suppose que l'outil est disponible dans `PATH`.

Au lieu d'exécuter un des programmes connus de l'outil de fusion, *git mergetool* peut être personnalisé pour exécuter un programme alternatif en spécifiant la ligne de commande à invoquer dans une variable de configuration `mergetool.<outil>.cmd`.

Lorsque *git mergetool* est invoqué avec cet outil (soit par l'option `-t` ou `--tool` ou par la variable de configuration `merge.tool`) la ligne de commande configurée sera invoquée avec `$BASE` défini au nom d'un fichier temporaire contenant la base commune pour la fusion, si disponible ; `$LOCAL` défini au nom d'un fichier temporaire contenant le contenu du fichier sur la branche courante ; `$REMOTE` défini au nom d'un fichier temporaire contenant le contenu du fichier à fusionner, et `$MERGED` défini au nom du fichier dans lequel l'outil de fusion doit écrire le résultat de la résolution de la fusion.

Si l'outil de fusion personnalisé indique correctement le succès d'une résolution de fusion dans son code de sortie, alors la variable de configuration `mergetool.<outil>.trustExitCode` peut être définie à `true`. Sinon, *git mergetool* demandera à l'utilisateur d'indiquer le succès de la résolution après que l'outil personnalisé soit sorti.

--tool-help

Afficher une liste des outils de fusion utilisables avec `--tool`.

-y
--no-prompt

Ne pas demander avant chaque invocation du programme de résolution de fusion. C'est la valeur par défaut si le programme de résolution de fusion est explicitement spécifié avec l'option `--tool` ou avec la variable de configuration `merge.tool`.

--prompt

Demander avant chaque invocation du programme de résolution de fusion pour donner à l'utilisateur une chance de sauter le chemin.

-g
--gui

Lorsque *git-mergetool* est invoqué avec l'option `-g` ou `--gui`, l'outil de fusion par défaut sera lu à partir de la variable `merge.guitool` configurée au lieu de `merge.tool`. Si `merge.guitool` n'est pas défini, nous reviendrons à l'outil configuré sous `merge.tool`.

--no-gui

Ceci prend le pas sur un réglage précédent de `-g` ou `--gui` et lit que l'outil de fusion par défaut à partir de la variable `merge.tool` configurée.

`-O<fichier-d-ordre>`

Traiter les fichiers dans l'ordre spécifié dans le `<fichier d'ordre>`, qui a un motif glob shell par ligne. Ceci remplace la variable de configuration `diff.orderFile` (voir [git-config\[1\]](#)). Pour annuler `diff.orderFile`, utilisez `-O/dev/null`.

FICHIERS TEMPORAIRES

`git mergetool` crée des fichiers de sauvegarde `*.orig` lors de la résolution des fusions. Il est possible de les supprimer en toute sécurité une fois qu'un fichier a été fusionné et que sa session `git mergetool` est terminée.

Définir la variable de configuration `mergetool.keepBackup` sur `false` fait que `git mergetool` supprime automatiquement la sauvegarde lorsque les fichiers sont correctement fusionnés.

GIT

Fait partie de la suite [git\[1\]](#)

TRADUCTION

Cette page de manuel a été traduite par Jean-Noël Avila `<jn.avila AT free DOT fr>` et les membres du projet `git-manpages-l10n`. Veuillez signaler toute erreur de traduction par un rapport de bogue sur le site <https://github.com/jnavila/git-manpages-l10n>.

[About this site](#)

Patches, suggestions, and comments are welcome.

Git is a member of [Software Freedom Conservancy](#).