NAME

git-mergetool − Run merge conflict resolution tools to resolve merge conflicts

SYNOPSIS

git mergetool [−−tool=<tool>] [−y | −−[no−]prompt] [<file>...]

DESCRIPTION

Use **git mergetool** to run one of several merge utilities to resolve merge conflicts. It is typically run after *git merge*.

If one or more <file> parameters are given, the merge tool program will be run to resolve differences on each file (skipping those without conflicts). Specifying a directory will include all unresolved files in that path. If no <file> names are specified, *git mergetool* will run the merge tool program on every file with merge conflicts.

OPTIONS

−t <tool>, −−tool=<tool>

Use the merge resolution program specified by <tool>. Valid values include emerge, gvimdiff, kdiff3, meld, vimdiff, and tortoisemerge. Run **git mergetool −−tool−help** for the list of valid <tool> settings.

If a merge resolution program is not specified, *git mergetool* will use the configuration variable **merge.tool**. If the configuration variable **merge.tool** is not set, *git mergetool* will pick a suitable default.

You can explicitly provide a full path to the tool by setting the configuration variable **mergetool.<tool>.path**. For example, you can configure the absolute path to kdiff3 by setting **mergetool.kdiff3.path**. Otherwise, *git mergetool* assumes the tool is available in PATH.

Instead of running one of the known merge tool programs, *git mergetool* can be customized to run an alternative program by specifying the command line to invoke in a configuration variable **mergetool.<tool>.cmd**.

When *git mergetool* is invoked with this tool (either through the **−t** or **−−tool** option or the **merge.tool** configuration variable) the configured command line will be invoked with **$BASE** set to the name of a temporary file containing the common base for the merge, if available; **$LOCAL** set to the name of a temporary file containing the contents of the file on the current branch; **$REMOTE** set to the name of a temporary file containing the contents of the file to be merged, and **$MERGED** set to the name of the file to which the merge tool should write the result of the merge resolution.

If the custom merge tool correctly indicates the success of a merge resolution with its exit code, then the configuration variable **mergetool.<tool>.trustExitCode** can be set to **true**. Otherwise, *git mergetool* will prompt the user to indicate the success of the resolution after the custom tool has exited.

−−tool−help

Print a list of merge tools that may be used with **−−tool**.

−y, −−no−prompt

Don’t prompt before each invocation of the merge resolution program. This is the default if the merge resolution program is explicitly specified with the **−−tool** option or with the **merge.tool** configuration variable.

−−prompt

Prompt before each invocation of the merge resolution program to give the user a chance to skip the path.

−g, −−gui

When *git−mergetool* is invoked with the **−g** or **−−gui** option the default merge tool will be read from

the configured **merge.guitool** variable instead of **merge.tool**. If **merge.guitool** is not set, we will fallback to the tool configured under **merge.tool**.

−−no−gui
: This overrides a previous **−g** or **−−gui** setting and reads the default merge tool will be read from the configured **merge.tool** variable.

−O<orderfile>
: Process files in the order specified in the <orderfile>, which has one shell glob pattern per line. This overrides the **diff.orderFile** configuration variable (see **git-config**(1)). To cancel **diff.orderFile**, use **−O/dev/null**.

CONFIGURATION

mergetool.<tool>.path
: Override the path for the given tool. This is useful in case your tool is not in the PATH.

mergetool.<tool>.cmd
: Specify the command to invoke the specified merge tool. The specified command is evaluated in shell with the following variables available: *BASE* is the name of a temporary file containing the common base of the files to be merged, if available; *LOCAL* is the name of a temporary file containing the contents of the file on the current branch; *REMOTE* is the name of a temporary file containing the contents of the file from the branch being merged; *MERGED* contains the name of the file to which the merge tool should write the results of a successful merge.

mergetool.<tool>.hideResolved
: Allows the user to override the global **mergetool.hideResolved** value for a specific tool. See **mergetool.hideResolved** for the full description.

mergetool.<tool>.trustExitCode
: For a custom merge command, specify whether the exit code of the merge command can be used to determine whether the merge was successful. If this is not set to true then the merge target file timestamp is checked and the merge assumed to have been successful if the file has been updated, otherwise the user is prompted to indicate the success of the merge.

mergetool.meld.hasOutput
: Older versions of **meld** do not support the **−−output** option. Git will attempt to detect whether **meld** supports **−−output** by inspecting the output of **meld −−help**. Configuring **mergetool.meld.hasOutput** will make Git skip these checks and use the configured value instead. Setting **mergetool.meld.hasOutput** to **true** tells Git to unconditionally use the **−−output** option, and **false** avoids using **−−output**.

mergetool.meld.useAutoMerge
: When the **−−auto−merge** is given, meld will merge all non−conflicting parts automatically, highlight the conflicting parts and wait for user decision. Setting **mergetool.meld.useAutoMerge** to **true** tells Git to unconditionally use the **−−auto−merge** option with **meld**. Setting this value to **auto** makes git detect whether **−−auto−merge** is supported and will only use **−−auto−merge** when available. A value of **false** avoids using **−−auto−merge** altogether, and is the default value.

mergetool.hideResolved
: During a merge Git will automatically resolve as many conflicts as possible and write the *MERGED* file containing conflict markers around any conflicts that it cannot resolve; *LOCAL* and *REMOTE* normally represent the versions of the file from before Git's conflict resolution. This flag causes *LOCAL* and *REMOTE* to be overwriten so that only the unresolved conflicts are presented to the merge tool. Can be configured per−tool via the **mergetool.<tool>.hideResolved** configuration variable. Defaults to **false**.

mergetool.keepBackup
: After performing a merge, the original file with conflict markers can be saved as a file with a **.orig** extension. If this variable is set to **false** then this file is not preserved. Defaults to **true** (i.e. keep the backup files).

mergetool.keepTemporaries
: When invoking a custom merge tool, Git uses a set of temporary files to pass to the tool. If the tool returns an error and this variable is set to **true**, then these temporary files will be preserved, otherwise they will be removed after the tool has exited. Defaults to **false**.

mergetool.writeToTemp
: Git writes temporary *BASE*, *LOCAL*, and *REMOTE* versions of conflicting files in the worktree by default. Git will attempt to use a temporary directory for these files when set **true**. Defaults to **false**.

mergetool.prompt
: Prompt before each invocation of the merge resolution program.

TEMPORARY FILES

git mergetool creates ***.orig** backup files while resolving merges. These are safe to remove once a file has been merged and its **git mergetool** session has completed.

Setting the **mergetool.keepBackup** configuration variable to **false** causes **git mergetool** to automatically remove the backup as files are successfully merged.

GIT

Part of the **git**(1) suite