git-gui(1) Manual Page

NAME

git-gui - A portable graphical interface to Git

SYNOPSIS

```
git gui [<command>] [arguments]
```

DESCRIPTION

A Tcl/Tk based graphical user interface to Git. *git gui* focuses on allowing users to make changes to their repository by making new commits, amending existing ones, creating branches, performing local merges, and fetching/pushing to remote repositories.

Unlike *gitk*, *git gui* focuses on commit generation and single file annotation and does not show project history. It does however supply menu actions to start a *gitk* session from within *git gui*.

git gui is known to work on all popular UNIX systems, Mac OS X, and Windows (under both Cygwin and MSYS). To the extent possible OS specific user interface guidelines are followed, making *git gui* a fairly native interface for users.

COMMANDS

blame
: Start a blame viewer on the specified file on the given version (or working directory if not specified).

browser
: Start a tree browser showing all files in the specified commit (or *HEAD* by default). Files selected through the browser are opened in the blame viewer.

citool
: Start *git gui* and arrange to make exactly one commit before exiting and returning to the shell. The interface is limited to only commit actions, slightly reducing the application's startup time and simplifying the menubar.

version
: Display the currently running version of *git gui*.

Examples

`git gui blame Makefile`
: Show the contents of the file *Makefile* in the current working directory, and provide annotations for both the original author of each line, and who moved the line to its current location. The uncommitted file is annotated, and uncommitted changes (if any) are explicitly attributed to *Not Yet Committed*.

`git gui blame v0.99.8 Makefile`
: Show the contents of *Makefile* in revision *v0.99.8* and provide annotations for each line. Unlike the above example the file is read from the object database and not the working directory.

`git gui blame --line=100 Makefile`
: Loads annotations as described above and automatically scrolls the view to center on line *100*.

`git gui citool`
: Make one commit and return to the shell when it is complete. This command returns a non-zero exit code if the window was closed in any way other than by making a commit.

`git gui citool --amend`
: Automatically enter the *Amend Last Commit* mode of the interface.

`git gui citool --nocommit`
: Behave as normal citool, but instead of making a commit simply terminate with a zero exit code. It still checks that the index does not contain any unmerged entries, so you can use it as a GUI version of git-mergetool(1)

`git citool`
: Same as `git gui citool` (above).

`git gui browser maint`
: Show a browser for the tree of the *maint* branch. Files selected in the browser can be viewed with the internal blame viewer.

SEE ALSO

gitk(1)
: The git repository browser. Shows branches, commit history and file differences. gitk is the utility started by *git gui*'s Repository Visualize actions.

Other

git gui is actually maintained as an independent project, but stable versions are distributed as part of the Git suite for the convenience of end users.

A *git gui* development repository can be obtained from:

```
git clone git://repo.or.cz/git-gui.git
```

or

```
git clone http://repo.or.cz/r/git-gui.git
```

or browsed online at http://repo.or.cz/w/git-gui.git/.

GIT

Part of the git(1) suite

Last updated 2013-02-15 15:43:56 UTC