What Is Git Bash and How Do You Use It?

This tool aimed at Windows users comes with the power of a Unix shell and built-in support for Git.

BY KADEISHA KEAN PUBLISHED JUN 24, 2022

Every Windows PC comes with Command Prompt (cmd), a command-line textual interface to your operating system. Using Command Prompt, you can install programs, create new users, and run scripts.

The closest equivalent to Command Prompt for Unix-like machines is Bash. Bash is often seen as superior because of its powerful features. These include arithmetic, array variables, loops, and branches.

Git Bash provides Windows users with access to Bash and its advanced features.

What Is Git?

Link copied to clipboard

Git is a version control system. It tracks the changes you make to a collection of files using commits. Commits allow you to capture the state of a project at a particular point in time. And Git allows you to go back to a previous commit whenever you want.

You can install and run Git locally or use one of its online hosts (such as GitHub or Bitbucket). But Git is fundamentally a Unix-style command-line utility program. It even comes installed on some macOS and Linux systems.

Git is probably the most popular version control system in the world today. This is due to its collaborative features. Branching allows you to create independent local versions of a codebase that you can later merge with others. This is one of the main reasons why programmers use Git to share their contributions to source code development.

Git is also open source, free to use, and easy to learn.

What Is Bash?

The acronym Bash stands for Bourne Again Shell. The name is a pun on the Bourne shell which it replaced. Bash has all the Bourne shell core features such as grammar and variable expansion. What makes it “born again” are several additional features, including:

- Multi-character invocation options
- Command-line editing
- Timestamped command-line history
- One-dimensional built-in array variables
- For loop expressions
- Job control
- Aliases

What Is Git Bash?

Link copied to clipboard

Although Git is a collection of command-line utility programs, you can use it on Windows via a GUI-based app. Bash is only available on Unix-like operating systems, like macOS and Linux.

Git Bash is strictly for Microsoft Windows users. It provides an emulation of both Git and the Bash command-line environment. Installing Git Bash on your Windows machine gives you access to a shell environment that is native to macOS and Linux users.

Popular Git Bash Commands

Before you start using Git Bash, you should note that the Bash commands that you can use in this command-line interface are case-sensitive. This is in contrast with the Windows Command Prompt, many parts of which are case-insensitive.

cd Command

The acronym **cd** stands for “change directory”. It lets a Git Bash user navigate from one directory to another. All you need to provide is the path to the destination directory:

```
cd [directory_path]
```

If you do not provide a directory path after the **cd** command, it will take you to your home directory. You can also easily navigate up one level, to the parent directory of the one you’re currently in:

```
cd ..
```

mkdir Command

mkdir stands for make directory and, as the name suggests, it allows you to create a new directory. All you need to provide is the desired directory name:

```
mkdir [directory_name]
```

This will create a new subdirectory of the directory you are currently in.

Link copied to clipboard

rmdir Command

The **rmdir** (remove directory) command removes empty directories. The **rmdir** command can delete one or more directories at a time, the only requirement is that they should be empty. If you want to delete many separate empty directories, the order can be important. Make sure you delete child directories before any of their parents, or the parent directories will not be empty when rmdir tries to remove them:

```
rmdir [main_directory/sub_directory] [main_directory]
```

rm Command

The **rm** (remove) command allows you to delete specific files from directories. All it requires is the path to the file you want to delete:

```
rm [directory_path/file]
```

It also allows you to delete populated directories using specific options. To do so, use one of the following options:

- -r
- -R
- --recursive

All these options are equivalent and they will delete files recursively. This means that rm will delete any given directories, and all the files underneath them, including any subdirectories.

mv Command

The **mv** command allows you to move a file or folder to any directory. It takes two arguments: a file name (or directory name) and a directory path.

```
Link copied to clipboard tory_path]
```

You can also use mv to rename a file since that operation is a special case of moving it:

```
mv file1.txt file2.txt
```

ls Command

The **ls** command allows you to list all the files and folders in a directory:

```
ls [directory_name]
```

echo Command

You can use the **echo** command to print a string to the Git Bash terminal:

```
echo ["random string"]
```

cat Command

The **cat** (concatenate) command uses several options and has three main functions.

It allows you to create and append to a file:

```
cat [>file.txt]
```

After you execute the command above Git Bash will remain in the newly created file and allow you to append to it. If you want to exit the file, press **Ctrl + C**.

cat also allows you to view the contents of a file:

Link copied to clipboard

You should always include the extension of the file you want to read from, or else Git Bash will not locate the file.

Finally, cat allows you to append one file to the end of another:

```
cat [file1.txt >> file2.txt]
```

In this example, Git Bash appends the content of **file1.txt** to the end of **file2.txt**. It is important to include the extension of both the files that you want to write to and from. If you do not include the extension for the first file, Git Bash will not locate it. If you forget the extension for the second file, Git Bash will create a new file.

Git Bash Perks

Git Bash allows a Windows user to experience shell scripting at its finest. However, advanced Bash features are not the only things you gain access to when you install Git Bash. You also gain access to Git and all its features.

RELATED TOPICS

PROGRAMMING GIT LINUX BASH SHELL COMMAND PROMPT

ABOUT THE AUTHOR

Kadeisha Kean

(90 Articles Published)

Link copied to clipboard

Kadeisha is a Full-Stack Software Developer and Technical/Technology Writer. She has a Bachelor of Science in Computing, from the University of Technology in Jamaica.

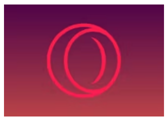
ARTIFICIAL INTELLIGENCE IPHONE 15

What Is Forefront AI and Is It Better Than ChatGPT?

2 days ago

What Is Zero Shot Learning and How Can It Improve AI?

2 days ago



How to Enable and Use Aria AI in Opera GX

2 days ago

See More