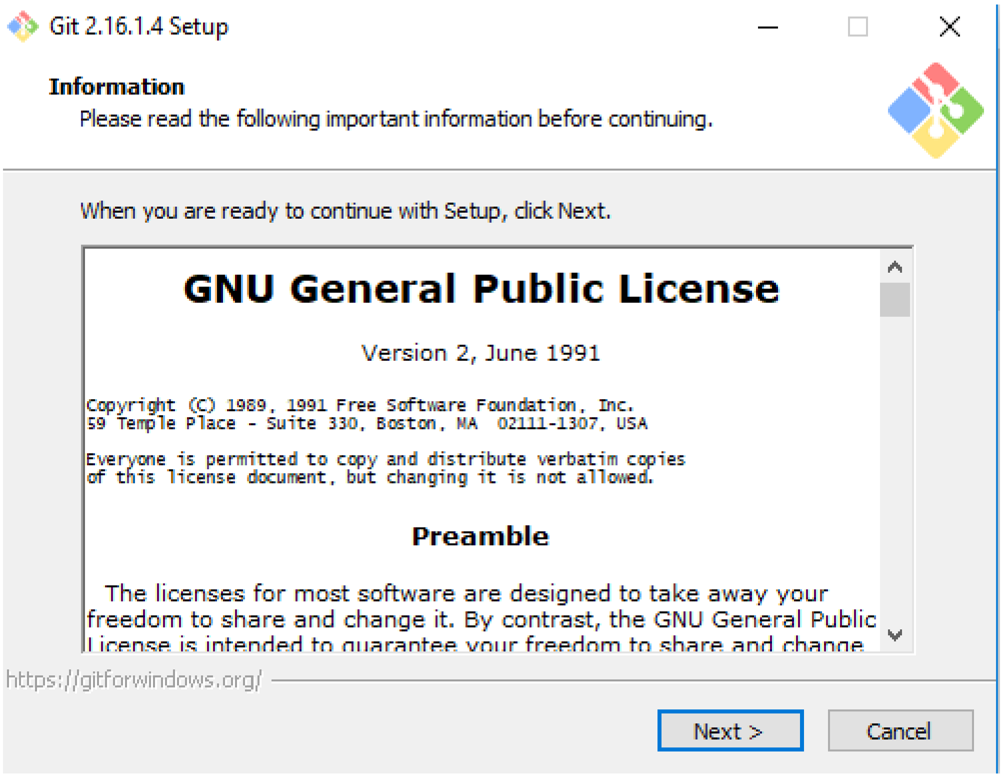
INSTALLER GIT

- Télécharger la dernière version de Git pour Windows : https://gitforwindows.org



Git 2.16.1.4 Setup

Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

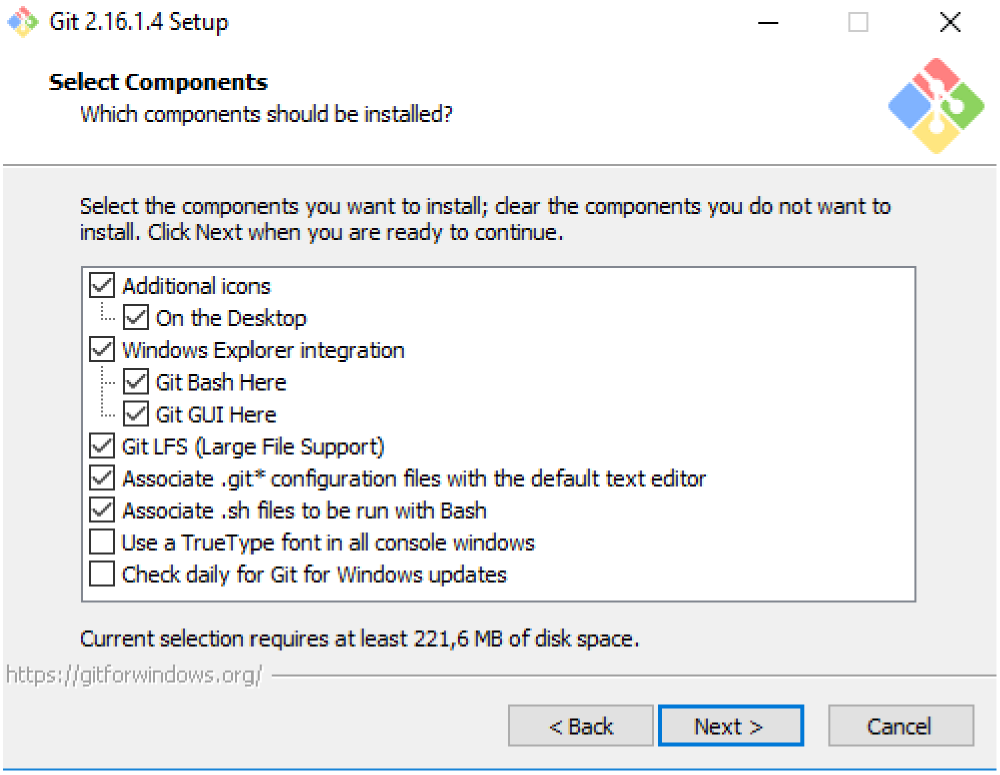
```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

https://gitforwindows.org/



Git 2.16.1.4 Setup

Select Components

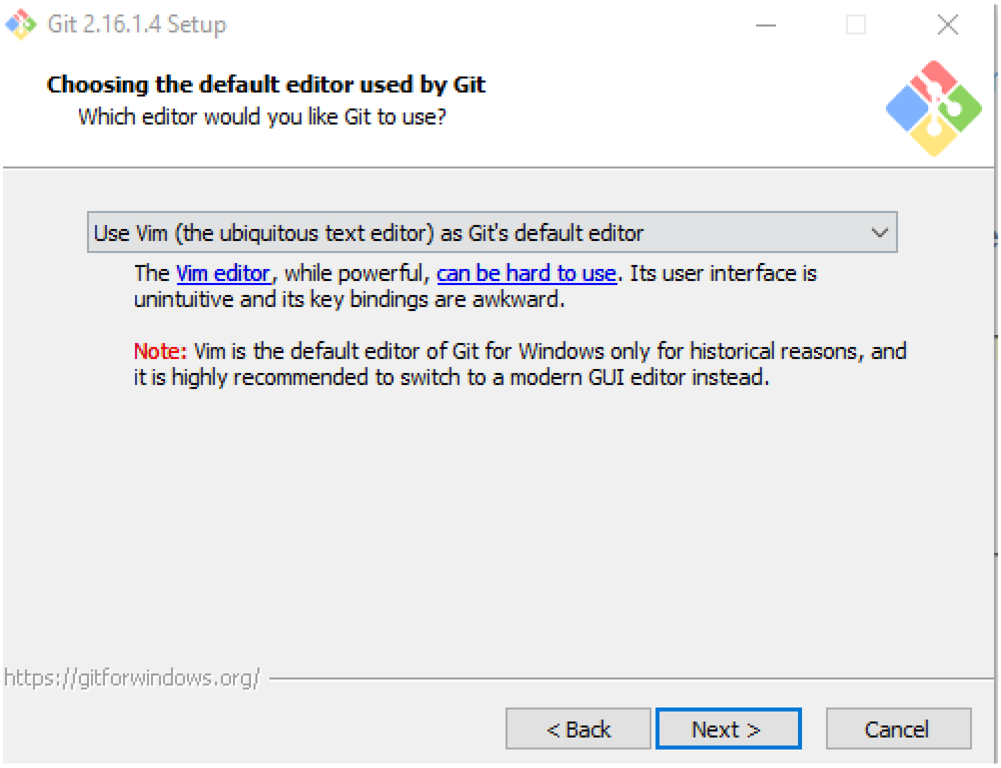
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

☑ Additional icons
  ☑ On the Desktop
☑ Windows Explorer integration
  ☑ Git Bash Here
  ☑ Git GUI Here
☑ Git LFS (Large File Support)
☑ Associate .git* configuration files with the default text editor
☑ Associate .sh files to be run with Bash
☐ Use a TrueType font in all console windows
☐ Check daily for Git for Windows updates

Current selection requires at least 221,6 MB of disk space.

https://gitforwindows.org/



Git 2.16.1.4 Setup

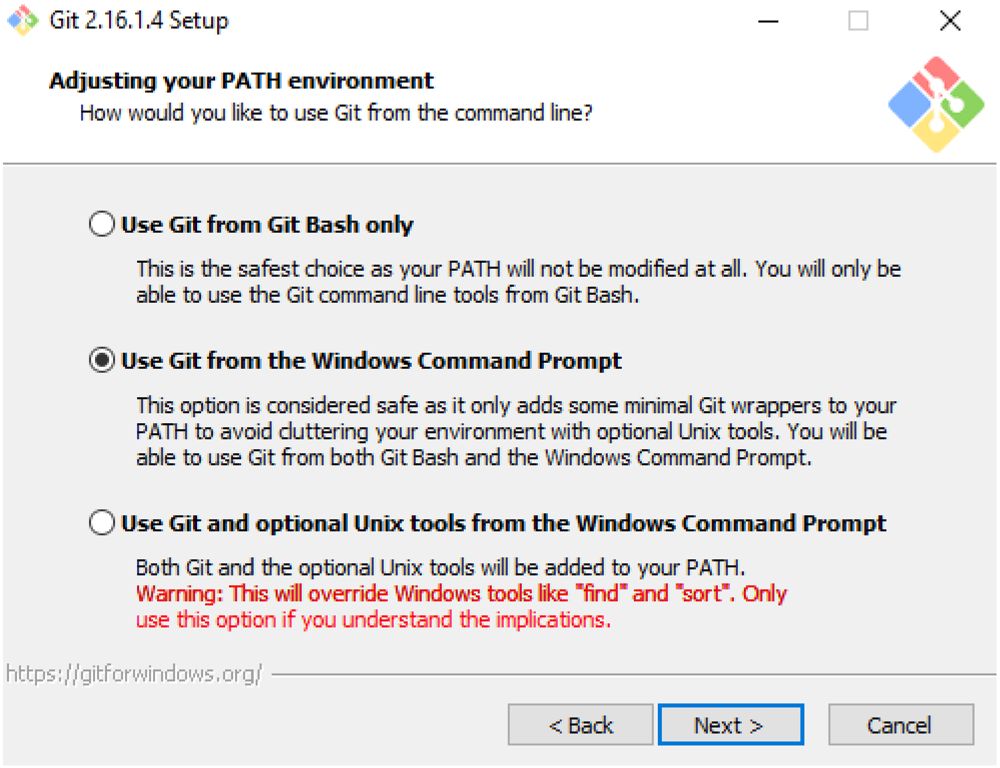
Choosing the default editor used by Git

Which editor would you like Git to use?

Use Vim (the ubiquitous text editor) as Git's default editor

The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.



Git 2.16.1.4 Setup

Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

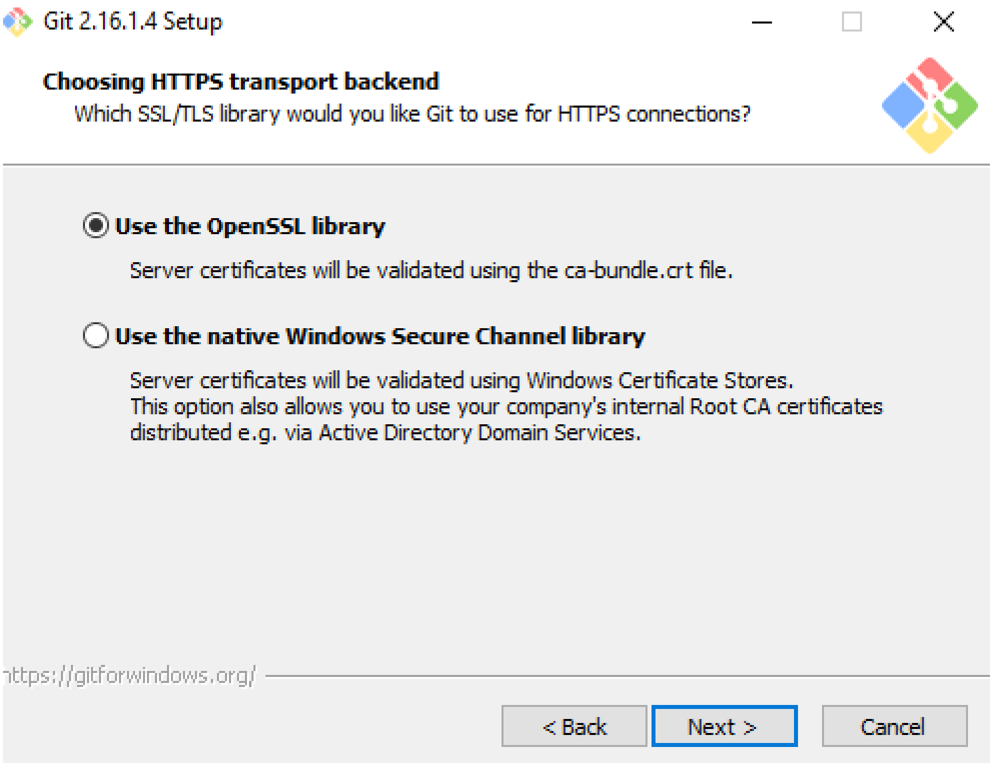
This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

● **Use Git from the Windows Command Prompt**

This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.

○ **Use Git and optional Unix tools from the Windows Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.



Choosing HTTPS transport backend

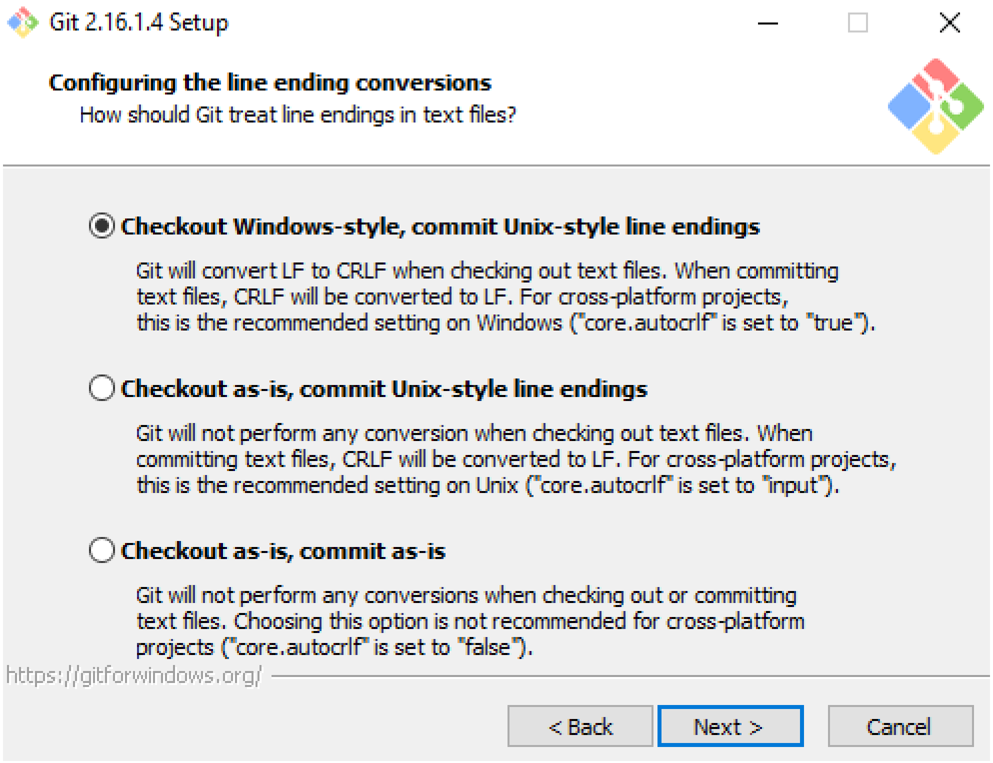
Which SSL/TLS library would you like Git to use for HTTPS connections?

● **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates
distributed e.g. via Active Directory Domain Services.



Configuring the line ending conversions

How should Git treat line endings in text files?

● **Checkout Windows-style, commit Unix-style line endings**

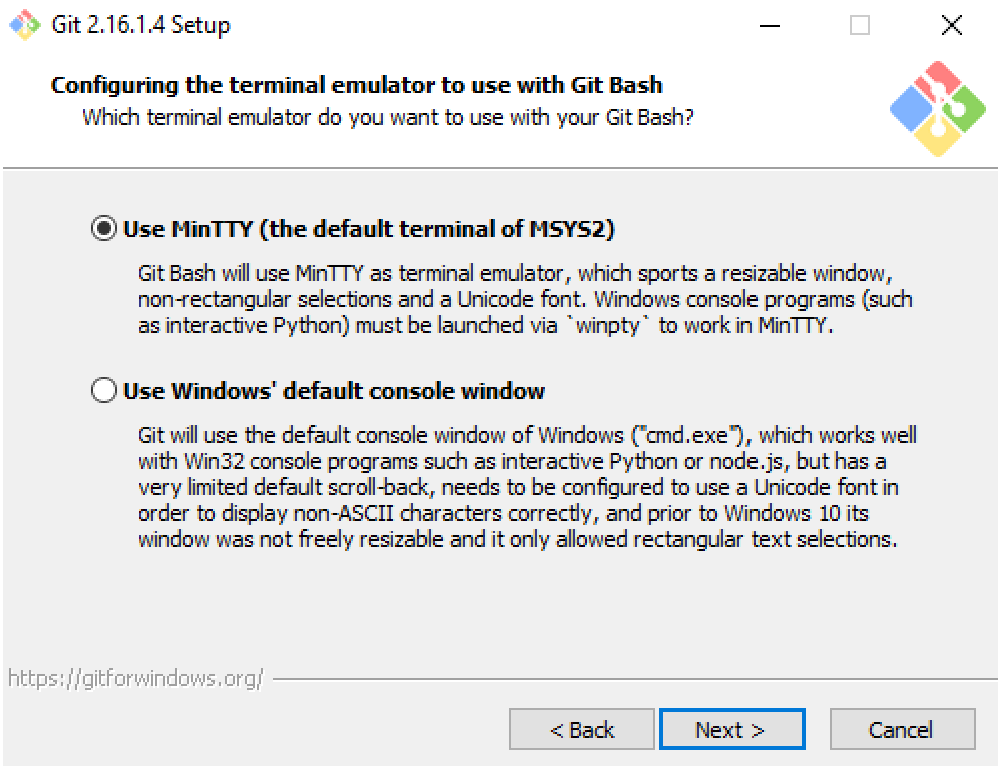
Git will convert LF to CRLF when checking out text files. When committing
text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Windows ("core.autocrlf" is set to "true").

○ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When
committing text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing
text files. Choosing this option is not recommended for cross-platform
projects ("core.autocrlf" is set to "false").



Configuring the terminal emulator to use with Git Bash

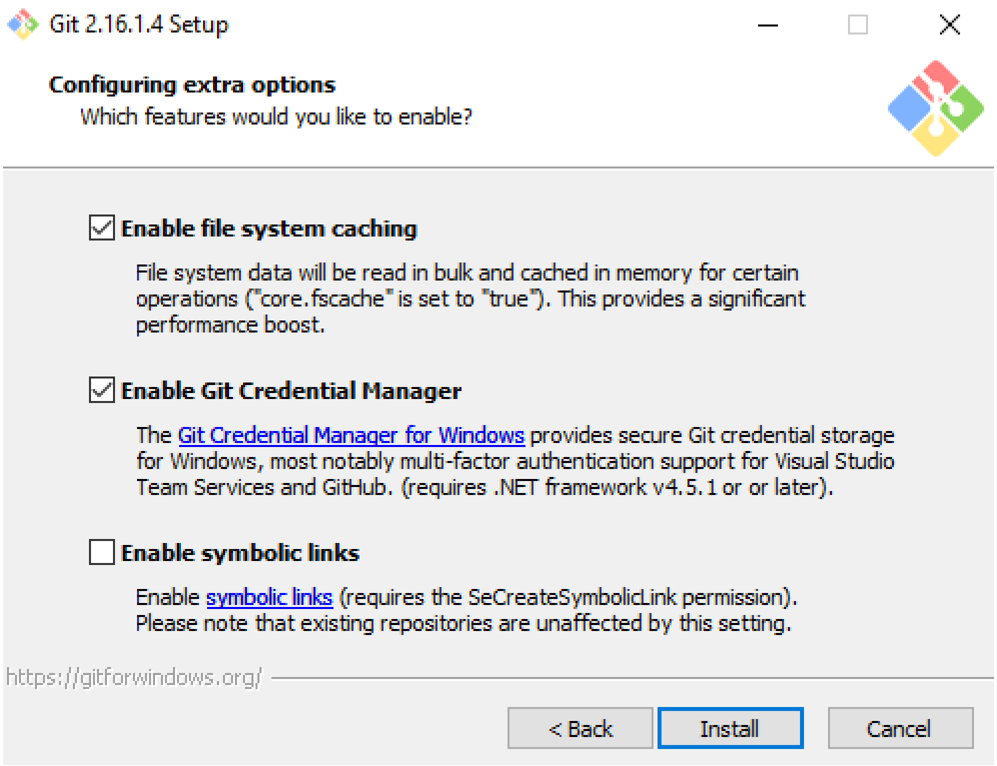
Which terminal emulator do you want to use with your Git Bash?

● **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.



Configuring extra options

Which features would you like to enable?

☑ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☑ **Enable Git Credential Manager**

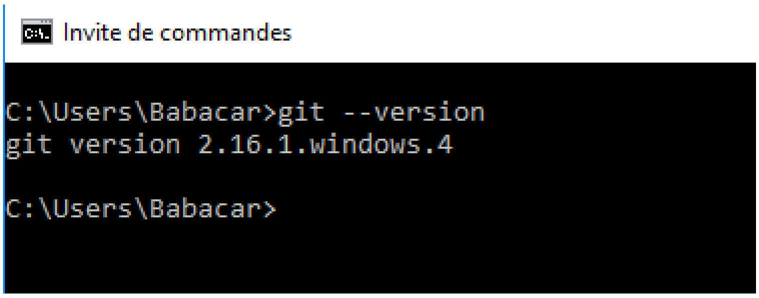
The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or or later).

□ **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

Puis cliquez sur « Install »

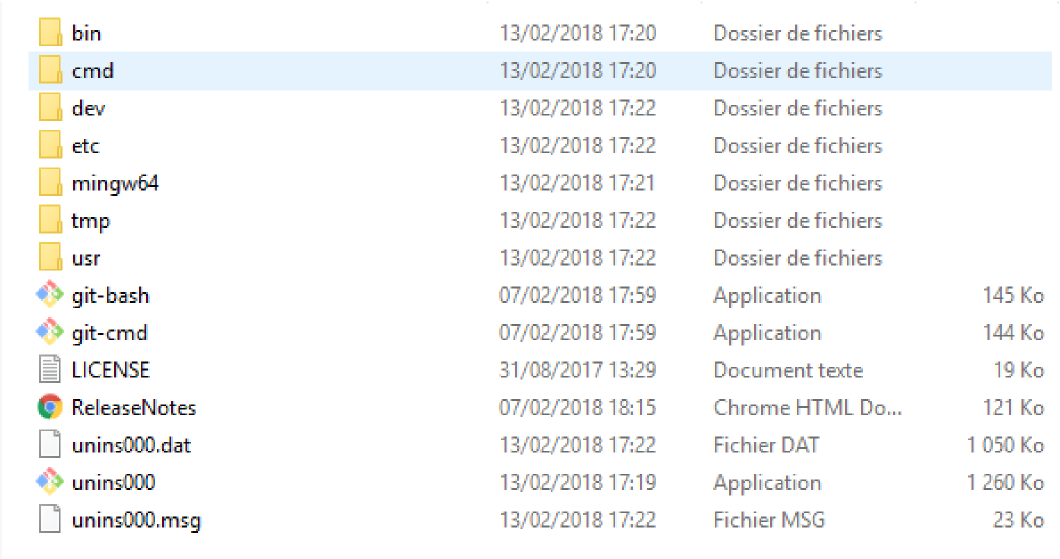
Allez dans l'invite de commandes et tapez : « **git - -version** » et vous obtenez :



```
C:\Users\Babacar>git --version
git version 2.16.1.windows.4

C:\Users\Babacar>
```

Git est bien installé sur votre ordinateur. Dans l'explorateur (C:), vous serez en présence d'un dossier git contenant :



bin	13/02/2018 17:20	Dossier de fichiers	
cmd	13/02/2018 17:20	Dossier de fichiers	
dev	13/02/2018 17:22	Dossier de fichiers	
etc	13/02/2018 17:22	Dossier de fichiers	
mingw64	13/02/2018 17:21	Dossier de fichiers	
tmp	13/02/2018 17:22	Dossier de fichiers	
usr	13/02/2018 17:22	Dossier de fichiers	
git-bash	07/02/2018 17:59	Application	145 Ko
git-cmd	07/02/2018 17:59	Application	144 Ko
LICENSE	31/08/2017 13:29	Document texte	19 Ko
ReleaseNotes	07/02/2018 18:15	Chrome HTML Do...	121 Ko
unins000.dat	13/02/2018 17:22	Fichier DAT	1 050 Ko
unins000	13/02/2018 17:19	Application	1 260 Ko
unins000.msg	13/02/2018 17:22	Fichier MSG	23 Ko