# **Tutoriel installation Git**

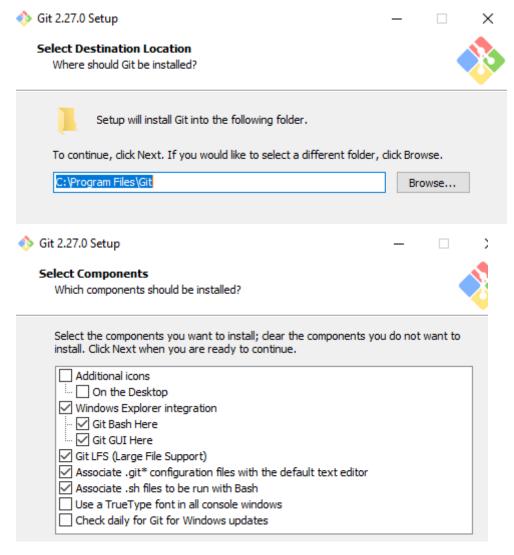
Git est un système de suivi de version et de partage de code, utilisé notamment par le très connu site GitHub.

Pour pouvoir utiliser Git, il faut un client git (un navigateur ne pouvant servir qu'à récupérer le code source, pas à le publier).

Divers utilitaires sont nécessaires pour avoir Git sous Windows.

Téléchargez l'archive : <a href="https://iutdijon.u-bourgogne.fr/intra/info/softs/files/git4win.zip">https://iutdijon.u-bourgogne.fr/intra/info/softs/files/git4win.zip</a>

Ouvrez-la et exécutez l'installateur. Il installera le client Git.



Pour l'éditeur avancé, vous avez le choix, mais il est conseillé d'utiliser Notepad++, Visual Studio Code ou Sublime Text (suivant votre éditeur préféré).



# Choosing the default editor used by Git

Which editor would you like Git to use?



Use Notepad++ as Git's default editor

(NEW!) Notepad++ is a popular GUI editor that can be used by Git.

This editor is popular in part due to the vast number of available plugins; However, when configured via this option, Git will call Notepad++ with plugins disabled (to open the editor as quickly as possible).

Bien sélectionner « Git from command line » comme sur la capture ci-dessous :

Use Git from Git Bash only
 This is the most cautious 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.

 Git from the command line and also from 3rd-party software
 (Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

 Use Git and optional Unix tools from the 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

Choisir « native windows » comme source de certificats.

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?



What oblited hordry Would you like die to use for this oblineedons.	
○ 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 certificat distributed e.g. via Active Directory Domain Services.	es
https://gitforwindows.org/	
Back Next	Cancel

L'option suivante dépend du type de projet utilisé ; la première option est recommandée.

Git 2.27.0 Setup

## 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").

## O 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").

# Ocheckout 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").

https://gitforwindows.org/

Utilisez MinTTY comme terminal.

## 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.

#### O 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.

#### Choose the default behavior of 'git pull'

What should 'git pull' do by default?



## Default (fast-forward or merge)

This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

## O Rebase

Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

# Only ever fast-forard

Fast-forward to the fetched branch. Fail if that is not possible.

Git 2.27.0 Setup

#### 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 <u>Git Credential Manager for Windows</u> 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 later).

## Enable symbolic links

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

Et finalisez l'installation du client Git.