

TUTORIEL D'INSTALLATION ET CONFIGURATION DU LOGICIEL GIT

DESCRIPTION DU THÈME

Propriétés	Description
Intitulé long	Tutoriel d'installation et configuration du logiciel GIT
Formation(s) concernée(s)	<input type="checkbox"/> Classes de première Sciences et technologies du management et de la gestion (STMG) <input type="checkbox"/> Terminale STMG Système d'information de gestion (SIG) <input type="checkbox"/> BTS Services Informatiques aux Organisations
Matière(s)	<input type="checkbox"/> Sciences de gestion <input type="checkbox"/> SIG <input type="checkbox"/> Bloc 1 – Support et mise à disposition de services informatiques <input type="checkbox"/> Bloc 2 SISR – Administration des systèmes et des réseaux <input type="checkbox"/> Bloc 3 SLAM – Cybersécurité des services informatiques
Présentation	Ce document décrit pas-à-pas le téléchargement, l'installation et la configuration des commandes GIT sous Windows.
Savoirs	Gestion de version
Compétences	Pour certains types de ressources : labo, exolab
Transversalité	SLAM/SISR
Prérequis	
Outils	git
Mots-clés	git, tutoriel, installation, configuration, commandes
Durée	Indicative et non obligatoire
Auteur·e·s	David ROUMANET
Version	v 4
Date de publication	15 Septembre 2020

DERNIÈRES RÉVISIONS

Ce tableau contient les modifications apportées au document après sa publication uniquement.

Date	Auteur·e	Description

SOMMAIRE

A Installer git.....	4
1 Téléchargement et installation.....	4
2 Installation sous Windows.....	4
3 Configuration sous Windows.....	7
B Vérification.....	8
1 Version de GIT.....	8
2 Aide.....	8
C Configuration.....	9
1 Identifiants.....	9
2 Liste des paramètres.....	9




Avertissement : ce tutoriel fonctionne avec la version 2.26 de GIT. Chaque mois, les logiciels évoluent et vous pouvez rencontrer des changements et des différences par rapport à ce document. Il est recommandé de laisser les options par défaut s'il n'y a rien de spécifié.

A INSTALLER GIT

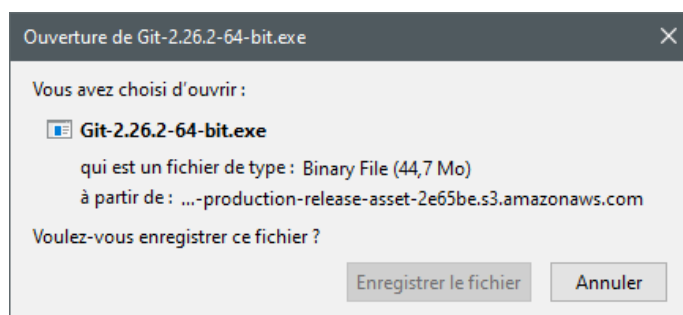
1 TÉLÉCHARGEMENT ET INSTALLATION

L'outil GIT est présent sur le site suivant :

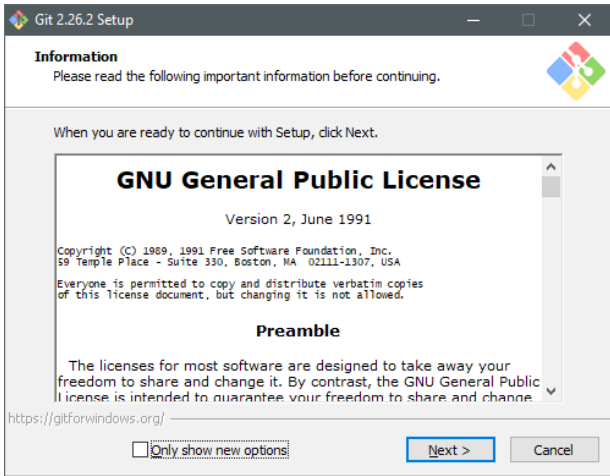
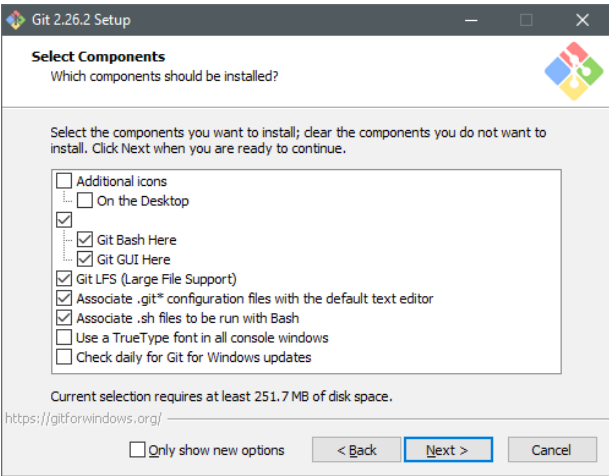
	<p>Windows : https://git-scm.com/download/win</p> <p>Linux : https://git-scm.com/download/win</p> <p>Mac OS : https://git-scm.com/download/mac</p>
---	---

2 INSTALLATION SOUS WINDOWS

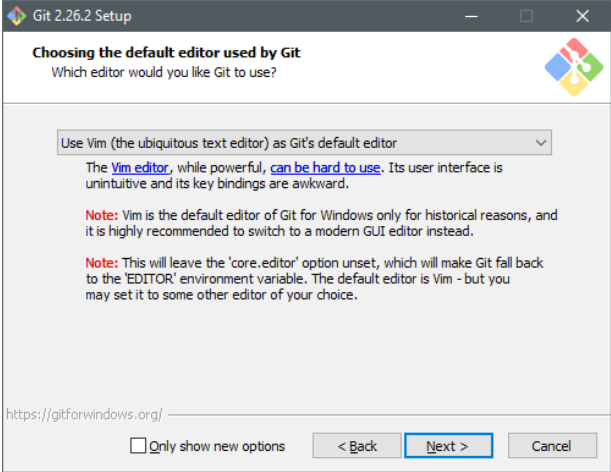
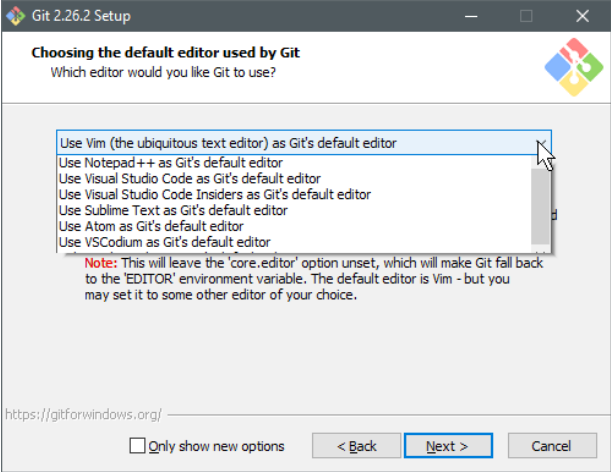
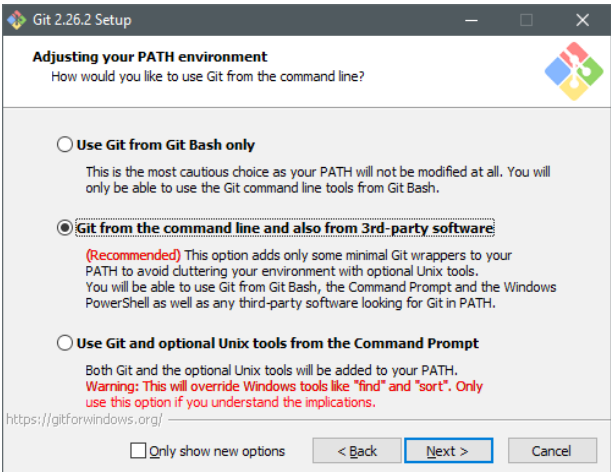
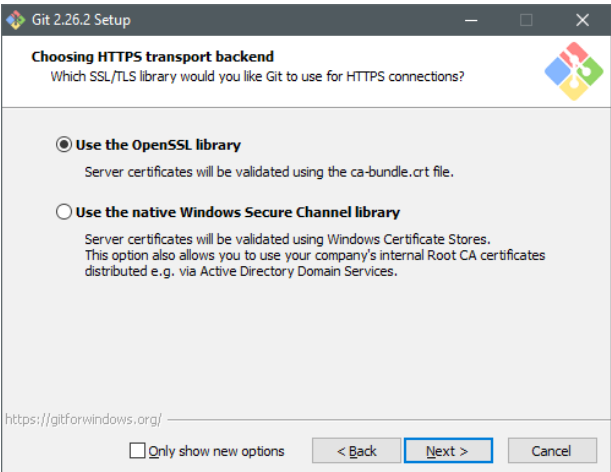
Le navigateur vous propose d'ouvrir ou d'enregistrer le fichier exécutable, d'environ 45 Mo.

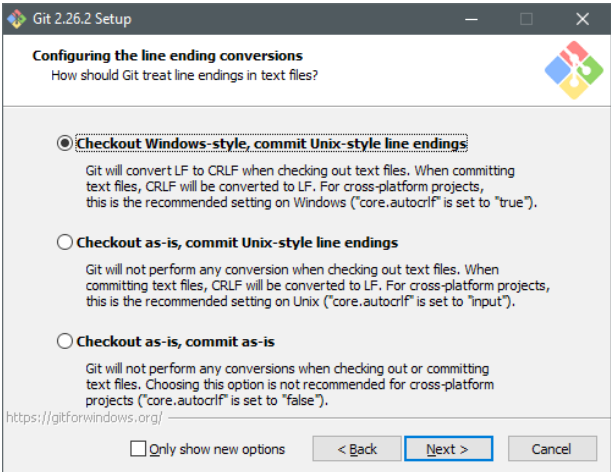
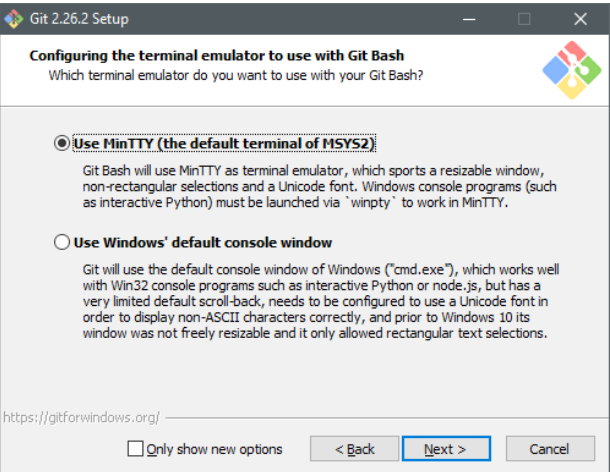
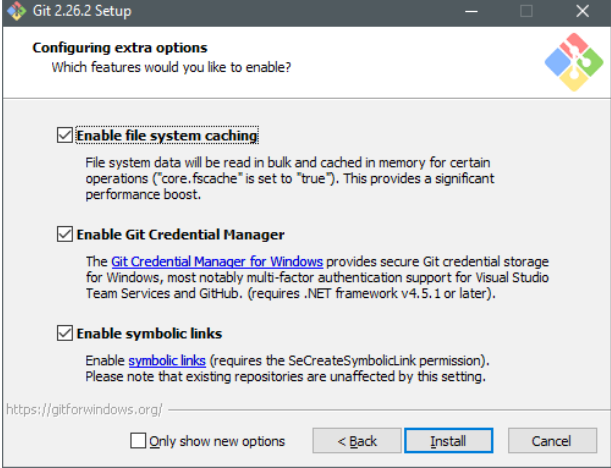
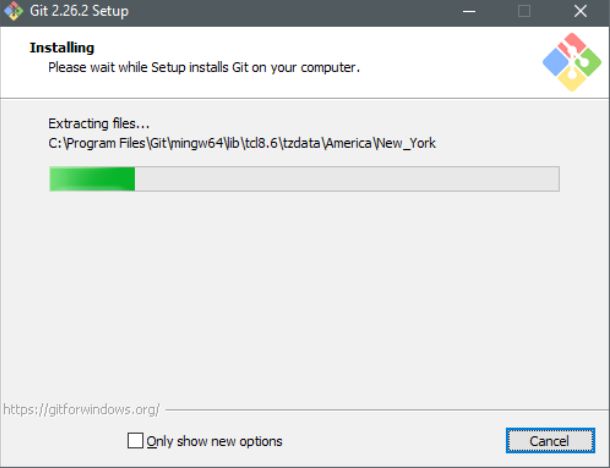


Ensuite, plusieurs fenêtres de dialogue apparaissent :

Étape 1	Étape 2
 <p>Pas de changement</p>	 <p>Pas de changement</p>

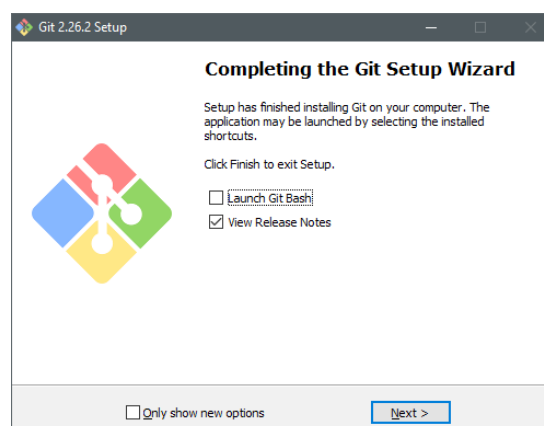
L'étape suivante dépend de votre éditeur de texte préféré :

Étape 3	Étape 4
	
Le choix par défaut laisse l'OS choisir...	... sinon, choisissez votre éditeur dans la liste.
Étape 4	Étape 5
	
Laissez le choix recommandé	Déterminez quelle bibliothèque SSL sera utilisée pour se connecter à un serveur de dépôt en HTTPS. Conservez le choix par défaut.

Étape 6	Étape 7
 <p>Configuring the line ending conversions How should Git treat line endings in text files?</p> <p><input checked="" type="radio"/> 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").</p> <p><input type="radio"/> 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").</p> <p><input type="radio"/> 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").</p> <p>https://gitforwindows.org/</p> <p><input type="checkbox"/> Only show new options < Back Next > Cancel</p>	 <p>Configuring the terminal emulator to use with Git Bash Which terminal emulator do you want to use with your Git Bash?</p> <p><input checked="" type="radio"/> 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.</p> <p><input type="radio"/> 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.</p> <p>https://gitforwindows.org/</p> <p><input type="checkbox"/> Only show new options < Back Next > Cancel</p>
<p>Sous Windows, les fin de ligne sont avec 2 caractères (CR+LF) mais pas sur les autres systèmes.</p>	<p>Cette option n'est utile que lorsque GIT lance des commandes.</p>
Étape 8	Étape 9
 <p>Configuring extra options Which features would you like to enable?</p> <p><input checked="" type="checkbox"/> 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.</p> <p><input checked="" type="checkbox"/> 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 later).</p> <p><input checked="" type="checkbox"/> Enable symbolic links Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.</p> <p>https://gitforwindows.org/</p> <p><input type="checkbox"/> Only show new options < Back Install Cancel</p>	 <p>Installing Please wait while Setup installs Git on your computer.</p> <p>Extracting files... C:\Program Files\Git\mingw64\lib\tzdata\America\New_York</p> <p>https://gitforwindows.org/</p> <p><input type="checkbox"/> Only show new options Cancel</p>
<p>Pas de changement.</p>	<p>Installation...</p>

L'installation est réussie, si vous obtenez l'image suivante :

Il n'est pas nécessaire de lancer le bash git, il est préférable de vérifier la configuration.



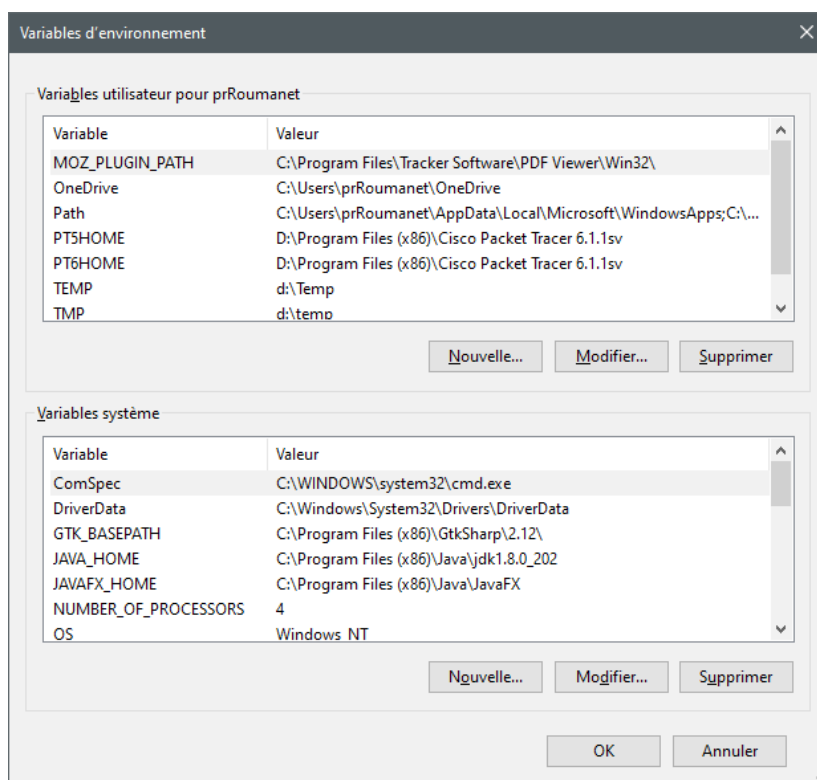
3 CONFIGURATION SOUS WINDOWS

Normalement, GIT s'installe dans le répertoire [C:\Program Files\Git](#).

La variable d'environnement %PATH% est donc modifiée pour permettre l'appel du programme GIT depuis n'importe où.

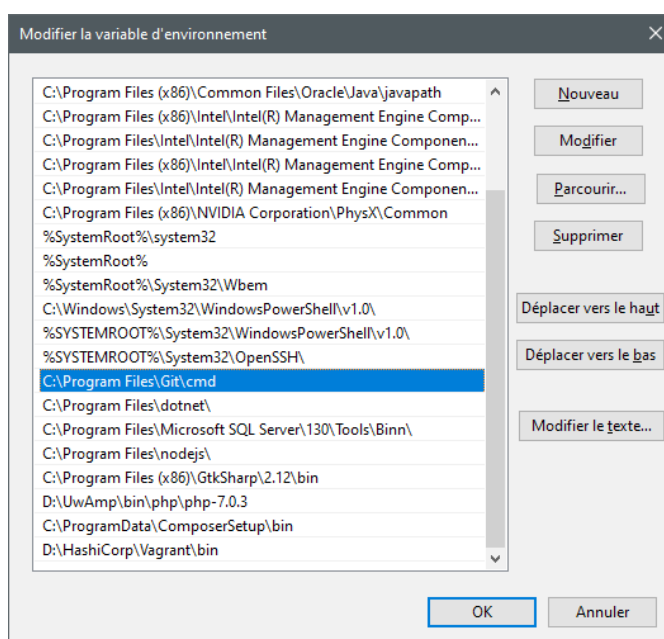
Pour vérifier cela, tapez [Win]+[Pause] et sélectionnez "Paramètres système avancés".

Cliquez sur le bouton [Variables d'environnement...]



Dans la partie haute, ce sont les variables d'environnement de l'utilisateur courant. Si vous sélectionnez la variable PATH du système (en bas), puis cliquez sur le bouton [Modifier...] vous devriez trouver le chemin vers GIT.

Dans le cas contraire, il faudra l'ajouter.



L'ordre des chemins peut avoir son importance en termes de performances. Conserver les chemins les plus fréquemment utilisés en haut de la liste.

B VÉRIFICATION

1 VERSION DE GIT

Si le chemin est bien présent, GIT sera disponible depuis n'importe quel répertoire.

Tapez [Win]+[R] et dans l'invite, écrivez "cmd" puis [Entrée].

En rédigeant une commande `git --version` vous devriez obtenir le numéro de version de votre programme GIT.

Résultat de commande

```
C:\Users\prRoumanet>git --version
git version 2.26.2.windows.1
```

2 AIDE

Git propose de nombreux mots clés. Pour retrouver le mot pour votre commande vous pouvez taper `git --help` ou `git help` pour obtenir l'aide générale.

Cependant, vous pouvez obtenir une aide plus détaillée sur une commande connue, en utilisant la commande `git help -a` :

résultat partiel

```
C:\Users\prRoumanet>git help -a
See 'git help <command>' to read about a specific subcommand
```

```
Main Porcelain Commands
  add          Add file contents to the index
  am           Apply a series of patches from a mailbox
  archive      Create an archive of files from a named tree
  bisect       Use binary search to find the commit that introduced a bug
  branch       List, create, or delete branches
  bundle       Move objects and refs by archive
  checkout     Switch branches or restore working tree files
```

Pour quitter ce mode, il suffit d'appuyer sur la lettre [Q].

C CONFIGURATION

GIT propose l'enregistrement de quelques paramètres dans un fichier, cependant, l'accès à la configuration se fait à l'aide de commandes, décrites ci-après.

1 IDENTIFIANTS

Avant de pouvoir utiliser les commandes GIT vers un serveur distant, il faut fournir à minima un courriel et un pseudonyme.

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

Le paramètre `--global` permet de conserver les informations pour tous les prochains projets.

2 LISTE DES PARAMÈTRES

La commande `git config -l` permet de vérifier l'ensemble des paramètres de configuration.

Extrait de configuration

```
git config -l
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
credential.helper=manager
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
...
core.autocrlf=true
...
user.name=droumanet
user.email=david.roumanet+tlp@ac-grenoble.fr
https.proxy=https://172.16.0.1:3128
```



Note : il est ici possible de configurer le serveur proxy pour pouvoir atteindre le dépôt distant. Les variables se nomment **http.proxy** et **https.proxy**.