

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"/> 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 SLAM – Conception et développement d'applications <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 13
Date de publication	28 Avril 2021

DERNIÈRES RÉVISIONS

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

Date	Auteur·e	Description
04/05/2021	D. Roumanet	Ajout de l'installation sous Linux

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 TÉLÉCHÈGEMENT ET INSTALLATION

L'outil GIT est présent sur le site suivant :

	Windows : https://git-scm.com/download/win Mac OS : https://git-scm.com/download/mac
---	---

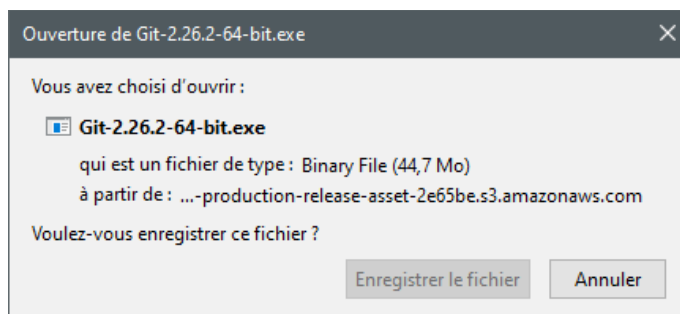
2 INSTALLATION SOUS LINUX

Utilisez la commande suivante :

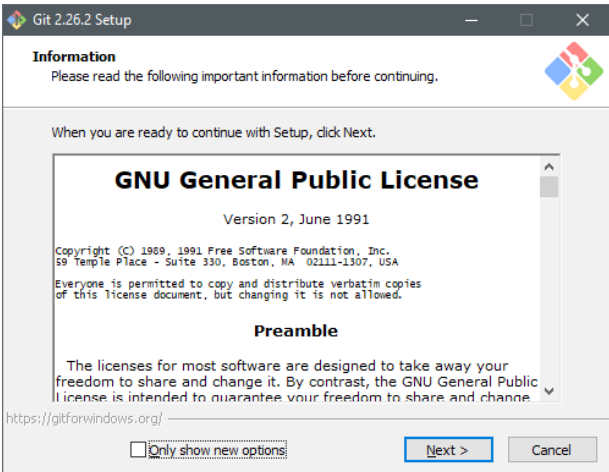
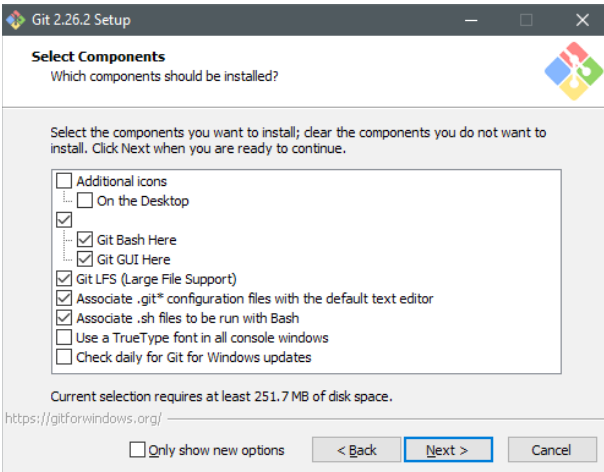
```
sudo apt install git
```

3 INSTALLATION SOUS WINDOWS

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

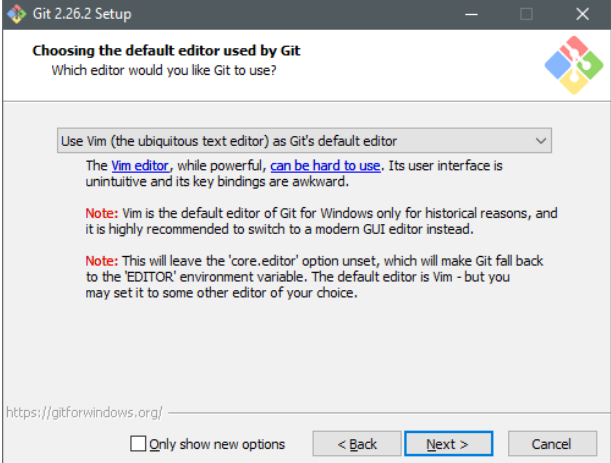
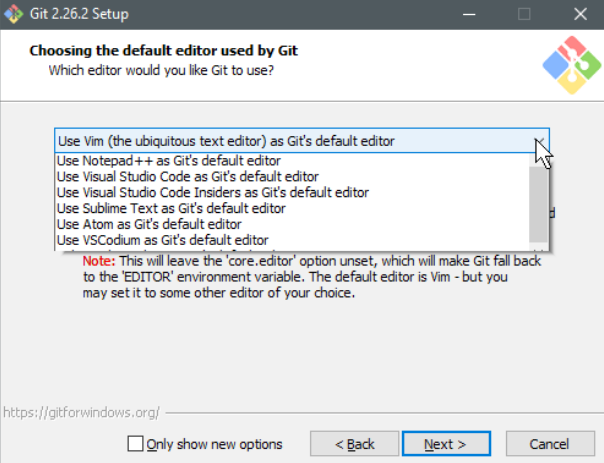
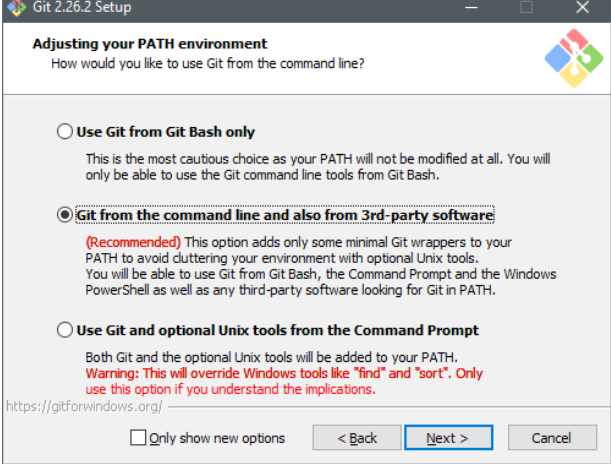
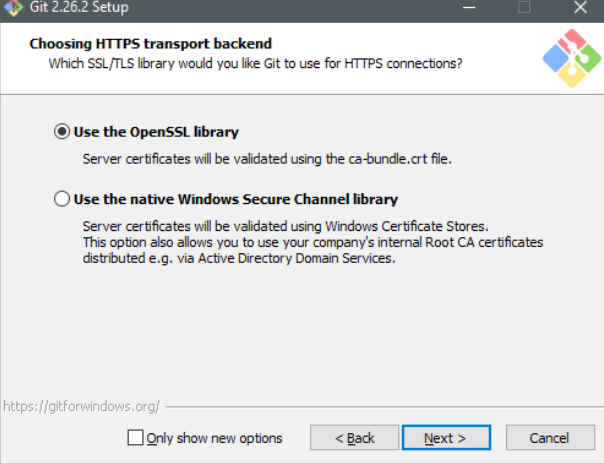


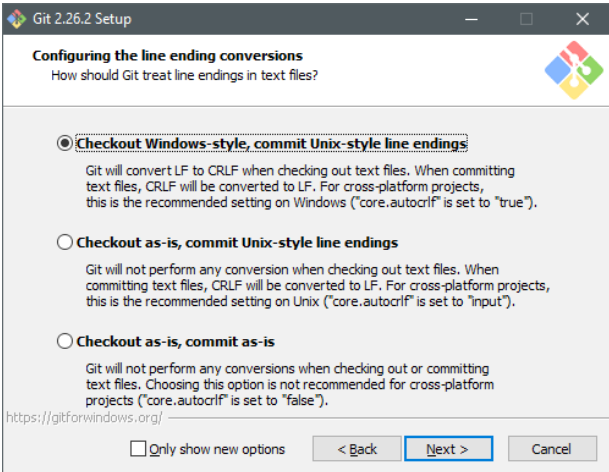
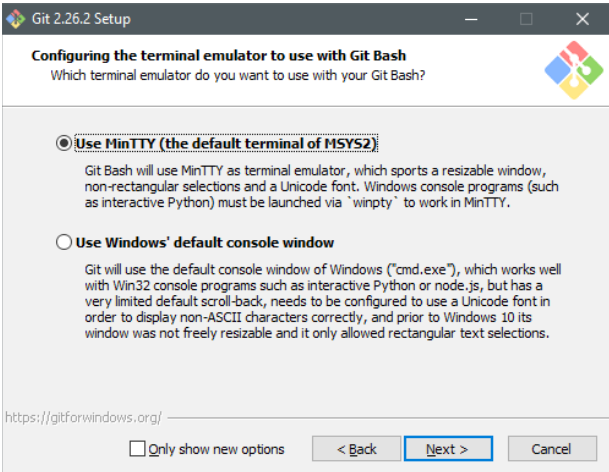
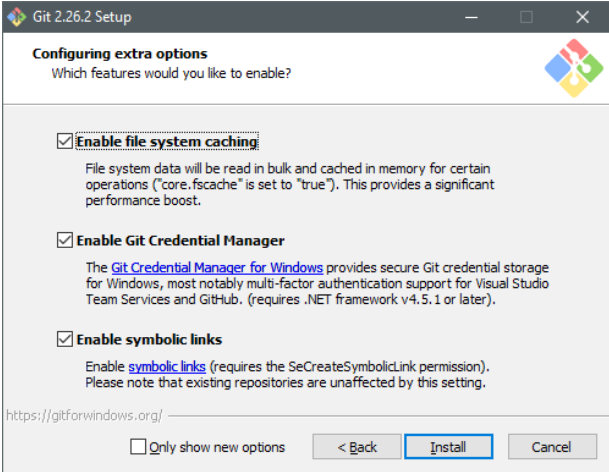
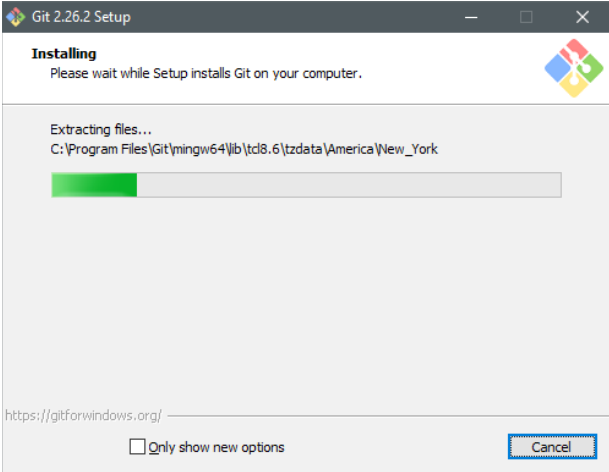
Ensuite, plusieurs fenêtres de dialogue apparaissent :

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

Notez que "Git bash here" et "git gui here" sont des options visibles dans le menu contextuel de l'explorateur de fichier. L'invite de commande qui pourra s'ouvrir utilise le bash de MinGW.

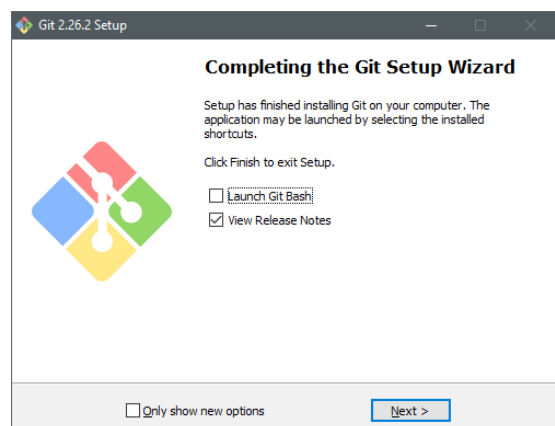
L'étape suivante dépend de votre éditeur de texte préféré :

Étape 3	Étape 4
 <p>Choosing the default editor used by Git Which editor would you like Git to use?</p> <p>Use Vim (the ubiquitous text editor) as Git's default editor</p> <p>The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.</p> <p>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.</p> <p>Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.</p> <p>https://gitforwindows.org/</p> <p><input type="checkbox"/> Only show new options < Back Next > Cancel</p>	 <p>Choosing the default editor used by Git Which editor would you like Git to use?</p> <p>Use Vim (the ubiquitous text editor) as Git's default editor</p> <p>Use Notepad++ as Git's default editor</p> <p>Use Visual Studio Code as Git's default editor</p> <p>Use Visual Studio Code Insiders as Git's default editor</p> <p>Use Sublime Text as Git's default editor</p> <p>Use Atom as Git's default editor</p> <p>Use VSCodium as Git's default editor</p> <p>Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.</p> <p>https://gitforwindows.org/</p> <p><input type="checkbox"/> Only show new options < Back Next > Cancel</p>
<p>Le choix par défaut laisse l'OS choisir...</p>	<p>... sinon, choisissez votre éditeur dans la liste.</p>
Étape 4	Étape 5
 <p>Adjusting your PATH environment How would you like to use Git from the command line?</p> <p><input type="radio"/> 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.</p> <p><input checked="" type="radio"/> 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.</p> <p><input type="radio"/> 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 use this option if you understand the implications.</p> <p>https://gitforwindows.org/</p> <p><input type="checkbox"/> Only show new options < Back Next > Cancel</p>	 <p>Choosing HTTPS transport backend Which SSL/TLS library would you like Git to use for HTTPS connections?</p> <p><input checked="" type="radio"/> Use the OpenSSL library Server certificates will be validated using the ca-bundle.crt file.</p> <p><input type="radio"/> 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.</p> <p>https://gitforwindows.org/</p> <p><input type="checkbox"/> Only show new options < Back Next > Cancel</p>
<p>Laissez le choix recommandé</p>	<p>Déterminez quelle bibliothèque SSL sera utilisée pour se connecter à un serveur de dépôt en HTTPS. Conservez le choix par défaut.</p>

Étape 6	Étape 7
 <p>Sous Windows, les fins de ligne sont avec 2 caractères (CR+LF) mais pas sur les autres systèmes.</p>	 <p>Cette option permet d'utiliser une invite de commande plus puissante que l'invite de base.</p>
Étape 8	Étape 9
 <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.



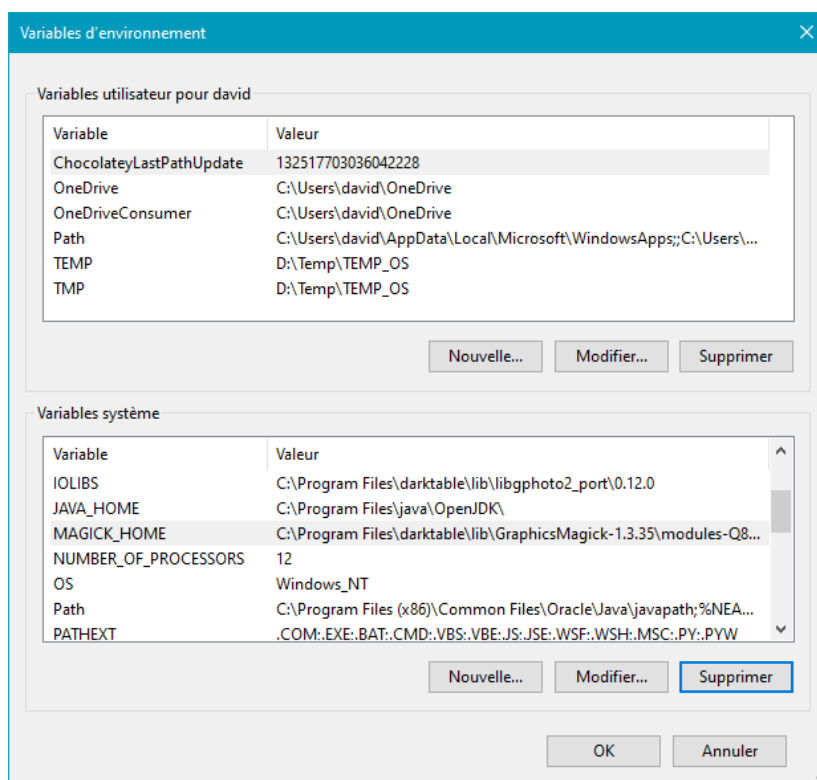
4 CONFIGURATION SOUS WINDOWS

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

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

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

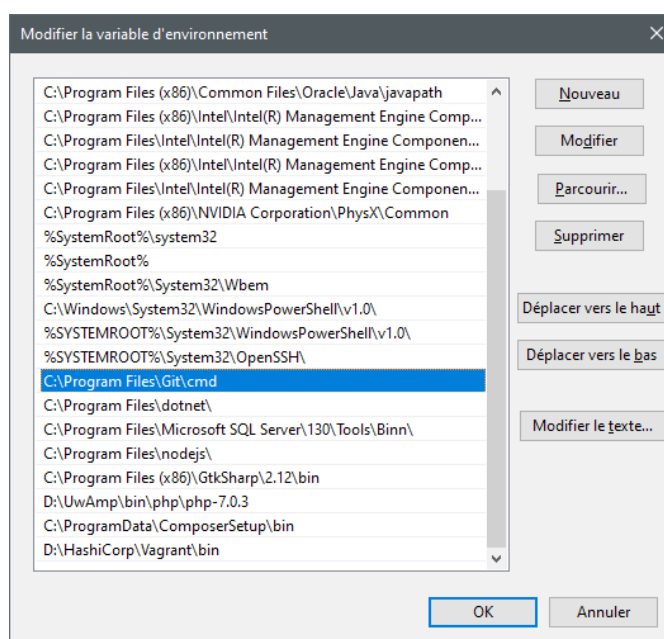
Cliquez sur le bouton [Variables d'environnement...]



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.

i 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 nombreuses commandes. Pour retrouver la syntaxe 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 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**.