

# Git Installation

Software Studio  
DataLab, CS, NTHU

# How to Install

- Download from the link below and select the correct OS based on your requirements.

<https://git-scm.com/downloads>



The screenshot shows the Git website's Downloads page. At the top left is the Git logo and the tagline "--local-branching-on-the-cheap". A search bar is located at the top right. The left sidebar contains navigation links for "About", "Documentation", "Downloads" (highlighted in red), "GUI Clients", "Logos", and "Community". Below the sidebar, there is a section for the "Pro Git book". The main content area features a "Downloads" heading, followed by three operating system options: macOS (with an Apple logo), Windows (with a Windows logo), and Linux/Unix (with a Linux logo). Below these options, a note states that older releases are available and the source repository is on GitHub. To the right of the OS options is a large monitor displaying the "Latest source Release 2.48.1" and a "Download for Windows" button. At the bottom of the page, there are two sections: "GUI Clients" and "Logos".

**git** --local-branching-on-the-cheap

Type / to search entire site...

**About**

**Documentation**

**Downloads**

GUI Clients

Logos

**Community**

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Downloads

 [macOS](#)  [Windows](#)

 [Linux/Unix](#)

Older releases are available and the Git source repository is on GitHub.

Latest source Release  
**2.48.1**  
[Release Notes \(2025-01-13\)](#)

[Download for Windows](#)

**GUI Clients**

Git comes with built-in GUI tools (**git-gui**,

**Logos**

Various Git logos in PNG (bitmap) and EPS

# Installation

- Simply follow the installation steps.
- However, please make sure you the following configurations are correctly set.



## Checkout Windows-style, commit Unix-style endings

**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").

<http://msysgit.github.io/>

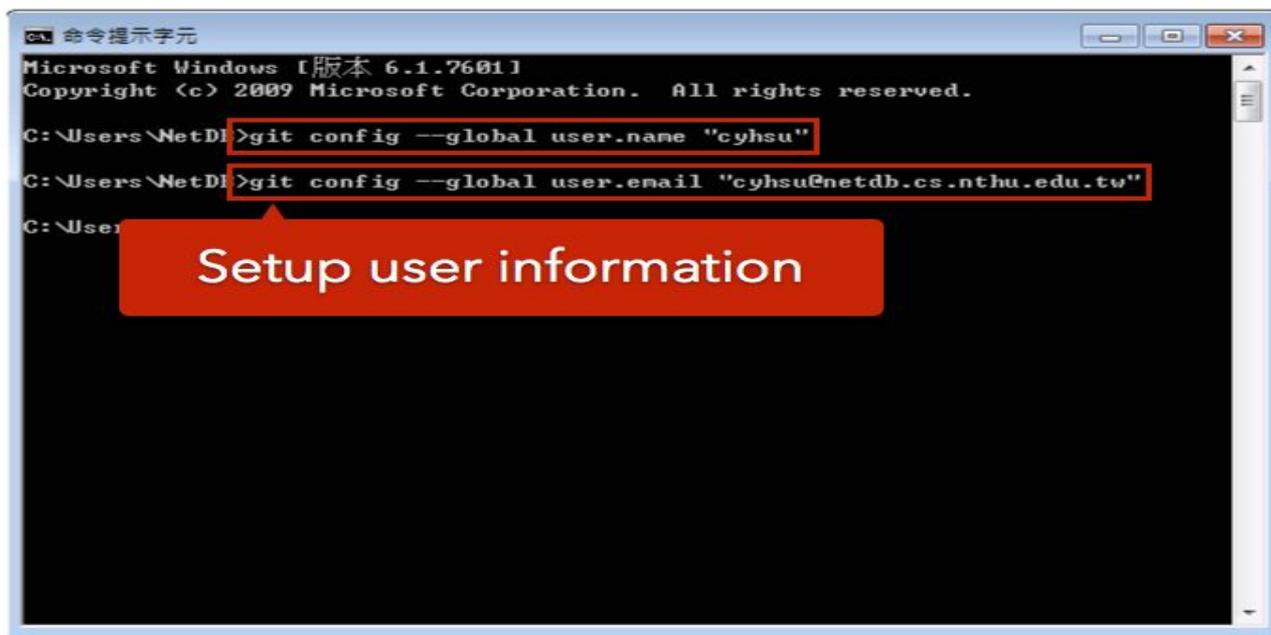
< Back

Next >

Cancel

# What's Next ?

- Setup your information in Git
  - Your name
  - Your email
- This is for identifying yourself in commits, allowing others to see who made each change in a repository.



A screenshot of a Windows command prompt window titled "命令提示字元". The window shows the following text: "Microsoft Windows [版本 6.1.7601] Copyright (c) 2009 Microsoft Corporation. All rights reserved. C:\Users\NetDI >git config --global user.name "cyhsu" C:\Users\NetDI >git config --global user.email "cyhsu@netdb.cs.nthu.edu.tw" C:\User". The two command lines are highlighted with red boxes. A red callout box with the text "Setup user information" points to the second command line.

```
命令提示字元
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\NetDI >git config --global user.name "cyhsu"
C:\Users\NetDI >git config --global user.email "cyhsu@netdb.cs.nthu.edu.tw"
C:\User
```

```
$ git config --global user.name "name"
$ git config --global user.email "email"
```

**GitLab** <gitlab@shwu10.cs.nthu.edu.tw>

to me ▾

Wed, Feb 25, 1:31PM (1 day ago)



**Welcome, 2026TA\_ACCOUNT!**

To get started, click the link below to confirm your account.

[Confirm your account](#)

**User Settings**

- Profile
- Account
- Applications
- Chat
- Access Tokens**
- Emails
- Password
- Notifications
- SSH Keys
- GPG Keys
- Preferences
- Active Sessions
- Authentication log

<< Collapse sidebar

User Settings > Access Tokens

Search settings

### Personal Access Tokens

You can generate a personal access token for each application you use that needs access to the GitLab API.

You can also use personal access tokens to authenticate against Git over HTTP. They are the only accepted password when you have Two-Factor Authentication (2FA) enabled.

#### Add a personal access token

Enter the name of your application, and we'll return a unique personal access token.

**Token name**  
ss

For example, the application using the token or the purpose of the token.

**Expiration date**  
2026-08-31

#### Select scopes

Scopes set the permission levels granted to the token. [Learn more.](#)

- api**  
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- read\_api**  
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- read\_user**  
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- read\_repository**  
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- write\_repository**

- User Settings
- Profile
- Account
- Applications
- Chat
- Access Tokens**
- Emails
- Password
- Notifications
- SSH Keys
- GPG Keys
- Preferences
- Active Sessions
- Authentication log

- read\_repository**  
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- write\_repository**  
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).
- read\_registry**  
Grants read-only access to container registry images on private projects.
- write\_registry**  
Grants write access to container registry images on private projects.
- sudo**  
Grants permission to perform API actions as any user in the system, when authenticated as an admin user.

Create personal access token

Active personal access tokens (0)

Personal access tokens are not revoked upon expiration.

This user has no active personal access tokens.

Feed token

Your feed token authenticates you when your RSS reader loads a personalized RSS feed or when your calendar application loads a personalized calendar. It is visible in those feed URLs.

It cannot be used to access any other data.

Feed token

\*\*\*\*\*

Keep this token secret. Anyone who has it can read activity and issue RSS feeds or your calendar feed as if they were you. If that happens, reset this token.

- User Settings
- Profile
- Account
- Applications
- Chat
- Access Tokens**
- Emails
- Password
- Notifications
- SSH Keys
- GPG Keys
- Preferences
- Active Sessions
- Authentication log

User Settings > Access Tokens

Your new personal access token has been created.

Search settings

### Personal Access Tokens

You can generate a personal access token for each application you use that needs access to the GitLab API.

You can also use personal access tokens to authenticate against Git over HTTP. They are the only accepted password when you have Two-Factor Authentication (2FA) enabled.

Your new personal access token  
ozQ3V1cV36RokYXX78th

Make sure you save it - you won't be able to access it again.

#### Add a personal access token

Enter the name of your application, and we'll return a unique personal access token.

##### Token name

For example, the application using the token or the purpose of the token.

##### Expiration date

##### Select scopes

Scopes set the permission levels granted to the token. [Learn more.](#)

- api**  
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- read\_api**  
Grants read access to the API, including all groups and projects, the container registry, and the package registry.

Please use this token along with your username to clone your first repository.

If you want to use SSH, you can refer to this [article](#)