

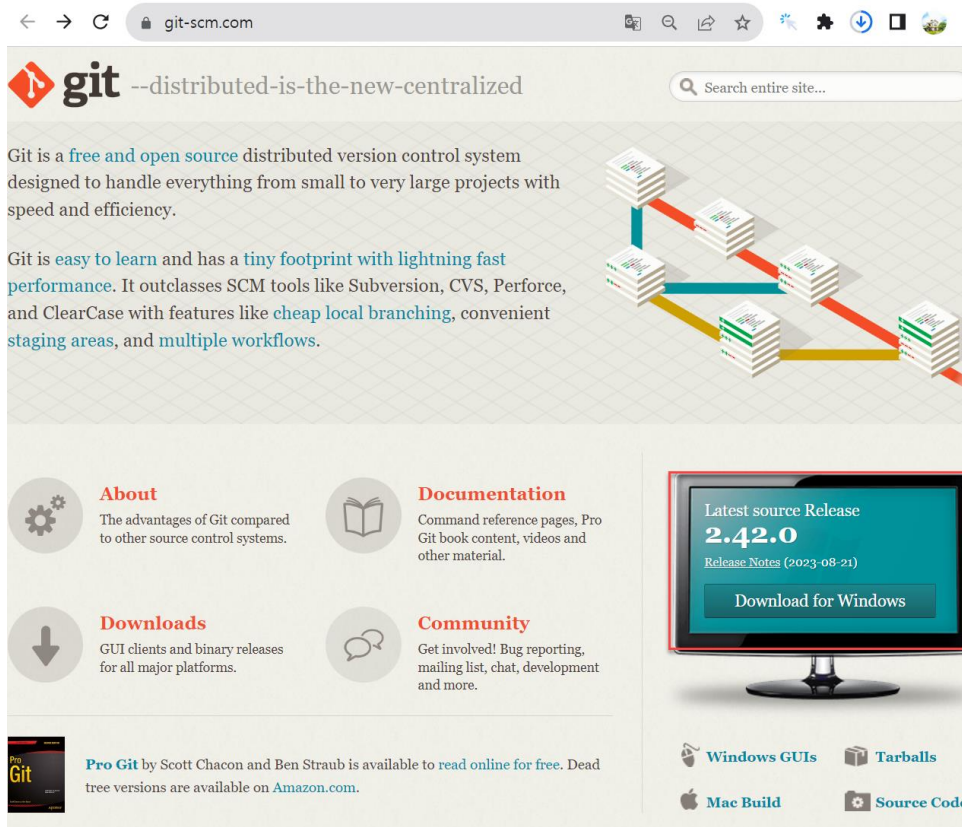
윈도우 10 Git 설치 하기

1. Git 설치

▶ 1. Git 설치 파일 다운로드

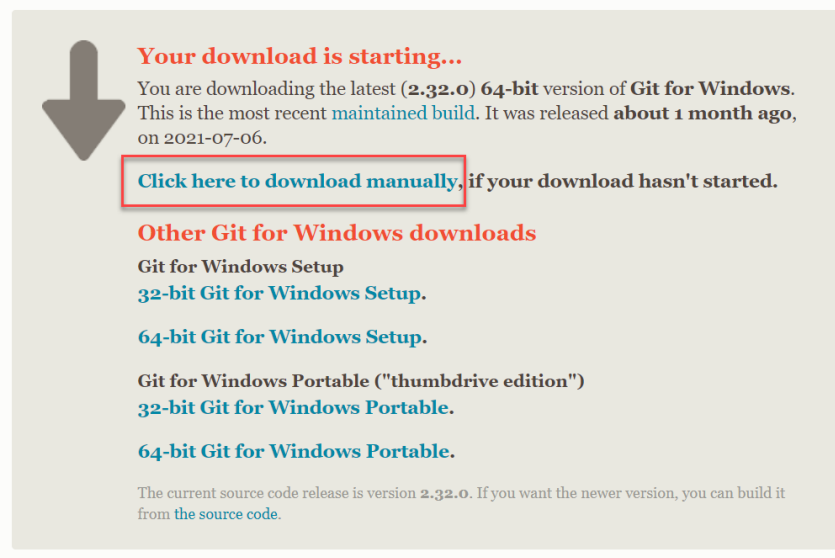
1.1 Git SCM 에 접속하여 설치 파일 다운로드

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



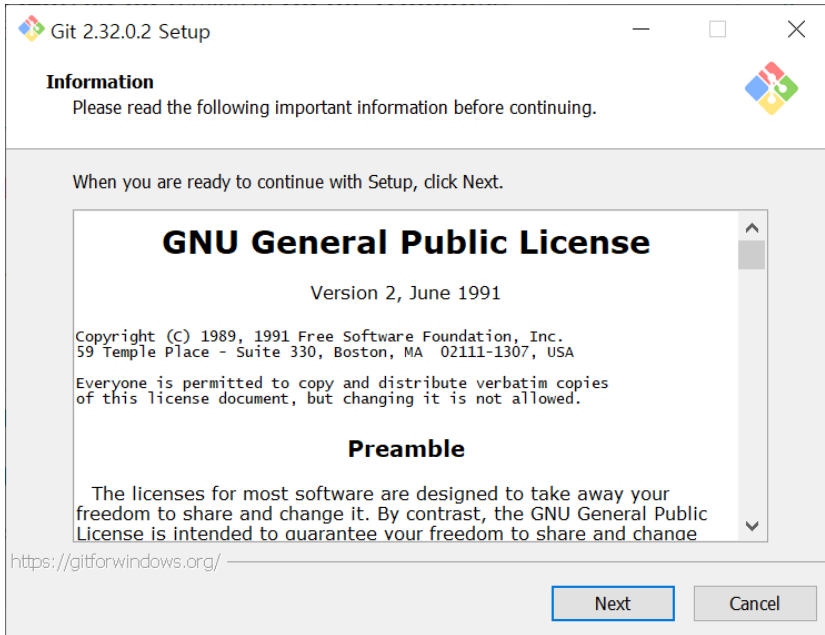
- 자동으로 다운로드가 안된다면 다음 클릭. (현재 기준 2.32.0 버전 다운로드)

Downloading Git

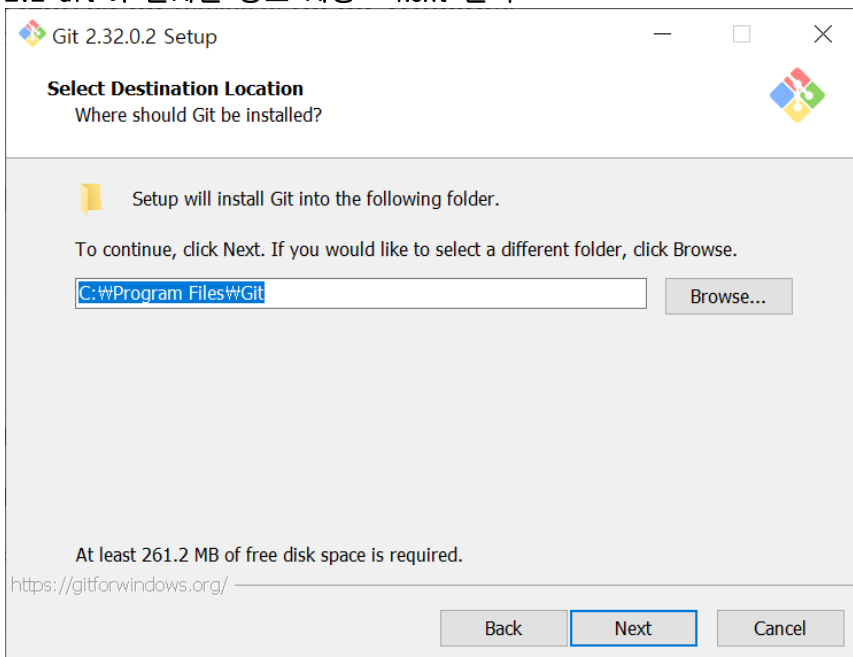


▶ 2. Git 설치

2.1 Next 클릭

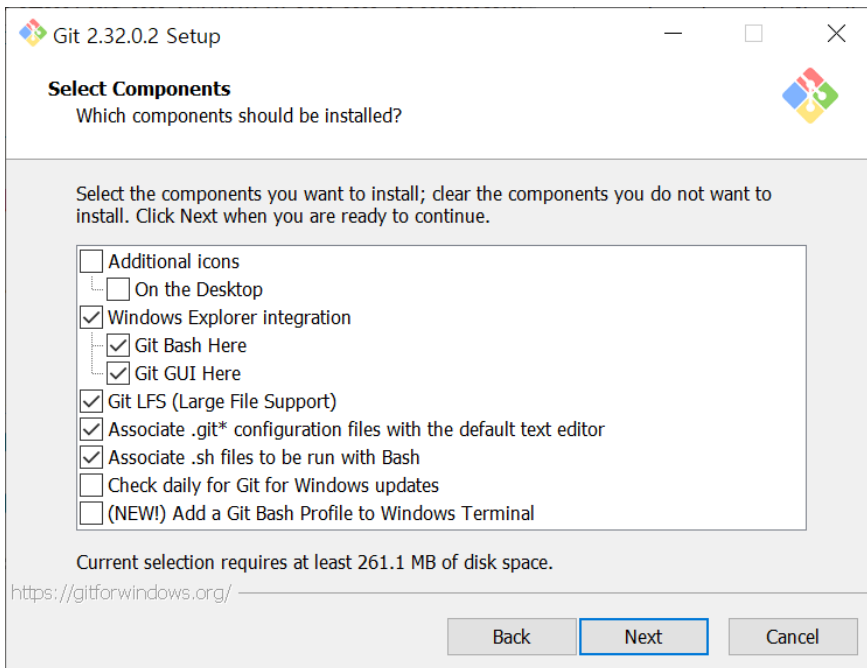


2.2 Git 이 설치될 경로 지정 > Next 클릭



2.3 설치할 구성요소 선택

- Default 선택되어있는 상태로 Next 클릭



※ 참고

1. Additional icons > On the Desktop

- 바탕화면에 아이콘 추가

2. Windows Explorer integration

- Git Bash Here : Git Bash 연결기능(폴더에서 Git 을 바로 연결할 수 있는 기능)
- Git GUI Here : Git GUI 연결기능(폴더에서 Git 을 바로 연결할 수 있는 기능)

해당 설치를 통해 폴더에서 마우스 우클릭 해보면 Git Bash Here, Git GUI Here 가 추가되어 있는 것을 볼 수 있다.

3. Git LFS (Large File Support)

- 용량이 큰 File 지원

4. Associate .git configuration files with the default text editor

- git 구성파일을 기본 텍스트 편집기와 연결

5. Associate .sh files to be run with Bash

- 확장자.sh 파일을 Bash 와 연결

6. Use a TrueType font in all console windows

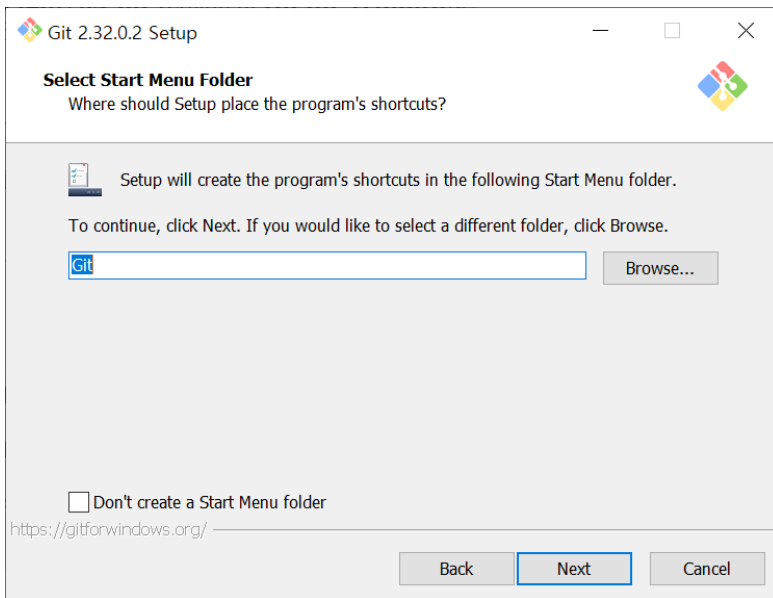
- 윈도우 콘솔창에서 TrueType 글꼴 사용

7. Check daily for Git for Windows updates

- 윈도우즈용 Git 업데이트를 매일 확인 할지 여부

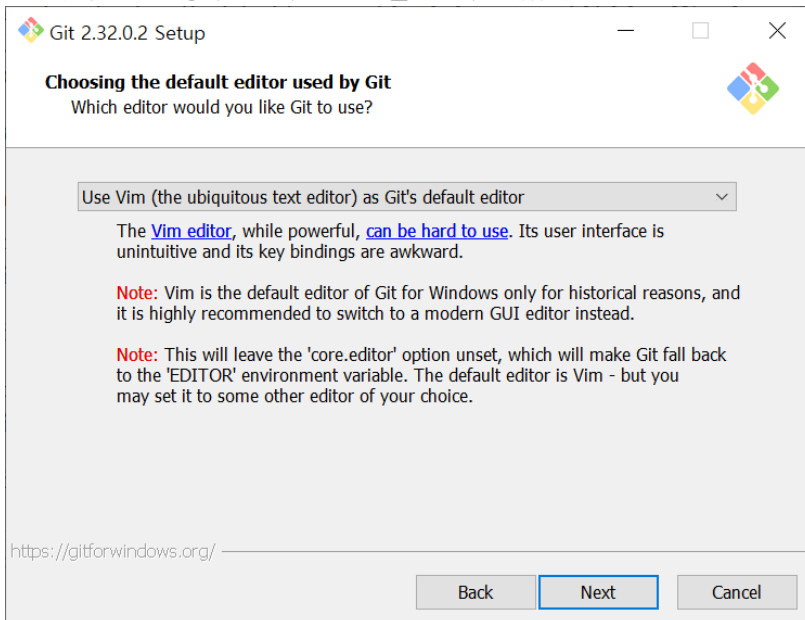
2.4 시작폴더 경로를 선택

- "Don't create a Start Menu folder" 를 체크하면 시작메뉴에 추가하지 않는다.

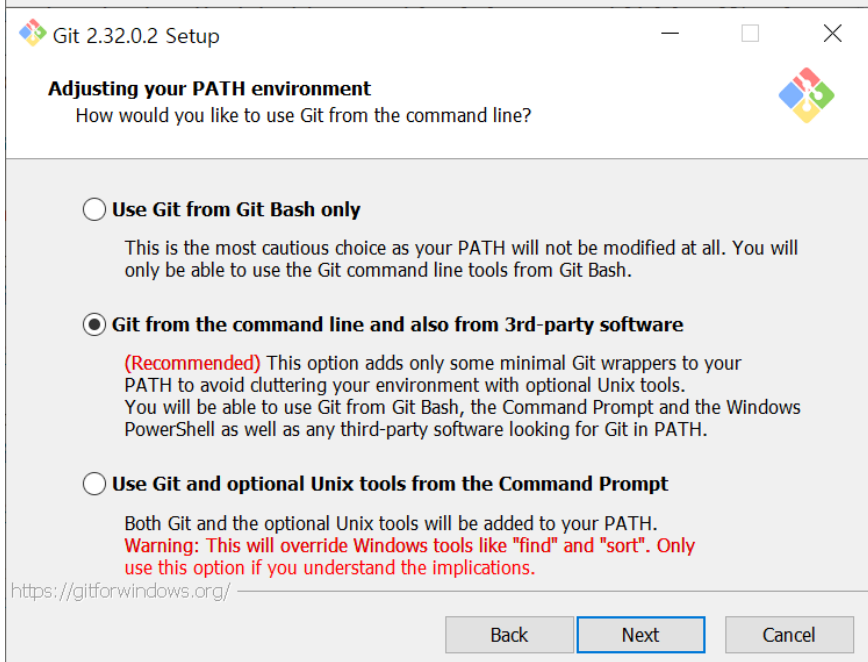
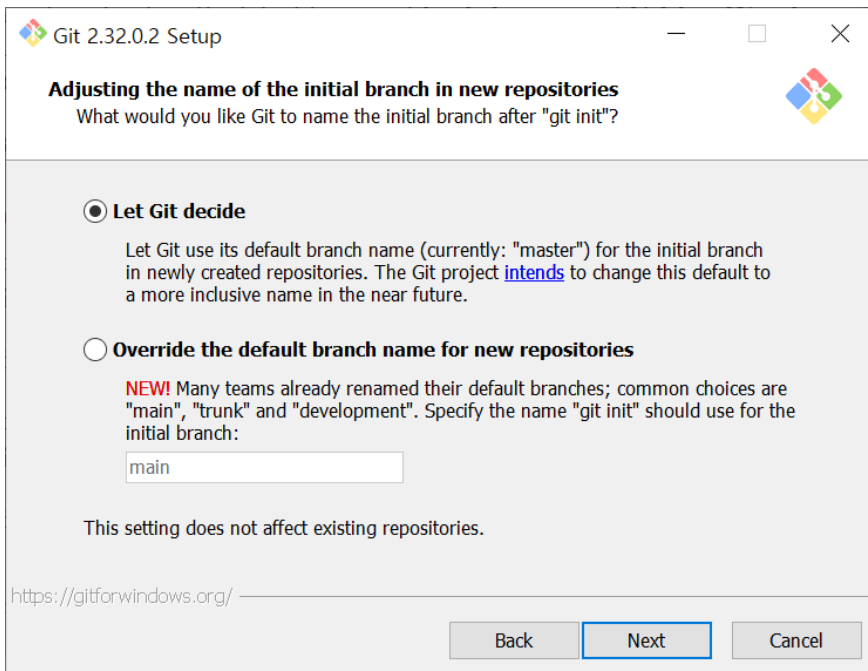


2.5 Git 을 사용할 기본 에디터 선택

- 나와 같은 경우는 기본 Vim 을 선택 하였다.



2.6 환경변수 옵션 설정



● Use Git from Git Bash only

- Git bash 에서만 Git Command 수행

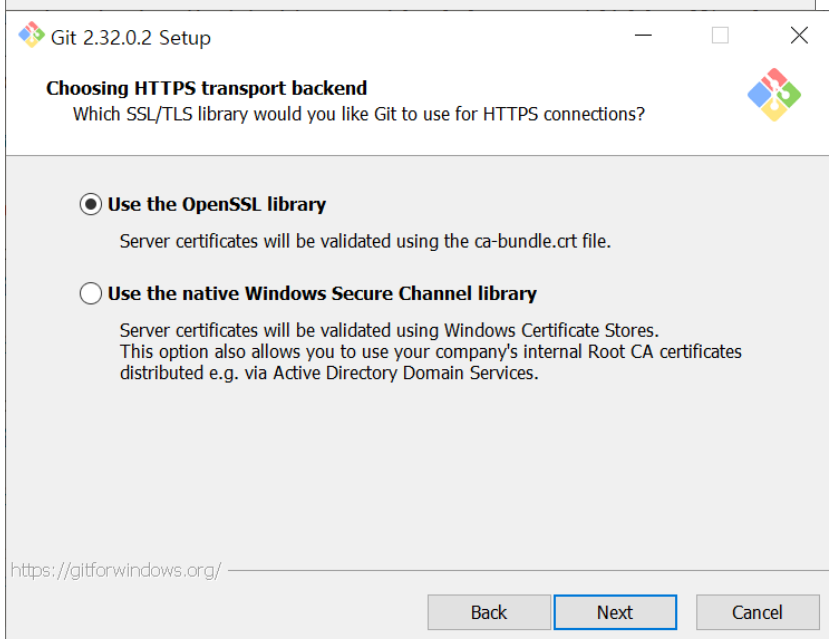
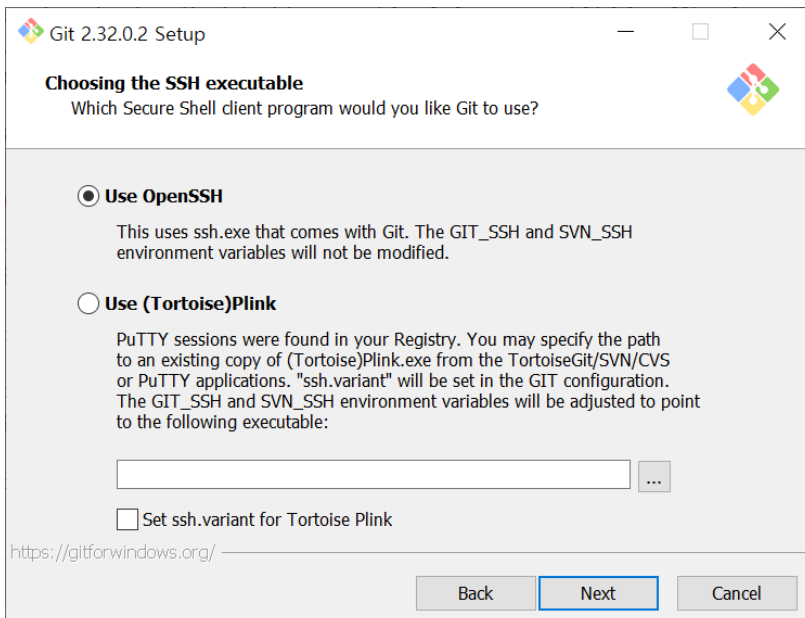
● Use Git from the Windows Command Prompt

- Git 를 환경변수에 등록하고 윈도우 cmd 등 에서도 Git 사용 가능

● Use Git and optional Unix tools from the Windows Command Prompt

- 윈도우 cmd 에서 Git 과 유닉스도구를 사용할 경우 환경변수에 추가

2.7 https 전송시 인증서 선택



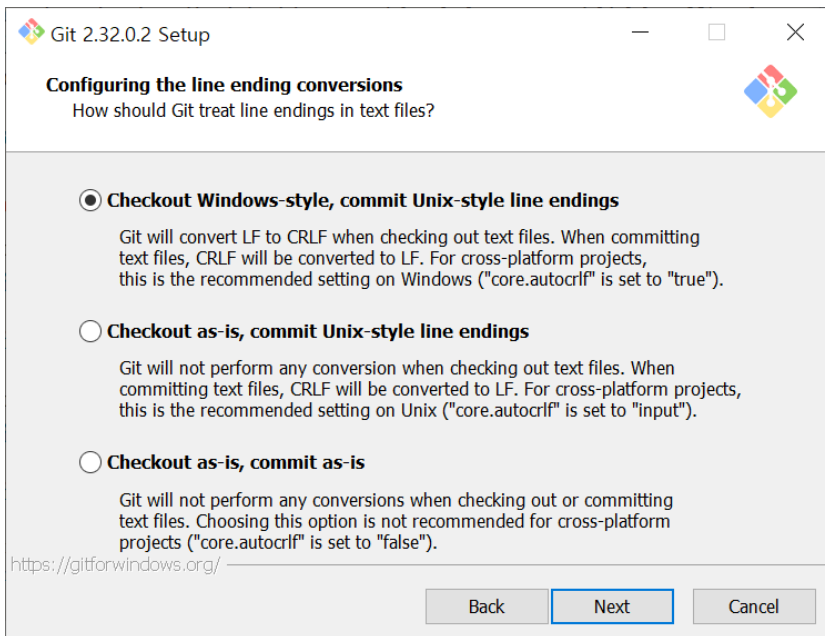
● Use the OpenSSL library

- OpenSSL 라이브러리 사용

● Use the native Windows Secure Channel library

- Windows 인증서 저장소를 사용하여 유효성 검사

2.8 Git 저장소에 체크인, 체크 아웃 할 때 줄 바꿈 옵션 선택



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

- 체크아웃시에는 윈도우 스타일, 커밋시에는 유닉스 스타일 적용

● Checkout as-is, commit Unix-style line endings

- 체크아웃시에는 스타일 변환 없다. 커밋시에만 유닉스 스타일 적용

● Checkout as-is, commit as-is

- 둘다 스타일 변환 없음

※ 개발 하다보면 줄바꿈 때문에 간혹 문제를 겪곤 한다.
다음 내용을 한번정도는 확인하고 넘어가면 좋을 것 같다.

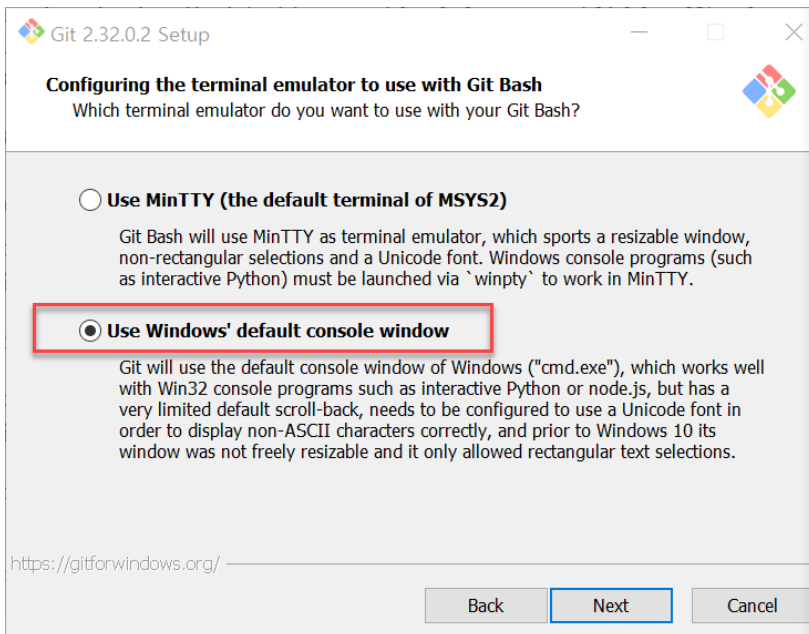
프로그래밍 하면서 "Wn" 또는 "WrWn"을 사용하여 줄바꿈 처리를 해본적이 있을 것이다.
여기서 Wr 을 캐리지 리턴(Carriage Return), Wn 을 라인 피드(Line Feed) 라고 한다.

> 윈도우 계열 : "WrWn" 사용

> 유닉스 계열(리눅스, 맥 등) : "Wn" 사용

Git 은 원래 리눅스 운영체제에서 사용하는 것을 전제로 만들어 졌기 때문에, 1 번을 체크 하면 Commit 수정시 라인 피드방식으로 "Wn"을 사용하도록 자동으로 바꿔주게 된다.

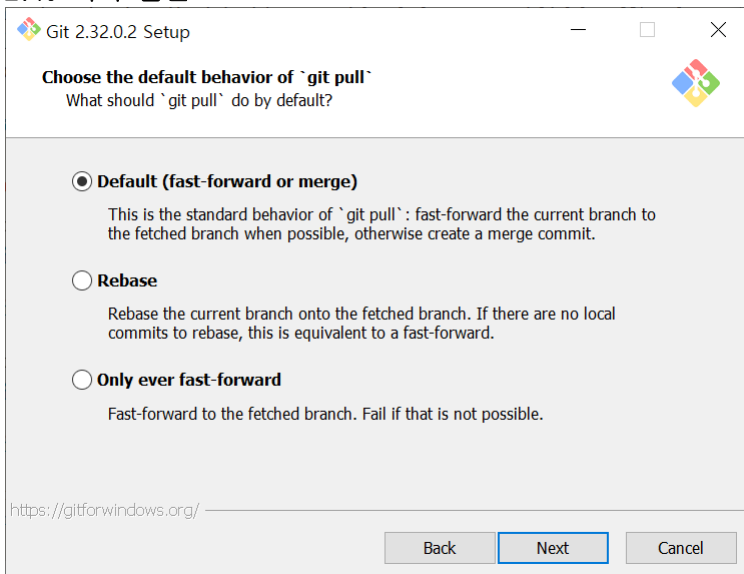
2.9 Git Bash 터미널 에뮬레이터의 설정

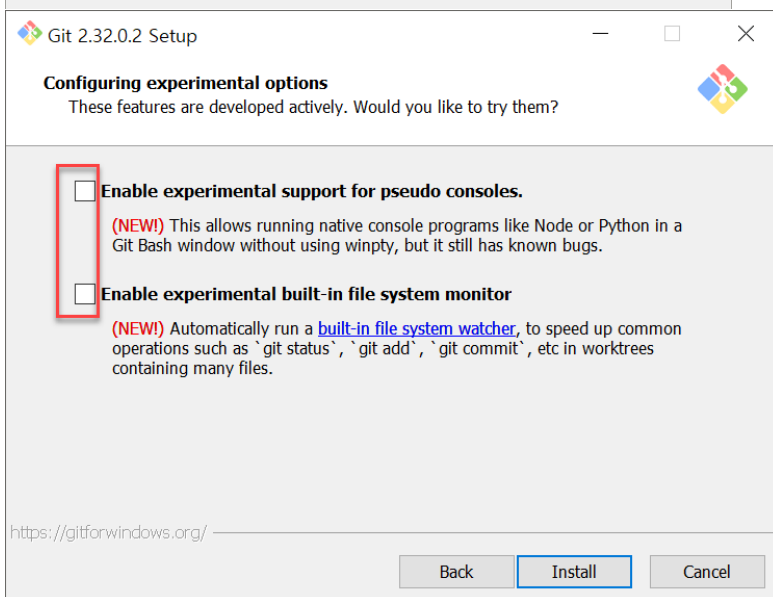
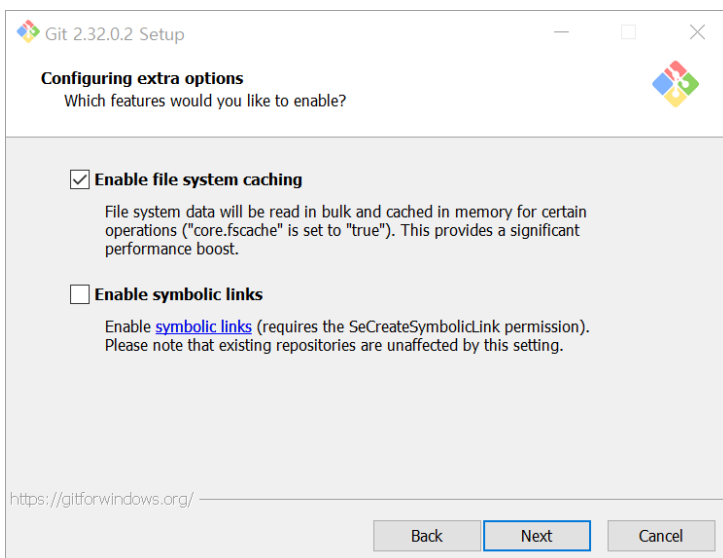
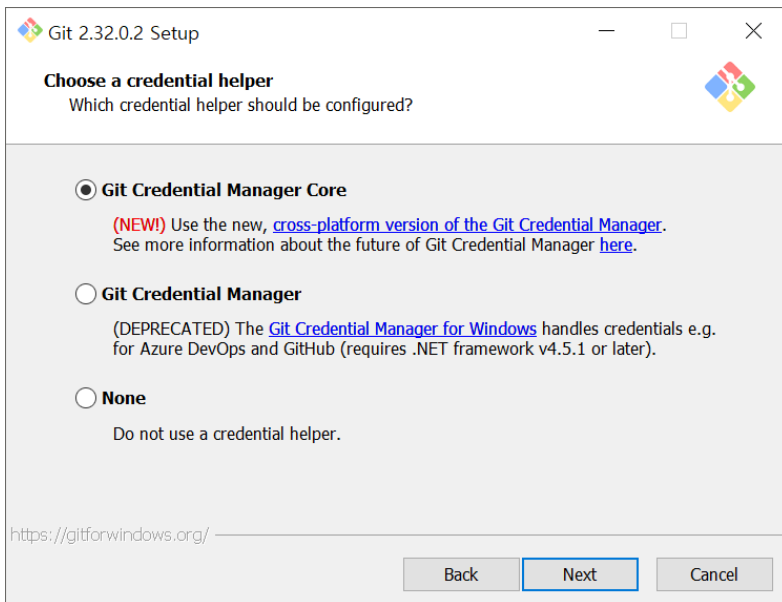


- Use MinTTY(the default terminal of MSYS2)
 - MinTTY terminal emulator 사용(Git bash 기본 터미널 사용)

- Use Windows' default console window
 - 윈도우 기본 콘솔 사용

2.10 기타 옵션





● Enable file system caching

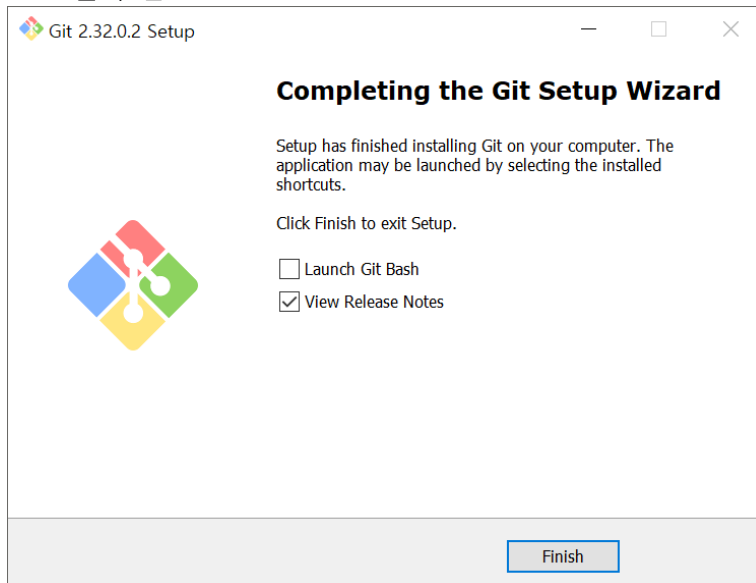
- 빠른 실행을 위해 파일 시스템 데이터를 메모리에 캐시

● Enable Git Credential Manager

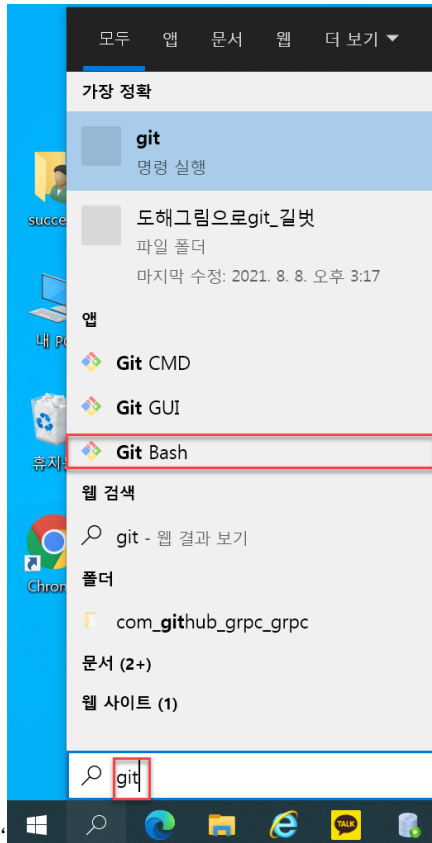
- Windows 용 Git 보안 자격증명 저장소를 사용하기 위하여, Git 인증관리자 활성화

- Enable symbolic links
 - symbolic links 활성화

2.11 설치 완료



- Launch Git Bash 에 체크하고 Finish 버튼을 클릭하면 Git Bash 가 실행 된다.
- 특정 폴더에서 우클릭시에도 Git Bash 를 실행할 수 있다.



운영체제와 상관없이 리눅스 방식의 명령을 사용한다.

‘git’ 입력하여 옵션 확인하기

```
MINGW64:/c/Users/success
success@serbible MINGW64 ~
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--super-prefix=<path>] [--config-env=<name>=<envvar>]
      <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone                Clone a repository into a new directory
  init                 Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add                  Add file contents to the index
```

git --version

```
success@serbible MINGW64 ~
$ git --version
git version 2.32.0.windows.2
```

2.12 Git 사용자 등록 및 확인

1. 사용자 등록

```
git config --global user.name "name"
git config --global user.email "god@god.com"
```

2. 확인

```
git config -list
```

```
MINGW64/c/Users/success
success@serbible MINGW64 ~
$ git config --global user.name "hong"
success@serbible MINGW64 ~
$ git config --global user.email "serbible@nate.com"
success@serbible MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=hong
user.email=serbible@nate.com
filter.lfs.required=true
...skipping
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=hong
user.email=serbible@nate.com
```

화면을 깨끗이 정리하기 : **clear**

[완료]