# CHAPTER III GIT

## General Theory

### **Git**

Git is a mature, actively maintained open source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel. A staggering number of software projects rely on Git for version control, including commercial projects as well as open source. Developers who have worked with Git are well represented in the pool of available software development talent and it works well on a wide range of operating systems and IDEs (Integrated Development Environments).

### **Advantages and Disadvantages of Using Git**

1. **Advantages**
   * + 1. **Performance**

The raw performance characteristics of Git are very strong when compared to many alternatives. Committing new changes, branching, merging and comparing past versions are all optimized for performance. The algorithms implemented inside Git take advantage of deep knowledge about common attributes of real source code file trees, how they are usually modified over time and what the access patterns are.

Unlike some version control software, Git is not fooled by the names of the files when determining what the storage and version history of the file tree should be, instead, Git focuses on the file content itself. After all, source code files are frequently renamed, split, and rearranged. The object format of Git's repository files uses a combination of delta encoding (storing content differences), compression and explicitly stores directory contents and version metadata objects.

* + - 1. **Security**

Git has been designed with the integrity of managed source code as a top priority. The content of the files as well as the true relationships between files and directories, versions, tags and commits, all of these objects in the Git repository are secured with a cryptographically secure hashing algorithm called SHA1. This protects the code and the change history against both accidental and malicious change and ensures that the history is fully traceable.

With Git, you can be sure you have an authentic content history of your source code.

* + - 1. **Flexibility**

One of Git's key design objectives is flexibility. Git is flexible in several respects: in support for various kinds of nonlinear development workflows, in its efficiency in both small and large projects and in its compatibility with many existing systems and protocols.

1. **Disadvantages**
   * + 1. Git is can be difficult to learn.
       2. Support for windows is limited than Linux

## Working Procedure

1. **Tool requirement for setup Git**

Before setup, we need to bring tools as follows:

Table 3.1 Tool Requirement

|  |  |  |
| --- | --- | --- |
| **No** | **Tool** | **Picture** |
| 1 | Laptop | Picture 3.1 Laptop |

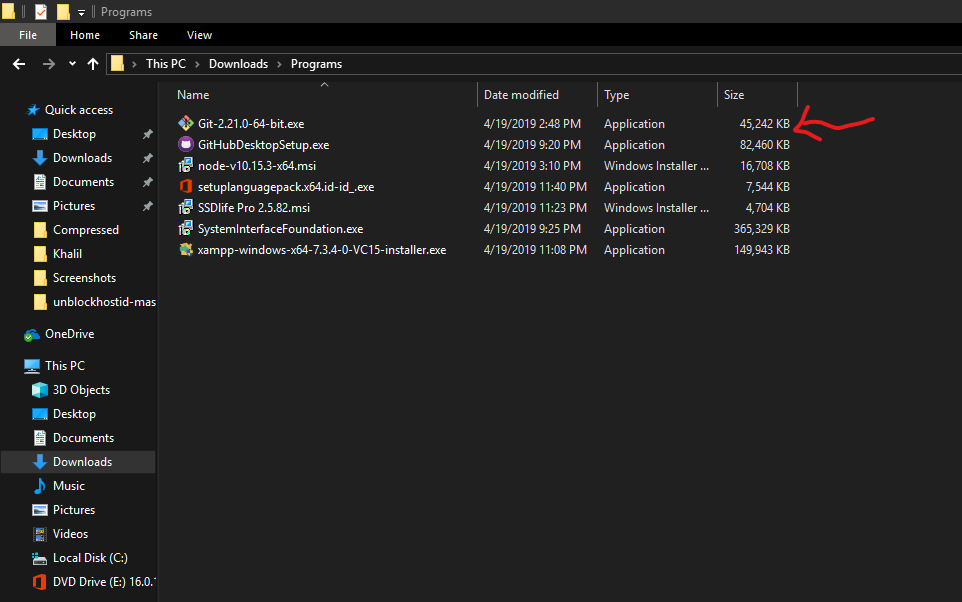
1. **Installation Git**

Download Git from official Website: <https://git-scm.com/>



Picture 3.2 Website Git

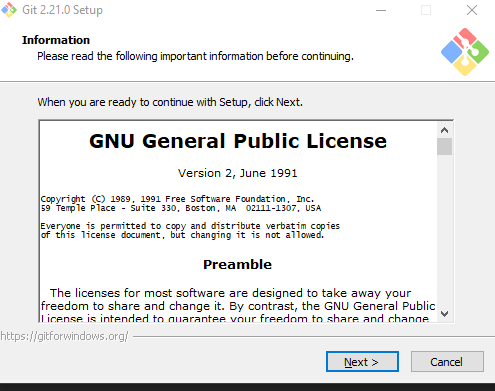
Double Click file to open file for installation git



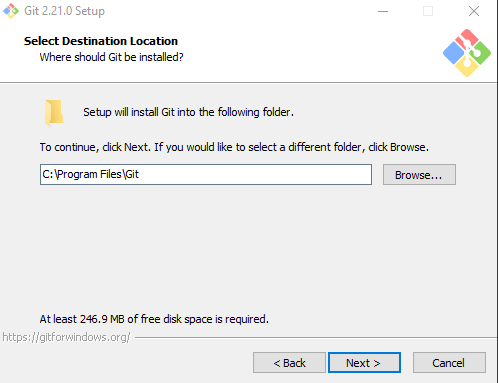
Picture 3.3 Open file for installation

On installer:

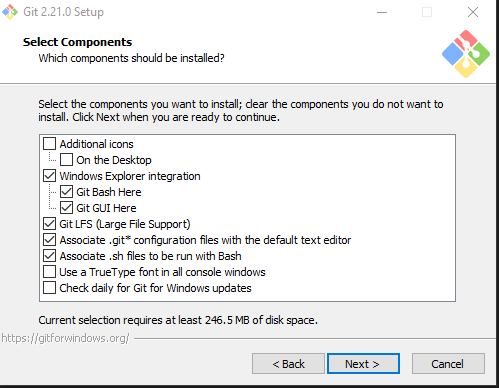
* Picture 3.4: Rule & Term after that click Next
* Picture 3.5: Set-up installation folder for Git after that click Next
* Picture 3.6: Set-up component that we want to install for Git, on this report we leave it default, then click Next
* Picture 3.7: Select Text editor that we want to integrated with Git, on this report, author use Visual Studio Code for default text editor for Git then click Next
* Picture 3.8: Name the shortcut we want to use for git on Start menu folder, on this report author use default name for shortcut, after that click Next
* Picture 3.9: Select one out of three choices for PATH Environment (Command line) on this report author use default (Git from command line and also from 3rd-party software) for choices then click Next
* Picture 3.10: Select one of two choices for HTTPS connection, on this report, author use the default choices for it, then click Next
* Picture 3.11: Select one of three choices for line conversion, we choice



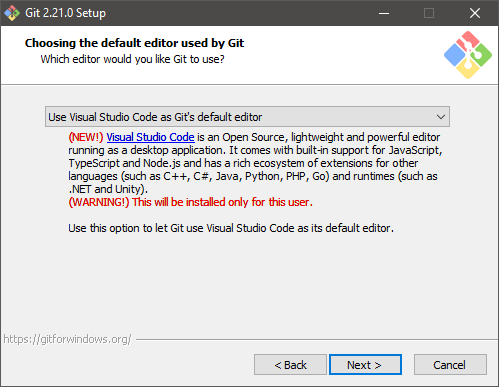
Picture 3.4 Git installer Page 1



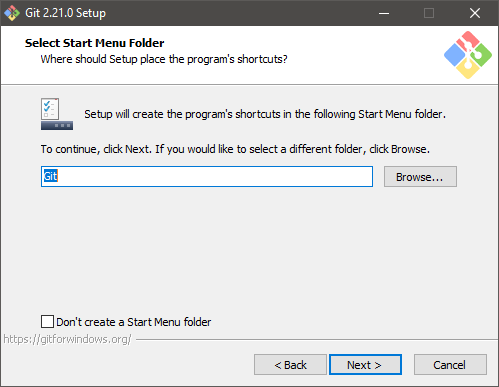
Picture 3.5 Git installer Page 2



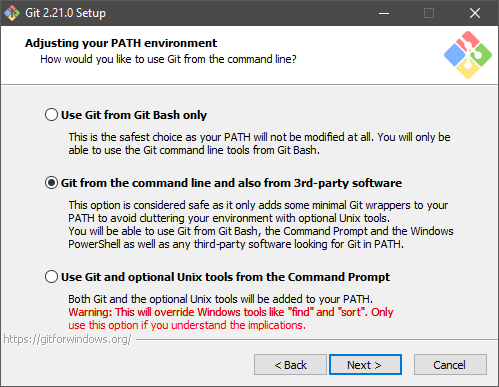
Picture 3.6 Git installer Page 3



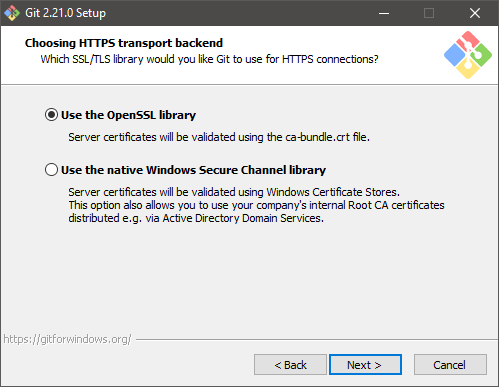
Picture 3.7 Git installer Page 4



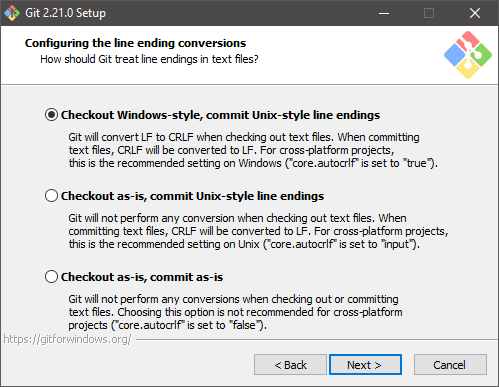
Picture 3.8 Git installer Page 4



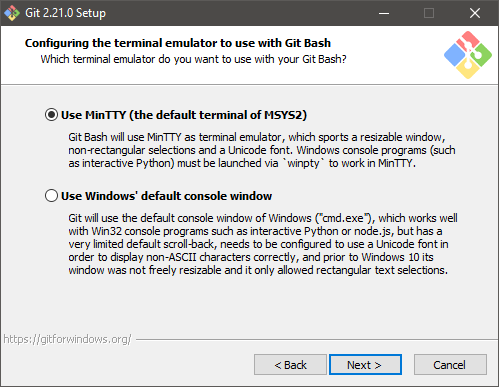
Picture 3.9 Git installer page 5



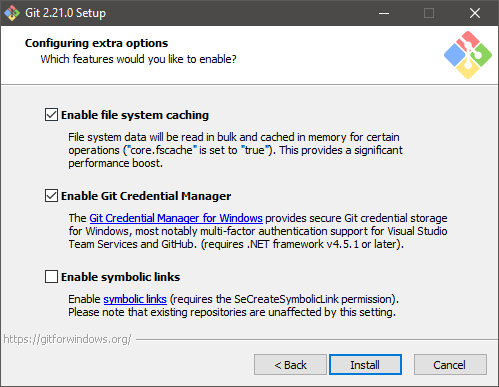
Picture 3.10 Git installer page 6



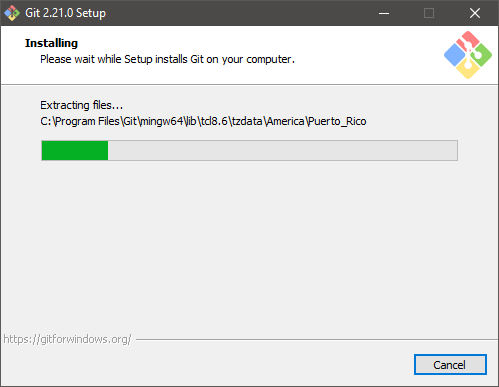
Picture 3.11 Git installer page 7



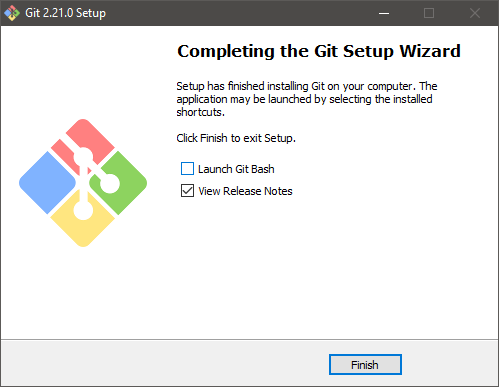
Picture 3.12 Git installer page 8



Picture 3.13 Git installer page 9



Picture 3.14 Git installer page 10



Picture 3.15 Git installer page 11

3. **Git** **Configuration**