**Source Control**

**Questions to companies:**

* What is the current source control server for undergoing projects?
* Is the new project developed on single or multiple platforms?
* How many people will interact with the revision repository of the new project?
* Are developers going to work in one site or different sites?
* Is this position responsible for the setup of source control server?
* What protocol had been applied on existing Git server?
* Are we using internal server or external hosting server like GitHub?
* Is code review needed?
* Why are there **two shell interfaces** for Git, Cygwin & msysGit?

Cygwin is a Linux shell, msysGit is a Windows shell.

* **Cygwin** **VS** **msysGit**?

|  |  |
| --- | --- |
| Cygwin | msysGit |
| **Support Chinese input and display perfectly** | Need specific configuration |
| Command line shell | **Provide visual shell TortoiseGit which integrated msysGit and windows file management** |

* **Traits of distributed revision control system** comparing with centralized revision control system.
* Revision repository is placed under the root of workspace.
* To secure repository, clone the repository and enable auto push.
* Editable username and email are not secure. Server repository can set up authentication (Gerrit).
* Why are **alias** used in Git?

Make it easier for CVS & SVN users to use CVS/SVN alike command in Git.

* Primary commands of **creating repository**

git init / git add / git commit / git config

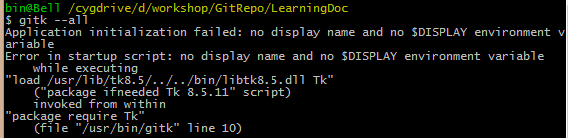
* Benefits of Stage in Git
* **Traits** of Git **repository** implementation
  + Tree structure
  + Pointer reference
  + Stage
  + SHA1 hash
* Why not use **index** to mark submitting in Git?

Submitting is not in sequence in distributed revision control system, only universal unique number can identify submitting occurred at any time in anywhere.

* **Visual user tool** for Git
  + Why not use gitk, gitg, qgit? – They requires Linux or QT environment.
  + Why choose TortoiseGit? – Familiar to SVN users; free SW in windows; easy to port SVN repository to Git.
  + Install **TortoiseGit**
    - Read and follow setup instruction in <http://code.google.com/p/tortoisegit/wiki/SetupHowTo>, prerequisites: Windows Installer and Git for Windows. Check if the computer already installed Windows Installer: run->command window->msiexec.
    - Download installation package in <http://code.google.com/p/tortoisegit/wiki/Download> and execute it.
    - Refer to instruction in Youdao notes.
  + Install **gitk**
    - Much easier way is installing it in the Cygwin installation, just tick the “tcl –tk: Tcl X11 toolkit” in the package list.
    - The other way is downloading ActiveTcl Community Edition at <http://www.activestate.com/downloads#4> which is for windows and free.
    - Run the .exe to perform installation.
    - An error during installation, not sure to what extent it would affect the function, note down it here for future reference:



* Troubleshooting of running **gitk**
  + Run “$ gitk –all” in Cygwin, below error is displayed:



To solve this issue, install “XInit” in Cygwin installation by selecting the “XInit” package. Refer to Youdao notes for details.

* **Protocols** supported in Git and the corresponding URL. SSH, GIT and local are intelligent protocols.



* A bare repository will be initiated under **Cygwin root drive** “C:\cygwin64” if only relative path is specified.



* **Ssh: could not resolve host name**

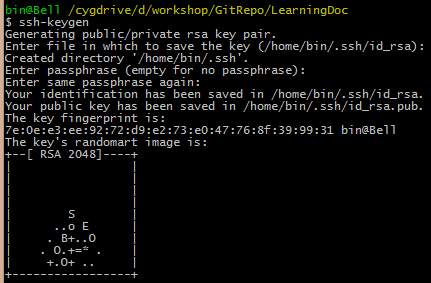


The issue occurred when push updates to remote repository. Reason is Cygwin recognises only paths in Cygwin format, the path of the remote repository was <file:///path/to/repo/shared.git>, later it was changed to C:\cygwin64\path\to\repo\shared.git when I clone repository for user2 in TortoiseGit.

To solve the issue, just specify the path in Cygwin:

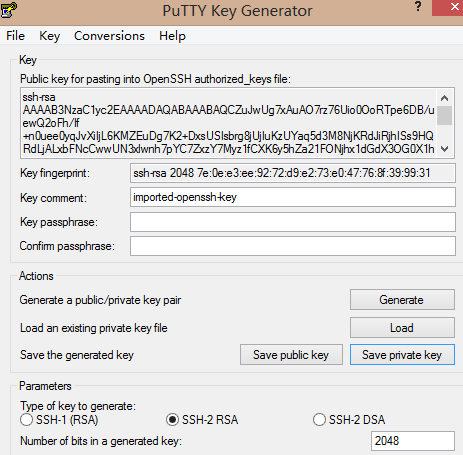


* Disable non-fast-forward: $ git –git-dir=/path/to/repo/shared.git config \ receive.denyNonFastForwards true (failed in trial)
* Source control server management tool **Gitolite**. – to be studied later.
* Create unique entry for users to visit repository server
* Manage users’ access rights to repositories or branches
* Gerrit – code review server constructed on top of source control server
* Download the .war file at <http://gerrit-releases.storage.googleapis.com/index.html>, then unzip it in winrar.
* More info refer to **Chapter 32**.
* **Host** Git project in an external dedicated hosting site
  + Quick to set up and easy to start project on
  + No server maintenance or monitoring
  + Best choice for open source project as people can find it easily
  + Most popular one **GitHub** – support open source projects and also private project
    - <https://github.com>, bexie
    - **Deploy key/cert** in GitHubas soon as account created, only then you can push local repository to GitHub. Refer to next subject of how to generate key/cert.
* Set up **key/certificate** for SSH/HTTP connection to remote Git server
  + **Public/private key pair** for SSH protocol
    - Check if folder “.ssh” exist under C:\cygwin64\home\bin, if not, then go to next step.
    - Generate public/private key pair by executing “$ ssh-keygen” in Cygwin console:



Or directly use PuTTY keygen to generate public/private key pair.

* + - Set PuTTY as SSH client in Git by “export GIT\_SSH=/cygdrive/C/Program\ Files\ \(x86\)/PuTTY/plink.exe”.
    - Launch PuTTY, load the private key generated above and save it as .ppk key.



* + - Launch Pageant from PuTTY folder, execute context menu “add key” in Pageant, load the .ppk private key we saved in above step.
* **Certificate** for HTTP protocol
* Source control tool cooperates with requirement and bug track tool?

Redmine?

**Chapter 5/6/7/8/9/10 to be read thoroughly to understand details of branches.**

**Chapter 7 – Git Reset -- unread.**

**Chapter 8 – Git Checkout – brief read.**

**Chapter 9 – Resume progress – brief read.**