**BÁO CÁO**

TÌM HIỂU PHẦN MÊM QUẢN LÝ NHÓM

SUBVERSION (SVN)

## **Khái niệm**

Hệ thống *SubVersion* (SVN) là một hệ thống quản lý phần *tài nguyên*(code, hình ảnh, video …)của một *dự án*.

Hệ thống có khả năng cập nhật, so sánh và kết hợp tài nguyên mới vào phần tài nguyên cũ, được giới thiệu vào năm 2000 bởi công ty CollabNet.

Đây là hệ thống hỗ trợ làm việc theo nhóm rất hiệu quả.

## **Chức năng**

SVN hoạt động theo phương thức *Client/Server*, code project sẽ được lưu trữ trên server (SVN hosting, GoogleCode, …). Các client có thể thao tác, edit trực tiếp trên đó, mọi thay đổi của từng client sẽ được lưu lại. Do đó SVN là một giải pháp làm việc nhóm từ xa hiệu quả, an toàn.

Khi một nhóm làm việc trên cùng một project, việc nhiều người cùng chỉnh sửa nội dung của một file là điều không thể tránh khỏi. SVN cung cấp các chức năng để có thể thực hiện việc này một cách đơn giản và an toàn.

* Subversion là hệ thống quản lý Source Code tập trung (Centralized).
* Subversion là hệ thống quản lý phiên bản mạnh mẽ, hữu dụng, và linh hoạt.
* Subversion quản lý tập tin và thư mục theo thời gian.

SVN giống như một hệ thống file server mà các Client có thể Download và Upload file một cách bình thường.

Điểm *đặc biệt* của SVN là nó lưu lại tất cả những gì thay đổi trên hệ thống file: file nào đã bị thay đổi lúc nào, thay đổi như thế nào và ai đã thay đổi nó.

SVN cũng cho phép *recover* lại những version cũ một cách chính xác.

Thông thường, *client* và *server* kết nối thông qua mạng LAN hoặc Internet. Client và server có thể cùng chạy trên một máy trong trường hợp SVN có nhiệm vụ theo vết lịch sử của dự án, điều này do các nhà phát triển phần mềm phát triển trong nội bộ.

Subversion hỗ trợ khá nhiều giao thức để kết nối giữa client và server.

Việc cho phép server hỗ trợ giao thức nào phụ thuộc vào lúc cấu hình.

## **Một số khái niệm trong SVN**

**Reposiroty**: Nó là khu vực trung tâm để các lập trình viên giữ tất cả công việc của họ. Kho lưu không chỉ lưu các file mà còn lưu giữ cả lịch sử làm việc. Kho lưu được truy cập thông qua mạng, thực hiện vai trò như một máy chủ và công cụ điều khiển phiên bản là một máy khách. Các máy khách có thể kết nối với repository và sau đó họ có thể giữ/lấy lại những thay đổi họ thực hiện tới/từ repository.

**HEAD**: Phiên bản mới nhất trên server.

**Master**: Thông thường khi tạo SVN, có một branchchính được tạo ra và gọi là *master*, các *branch*phụ thường được đặt theo tên feature mà branchđó được tạo.

**Change**: Mô tả sự thay đổi cụ thể của một revisionso với revisiontrước đó.

**Working** **copy**: Bản copy của toàn bộ SVN tại máy của developer.

**Conflict**: Khi bạn cập nhật/sáp nhập các tập tin của bạn từ kho hoặc khi bạn chuyển bản sao của bạn làm việc đến một URL khác nhau.

Có hai loại xung đột:

* *Xung đột tập tin:* Sự xung đột giữa các tập tin xảy ra nếu có hai (hoặc nhiều hơn) nhà phát triển đã thay đổi một vài dòng lệnh của tập tin.
* *Các xung đột cây:* Sự xung đột giữa các cây xảy ra khi một nhà phát triển di chuyển/đổi tên/xóa một tập tin hoặc thư mục mà một nhà phát triển khác hoặc cũng đã di chuyển/đổi tên/xóa hoặc chỉ sửa đổi.

**Checkout**là hành động lấy toàn bộ dữ liệu được chia sẻ trên **Repository**xuống máy địa phương lần đầu tiên, sau đó dữ liệu được sửa đổi tại địa phương. Việc đưa dữ liệu đã thay đổi lên **Repository**được gọi là **Commit**, và việc lấy dữ liệu đã thay đổi về là hành động **Update**.

**Update**: Khi một thành viên muốn cập nhật những thay đổi của các file thuộc thành viên khác trên Repository về máy cục bộ của mình.

**Merge**: Nhiều thành viên cùng tiến hành cập nhật trên một tập tin.

**Revert**: Sau khi tiến hành thay đổi, developer cảm thấy họ đã sai và muốn xóa đi làm lại từ đầu, việc revertlà hành động để khôi phục trạng thái của một hay nhiều document về một revisionnào đó, thường thì sẽ revertvề revisionhiện tại mà user đang làm việc.

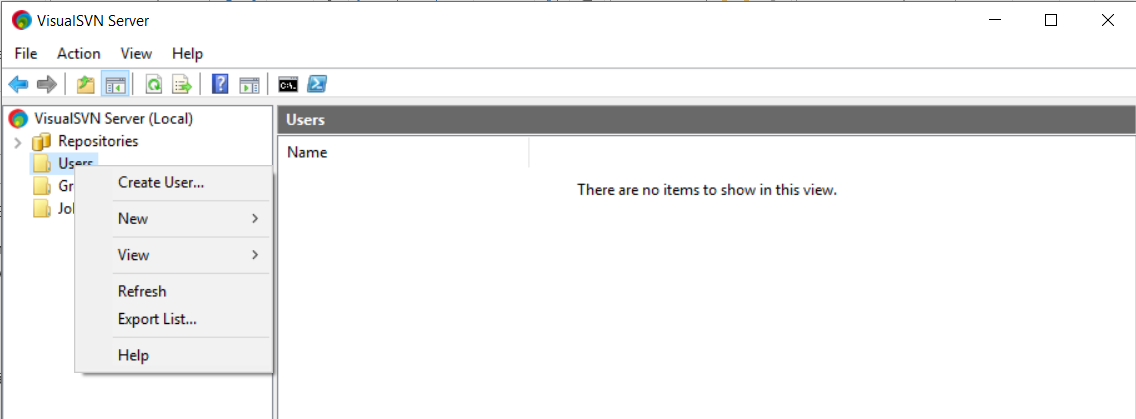
## **Cài đặt**

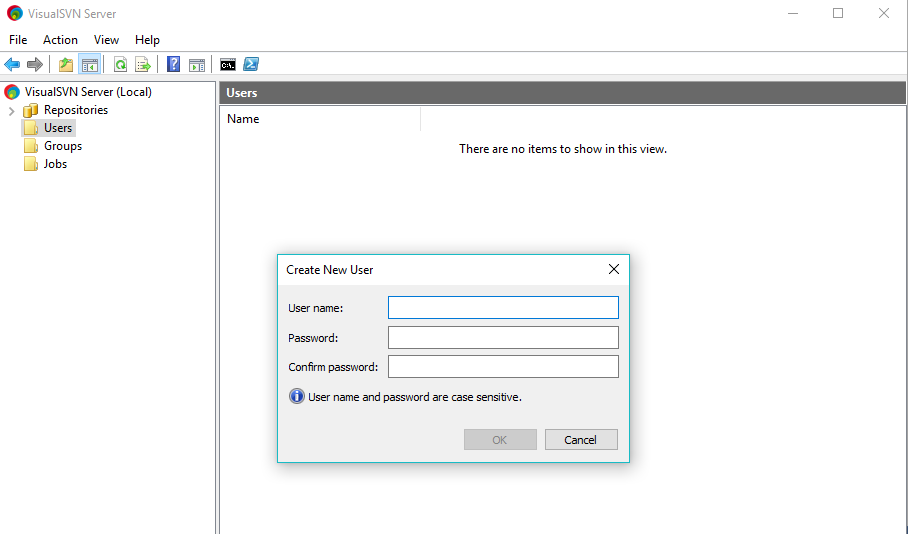
* Download và cài đặt:

**Tortoisesvn:** <https://tortoisesvn.net/downloads.html>

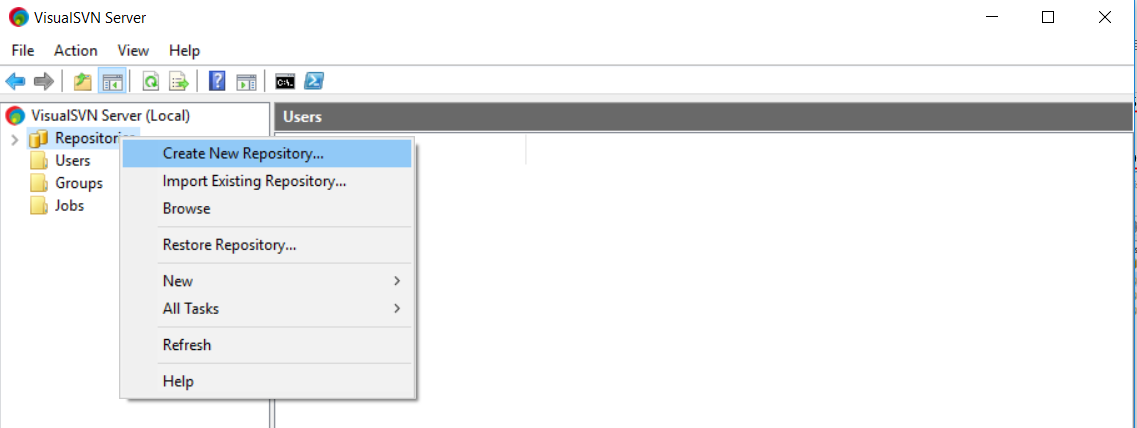
**Visualsvn:** <https://www.visualsvn.com/server/download>

* Tạo tài khoản cho người sử dụng: username và password

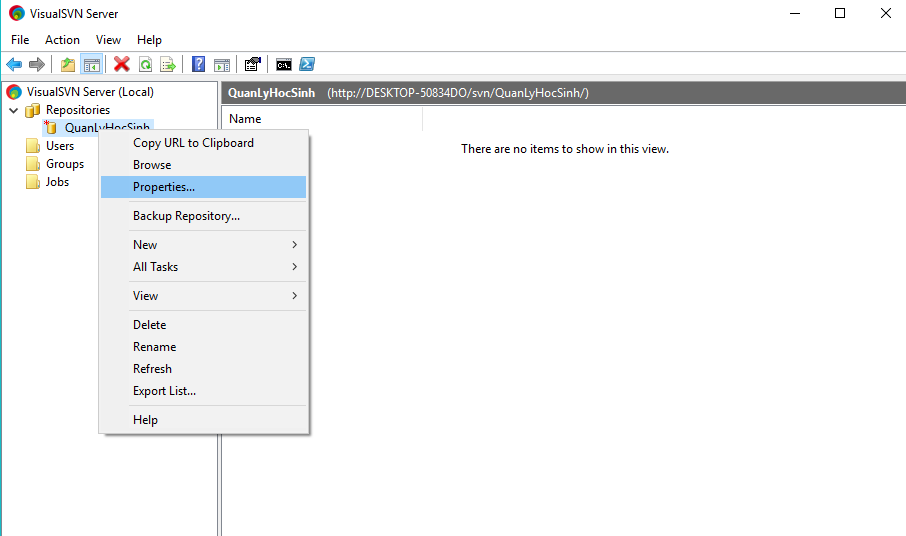


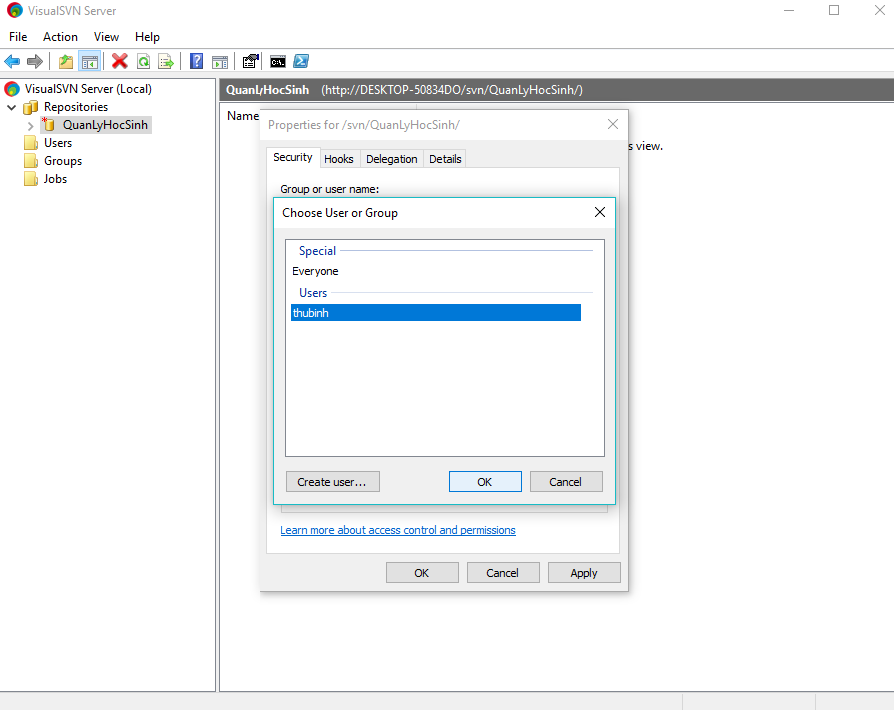
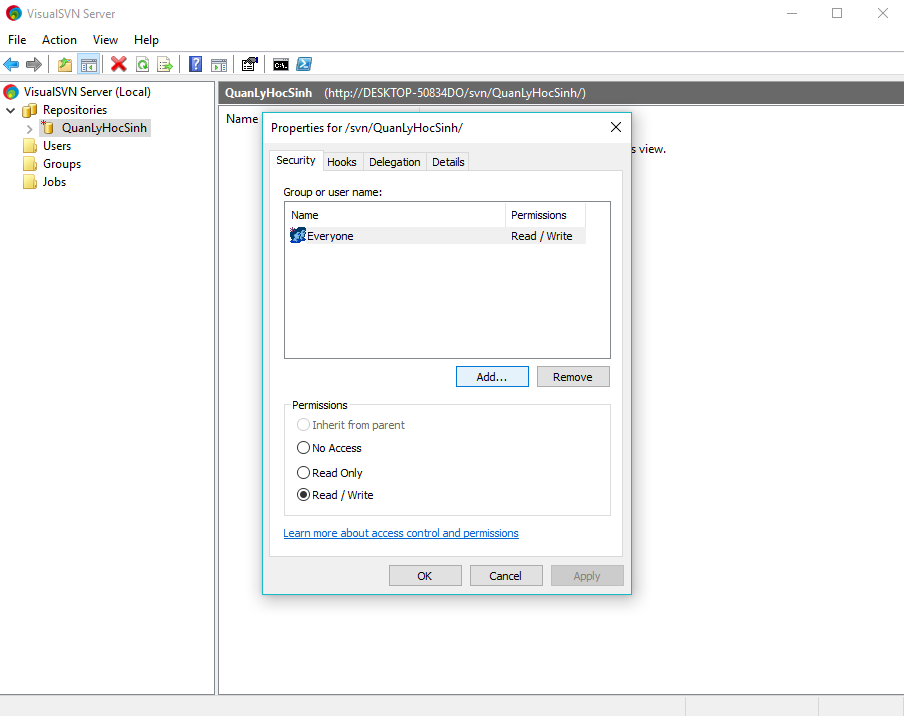


* Tạo 1 repository:



* Tạo user: Dùng để kiểm soát giới hạn quyền cho các thành viên trong dự án. Nhấp chuột phải vào **Repository🡪 Properties.** Nhấn Add để thêm user có quyền truy cập vào **Repository** này. Chọn quyền truy cập (Permissions): No Access hoặc Read Only hoặc Read/Write.

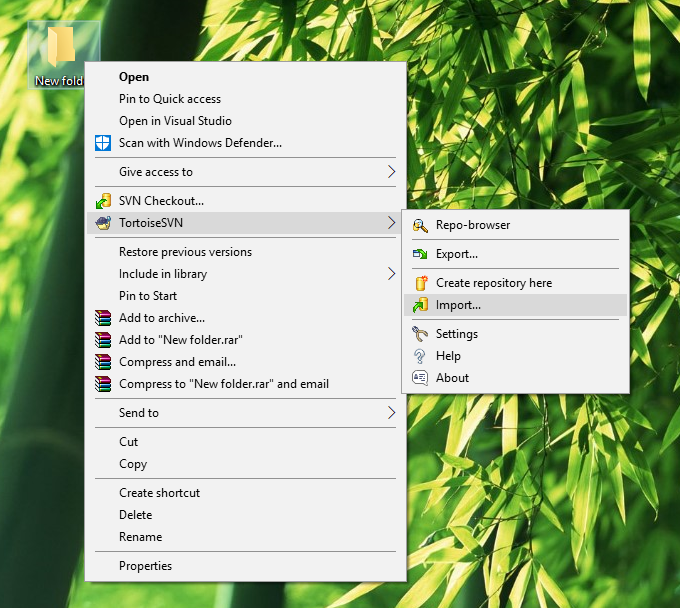




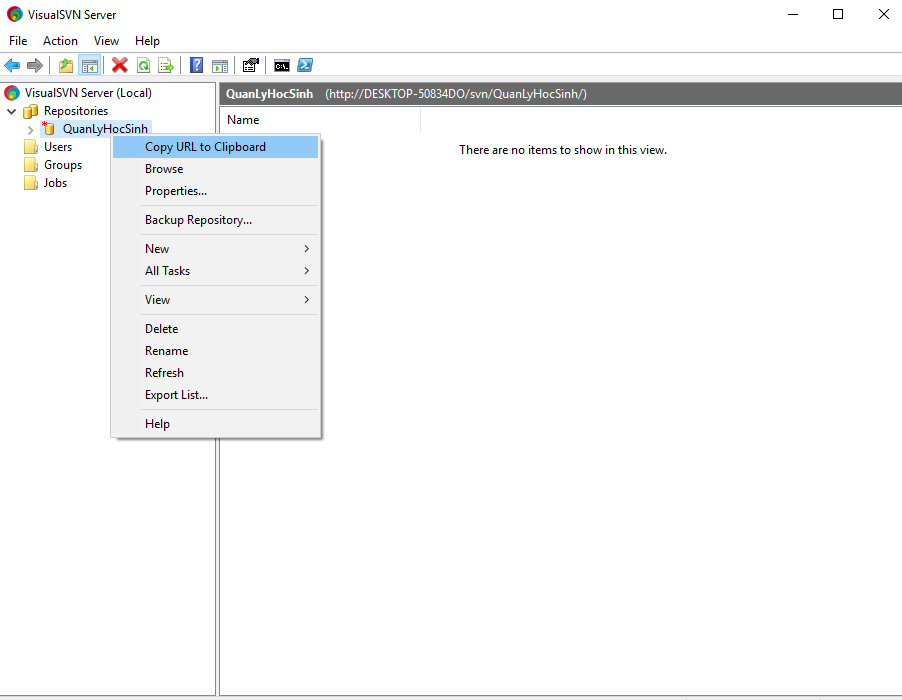
## **Cách lưu trữ dữ liệu trên SVN, upload dữ liệu và download dữ liệu về từ Server**

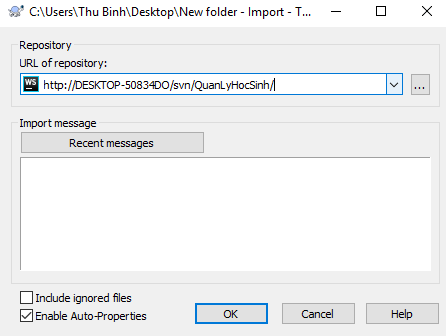
### Import dữ liệu lên server:

* Tại thư mục muốn đẩy dữ liệu lên server nhấp chuột phải chọn TortoiseSVN 🡪 Import…

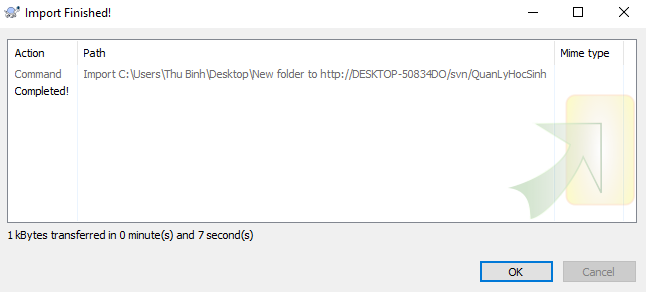


* Copy URL paste vào yêu cầu nhập đường dẫn của SVN Repository.



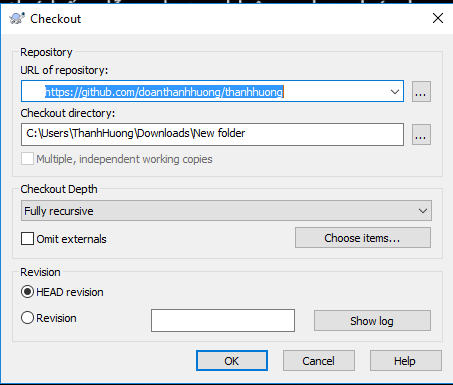


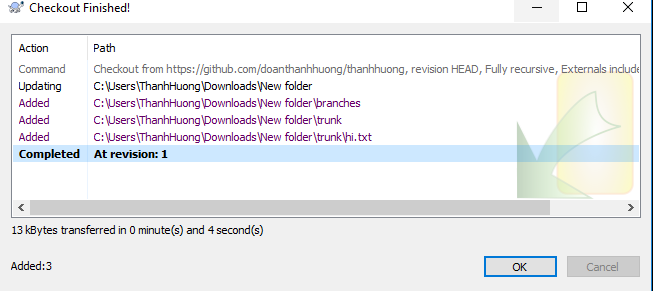
* Import hoàn thành :



### Sử dụng

* **Checkout** source từ github:
* Tạo một folder để lưu trữ.
* Mở github, lấy đường link của dự án.
* Right click lên folder, chọn SVN checkout.



Quá trình checkout thành công.

* Xem sự thay đổi:

Mở file ra, thêm 1 số nội dung vào, ta thấy sự thay đổi: file chyển sang có dấu chấm màu đỏ. Ngoài folder cũng hiện dấu chấm màu đỏ.

* Commit:

Right click chọn SVN commit.Sau khi commit thành công, file đã thay đổi, là file commit lên GITHUB.

