

# Subversion(SVN)

## 1. Giới thiệu

**Subversion** là một phần mềm mã nguồn mở dùng để quản lý và kiểm tra các phiên bản mã nguồn khác nhau trong quá trình phát triển phần mềm. Subversion cũng còn được gọi là **SVN**. Svn là lệnh dùng để thực hiện các chức năng Subversion trong môi trường thi hành lệnh trên các máy vi tính. Subversion được thiết kế với mục đích thay thế phần mềm quản lý phiên bản *Concurrent Versioning System* (**CVS**) đã cũ và có nhiều nhược điểm. Một số các lập trình viên chính cho dự án CVS hiện đang tham gia phát triển Subversion.

Subversion rất phổ biến trong giới mã nguồn mở và được dùng để quản lý mã nguồn cho nhiều ứng dụng tự do như: [Apache Software Foundation](#), [KDE](#), [GNOME](#), [Free Pascal](#), [FreeBSD](#), [GCC](#), [Python](#), [Django](#), [Ruby](#), và [Mono](#). [SourceForge.net](#) và [Tigris.org](#), nơi chứa miễn phí chuyên nghiệp dành cho các ứng dụng tự do, đều có hỗ trợ Subversion cho các ứng dụng được lưu trữ trong máy chủ. Ngoài ra [Google Code](#) và [BountySource](#) chỉ sử dụng Subversion để quản lý các phiên bản mã nguồn.

Subversion được phát hành với giấy phép [Apache License](#). Do đó Subversion là một [phần mềm nguồn mở](#)(FOSS) và tự do.

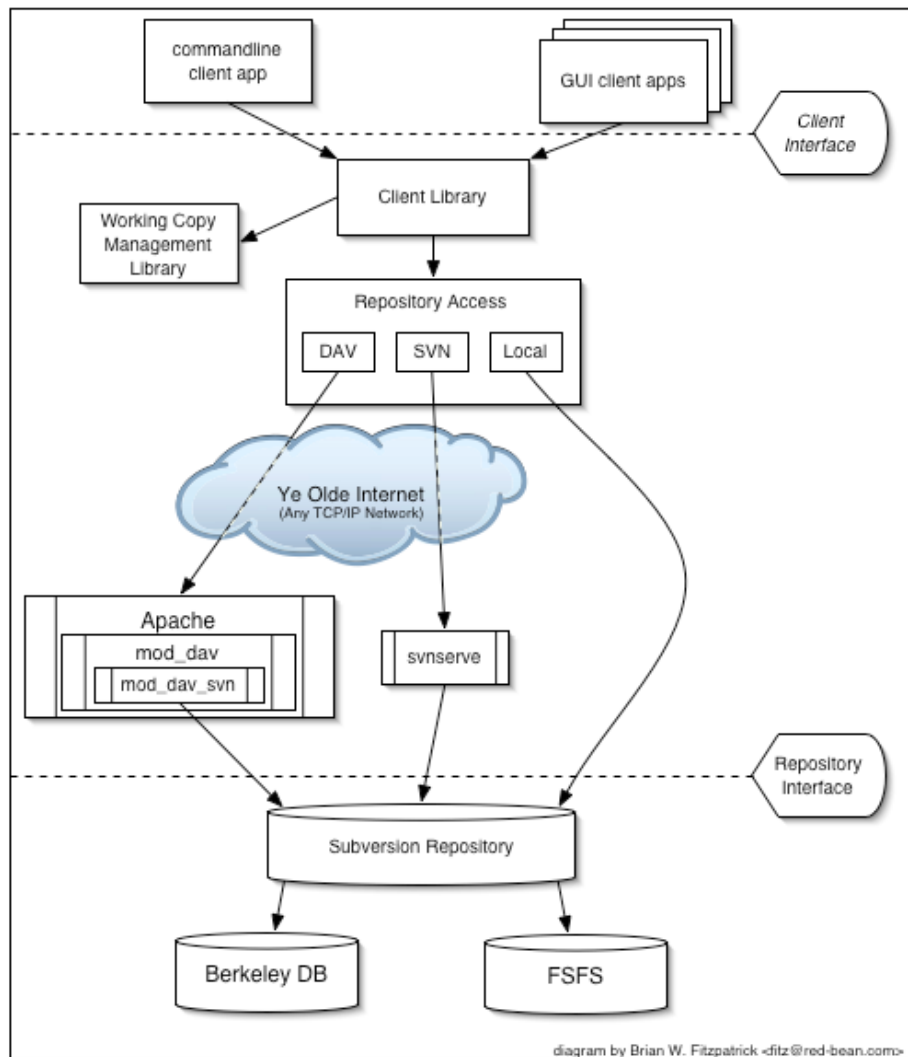
(Theo [Wikipedia](#))

Trang chủ của dự án Subversion: <http://subversion.tigris.org/>

Tại đây bạn có thể download bộ cài, mã nguồn của SVN dành cho các hệ điều hành khác nhau(MS Windows, Linux, MAC OS,...). Ngoài ra tại đây cũng cung cấp các phần mềm dùng cho client(TortoiseVN, eSVN, KDESvn,...) để làm việc với SVN và nhiều hỗ trợ khác: Book, Help, Report a Problem, Development, ...

## 2. SVN Server

### a. SVN Diagram



(Nguồn: <http://subversion.tigris.org>)

### b. Cấu hình và cài đặt

- i. [Trên Windows](#)
- ii. [Trên Ubuntu](#)

## 3. SVN Clients

Dưới đây là một số các chương trình khách hay giao diện người dùng có khả năng truy cập các máy chủ chạy Subversion.

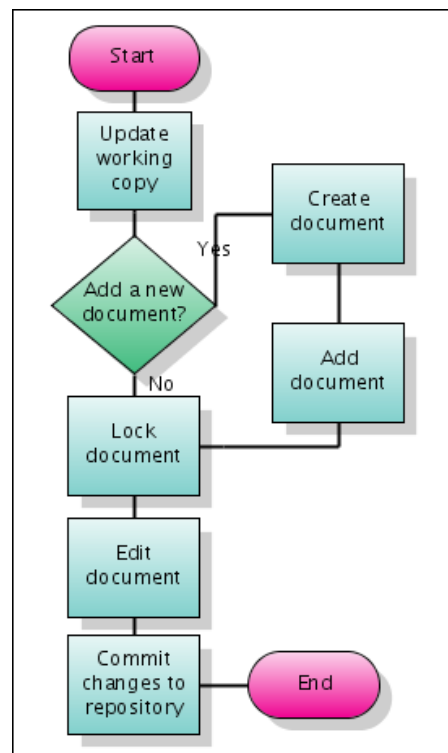
- [AnkhSVN](#) là một phần mềm phụ, bổ sung khả năng truy cập Subversion cho môi trường lập trình Visual Studio .NET.
- [eSvn](#), chương trình khách dựa trên thư viện Qt.
- [Insurrection](#), giao diện Web cho Subversion.
- [JSVN](#), chương trình khách viết bằng Java.
- [kdesvn](#), chương trình khách cho [KDE](#).
- [psvn.el](#), giao diện Subversion cho [Emacs](#).
- [PushOk SVN SCC PlugIn](#) là một phần mềm phụ, bổ sung khả năng truy cập Subversion cho môi trường lập trình Visual Studio .NET.
- [RapidSVN](#), chương trình khác đa hệ điều hành dựa trên thư viện [wxWidgets](#).
- [SCPlugin](#), phần mềm phụ cho [Macintosh Finder](#).
- [SmartSVN](#) là một chương trình khác đa hệ điều hành, có hai bản quyền: tự do và thương mại, được viết bằng Java.
- [Subcommander](#), chương trình khách đa hệ điều hành dựa trên thư viện Qt
- [svnX](#), chương trình khách cho Mac OS
- [TortoiseSVN](#) là một phần mềm làm việc trên nền [Windows Explorer](#).
- [TurtleSVN](#) đi kèm với TortoiseSVN để hỗ trợ môi trường lập trình Visual Studio .NET.
- [WebSVN](#), là một giao diện Web cho Subversion, viết bằng [PHP](#)
- [WebClient for SVN](#) là một chương trình khách mã nguồn mở viết bằng Java/JSP, thường được dùng cho các thao tác đơn giản.
- [ZigVersion](#) là một chương trình khác thương mại cho [Mac OS X](#) Subversion.
- [sventon](#) Web GUI

(Theo [WikipediA](#))

**[TortoiseSVN](#) đang được chúng ta sử dụng cho dự án fRun vì đây là SVN Client chạy trên nền MS Windows.**

## 4. Học cách sử dụng SVN với TortoiseSVN

TortoiseSVN là một chương trình phía client của Subversion system. Chương trình có thể download miễn phí:



<http://jaist.dl.sourceforge.net/sourceforge/tortoisesvn/TortoiseSVN-1.4.8.12137-win32-svn-1.4.6.msi>

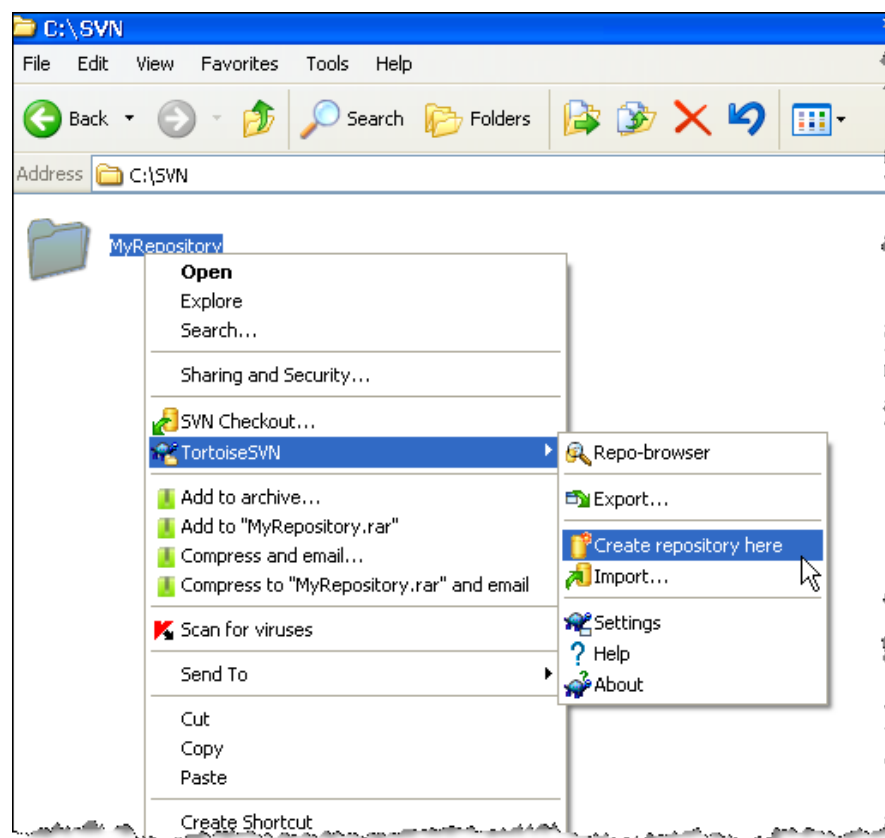
Sau khi cài đặt xong, các bạn có thể dùng thử TortoiseSVN theo hướng dẫn dưới đây

Đây chỉ là hướng dẫn để làm quen với khái niệm và các thao tác trên TortoiseSVN. Cách setup một SVN server thực sự và các chức năng cụ thể của TortoiseSVN xin tham khảo file help trong thư mục cài đặt. **Để đơn giản, bài viết sẽ tạo cả server lẫn client trên cùng một máy tính.**

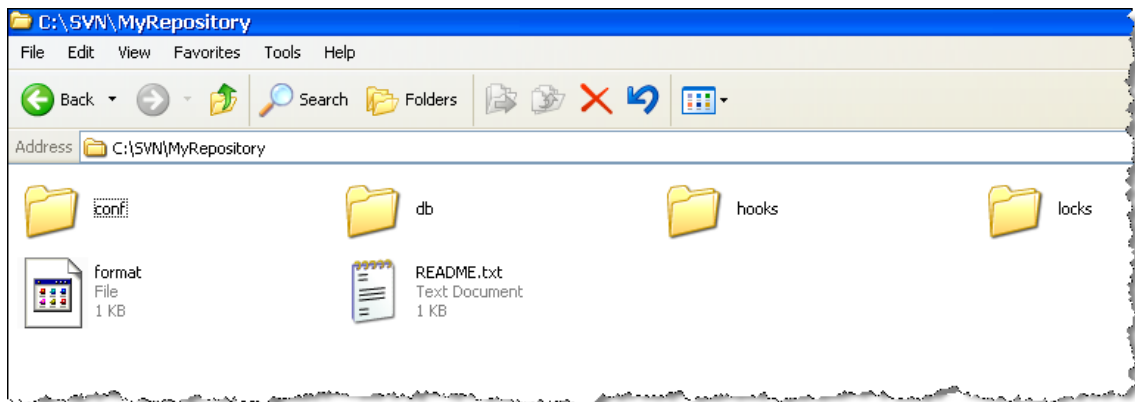
**a)** Tạo thư mục Repository (*thư mục để chứa các file trên server*)

Tạo một thư mục tên bất kì tại nơi bạn muốn đặt server. Ví dụ: C:/SVN/MyRepository.

Click vào thư mục vừa tạo và click chuột phải, chọn TortoiseSVN→Create Repository.



Click OK ở bước tiếp theo

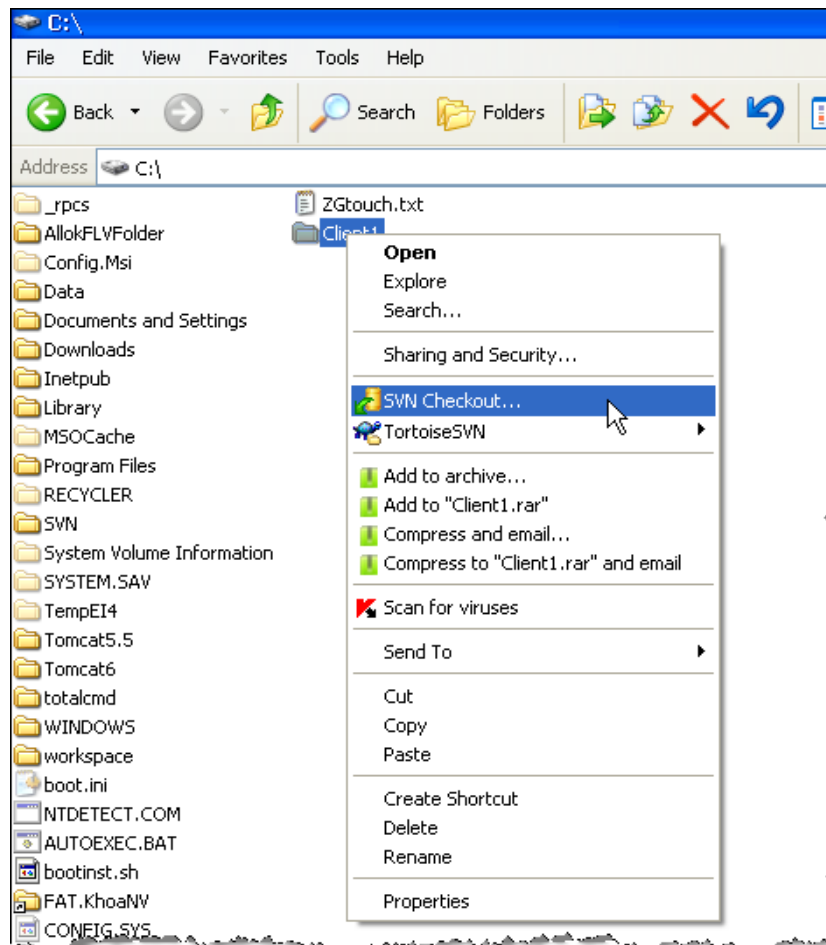


Repository đã được tạo xong. Bạn không nên(và không được) tự mình chỉnh sửa những file trong đó.

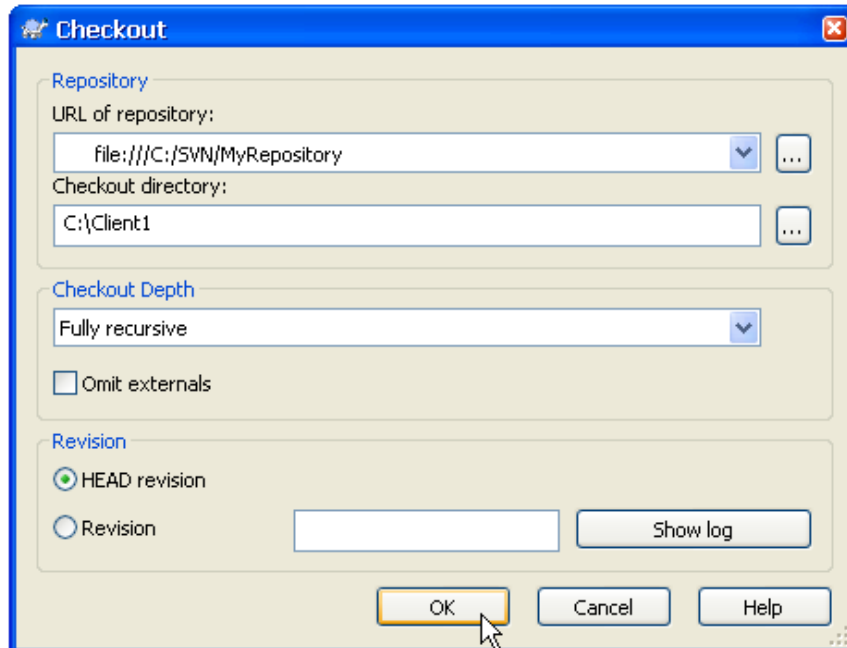
### ***b)*** Access vào Repository như một client

Tạo một thư mục bất kì và di chuyển vào đó(ví dụ C:\Client1)

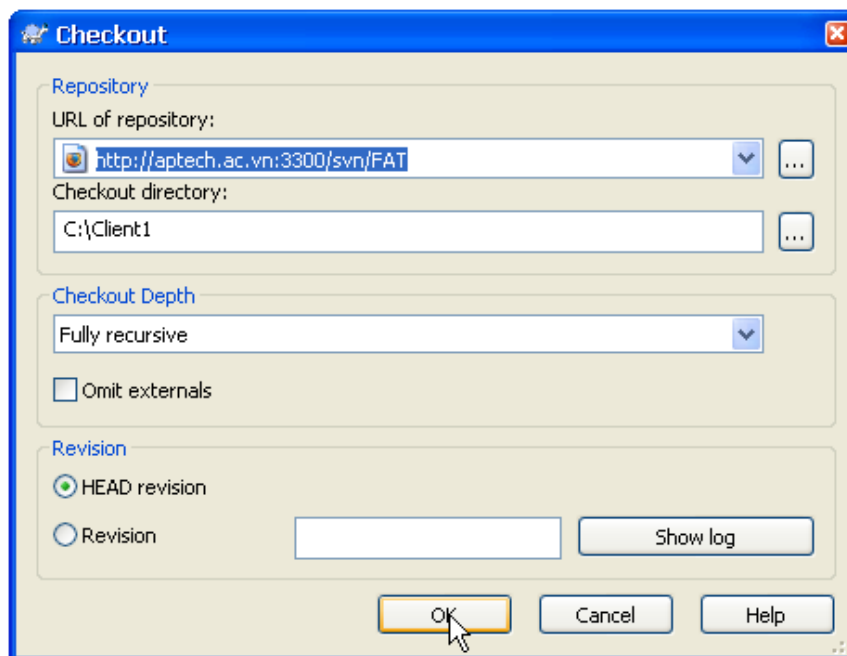
Click chuột phải, chọn SVN Checkout...



Nhập URL repository trên server(trong trường hợp này là <file:///C:/SVN/MyRepository>); nếu bạn có sẵn respo trên mạng bạn điền thông tin đầy đủ vào VD: <http://aptech.ac.vn:3300/svn/FAT> , click OK.

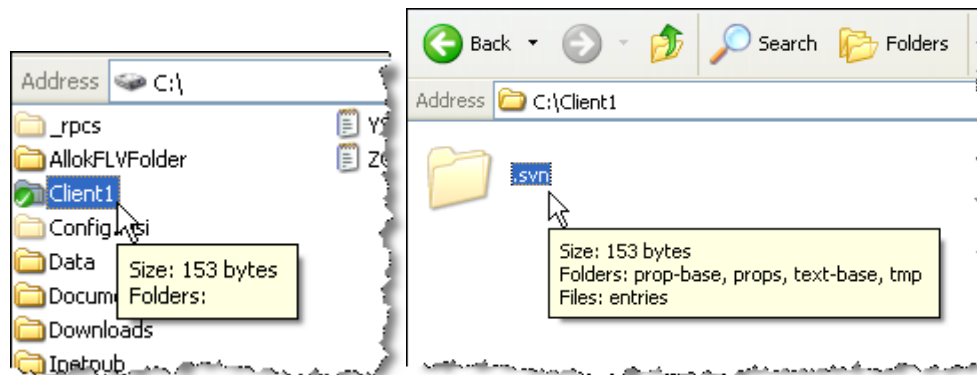


Nếu Respo từ xa ta sẽ có như sau:



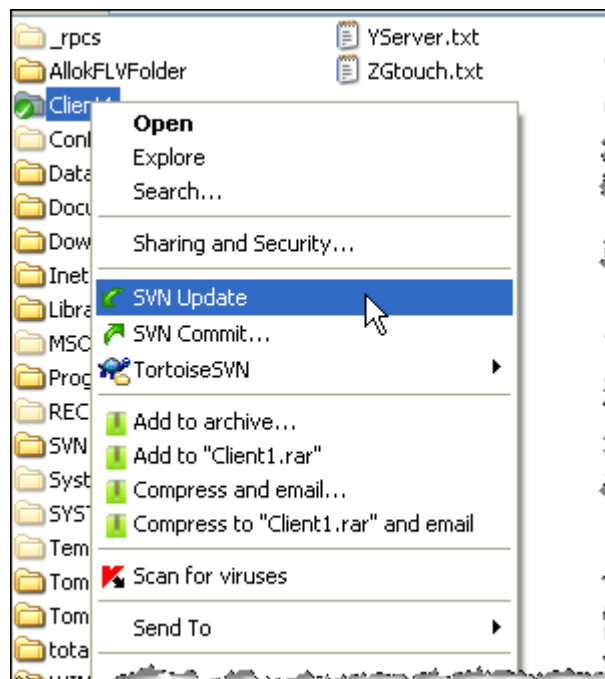


Nếu check out thành công, bạn sẽ thấy trong thư mục xuất hiện thư mục .svn(thư mục ẩn) cùng tất cả các file có trên server (hiện tại trên server chưa có file nào nên chỉ có thư mục .svn xuất hiện). Bạn có thể chỉnh sửa cũng như tạo mới, xóa bớt các file trong đó.



### c) Update từ SVN server

Để lấy các tài nguyên trên SVN server bạn bấm chuột phải vào thư mục Client1 đã checkout thành công ở trên và chọn Update:

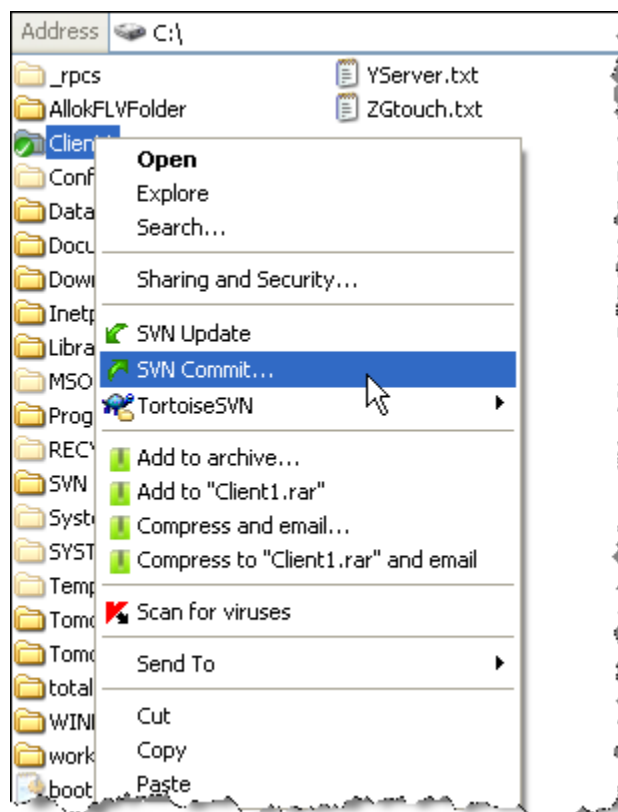


Nếu respo trên Server có thay đổi bạn sẽ có thêm các thông tin mới được cập nhật trong thư mục Client1:

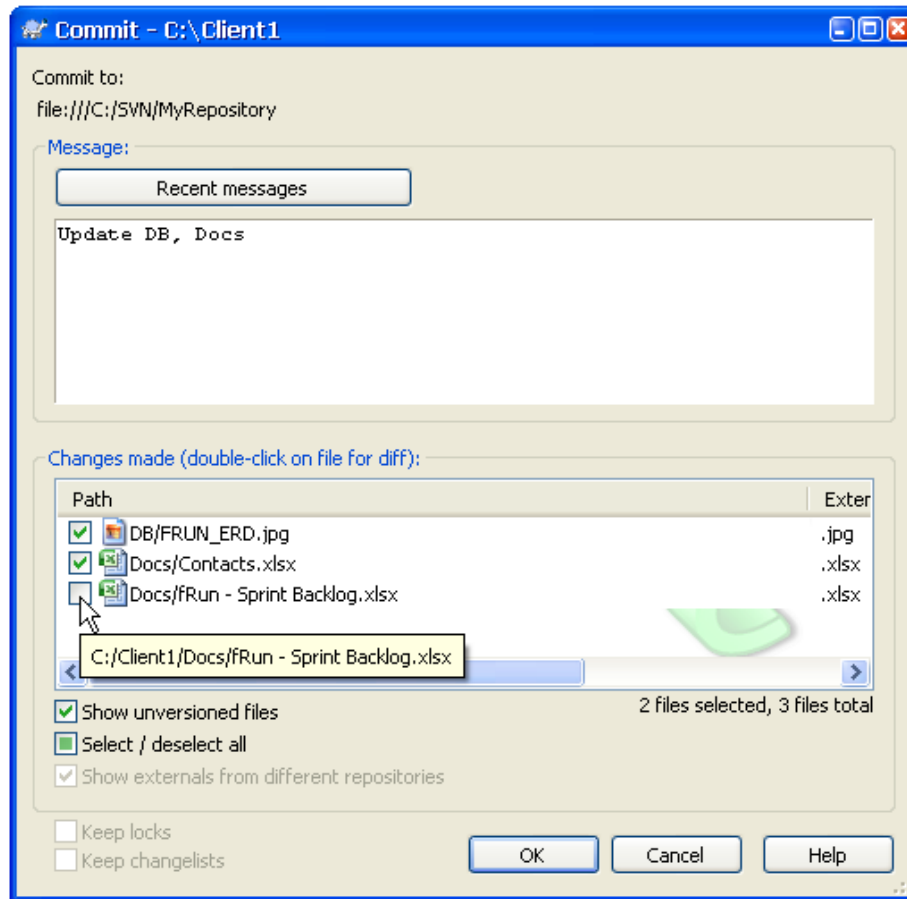


#### d) Upload file lên server (commit)

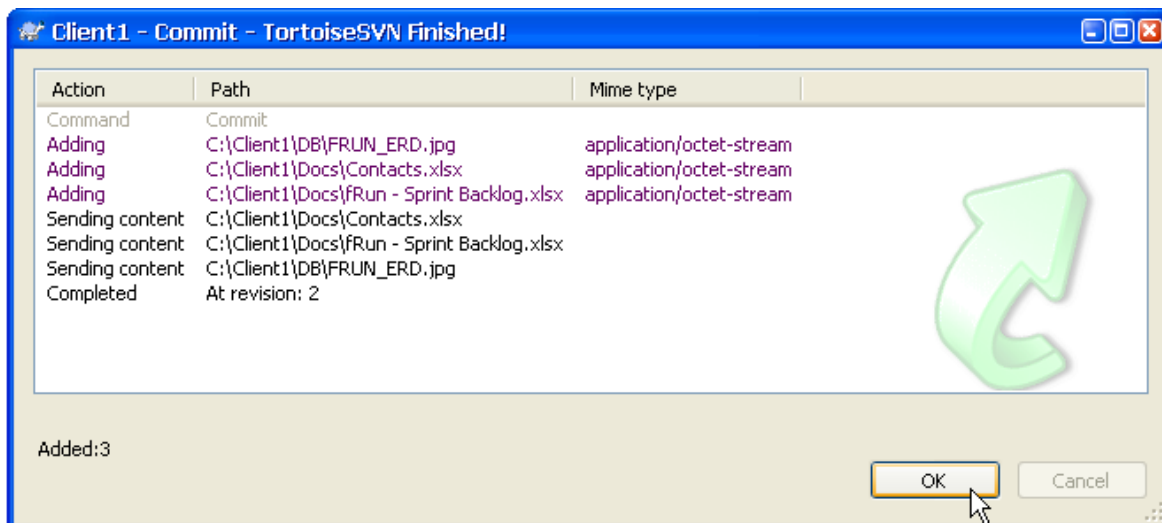
Sau khi đã chỉnh sửa file, các bạn có thể commit những thay đổi lên server bằng cách click chuột phải, chọn SVN commit..., check những file cần commit, viết thêm ghi chú về thay đổi (nếu cần) vào phần log message rồi click ok. Ở bước này các bạn chú ý là chỉ nên commit những file chung cho project, những file chỉ sử dụng cho riêng mình (ví dụ các file test, backup...) các bạn không nên commit.



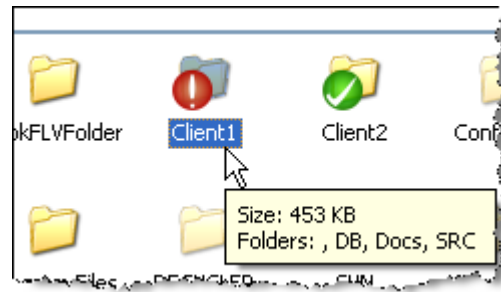
Các thông tin thay đổi trong thư mục Client1 sẽ được thông báo để bạn chọn và commit:



Thông tin nhận được sau khi commit:



Biểu tượng thông báo có thay đổi, cần commit lên server:



### e) Xử lý khi bị conflict

Conflict xảy ra khi 2 client đồng thời thay đổi nội dung của của 1 file. Để hiểu rõ hơn về conflict, các bạn làm theo ví dụ sau:

Trong thư mục Client1/SRC, check out, sau đó tạo file bất kì (ví dụ test.java).

Commit file test.java.

Edit file test.java, thêm vào dòng “public static void main...”. Save file. (chưa commit)

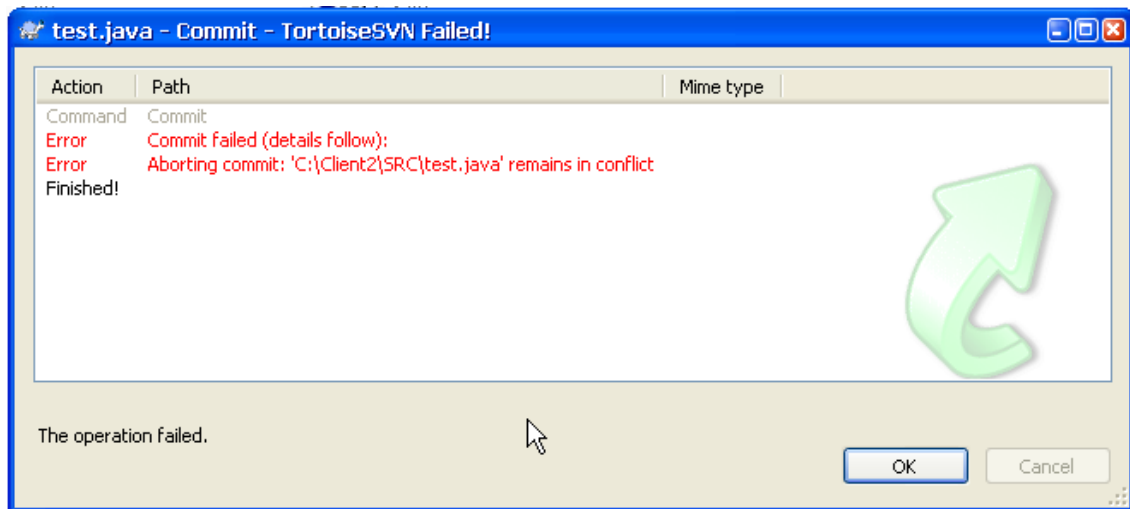
Ra khỏi thư mục Client1, tạo thư mục Client2 (giả định đây là 1 client khác cùng tham gia 1 project với Client1). Check out as Client2.

Edit file test.java, thêm vào dòng “public test()...”.

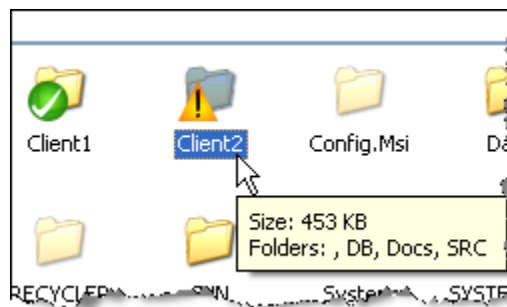
Trở lại thư mục Client1. Commit file test.java.

Trở lại thư mục Client2. Commit file test.java.

Bảng hiển thị lỗi sẽ hiện ra và bạn sẽ không thể commit lên server.

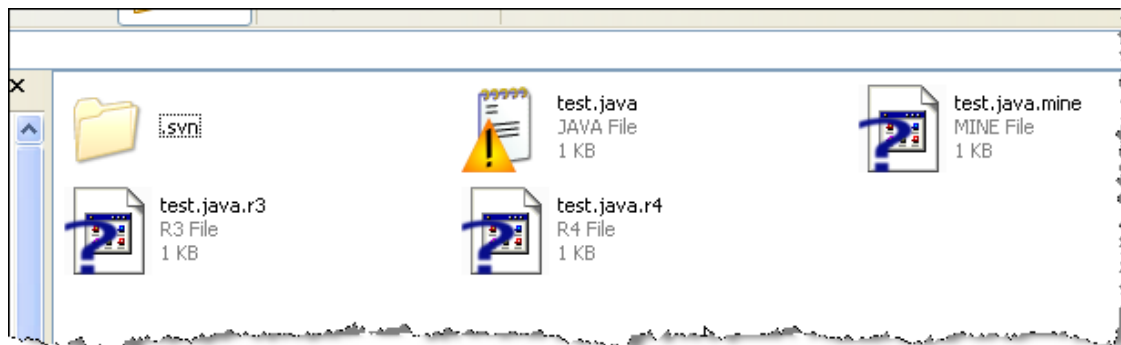


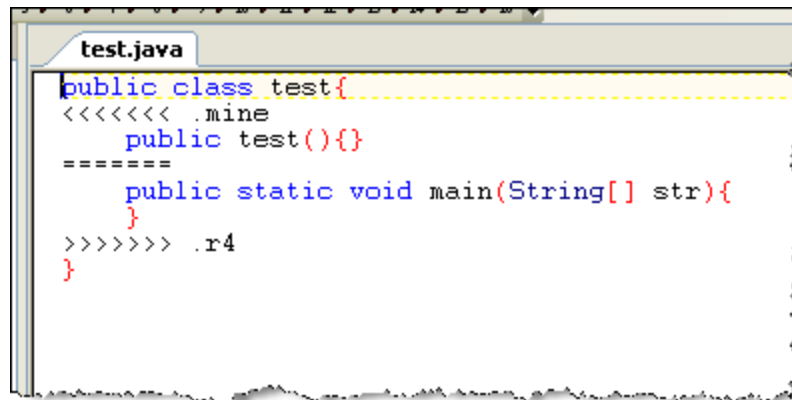
Biểu tượng như sau thông báo về conflict:



**Cách khắc phục:**

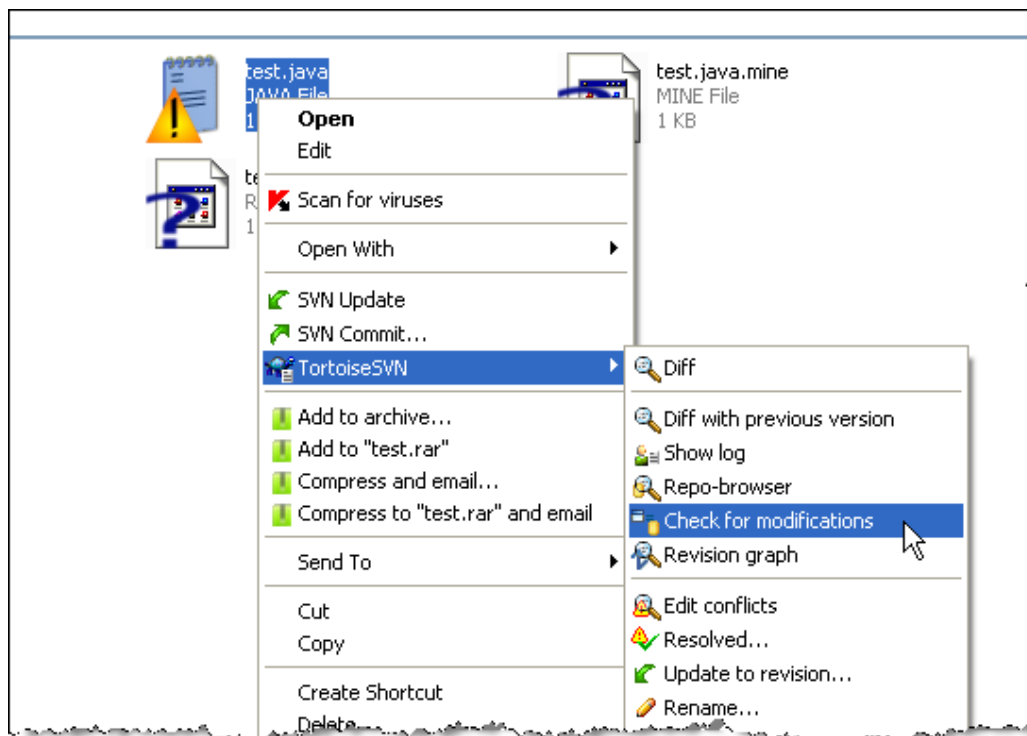
Trước tiên bạn phải update. Click chuột phải, chọn SVN update. File test.java của bạn sẽ được copy qua test.java .mine, còn file test.java sẽ có nội dung mà cả 2 client đã thay đổi.



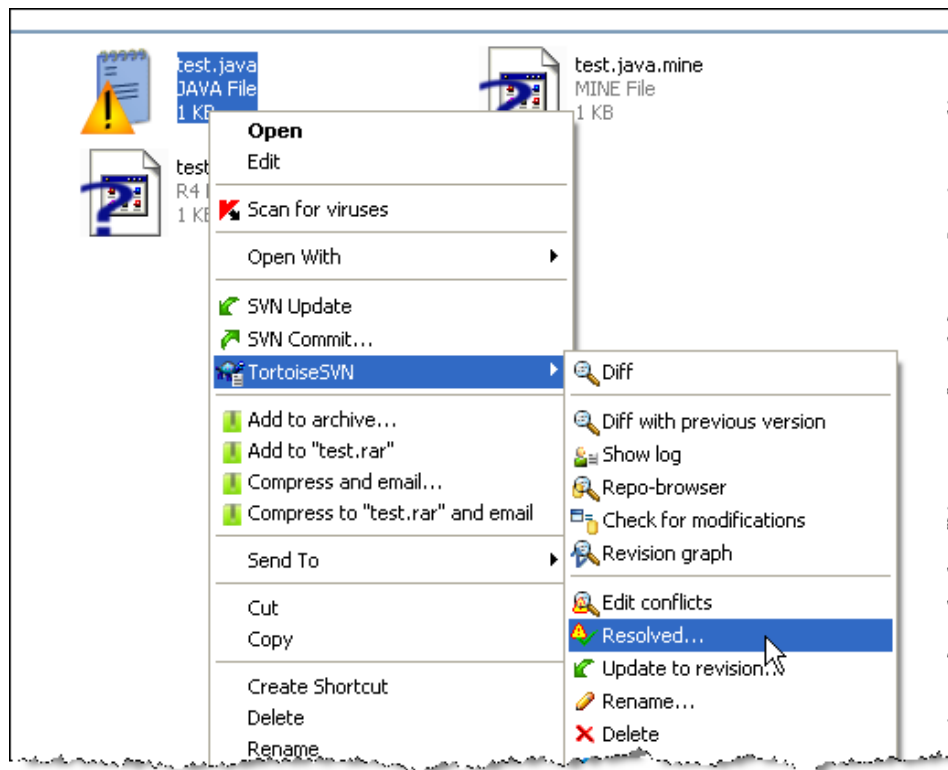


```
test.java
public class test{
<<<<<<< .mine
    public test(){
=====
    public static void main(String[] str){
    }
>>>>>>> .r4
}
```

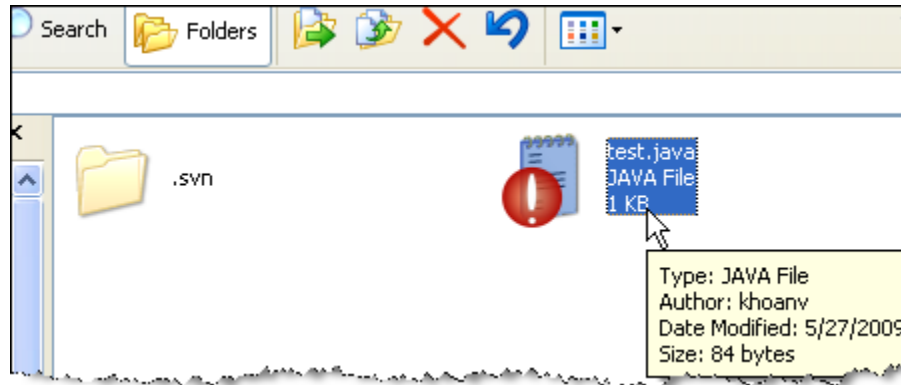
Xem file test.java để “đoán” xem chuyện gì đã xảy ra và chỉnh sửa lại. Hoặc bạn có thể dùng check modification tool(click chuột phải, chọn TortoiseSVN→Check for modification).



Sau khi đã chỉnh sửa xong file test.java, bạn phải thông báo cho SVN rằng conflict đã được fix bằng cách click chuột phải, chọn TortoiseSVN→Resolved... Sau đó mới có thể commit được.



Kết quả:



Giờ thì bạn có thể Commit file test.java được rồi!

(Nguồn: <http://asteriskese.wordpress.com>)

## 5. Liên hệ

Họ và tên: Nguyễn Việt Khoa

Đơn vị: FAT

Email: [nguyenvietkhoa@gmail.com](mailto:nguyenvietkhoa@gmail.com)

Điện thoại: 0912.151.629