CVS là từ viết tắt của **Concurrent Versioning System**, một cách sử dụng thông thường quản lý mã nguồn đối với các dự án phần mềm lớn. CVS giữ tất cả các phiên bản của tất cả các file vì thế không cái nào bị mất bao giờ, và chúng được kiểm tra bởi nhiều người. Nó cũng cung cấp nhiều cách kết hợp mã nếu hai hoặc nhiều người đang làm việc trên file tương tự. Tất cả các mã và tất cả các phiên bản được lưu trữ trên một server trung tâm.

Các môđun trong CVS:

Trong CVS, từ "môđun" đề cập đến việc phân chia tập hợp mã. Trong Moodle chúng ta có các môđun sau trong kho lưu trữ của chúng ta:

**moodle** - mã nguồn Moodle chính

**contrib** - các đóng góp người dùng và sự đa dạng của các loại mã trong việc phát triển

**mysql** - một phpMyAdmin tuỳ chọn để đưa vào trong Moodle đối với cơ sở dữ liệu admin

**windows-cron** - một gói nhỏ mà có khả năng tạo cron trên hệ thống Windows

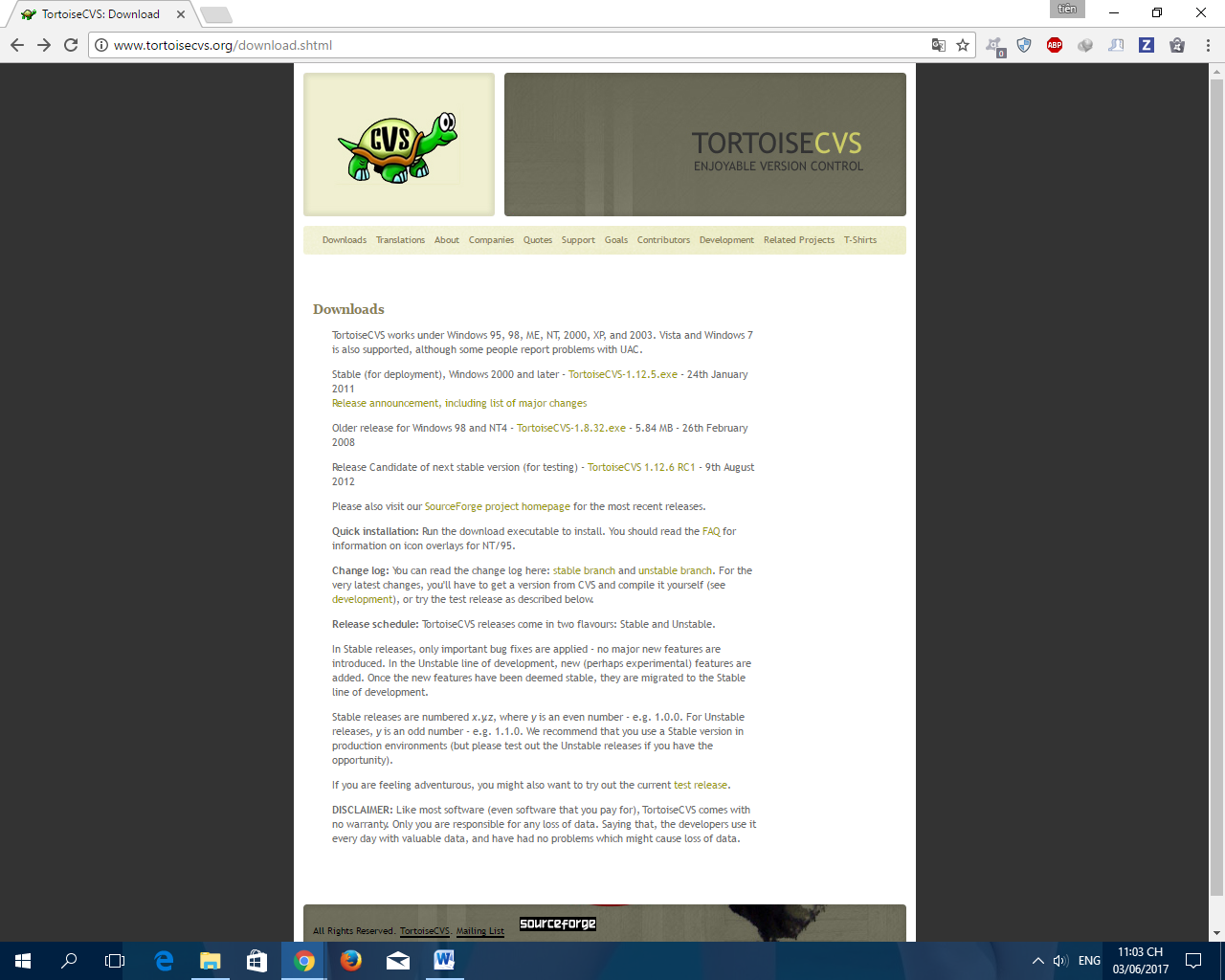
**docs** - Tài liệu được đóng góp bởi nhiều người dùng khác nhau

Tất cả mọi người đang làm việc trên các đặc tả đã tồn tại trong môđun **moodle** , nhưng nhiều cái cũng được đóng góp bởi các ý kiến mới trong các môđun **contrib**. Một mã tìm kiếm ở một mức độ nào đó trong vùng **contrib**, nó có thể di cư toàn bộ sang cây thư mục **moodle** chính.

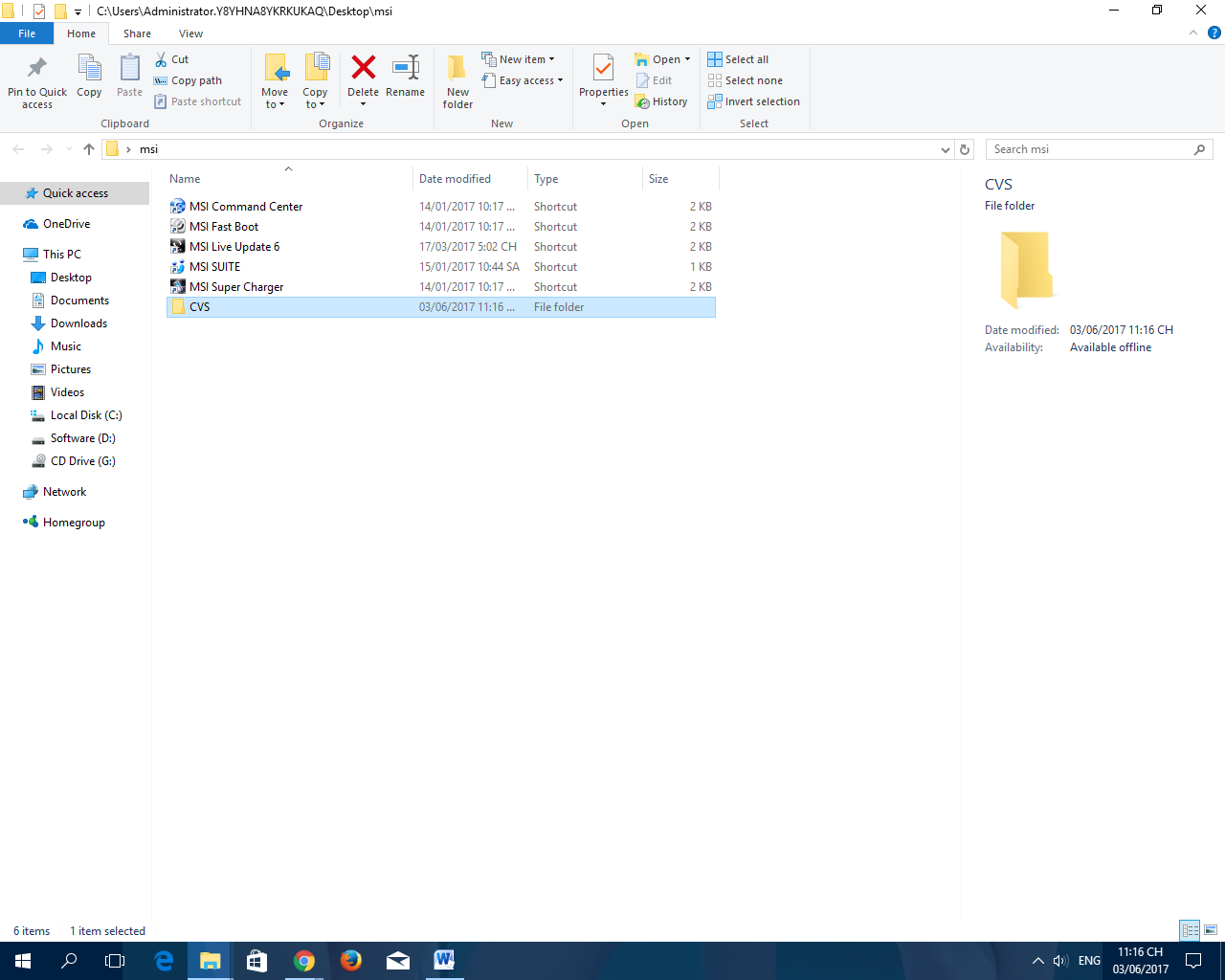
Các lệnh cơ bản của CVS:

Đầu tiên bạn cần download một bản copy Moodle đầy đủ mới sử dụng tài khoản người phát triển.

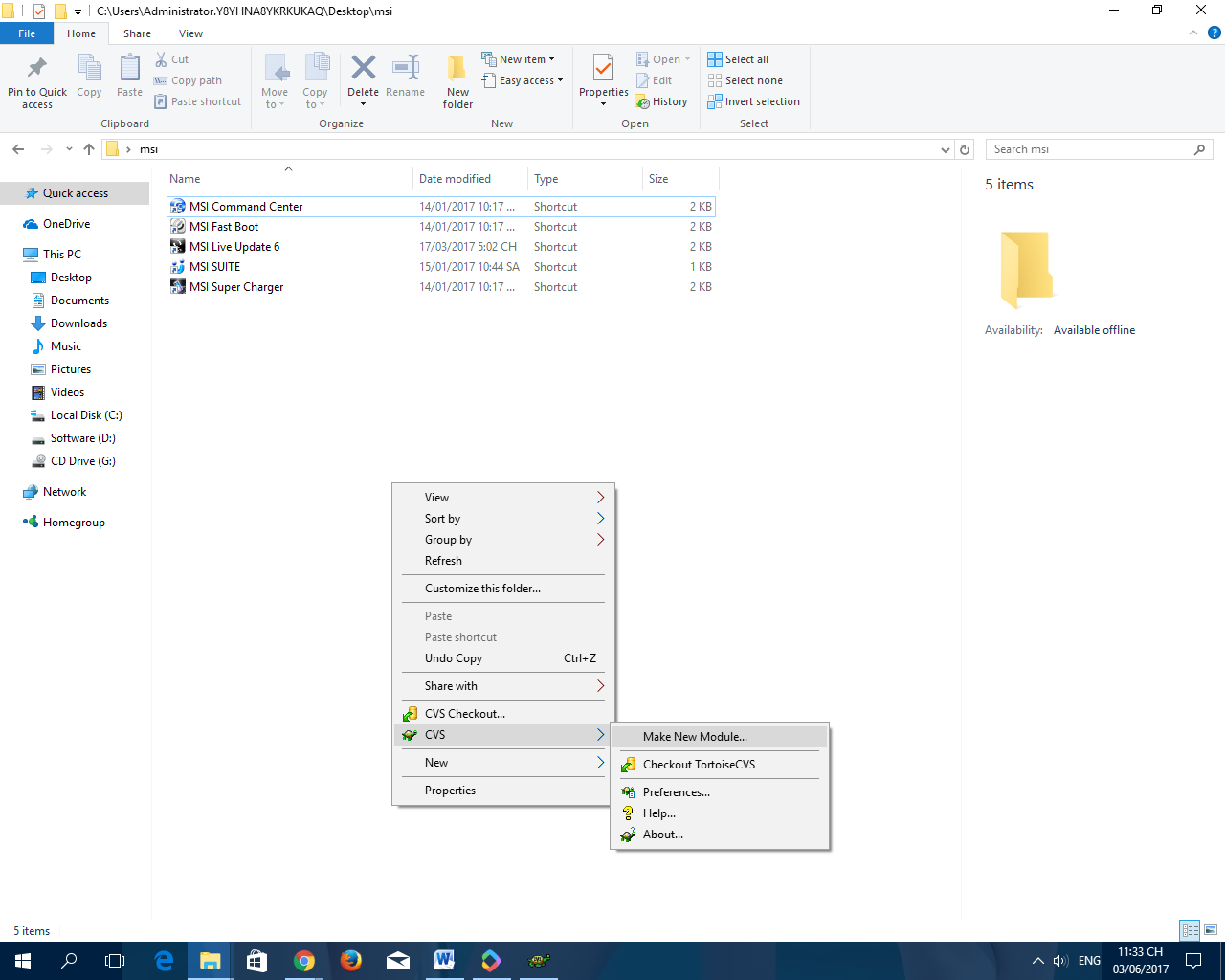
* + 1. Lấy được TortoiseCVS từ [tortoisecvs.org](http://www.tortoisecvs.org/) và cài đặt nó, sau đó khởi động lại.



* + 1. Tìm kiếm hoặc tạo một folder mới ở đâu đó mà bạn muốn Moodle được downloaded xuống.

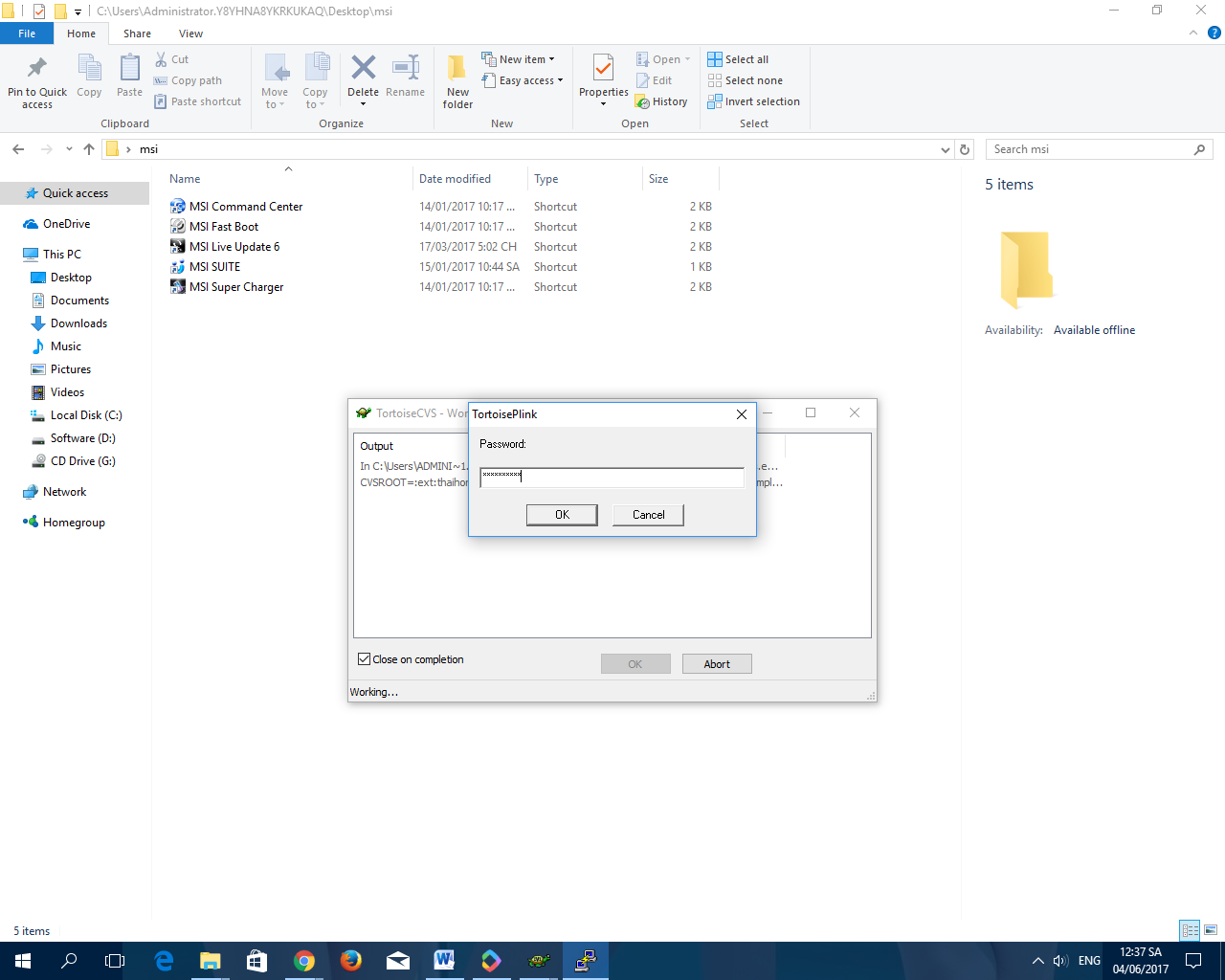


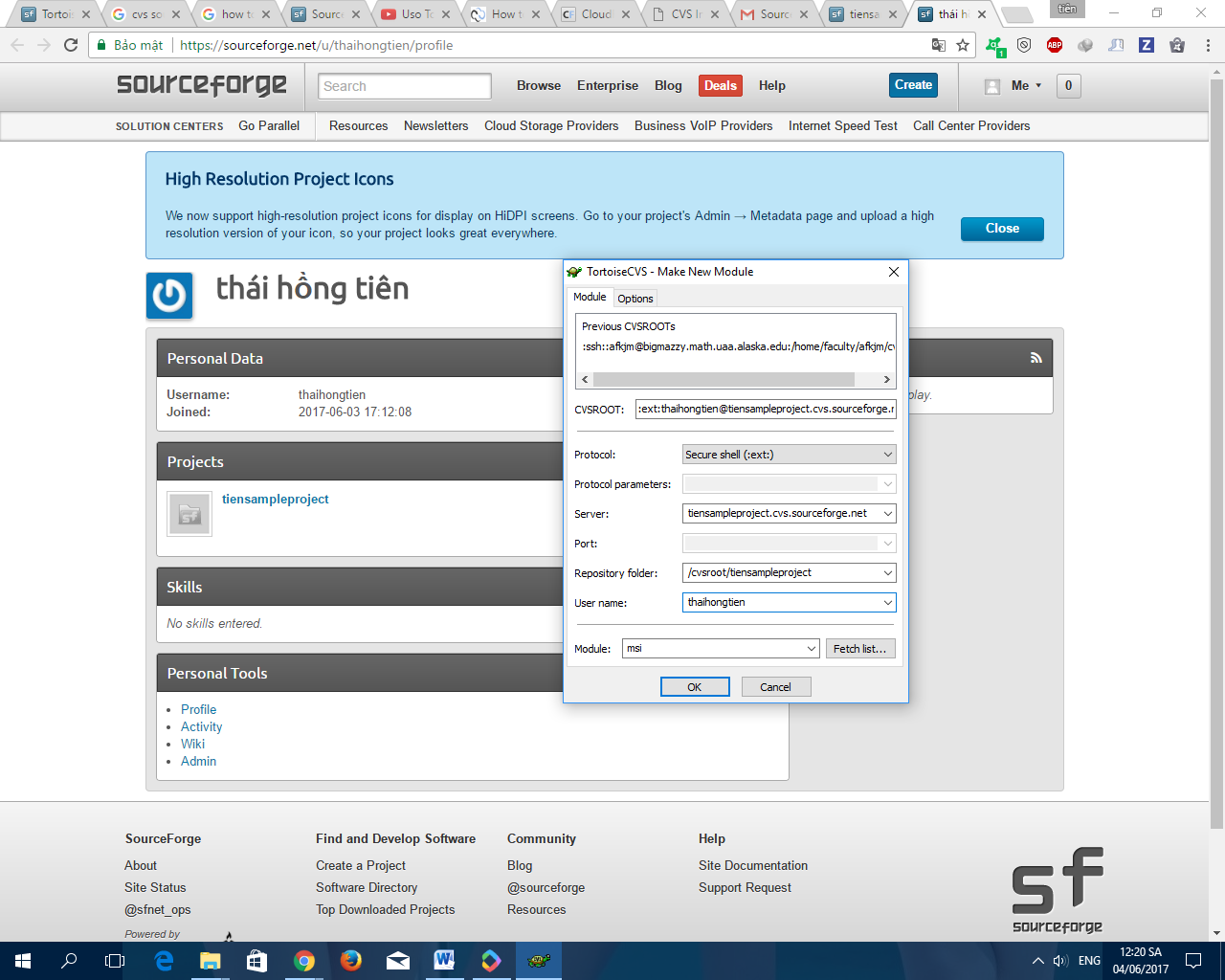
* + 1. Kích chuột phải vào folder đó và chọn "**CVS"->”make new module”** từ danh sách thả xuống. Bạn sẽ nhìn thấy một hộp thoại.



* + 1. Copy văn bản này vào trong trường CVSROOT (sử dụng username của riêng bạn!):

:ví dụ:myusername@cvs.sourceforge.net:/cvsroot/moodle





* + 1. Trong trường "Môđun ", gõ "**moodle**" để nhận phiên bản phát triển mới nhất của Moodle, "**contrib**" để nhận được thư mục các đóng góp, hoặc "**mysql**" để nhận được môđun MySQL Admin.
    2. Nhấn nút: "**OK**"và mọi thứ sẽ được tải xuống.

Một hộp thoại sẽ hiển thị tất cả các trường đang được tải xuống, và sau đó bạn sẽ có một bản copy đầy đủ của Moodle. Sau lần kiểm tra đầu tiên này, bạn có thể tìm nạp các file được cập nhật mới nhất từ CVS server:

* + 1. Kích chuột phải trên folder Moodle của bạn (hoặc bất kỳ file nào) và chọn "**Cập nhật CVS**".
    2. Xem các bản ghi cuộn xuống. Xem chú ý về các xung đột mà có thể xuất hiện khi mã cục bộ cục bộ của bạn thay đổi mà nó xung đột với các phiên bản đang đưa vào - bạn sẽ cần soạn thảo những file này và giải quyết các xung đột đó bằng tay.

Sau khi thay đổi các file( bạn sẽ thông báo thay đổi các biểu tượng từ màu xanh sang màu đỏ !), you can commit them back to the CVS server like this:

* + 1. Kích chuột phải vào folder Moodle (hoặc bất kỳ file nào ) và chọn "**CVS Commit...**".
    2. Trong hộp thoại, gõ một mô tả một cách rõ ràng những thay đổi bạn đang cam kết .
    3. Nhấn chuột vào nút "OK". Những thay đổi của bạn sẽ được gửi tới server.

4.Làm việc với các nhánh

Để xem tất cả các thẻ hiện tại và các nhánh mà nó có sẵn, sử dụng lệnh này trên bất kỳ file thân thuộc nào(ví dụ file index.php trong thư mục moodle):

**cvs status -v index.php**

Một số nguyên tắc tagging:

* Các tên Tag và nhánh sẽ luôn luôn là chữ hoa.
* Các Tag và các nhánh sẽ luôn luôn được ứng dụng đối với**toàn bộ môđun** (tất cả các môđun của Moodle). Đừng gắn các file hoặc các thư mục một cách riêng rẽ.
* Chúng tôi không cho phép đổi tên các tags bởi vì mọi người có thể đang hồi âm chúng, cho nên nhận chúng đúng lần đầu tiên so get them right the first time!

4.1 Phát triển phần thân

Phần thân của CVS là phiên bản phát triển chính của Moodle. Trong CVS nó cũng được biết đến như **HEAD**, hoặc nhánh mặc định.

Các nhà phát triển Moodle cố gắng duy trì tính ổn định này chừng nào có thể, nhưng bởi vì nó thường chứa mã mới nó hầu như có các lỗi và tính không ổn định nhỏ.

Mọi thứ bây giờ và sau đó chúng ta quyết định sản phẩm có đủ đặc trưng để tạo nên một phiên bản. Ở thời điểm này, phần thân được gắn với một thẻ **MOODLE\_XX\_BETA** (trong trường hợp chúng ta muốn quay trở lại n case we ever want to roll back to that point) và một nhánh mới được tạo thành đối với phiên bản, được gọi là MOODLE\_XX\_STABLE.

Một gói Beta cũng được phát hành ở thời điểm này - nó dành cho những người thử những người mà không sử dụng CVS nhưng muốn thử nghiệm các đặc trưng mới nhất và các thông báo lỗi.

4.2 Các nhánh ổn định đối với mỗi phiên bản

Ngay khi một nhánh ổn định **MOODLE\_XX\_STABLE** được tạo ra, các lỗ lực cố gắng phát triển sẽ phân nhánh thành hai hướng đối với mỗi khoảng thời gian. Mọi người có thể tiếp tục làm việc trên các đặc trưng mới trong phần thân đối với các phiên bản tiếp theo, nhưng tất cả các nhà phát triển nên cố gắng tập trung sử dụng nhánh**ổn đinh**hiện tại và cố định các lỗi mà được tìm thấy trong nó.

Bạn có thể chuyển đổi bản copy Moodle cục bộ của bạn thành phiên bản ổn định sử dụng lệnh sau trong Unix từ thư mục gốc :

**cvs update -dP -r MOODLE\_XX\_STABLE**

Sau khi làm tất cả xong, tất cả các lệnh được mô tả ở trên sẽ áp dụng cho phiên bản ổn định đó. Quay trở lại với vấn đề phần thân :

**cvs update -dPA**

Trên Windows clients bạn nên có một menu từ đó bạn có thể chọn nhánh .

Một khi nhánh bảng mới thực sự ổn định, một phiên bản có thể được có thể được khai báo. Các gói được tạo để phân phối và nhánh sẽ được gắn thẻ (bởi Martin) với một thẻ đặt tên là: **MOODLE\_XXX**

Một cách định kỳ, lỗi cố định trong nhánh bảng sẽ được kết hợp lại vào trong phần thân vì thế chúng trở nên có sẵn trong các phiên bản tương lai của Moodle. Một thẻ di chuyển được gọi là MOODLE\_XX\_MERGED sẽ được duy trì để theo dõi sự kết hợp cuỗi cùng. Quy trình đối với mỗi lần kết hợp này là như sau:

* 1. Get out the very latest trunk version.

**cvs update -dPA**

* 1. Kết hợp mọi thứ vào trong nhánh từ nhánh cuối cùng, đưa vào phần thân của bạn

**cvs update -kk -j MOODLE\_13\_MERGED -j MOODLE\_13\_STABLE**

* 1. Xem một cách cẩn thận các bản ghi được cập nhật đối với các xung đột, và cố định mọi file mà bạn nhìn thấy có xung đột
  2. Check the merged copy back into CVS trunk version

**cvs commit**

* 1. Quay trở lại phiên bản nhánh Go back to the branch version

**cvs update -dPr MOODLE\_13\_STABLE**

* 1. Cập nhật thẻ kết hợp đang thay đổi vì thế quy trình xử lý này có thể được lặp lại thời gian tới Update the floating merge tag so that this process can be repeated next time

**cvs tag -RF MOODLE\_13\_MERGED**

Cuối cùng các kết quả đối với *$version* trong tất cả các file version.php của Moodle trong nhánh ổn định không nên được cập nhật ở tất cả khi có thể( ngoại trừ số cuối cùng nếu cần thiết ). Nguyên nhân là ai đó đang cập nhật từ một phiên bản rất ổn định thành phiên bản ổn định tiếp theo có thể thiếu các nâng cấp cơ sở dữ liệu điều đó được xảy ra trên phần thân.

4.3 Các nhánh đặc trưng đối với các thay đổi lớn

Thỉnh thoảng, có thể có một đặc trưng rất lớn mà cần được kiểm tra vì thế một số người có thể làm việc trên nó, nhưng nó quá không ổn định để tính đến trong phần thân phát triển chính.

Trong những trường hợp này một nhánh ngắn có thể được tạo ra để làm việc trên đặc trưng, và sau đó kết hợp trở lại vào trong phần thân chính ngay khi có thể. Một ví dụ được gọi là nhánh **MOODLE\_14\_WIDGET** có thể được xem trong biểu đồ ở trên.

Nếu bạn cần làm điều này cho đặc trưng WIDGET mới của bạn, làm theo các bước sau :

* 1. Thảo luận với các nhà phát triển khác để đảm bảo rằng nó là cần thiết !
  2. Tạo một thẻ mới trên **phần thân** (đối với tất cả Moodle) được gọi là **MOODLE\_XX\_WIDGET\_PRE**

**cvs tag -R MOODLE\_XX\_WIDGET\_PRE**

* 1. Tạo nhánh của bạn được gọi là **MOODLE\_XX\_WIDGET**

**cvs tag -Rb MOODLE\_XX\_WIDGET**

* 1. Làm việc trong nhánh đó cho tới khi đặc trưng là ổn định. Cam kết khi cần thiết.

**cvs commit**

* 1. Khi đã sẵn sàng, kết hợp toàn bộ nhánh vào trong phần thân, cố định các xung đột, cam kết nó vào phần thân và sau đó loại bỏ nhánh.

**cvs update -dPA  
cvs update -kk -j MOODLE\_XX\_WIDGET  
cvs commit**

một số lệnh cơ bản nhất của **CVS**.

**1. Lấy mã nguồn về**  
Lệnh thực hiện điều này trong phần lớn hệ thống mã nguồn là "**checkout**" và **CVS** cũng thế. Ví dụ để lấy mã nguồn của dự án "tên\_dự\_án" về từ **gnu.org** bạn làm như sau:

$ cvs -z3 -d:pserver:[anonymous@cvs.sv.gnu.org](mailto:anonymous@cvs.sv.gnu.org):/sources/tên\_dự\_án checkout tên\_dự\_án​

Thực ra phần lớn các dự án phần mềm nguồn mở đều cho phép bạn lấy mã nguồn về từ hệ thống quản lý mã nguồn. Đồng thời họ cũng sẽ chỉ cho bạn "**cách**" để lấy nó về, tôi muốn nói là họ sẽ nêu cả câu lệnh như trên trong trang dự án, và ta chỉ việc **copy**mà thôi. Dòng lệnh trên tôi cũng "**copy**" như vậy đấy!  
Sau khi chạy lệnh trên thành công sẽ tạo ra thư mục tên\_dự\_án. Hãy di chuyển vào đó.

$ cd tên\_dự\_án​

Trong thư mục này ta thấy xuất hiện thư mục **CVS**. Đây chính là nơi cấu hình cho **CVS**bao gồm một số thông tin như:

* Entries: chứa tên các file, thư mục đồng thời cả ngày tháng của chúng nữa
* Repository: tên "kho", thực chất là tên của thư mục, tên của dự án
* Root: chứa thông tin về URL của server

**2. Xem trạng thái**  
Để xem tình trạng hiện tại của mã nguồn bạn sử dụng:

$ cvs status​

**3. Lấy về mã nguồn mới nhất**  
Do đã có thông tin đầy đủ rồi nên bạn chỉ cần chạy lệnh update mà không cần gì thêm

$ cvs update​

**4. Gửi các thay đổi đi**  
Lệnh này chỉ dùng khi và chỉ khi bạn cũng là thành viên phát triển phần mềm của dự án. Bạn sửa mã nguồn và muốn giửi các thay đổi đó đi? Trước hết bạn phải **update**trước đã để trộn các thay đổi với người khác, nếu có, sau đó dùng lệnh:

$ cvs commit -m "Chú thích của bạn về lần sửa này."​

**5. Cài đặt CVS**  
Các bản phân phối Linux thường không cài sẵn **CVS**, bạn phải tự cài lấy. Trong **Ubuntu**bạn sử dụng lệnh:

$ sudo apt-get install cvs​

Tính năng

## CVS sử dụng [kiến trúc client-server](https://vi.wikipedia.org/w/index.php?title=Ki%E1%BA%BFn_tr%C3%BAc_client-server&action=edit&redlink=1), trong đó phía server lưu trữ các phiên bản hiện tại của [dự án](https://vi.wikipedia.org/wiki/D%E1%BB%B1_%C3%A1n) và lịch sử của nó. Trong khi đó, các [client](https://vi.wikipedia.org/w/index.php?title=Client&action=edit&redlink=1) (tạm hiểu là các nhà phát triển) kết nối tới [server](https://vi.wikipedia.org/wiki/M%C3%A1y_ch%E1%BB%A7) để lấy về một phiên bản copy đầy đủ dự án, làm việc trên phiên bản đó và sau đó lưu lại nhưng thay đổi của họ. Thông thường, client và server kết nối thông qua mạng [LAN](https://vi.wikipedia.org/wiki/LAN) hoặc [Internet](https://vi.wikipedia.org/wiki/Internet). Client và server có thể cùng chạy trên một máy nếu CVS có nhiệm vụ theo vết lịch sử của dự án do các nhà phát triển phần mềm phát triển trong nội bộ. Phần mềm server thường chạy trên [Unix](https://vi.wikipedia.org/wiki/Unix), trong khi CVS client có thể chạy trên nhiều [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) khác nhau.

## Nhiều nhà phát triển phần mềm có thể làm việc đồng thời trên cùng một dự án. Trong đó, mỗi nhà phát triển thay đổi nội dung các [tập tin](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin) bên trong phiên bản copy của dự án của chính họ và sau đó gởi những thay đổi của họ về máy server. Để tránh việc người này ghi đè lên những thay đổi của người khác, server chỉ chấp nhận những thay đổi đối với phiên bản gần đây nhất của một file. Điều này cũng giúp cho các nhà phát triển dự án luôn làm việc trên các bản copy mới nhất bằng cách kết hợp với những thay đổi của người khác dựa trên những nguyên tắc cơ bản. Điều này được thực hiện một cách tự động bởi CVS client.

Nếu client lấy nội dung của dự án thành công thì định danh (thường là con số thể hiện phiên bản) của tất cả các tập tin được tự động tăng lên và server có nhiệm vụ lưu lại tất cả các thao tác này trong các tập tin gọi là tập tin nhật ký (*logs file*) bao gồm các nội dung như ngày tháng, tên của nhà phát triển phần mềm, các mô tả tóm lược...

Các client cũng có thể so sánh các [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n), yêu cầu lịch sử đầy đủ của sự thay đổi. Nhiều dự án mã nguồn mở cũng cho phép kiểu truy cập nặc danh (*anonymous read access*) để đọc, ý tưởng này được đưa ra bởi [OpenBSD](https://vi.wikipedia.org/w/index.php?title=OpenBSD&action=edit&redlink=1).

Client cũng có thể sử dụng lệnh cập nhật để cập nhật bản copy của họ với phiên bản mới nhất trên server. Điều này loại bỏ được khả năng download lặp lại toàn bộ dự án.

CVS sử dụng phương pháp nén [delta compression](https://vi.wikipedia.org/w/index.php?title=Delta_compression&action=edit&redlink=1) để tiết kiệm chi phí lưu trữ các phiên bản khác nhau của cùng một tập tin.