

## آشنایی با Subversion بخش هفتم

### بروز کردن نسخه کاری

همانطور که ذکر شد چون شما تنها کاربر موجود در پروژه نیستید، هر لحظه ممکن است کاربران دیگر نسخه کاری خود که یک سری فایل در آن تغییر یافته‌اند را بر روی مخزن فایل اصلی commit نمایند. در اینصورت شما نیز باید این تغییرات را بر روی نسخه کاری خود داشته باشید تا همیشه در جریان پروژه قرار داشته و جدیدترین فایلها را در اختیار داشته باشید. برای اینکار از دستور svn update مانند مثال قبل استفاده می‌کنیم. اجازه دهید چهار حالت مختلفی که به هنگام بروز کردن commit کردن نسخه کاری پیش می‌آید را بررسی نماییم:

- فایل موجود در نسخه کاریتان از آخرین لحظه checkout تا بحال تغییر نکرده و فایل موجود در مخزن فایل نیز تغییر نکرده است، به عبارتی این دو فایل عین هم هستند. svn commit هیچ کاری را انجام نمی‌دهد، svn update هم هیچ کاری را انجام نمی‌دهد.
- فایل شما در نسخه کاریتان تغییر کرده است، ولی فایل موجود در سرور هیچ تغییری نکرده است، در اینصورت svn commit باعث می‌شود تغییرات شما بر روی سرور منتقل شود و svn update هیچ کاری را انجام نمی‌دهد.
- فایل در نسخه کاری تغییر نکرده، ولی در مخزن فایل اصلی سرور تغییر یافته است، در اینصورت svn commit هیچ کاری را انجام نمی‌دهد، ولی svn update نسخه جدید را از سرور گرفته و فایل شما را بروز می‌کند.
- فایل هم در نسخه کاری و هم در مخزن فایل اصلی تغییر کرده است! در این حالت دستور svn commit پیغام خطای اینکه فایل شما به‌روز نیست را برمی‌گرداند که برای رفع این خطا باید از دستور svn update استفاده نمایید. پس از اجرای svn update، ابتدا subversion تلاش می‌کند خود تصادم (Conflict) ایجاد شده را با ترکیب این دو فایل رفع نماید، اگر تغییرات در این دو نسخه به گونه‌ای باشد که subversion نتواند این کار را انجام دهد، وظیفه انجام اینکار بر دوش کاربر گذاشته می‌شود. تنها در صورتی subversion می‌تواند خود عمل ترکیب (merge) این دو فایل را انجام دهد که تغییرات ایجاد شده در آنها بسیار ساده باشد، مثلاً فایل موجود در سرور در خطوط اولیه تغییر کرده باشد و فایل موجود در نسخه کاری در خطوط آخر، در غیر اینصورت subversion نمی‌تواند این کار را انجام دهد.

همانطور که دیدید بدترین حالت گزینه آخر است، یعنی هنگامی که شما نسخه کاری را خود را update کرده‌اید و در حال کار بر روی یک فایل هستید، بعد از چند وقت که بخواهید تغییرات خود را دوباره در سرور قرار دهید، ببینید که شخص دیگری نیز همزمان با شما مشغول به کار بر روی همان فایل بوده، منتها او فایل خود را زودتر commit کرده است!

فرض کنید همین اتفاق برای فایل someOtherFile.php بیفتد. اگر خاطرتان باشد بار اولی که این فایل را ایجاد کردیم و در مخزن فایل commit نمودیم، محتوای آن بصورت زیر بود:

```
<?PHP
print "hello hello hello";
?>
```

و اگر باز هم خاطرتان باشد، بعد از انجام ویرایشاتی بر روی آن، محتوای آن به صورت زیر تغییر پیدا کرد:

```
<?PHP
if ( $ILikeYou )
    print "hello hello hello";
else
    print "go to hell!";
for ($i=0; $i<2000; $i++)
    echo "I should go to military service...";
?>
```

حال فرض کنید کاربری همان موقع که ما نسخه اولیه را در مخزن فایل commit نمودیم، این نسخه را دریافت کرده باشد. او پس از دریافت این نسخه، تغییرات خودش را بر روی فایل اعمال کرده است، تغییراتی مشابه زیر:

```
<?PHP
for ($i=5; $i<10; $i++)
{
    print "hello hello hello";
    $j=0;
}
?>
```

کاربر بیچاره ما خبر ندارد که در هنگامی که او در حال انجام ویرایشات خود بوده، آن نسخه جدیدتر ما در مخزن فایل commit شده است (نمونه کد شماره ۲ واقع در وسط صفحه قبل). او بعد از انجام تغییرات خود می‌خواهد که کار خود را commit نماید که با پیغام زیر روبرو می‌شود:

### \$ svn commit -m 'Deliberately making a conflict!'

Sending someOtherFile.php

svn: Commit failed (details follow):

svn: Out of date: '/someOtherFile.php' in transaction 'f'

کاربر سرشکسته ما که فهمیده است دلیل بروز این پیغام همانا انجام ویرایشات او بر روی فایل قدیمی نسبت به آخرین نسخه موجود در مخزن فایل بوده است، سعی می‌کند با اجرای svn update فایل جدیدتر را دریافت کند، که با اینکار در واقع از چاله به چاه می‌افتد:

\$ svn update

C someOtherFile.php

Updated to revision 12.

علامت C در کنار نام فایل همان Conflict است که نشان می‌دهد subversion خود نتوانسته است تغییرات این دو فایل را ترکیب نماید. اگر subversion خود موفق به انجام اینکار می‌شد، آنگاه علامت G که همان merGed است را می‌دیدیم. با دیدن علامت C در کنار نام فایلی بدانید که آن فایل در واقع update نشده است و همینطور نمی‌توان آنرا در مخزن فایل نیز commit کرد، چرا که خطر این وجود دارد که تغییرات یکی از طرفین از بین برود. پس تکلیف چیست؟ خوب، شما باید خود بصورت دستی اینکار را انجام دهید. به هنگام بروز چنین مشکلی، subversion سه کار جالب را برای آگاه‌سازی هر چه بیشتر شما انجام می‌دهد:

- علامت C را کنار فایل چاپ می‌کند و به خاطرش می‌ماند که این فایل مشکل دارد.
- اگر subversion تشخیص دهد که در چه نقاطی از فایل خطر تصادم تغییرات کاربر با تغییرات موجود در مخزن فایل وجود دارد، با گذاشتن علامتی که بعداً با هم می‌بینیم در فایل که در اینجا همانا someOtherFile.php می‌باشد، این نقاط را مشخص می‌کند تا کار برای شما که می‌خواهید دو فایل را به طور صحیح با هم ترکیب کنید، آسان شود.
- برای هر فایلی که به مشکل conflict برخورد، ۲ و یا ۳ فایل کمکی زیر را ایجاد می‌کند:

filename.rOLDER\_VERSION

این فایلی است که در نسخه BASE شما وجود داشته، به بیان دیگر فایلی که بعد از آخرین checkout شروع به ایجاد تغییرات در آن کرده‌اید که الان همین تغییرات مشکل ساز شده‌اند.

filename.rNEWER\_VERSION

این فایلی است که در نسخه HEAD وجود داشته، به بیان دیگر این همان فایلی است که شما می‌خواستید از مخزن فایل به عنوان جدیدترین نسخه دریافت کنید، غافل از اینکه با تغییراتی که در نسخه خود داده‌اید باعث بروز مشکل خواهید شد.

filename.mine

این در واقع همان فایلی است که پس از انجام تغییرات بر روی نسخه BASE خود در دست داشتید و می‌خواستید در مخزن فایل اصلی commit کنید که مشکلات از آنجا شروع شد و ... فراموش نکنید که همانطور که ذکر شد در صورتی که subversion بتواند نقاط تفاوت فایلها را با علاماتی مشخص کند، این علامات را در خود فایل اصلی (در اینجا someOtherFile.php) می‌گذارد، به بیان دیگر شما به این فایل filename.mine نیاز دارید تا اصولاً آخرین کار خود را ببینید. منتها اگر subversion نتواند نقاط مشکل دار فایلها را مشخص کند، دیگر فایل شما را نیز تغییر نمی‌دهد، بنابر این دیگر filename.mine را نمی‌سازد.

تئوری کافیت! ببینیم که چه اتفاقی افتاد. در وهله اول اگر محتوای شاخه خود را با دستور ls نگاه کنید، می‌بینید که این ۳ فایلی که خدمتتان عرض شد، واقعا ایجاد شده‌اند. می‌توانید محتوای هر کدام از این فایلها را نیز مشاهده کنید تا صحت گفتار ما و حسن نیت ما بر شما ثابت شود! خوب، الان اگر فایل someOtherFile.php را با ویرایشگر محبوب خود باز نمایید، خطوط زیر را مشاهده خواهید کرد.

و این همان علامات جادویی هستند که ما قرار بود راجع به آنها صحبت کنیم. هر کجا که >>>> mine آمده بود یعنی اینکه آن قسمت مربوط به فایل موجود در نسخه کاری شما می‌باشد، و دقیقاً بعد از آن که آن قسمت تمام شد، قسمتهای موجود در فایل موجود در مخزن فایل می‌آید. اینگونه شما براحتی می‌توانید تغییرات را مشاهده کرده و آنها را به بهترین وجه که خود به عنوان برنامه‌نویس سیستم‌مطالعید با هم ترکیب کنید تا حاصل کار شما و آخرین کار با هم دیگر نسخه واحدی را بسازند. بعد از اینکه این تغییرات را انجام دادید، می‌توانید کل محتوای این فایل را در someOtherFile.php کپی کنید، و یا اصلاً فایل someOtherFile.php را پاک کرده و اسم این فایل را به someOtherFile.php تغییر دهید! خلاصه اینکه در انتها باید نتیجه کار خود را مجدد در فایلی به همان اسم اولیه، در اینجا someOtherFile.php قرار دهید. بعد از آنکه کار خود را تمام کردید، باید این ۳ فایلی که توسط subversion تولید شده‌اند را پاک کنید، آنگاه مجدد دستور svn commit را اجرا نمایید. تا هنگامی که این فایلها پاک نشده‌اند، subversion هنوز فکر می‌کند که شما مشکل تصادم را حل نکرده‌اید.

```
<<<<<<< .mine
<?PHP
for ($i=5; $i<10; $i++)
{
    print "hello hello hello";
    $j=0;
}
?>
=====
<?PHP
if ( $ILikeYou )
    print "hello hello hello";
else
    print "go to hell!";
for ($i=0; $i<2000; $i++)
    echo "I should go to military service...";
?>
>>>>>>> .r12
```

چگونه باید این فایلها را پاک کرد؟ آیا توسط سیستم عامل و بصورت دستی باید اینکار را انجام دهیم؟ می‌توانید همچنین کاری بکنید، ولی چه لزومی دارد از خود subversion عزیز و دستور svn resolved که مختص به اینکار ایجاد شده است استفاده نکنیم؟

```
$ svn resolved someOtherFile.php
Resolved conflicted state of 'someOtherFile.php'
```

با اجرای این دستور مشکل conflict ما حل شده و می‌توانیم با خیال راحت commit نماییم.

**بیژن هومند**

[1] <http://subversion.tigris.org>