

# مدیریت فایل‌های یک پروژه نرم افزاری با استفاده از Subversion

پژوهش و نگارش :

وحید نصیری

**Vahid\_Nasiri@Yahoo.com**

چاپ عمومی غیر رایگان این مطالب بدون مجوز کتبی از طرف نویسنده به هر نحوی غیرمجاز است.  
انتشار این مطالب بر روی اینترنت و یا استفاده از آن به صورت مستقیم و یا غیر مستقیم در نشریات الکترونیکی با ذکر مأخذ بلا مانع است.

صفحه	عنوان
۳	مقدمه
۴	فصل ۱ - نصب و راه اندازی Subversion
۲۲	فصل ۲ - انجام کارهای متداول کنترلی با TortoiseSVN
۴۳	فصل ۳ - آشنایی با مفاهیم Branching & Tagging
۵۲	فصل ۴ - یکپارچگی SVN با محیط‌های توسعه مجتمع
۵۷	فصل ۵ - تهیه پشتیبان از مخزن‌های کدها
۵۹	فصل ۶ - آشنایی با قلاب‌ها در SVN
۶۴	فصل ۷ - مرور بهترین تجربه‌های کاری با SVN
۶۷	فصل ۸ - یکپارچگی SVN با سایر ابزارها
۸۲	فصل ۹ - یکپارچگی مداوم و پیوسته (Continuous Integration)
۸۹	فصل ۱۰ - دسترسی به SVN با برنامه نویسی
۹۲	منابع و مآخذ

آیا شما هم جزو آن دسته از برنامه نویسانی هستید که هنوز به ازای هر نگارش از برنامه، سورس کدهای خود را فشرده کرده و آنها را بر اساس تاریخ آرشیو می‌کنند؟ آیا می‌توانید گزارشی مقایسه‌ای را از تغییرات انجام شده در سورس کدهای دو نگارش اخیر برنامه خود ارائه دهید؟ آیا می‌دانید فایلی را که هم اکنون بر روی آن مشغول به کار هستید در نگارش قبلی آن چه وضعیتی داشته و مقایسه آن با فایل فعلی شامل چه موارد و تغییراتی خواهد شد؟ آیا امکان بازگشت سریع و تبدیل سورس کدهای جاری برنامه را به سه نگارش قبل که از نظر شما پایدارتر هستند، بدون اضطراب و تشنج عصبی و با سرعت هرچه تمام‌تر و بدون خطا دارید؟ آیا می‌دانید بر روی یک فایل ویژه پروژه چه کسانی کار کرده‌اند و چه تغییراتی را اعمال نموده‌اند؟ آیا می‌توانید در یک تیم برنامه نویسی در کنار سایر برنامه‌نویس‌ها بر روی یک پروژه واحد کار کنید و به‌سادگی تداخلات رخ داده را مرتفع و یا آخرین تغییرات انجام شده را ادغام نمایید؟ آیا می‌توانید همواره آخرین نگارش تأیید شده پروژه را به سرعت و با دقت تمام به افراد تیم نرم افزاری خود تحویل دهید؟ آیا مکان مجتمعی را جهت نگهداری سورس کدهای پروژه‌های مختلف، در سازمان خود دارید؟

به همین جهت امروزه در دنیای برنامه نویسی مدرن استفاده از یک سورس کنترل، نه تنها یک ویژگی محسوب نمی‌شود بلکه یک باید به‌شمار می‌رود. محصولات بسیاری توسط شرکت‌های بزرگ در این راستا جهت استفاده برنامه‌نویس‌ها تهیه شده‌اند مانند **Microsoft Source Safe** و یا اخیراً **Team Foundation Server**. برنامه **Microsoft Source Safe** بدلائل بسیاری بین برنامه نویس‌ها با اقبال چندانی روبرو نشد و **Team Foundation Server** بدلیل هزینه‌های سنگین آن جهت یک پروژه با بودجه متوسط مناسب نیست. دنیای سورس باز، راه‌حلی مجانی، بسیار پخته و بی‌نظیری را به نام **Subversion** یا **SVN** ارائه داده است که امروزه به‌عنوان پرطرفدارترین محصول سورس کنترل در دنیا و تحت سکوهایی کاری مختلف مانند ویندوز، لینوکس و غیره شناخته می‌شود. **TortoiseSVN** یک کلاینت **SVN** به‌شمار آمده و با یکپارچه شدن با **Windows Explorer** رابط کاربر گرافیکی را برای **SVN** جهت کاربران ویندوزی فراهم می‌آورد. پس از نصب **TortoiseSVN** با استفاده از یک سری افزودنی می‌توان یکپارچگی کاملی را در محیط‌های مجتمع توسعه مانند **Visual studio** و غیره در اختیار داشت. نصب و راه اندازی **SVN**، یکپارچگی آن با ویندوز و محیط‌های مجتمع توسعه و همچنین نحوه استفاده از آن موضوعات اصلی کتاب جاری را تشکیل می‌دهند و این مباحث محدود به زبان یا محیط برنامه نویسی خاصی نیستند.

وحید نصیری

پائیز ۱۳۸۷

## فصل ۱ - نصب و راه اندازی Subversion

### مقدمه

Subversion یک برنامه کنترل سورس کد است که توسط جامعه سورس باز پشتیبانی می‌شود. از این برنامه در بسیاری از پروژه‌های بزرگ دنیا استفاده می‌شود که دلایل چندی را می‌توان برای آن برشمرد:

Subversion بسیار سریع است و کاراً، تحت سیستم عامل‌های مختلفی در دسترس است، مراحل نصب و راه اندازی بسیار ساده‌ای دارد و همچنین استفاده از آن مجانی است. تمامی عملیات کنترل سورس در این برنامه توسط فایل‌ها صورت گرفته و از ساختارهای پوشه‌های متداول سیستم عامل استفاده می‌شود. به همین جهت کپی و تهیه پشتیبان از آن نیز بسیار ساده می‌باشد. در این سیستم هیچ شخصی مالک یک فایل تحت کنترل نبوده و به همین جهت هیچ فایلی جهت استفاده یک شخص خاص در تیم قفل نخواهد شد (مشکلی که با source safe مایکروسافت وجود دارد). مدیریت نهایی را Subversion انجام داده و در حالت یکی سازی نسخه‌های کاری مختلف، تداخل‌ها را گوشزد می‌کند (عموماً تغییر یک خط مشخص از کد توسط چندین برنامه نویس مختلف).

در ادامه جزئیات نحوه نصب Subversion را مرور خواهیم کرد. امکان نصب تمامی مواردی که در این فصل بررسی خواهند شد تنها توسط یک بسته نرم افزاری به نام Svn1Click نیز مهیا است. برای دریافت Svn1Click می‌توان به آدرس زیر مراجعه نمود:

<http://svn1clicksetup.tigris.org>

این بسته کار نصب SVN، TortoiseSVN (TSVN)، سرویس مربوطه، مستندات محصول و همچنین ایجاد اولین مخزن (repository) کد را به صورت خودکار انجام می‌دهد. اما از این روش در این کتاب استفاده نخواهد شد زیرا قصد داریم با جزئیات کاری مربوطه با دقت آشنا شویم تا اگر در آینده به مشکلی برخوردیم با داشتن درک صحیحی از نحوه عملکرد سیستم نصب شده، بتوانیم به خطایابی آن به پردازیم.

### برتری‌های Subversion نسبت به SourceSafe مایکروسافت

- دسترسی offline مدلی است که SVN بر مبنای آن طراحی شده است و برنامه نویس در اینجا هیچگاه نگران قطعی ارتباط با مخزن کد و offline شدن نخواهد بود. این نکته را باید در نظر داشت که Subversion بعنوان جایگزینی جهت سیستم کنترل سورس کد CVS مطرح است. CVS جهت مدیریت پروژه‌هایی با گستره جغرافیای بسیار زیاد طراحی شده است.
- دسترسی از راه دور به سرور مخزن کد SVN بسیار ساده است. تنها کافی است port مورد نظر را به فایروال سیستم معرفی نمود (در این مورد در ادامه بحث خواهد شد).
- سرعت بارگذاری پروژه در مقایسه با SourceSafe بسیار بالاتر است.
- همانطور که عنوان شد در SVN به صورت پیش فرض هیچ برنامه نویسی به‌عنوان مالک فایل معرفی نشده و فایلی قفل نخواهد شد (برخلاف SourceSafe که جهت ویرایش فایل‌ها باید آنها را قفل کرد). هر چند در SVN نیز در صورت تمایل می‌توان فایل‌ها را قفل کرد.
- سرعت همگام سازی پروژه با سرور مخزن کد بسیار بالاتر از نمونه مشابه در SourceSafe است.
- تغییرنام و یا انتقال مکان فایل‌ها در SVN بهتر مدیریت می‌شود (در فصل بعد در این باره توضیحات بیشتری ارائه خواهد شد).
- یکپارچگی TortoiseSVN با windows explorer بسیار ارزشمند بوده و مدیریت پروژه را بسیار ساده می‌نماید.
- داشتن چندین نگارش همزمان از یک پروژه در SVN امری بسیار ساده است و سوئیچ کردن بین آنها نیز ساده بوده و اثرات جانبی ناگواری را به همراه نخواهد داشت (مبحث انشعابات و برچسب‌ها که در طی یک فصل به آن خواهیم پرداخت در SourceSafe پشتیبانی نمی‌شود).

- مبحث قلاب‌های (hooks) سمت سرور در SourceSafe پشتیبانی نمی‌شود (قلاب‌ها برنامه‌هایی هستند که در صورت وقوع یک رخداد خاص اجرا می‌شوند. به این مورد نیز در طی فصلی جداگانه خواهیم پرداخت).
- مطابق آمارهای موجود، مستندات، راهنماها و آموزش‌های بیشتری را در مورد SVN می‌توان یافت.
- مطابق تجربیات کاری بسیاری از برنامه نویس‌ها، با بزرگ شدن بانک اطلاعاتی SourceSafe، احتمال تخریب و از کار افتادن آن نیز بسیار بیشتر می‌شود (برای مثال Visual SourceSafe corruption را در گوگل جستجو نمایید).
- SourceSafe تنها تحت ویندوز مهیا است.
- پروژه SourceSafe از طرف مایکروسافت تقریباً خاتمه یافته تلقی شده و نگارش‌های جدید آن تنها به روز رسانی مختصر نگارش اصلی آن است که در دهه ۹۰ میلادی ارائه شد.

### مقایسه SVN و CVS (Concurrent Versions System)

- SVN اساساً بعنوان جایگزینی با طراحی بهتر جهت CVS مطرح شده است.
- SVN سریعتر است. در مقایسه با CVS اطلاعات کمتری را در شبکه رد و بدل می‌کند.
- امکان الصاق یک سری ویژگی‌ها به فایل‌ها در SVN موجود است (در این مورد در فصل‌های آتی بیشتر بحث خواهد شد). اما CVS تنها یک فایل را ذخیره و مدیریت می‌کند و دیگر هیچ.
- CVS تنها فایل‌ها را مدیریت می‌کند (برخلاف SVN که مدیریت پوشه‌ها نیز در آن میسر است).
- CVS اساساً جهت مدیریت فایل‌های متنی طراحی شده‌است و برای مدیریت فایل‌های غیرمتنی در آن باید تمهیدات ویژه‌ای را فراهم آورد. اما SVN به صورت پیش فرض مدیریت هرگونه فایلی را پشتیبانی می‌کند.
- CVS برخلاف SVN تراکنش‌هایی از نوع همه یا هیچ را پشتیبانی نمی‌کند (به آن ساختار اتمی نیز گفته می‌شود). در CVS اگر هنگام هماهنگ سازی فایل‌های پروژه با سرور تعدادی از آنها هماهنگ شده و تعدادی دیگر بدلیل وجود تداخلات قابل هماهنگ سازی نباشند، پروسه هماهنگ سازی متوقف نشده و این کار باید پس از رفع تداخلات مجدداً برای باقیمانده فایلها صورت گیرد که این مساله می‌تواند یکپارچگی مخزن کد را زیر سؤال ببرد. یا تصور کنید که در حین عملیات هماهنگ سازی با سرور، ارتباط با سرور قطع گردد. CVS بدلیل عدم پشتیبانی از تراکنش‌هایی از نوع همه یا هیچ، پروسه را برگشت نده و در وضعیتی کاملاً ناهماهنگ و غیرقابل اطمینان قرار خواهد گرفت.
- در SVN امکان انتقال و یا کپی فایل‌های تحت کنترل به‌سادگی مهیا است (CVS با توجه به پشتیبانی نکردن از کنترل پوشه‌ها، تغییراتی را که در یک پوشه مانند کپی و انتقال و غیره رخ می‌دهد نیز کنترل نمی‌نماید).
- در SVN با توجه به مکانیزم‌های file caching، بسیاری از امور را در حالت آفلاین نیز می‌توان انجام داد (برای مثال پشتیبانی از Offline diff).

اگر علاقمند به مشاهده لیست نسبتاً کامل و مفصلی درباره مقایسه برنامه‌هایی در این ردیف هستید می‌توان به آدرس زیر مراجعه نمود:

[http://en.wikipedia.org/wiki/Comparison\\_of\\_revision\\_control\\_software](http://en.wikipedia.org/wiki/Comparison_of_revision_control_software)

### دریافت و نصب SVN

جهت دریافت برنامه نصب SVN می‌توان به آدرس زیر مراجعه نمود (شکل ۱):

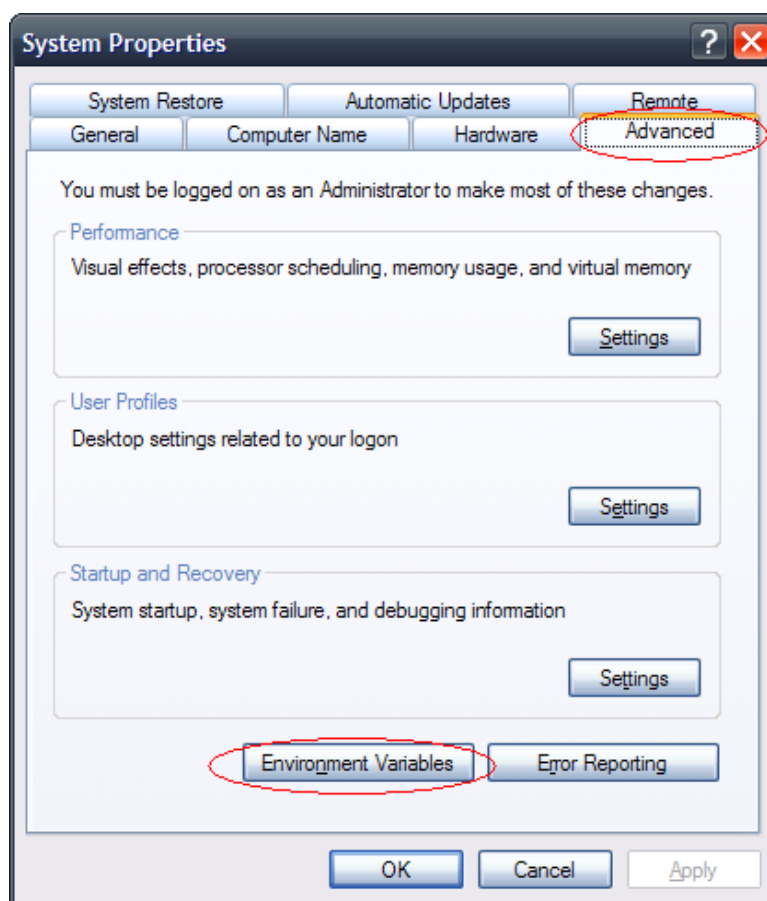
<http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91>

Name	Status	Modified by	Size	Reservations	Description	Info
Setup-Subversion-1.5.1.en-us.msi	Stable	ebswift on Friday, August 15, 2008 at 4:37:30 PM	7.02 MB		Windows MSI installer with the basic win32 binaries - includes bindings for Apache 2.0x and 2.2x	

شکل ۱- دریافت SVN از سایت [subversion.tigris.org](http://subversion.tigris.org).

نصب ابتدایی Subversion نکته خاصی ندارد و با چند کلیک بر روی دکمه Next به پایان می‌رسد (در هنگام نصب binding for Apache 2.2.x را انتخاب کنید. در ادامه در اینباره بیشتر توضیح داده خواهد شد). سپس یک پوشه جدید را در مسیری دلخواه برای مثال در `c:\svn-repos` بعنوان پوشه مخزن کد ایجاد نمایید.

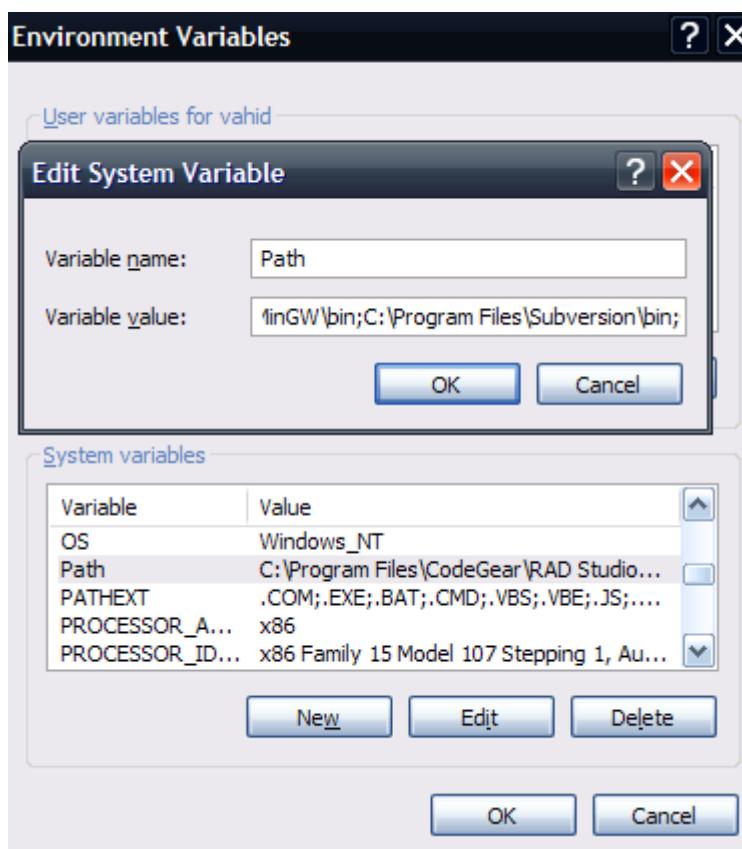
پس از نصب باید مسیر پوشه bin برنامه را به Environment Variables ویندوز اضافه نمود تا به‌سادگی بتوان توسط خط فرمان ویندوز در هر جایی از امکانات آن استفاده نمود. برای این منظور به control panel ویندوز مراجعه نموده و بر روی گزینه system دوبار کلیک نمایید. برگه advanced را انتخاب کرده و بر روی دکمه Environment Variables کلیک کنید (شکل ۲).



شکل ۲- افزودن پوشه bin برنامه به Environment Variables ویندوز.

در قسمت system variables، گزینه PATH را انتخاب کرده و با ویرایش آن مسیر پوشه bin برنامه Subversion را به انتهای آن اضافه نمایید (شکل ۳).

این عملیات عموماً توسط برنامه نصب SVN در نگارش‌های اخیر آن به صورت خودکار انجام می‌شود. بنابراین جهت بررسی صحت عملیات نصب می‌توان یکبار متغیر PATH را به ترتیبی که عنوان شد، ملاحظه نمود.



شکل ۳- افزودن مسیر برنامه به PATH ویندوز.

تا اینجا کار نصب ابتدایی SVN به پایان می‌رسد.

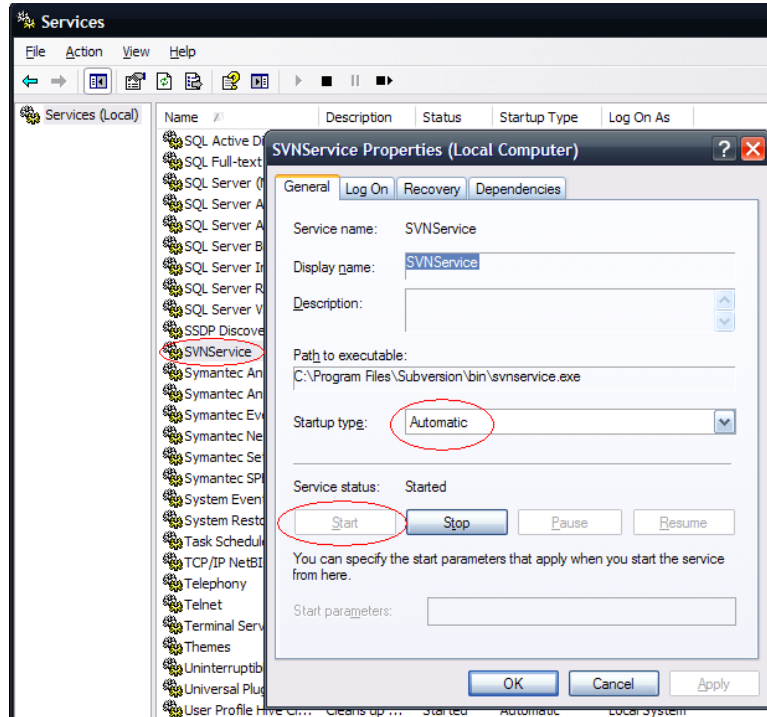
### نصب SVN بعنوان سرویس ویندوز

با نصب svnservice بعنوان سرویس ویندوز که در پوشه bin برنامه موجود است، می‌توان اطمینان حاصل نمود که با هر بار شروع ویندوز، برنامه به صورت خودکار اجرا خواهد شد و نیازی به دخالت مدیر سیستم جهت ورود به ویندوز و سپس اجرای دستی برنامه وجود نخواهد داشت. بدین منظور تنها کافی است دستور زیر را در خط فرمان ویندوز اجرا نمایید:

```
sc create "SVNService" binpath= "C:\Program Files\Subversion\bin\svnservice --service -r C:\svn-repos" displayname= "SVNService" depend= Tcpip start= auto
```

بدیهی است تنها تغییری که در دستور فوق باید ایجاد گردد، اصلاح مسیر svnservice مطابق مسیر نصب برنامه توسط شما است و همچنین اصلاح مسیر صحیح پوشه مخزن کد که در اینجا C:\svn-repos در نظر گرفته شده است. در این دستور، برنامه SC یا Service Control، برنامه استاندارد است که در ویندوزهای NT مانند XP، ۲۰۰۳ و غیره جهت نصب یک سرویس ویندوز به کار گرفته می‌شود. در هنگام اجرای دستور فوق باید دقت داشت که بین علامت مساوی و مقدار آن باید حتما یک فاصله وجود داشته باشد (در غیراینصورت موفق به اجرای دستور فوق نخواهید شد).

اکنون باید سرویس نصب شده را اجرا نمود (شکل ۴). برای این منظور در خط فرمان تایپ کنید: services.msc. پس از اجرای آن کنسول سرویس‌های ویندوز ظاهر می‌گردد. سرویس SVNService را یافته، نوع شروع آنرا بر روی automatic قرار داده (این مورد توسط خط فرمان نصب سرویس پیشتر تنظیم شده است) تا مطمئن شویم که پس از هر بار روشن شدن رایانه، سرویس مورد نظر ما به صورت خودکار و بدون دخالت مدیر سیستم اجرا می‌گردد و سپس بر روی دکمه start کلیک کنید.



شکل ۴- اجرای سرویس SVN و تنظیمات مربوطه

این مرحله را با استفاده از خط فرمان نیز می‌توان به صورت زیر انجام داد:

```
c:\>sc start SVNService
```

## نصب TortoiseSVN و ایجاد اولین مخزن کد

نصب SVN تا همینجا به پایان می‌رسد. توسط دستورات خط فرمان آن می‌توان مدیریت کاملی را بر روی مخزن کد خود داشت. برای مثال برای ایجاد یک مخزن کد جدید در پوشه اصلی مخازن (C:\svn-repos) می‌توان دستور زیر را در خط فرمان اجرا کرد:

```
svnadmin create c:\svn-repos\test1
```

با توجه به اینکه بیشتر پوشه bin برنامه در PATH ویندوز قرار گرفت، به‌سادگی می‌توان برنامه svnadmin را در خط فرمان فراخوانی کرد.

اما از آنجائیکه به‌خاطر سپاری دستورات خط فرمان در دنیای ویندوز آنچنان متداول نیست، برنامه‌نویس‌ها رابط‌های کاربر گرافیکی مختلفی را جهت استفاده از توانایی‌های SVN توسعه داده‌اند که یکی از پرطرفدارترین آنها برنامه TortoiseSVN می‌باشد. لیست سایر کلاینت‌های SVN را در آدرس زیر می‌توانید ملاحظه نمایید:

[http://en.wikipedia.org/wiki/Comparison\\_of\\_Subversion\\_clients](http://en.wikipedia.org/wiki/Comparison_of_Subversion_clients)

برنامه TortoiseSVN یکی از محبوب‌ترین کلاینت‌های SVN به‌شمار آمده و قابلیت یکپارچگی با Windows Explorer را دارا می‌باشد. امکانات آن با کلیک راست بر روی یک پوشه و یا یک فایل در ویندوز قابل دسترسی و استفاده است. همچنین اکثر افزودنی‌هایی (add-in) که برای محیط‌های مجتمع توسعه (IDE) مختلف نوشته شده‌اند، در حقیقت محصور کننده‌های (wrappers) توانایی‌های این برنامه می‌باشند. همچنین اگر SVN را بر روی یک سرور نصب کرده باشید، کلاینت‌ها با استفاده از TortoiseSVN می‌توانند از تمامی امکانات سرور مخزن کدها استفاده نمایند و نیازی به نصب SVN ندارند.

برای دریافت این برنامه می‌توان به آدرس بعد مراجعه نمود:

<http://www.tortoisesvn.net/downloads>

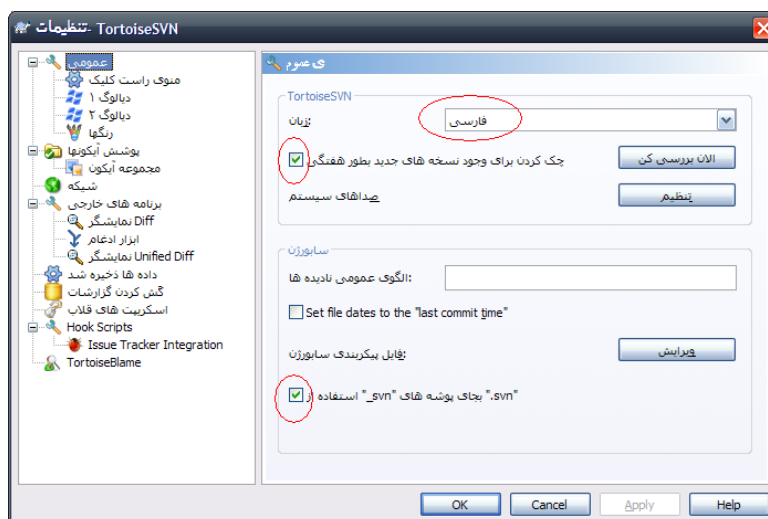


همچنین از آدرس فوق، بسته زبان فارسی برنامه نیز قابل دریافت است.

پس از نصب ساده این برنامه که تنها با چند کلیک بر روی دکمه Next میسر است و راه اندازی مجدد سیستم (جهت یکپارچگی کامل با windows explorer)، نوبت به تنظیمات آن می‌رسد. مسیر زیر را در ویندوز دنبال نمائید تا صفحه تنظیمات برنامه باز شود (گزینه settings با کلیک راست بر روی یک پوشه و یا فایل و انتخاب TortoiseSVN نیز در دسترس است):  
Start → All programs → TortoiseSVN → Settings

همانطور که عنوان شد اگر بسته زبان فارسی برنامه را نصب نمائید، در قسمت تنظیمات می‌توان این زبان را بعنوان زبان پیش فرض منوها و تنظیمات برنامه مورد استفاده قرار داد (شکل ۵). هرچند در این کتاب جهت آشنایی با یک سری مفاهیم اولیه از زبان انگلیسی پیش فرض استفاده خواهیم کرد.

همچنین اگر برنامه نویس دات نت هستید گزینه استفاده از svn بجای SVN. را جهت پوشه‌های مدیریتی svn انتخاب نمائید (این مورد صرفاً با پروژه‌های ASP.NET 1.x و VS.NET 2003 مشکل ساز است، اگر از نگارش‌های جدیدتر استفاده می‌کنید، مشکل خاصی وجود نخواهد داشت).



شکل ۵- تنظیمات اولیه برنامه TortoiseSVN.

در قسمت global ignore pattern یا الگوی عمومی نادیده‌ها (مطابق ترجمه فوق، یا فایل‌هایی که باید توسط برنامه ندید گرفته شده و وارد مخزن کد نشوند) می‌توان موارد زیر را اضافه کرد:

```
"*.dcu *.~* dcu temp *.exe *.zip *.bkm *.ddp *.cfg *.dof *.dsk *.ini *.hlp *.gid *.bmp *.png
*.gif ~* *.log bin debug release *.map *.chm *.bkf Thumbs.db *.mdb *.obj *.elf *.stat *.ddp
*.bpl *.map *.GID *.hlp *.opt *.dll *.raw *.BIN *.obj *.pdb *.scc Debug Release *.xml obj
*.~* *.backup *.INI *.ArmLog *.KeyLog *.NanoLog *.Stats *.PreARM *.old *.drc *.~*
*.doc *.pdf *.bmp *.jpg *.MRW *.NEF *.ORF *.psd *.X3F __history *.local *.identcache
*.bak Thumbs.db *.ldb *.dex *.rar DllDcu *.lck CVS cvs *.txt *.TXT *.jdbg *.HLP *.KWF
*.xls *.cnt *.dsm *.dti *.tmp *.lnk *.cbk *.mes *.suo *.ncb *.user _ReSharper.* [Bb]in obj
[Dd]ebug [Rr]elease *.aps *.eto"
```

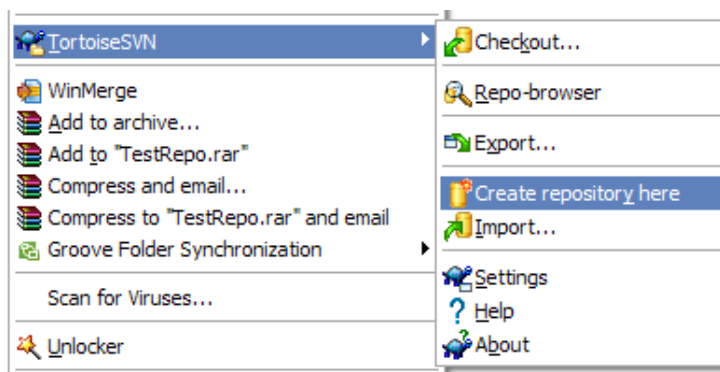
موارد فوق جهت کار افرادی که با VS.NET یا دلفی کار می‌کنند، مفید است.

باید دقت داشت که این الگوها حساس به حروف کوچک و بزرگ هستند. برای مثال الگوی بعد بحث حساس بودن به کوچکی و بزرگی حروف را پیاده سازی است:

\*[Bb][Li][Nn]  
\*[Oo][Bb][Jj]  
\*.[Ss][Uu][Oo]  
\*.[Cc][Ss][Pp][Rr][Oo][Jj].[Uu][Ss][Ee][Rr]  
\*.[Vv][Bb][Pp][Rr][Oo][Jj].[Uu][Ss][Ee][Rr]  
\*.[Ww][Ee][Bb][Li][Nn][Ff][Oo]

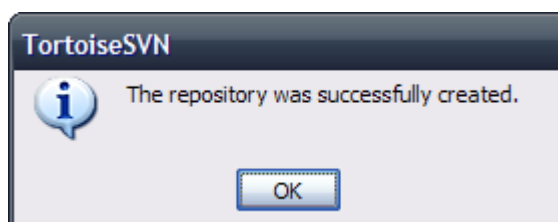
برای ایجاد اولین مخزن کد یک پوشه دلخواه را ایجاد نمایید، برای مثال  
C:\Repositories\TestRepo

روی پوشه کلیک راست کرده و گزینه زیر را انتخاب نمایید (شکل ۶):  
TortoiseSVN → create repository here



شکل ۶- ایجاد اولین مخزن کد.

پس از ایجاد مخزن کد، پیغام زیر را دریافت خواهید کرد:



شکل ۷- ایجاد موفقیت آمیز مخزن کد.

مخزن کد درحقیقت بانک اطلاعاتی مرکزی تغییرات انجام شده بر روی پروژه شما است که در ادامه به آن معرفی و اضافه خواهد شد. مستندات رسمی **subversion** توصیه می کنند که ساختار یک مخزن کد بهتر است به صورت زیر باشد:

/branches/  
/trunk/  
/tags/

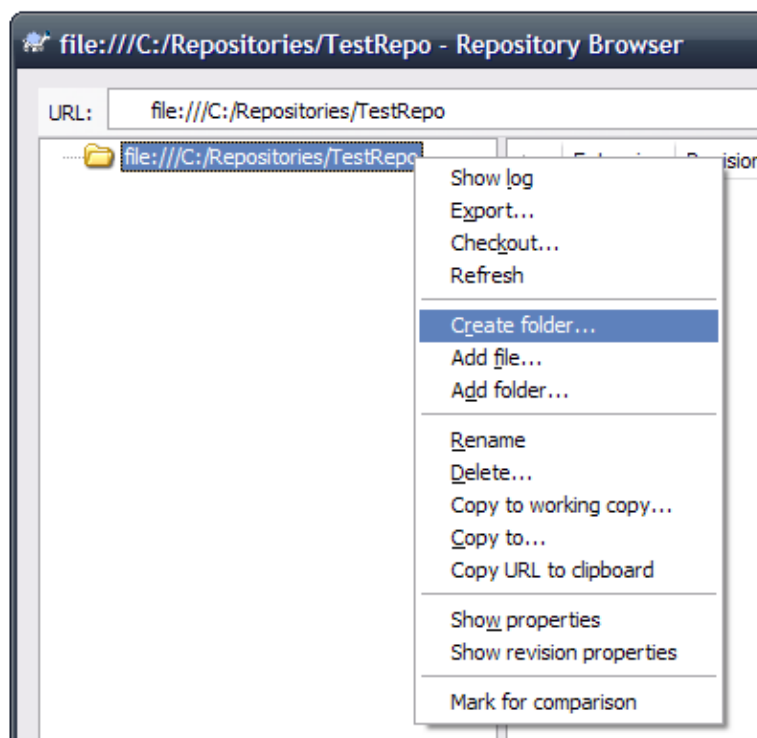
که در پوشه **trunk**، توسعه اصلی پروژه رخ خواهد داد. برای ایجاد این ساختار، بر روی پوشه مخزن کد ایجاد شده کلیک راست کرده و گزینه زیر را انتخاب کنید (شکل ۸):

TortoiseSVN → Repo-browser



شکل ۸- انتخاب مرورگر مخزن جهت ایجاد پوشه‌های کاری.

در صفحه مرورگر مخزن گشوده شده، بر روی لینک مخزن کلیک راست کرده (شکل ۹) و سه پوشه `branches`، `trunk` و `tags` را ایجاد نمایید.



شکل ۹- نحوه ایجاد فولدرهای جدید در مخزن کدها.

ایجاد این ساختار بدین شکل الزامی نبوده و کاملاً وابسته به تصمیم‌گیری شما است. برای مثال `tags` شامل کپی خاصی از `trunk` در یک زمان مشخص هستند:

```

|-- tags
|-- 1.0 Release
|-- 1.1 Beta
|-- 1.1 Release
    
```

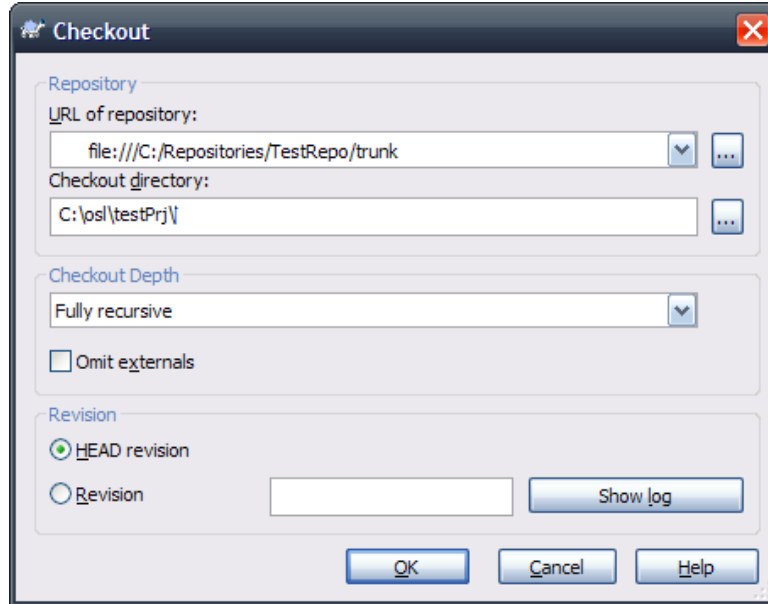
اکنون می‌خواهیم پروژه خود را تحت کنترل Subversion قرار داده و آنرا به مخزن کد ایجاد شده اضافه نماییم. فرض کنید پروژه ما در مسیر `C:\osl\testPrj` قرار دارد. بر روی پوشه آن کلیک راست نموده و گزینه زیر را انتخاب کنید (شکل ۱۰):

TortoiseSVN → Checkout

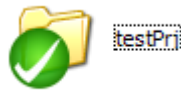


شکل ۱۰- معرفی پوشه پروژه به مخزن کد ایجاد شده

در صفحه باز شده مطابق تنظیمات شکل ۱۱ (انتخاب trunk مخزن کد و وارد کردن مسیر پوشه پروژه)، می‌توان پروژه را به مخزن کد افزود. پس از کلیک بر روی دکمه OK، اگر به پوشه پروژه مراجعه کنیم با یک سری فایل مخفی قرار گرفته در پوشه \_svn مواجه خواهیم شد که نشانگر تحت کنترل بودن این پوشه توسط Subversion می‌باشد. همچنین اگر به windows explorer مراجعه کنیم، آیکون این پوشه نیز تغییر کرده است (شکل ۱۲).



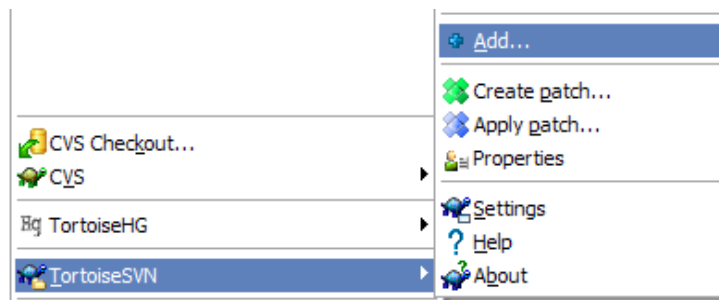
شکل ۱۱ - مراحل معرفی پروژه به مخزن کد



شکل ۱۲ - نمایی از یک پوشه که تحت کنترل Subversion است.

تنها کاری که تا بدین لحظه صورت گرفته است، اعلام تحت کنترل قرار دادن پوشه پروژه به Subversion است. جهت افزودن فایل‌ها به مخزن کد گزینه زیر را انتخاب کنید (شکل ۱۲)

TortoiseSVN → Add

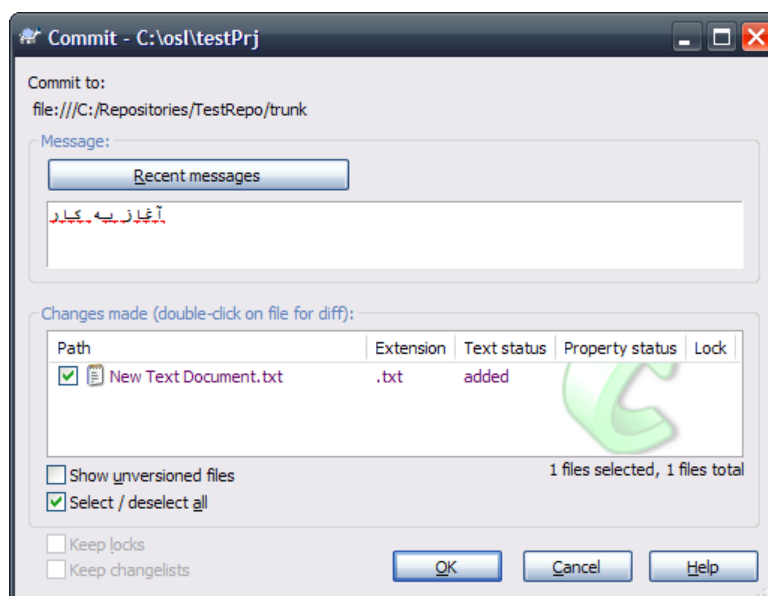


شکل ۱۲ - افزودن پوشه پروژه به مخزن کد

در اینجا می‌توان کل پوشه را به مخزن کد معرفی کرد یا تنها چند فایل و یا یک یا چند پوشه درون آنرا. در صورت نیاز، در صفحه ظاهر شده افزودن فایلها، با کلیک راست بر روی هر آیتم می‌توان آنها را به لیست موارد صرفنظر شونده اضافه کرد (Add to ignore list). پس از این مرحله آیکن پوشه و فایل‌های درون آن تغییر می‌کنند. این تغییر بدان معنا است که فایل‌ها به مخزن کد اضافه شده‌اند اما هنوز همگام با سرور و مخزن کد نیستند (ذخیره نشده‌اند). برای این همگام سازی روی پوشه پروژه کلیک راست کرده و گزینه زیر را انتخاب نمایید:

### TortoiseSVN → Commit

در صفحه باز شده توضیحاتی معنا دار (عموما خلاصه تغییرات انجام شده جهت مراجعات بعدی) را وارد کرده و بر روی دکمه OK کلیک نمایید (شکل ۱۳).



شکل ۱۳- همگام سازی تغییرات پروژه محلی با سرور مخزن کد.

جهت بررسی عملیات انجام شده، به پوشه مخزن کد اصلی مراجعه نموده و بر روی آن کلیک راست کنید. سپس گزینه نمایش مرورگر مخزن کد را انتخاب نمایید (TortoiseSVN → Repo-browser). در اینجا با مراجعه به پوشه trunk، می‌توان لیست فایل‌های افزوده شده به مخزن را مشاهده نمود.

بنابراین به صورت خلاصه :

- ابتدا یک پوشه در مخزن کد ایجاد کنید.
- پوشه پروژه مورد نظر را Checkout نمایید.
- با استفاده از TortoiseSVN، فایل‌های مورد نظر یا کل پوشه را به مخزن کد اضافه کنید (انتخاب گزینه Add).
- تغییرات نهایی را به سرور اعمال نمایید (Commit).

## تنظیمات سطوح دسترسی به مخزن کد


اولین مخزن کد ما در مسیر `C:\Repositories\TestRepo` ساخته شد. برای تنظیمات امنیتی و تعیین سطوح دسترسی به آن باید فایل‌های موجود در پوشه زیر را ویرایش کرد:

`C:\Repositories\TestRepo\conf`

با ویرایش فایل متنی `svnserve.conf` می‌توان نوع دسترسی را مشخص کرد:

```
[general]
anon-access = none
auth-access=write
password-db=passwd
realm=trunk
```

به این ترتیب (مطابق تنظیمات فوق) کاربران تعیین اعتبار نشده هیچگونه دسترسی به مخزن کد نخواهند داشت. کاربران تعیین اعتبار شده، دسترسی خواندن و نوشتن خواهند داشت.

 **نکته:** هنگام ویرایش این فایل نباید پیش از نام یک متغیر هیچگونه فاصله‌ای وجود داشته باشد، در غیراینصورت هنگام اتصال به مخزن با خطای زیر مواجه خواهید شد:

`svnserve.conf: 12: Option expected`


و با ویرایش فایل متنی `passwd` می‌توان کاربران مجاز را تعریف نمود (فاصله‌های اضافی را حذف نمائید):

```
[users]
user1=pass1
```

دقت نمائید که در اینجا نام کاربران و کلمات عبور، حساس به حروف کوچک و بزرگ هستند.

با ویرایش فایل `authz` که در این پوشه وجود دارد می‌توان دسترسی‌های بسیار محدود و ویژه‌ای را برای کاربران تعیین کرد. برای مثال به یک کاربر تنها دسترسی به یک `tag` مشخص را داد، برای مثال:

```
[projects:/spreadsheet/tags/release-1.0]
User1=rw
User2=r
```

 **نکته:** جهت بررسی عملکرد `svnserve` در مورد سرویس دهی مخزن‌های کد می‌توان از دستورات خط فرمان زیر استفاده کرد (شکل ۱۴)، و یا از `repo-browser` نیز می‌توان کمک گرفت. در صفحه مربوطه بجای URL مثلا وارد کنید:

`svn://localhost/test1`

در این حالت باید بتوان به مخزن دسترسی داشت.

```

Administrator: C:\Windows\system32\cmd.exe
C:\Users\Administrator>svn info svn://localhost/test1
Path: test1
URL: svn://localhost/test1
Repository Root: svn://localhost/test1
Repository UUID: ee4812de-a2f3-c944-8d9e-3d3bc14a146c
Revision: 0
Node Kind: directory
Last Changed Rev: 0
Last Changed Date: 2008-09-16 03:36:53 -0700 <Tue, 16 Sep 2008>

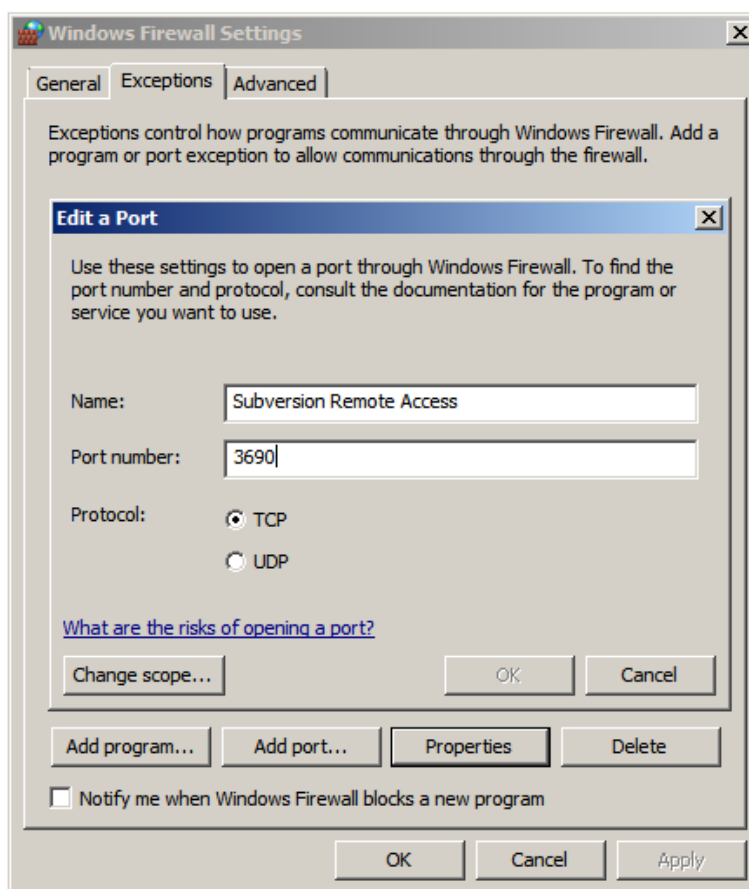
C:\Users\Administrator>svn info svn://localhost/testrepo
Path: testrepo
URL: svn://localhost/testrepo
Repository Root: svn://localhost/testrepo
Repository UUID: bb17661d-0d8a-4547-a028-bf8966a896ff
Revision: 0
Node Kind: directory
Last Changed Rev: 0
Last Changed Date: 2008-09-17 07:16:33 -0700 <Wed, 17 Sep 2008>

C:\Users\Administrator>
    
```

شکل ۱۴- بررسی عملکرد svnservice.

### فعال کردن دسترسی از راه دور به سرور مخزن‌های کد

برای فعال کردن دسترسی از راه دور به سرور مخزن‌های کد، باید port مربوط به SVN را در فایروال ویندوز (و یا هر فایروال دیگری که از آن استفاده می‌کنید) باز نمود. در اینجا نحوه انجام اینکار در فایروال ویندوز سرور ۲۰۰۸ بررسی می‌شود: در کنترل پنل ویندوز سرور، بر روی فایروال ویندوز دوبار کلیک کرده و در صفحه باز شده بر روی لینک change settings کلیک نمائید. در برگه exceptions بر روی دکمه add port کلیک نموده و port شماره ۳۶۹۰ را اضافه نمائید (شکل ۱۵).



شکل ۱۵- افزودن port برنامه SVN جهت دسترسی از راه دور به آن در فایروال ویندوز.

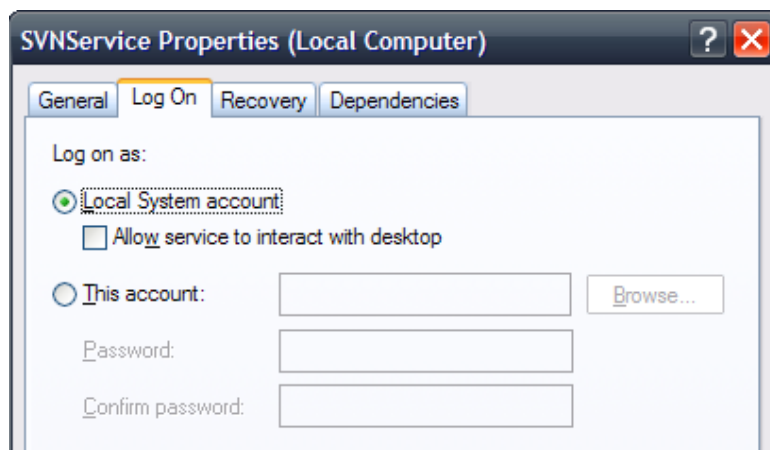
اگر می‌خواهید تنها رایانه‌های ویژه‌ای در شبکه شما دسترسی به SVN داشته باشند، در صفحه افزودن و یا ویرایش port، بر روی دکمه change scope کلیک نموده و لیست IP های این رایانه‌ها را وارد نمایید.

اکنون برای تست این مخزن کد در رایانه‌ای دیگر در شبکه، از خط فرمان زیر استفاده نمایید (و یا برنامه repo-browser):

```
svn info svn://Server-IP/test1
```

### امن کردن ویندوز سرور مخزن کدها

اگر هنگام نصب سرویس SVN دقت کرده باشید، از local system account بعنوان کاربری که مجوز اجرای این سرویس و برنامه را دارد به صورت پیش فرض استفاده شده است (شکل ۱۶).



شکل ۱۶- تعیین حساب کاربری مجاز جهت اجرای سرویس SVN.

همواره باید بخاطر داشت که این کاربر تنها باید دسترسی اجرای این برنامه را داشته باشد و نه هیچ کار دیگری در سیستم. بنابراین local system account با توجه به اینکه دسترسی کاملی بر روی تمامی برنامه‌های سرور دارد، جهت اینکار مناسب نیست. به همین جهت کاربر جدیدی را در ویندوز (یا شبکه) تعریف نموده (با کلمه عبوری قوی که هیچگاه منقضی نمی‌شود) و سپس از آن بعنوان حساب کاربری این سرویس استفاده نمایید. پس از اینکار باید به این کاربر دسترسی full control بر روی پوشه اصلی مخزن کد داده شود (بر روی پوشه مربوطه کلیک راست نموده و با انتخاب خواص و سپس مراجعه به برگه security می‌توان اینکار را انجام داد). هر چند به ظاهر این مورد کمی کار مدیر سیستم را بیشتر خواهد کرد، اما اگر باگی امنیتی در SVN کشف شود، با توجه به محدود شدن دسترسی آن، عملاً سطح امنیت بالاتری را در سیستم و شبکه تجربه خواهید کرد. پس از تغییر حساب کاربری سرویس SVN، باید یکبار این سرویس را متوقف و سپس اجرا نمود تا تغییرات انجام شده اعمال شوند.

### نصب وب سرور Apache و دسترسی به مخزن کدها از طریق مرورگر وب

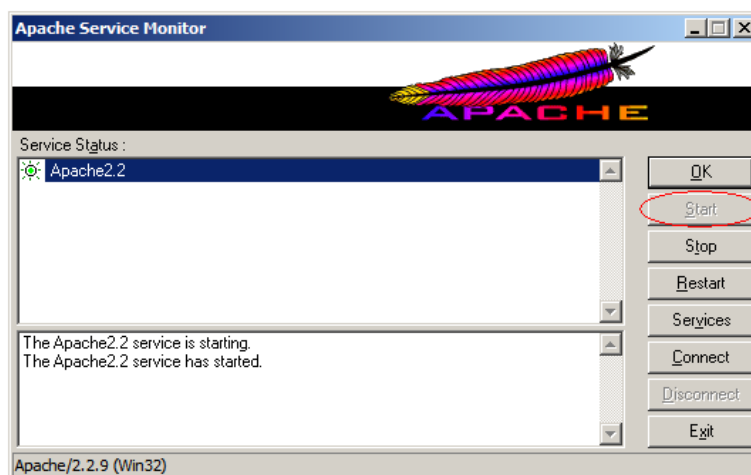
در ادامه قصد داریم نحوه استفاده از Subversion HTTP Module را مرور نماییم. ابتدا وب سرور آپاچی را از آدرس زیر دریافت کنید (Win32 Binary without crypto):  
<http://httpd.apache.org/download.cgi>



نصب ابتدایی آن نکته خاصی ندارد و پیش فرض‌های برنامه نصاب را بپذیرید. پس از نصب، اگر پیشتر بر روی سرور، IIS میکروسافت را نصب کرده باشید، با توجه به استفاده از پورت ۸۰ توسط IIS، وب سرور آپاچی آغاز به کار نخواهد کرد. برای رفع این تداخل، فایل متنی زیر را ویرایش کنید:

C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf

در آن سطر Listen 80 را یافته و برای مثال به Listen 81 تغییر دهید. پس از آن سرور را start کنید (شکل ۱۷).



شکل ۱۷- راه اندازی وب سرور آپاچی.

پس از راه اندازی وب سرور آپاچی، در مرورگر وب خود آدرس زیر را جهت آزمودن صحت عملیات نصب وارد کنید:

<http://localhost:81/>

پیغام **It works** را باید مشاهده نمایید.

اکنون به محل نصب **subversion** مراجعه کرده و دو فایل با پسوند **so** را یافته و در پوشه **modules** مربوط به وب سرور آپاچی کپی نمایید. این دوفایل عبارتند از :

C:\Program Files\Subversion\bin\mod\_authz\_svn.so

C:\Program Files\Subversion\bin\mod\_dav\_svn.so

آنها را در پوشه زیر کپی کنید:

C:\Program Files\Apache Software Foundation\Apache2.2\modules

اکنون باید با ویرایش فایل **httpd.conf** وب سرور آپاچی که پیشتر در مورد آن صحبت شد، نحوه بارگذاری این ماژول‌ها را تعیین نماییم. مجدداً فایل **httpd.conf** را گشوده و سه سطر زیر را به آن اضافه کنید:

LoadModule dav\_module modules/mod\_dav.so

LoadModule dav\_svn\_module modules/mod\_dav\_svn.so

LoadModule authz\_svn\_module modules/mod\_authz\_svn.so

همچنین به انتهای فایل خطوط زیر را نیز اضافه نمایید:

<Location /repos/>

DAV svn

SVNParentPath "C:\svn-repos"

</Location>

در اینجا SVNParentPath مسیر اصلی مخزن‌های کد است که سایر مخزن‌ها در آن واقع شده‌اند. همچنین توسط Location /repos ، وب سرور قادر به تفسیر URL هایی به شکل زیر خواهد شد:  
`http://localhost:81/repos/repository-name`

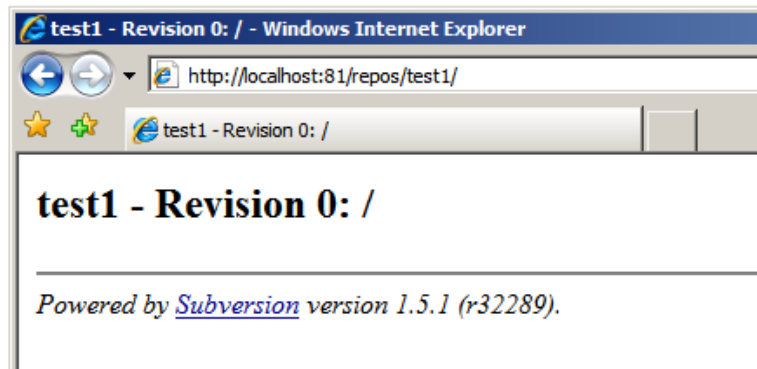
repos در اینجا یک نام دلخواه است.

پس از ذخیره فایل ، وب سرور آپاچی را یکبار stop/start نمائید تا تغییرات اعمال شوند. اینکار را با استفاده از Apache service monitor همانند شکل ۱۷ می‌توان انجام داد.

اکنون جهت مشاهده مخزن کد، آدرس زیر را در مرورگر وارد نمائید (test1 یکی از مخازن کد تعریف شده است):

`http://localhost:81/repos/test1/`

نتیجه کار باید مانند شکل ۱۸ باشد.



شکل ۱۸- دسترسی به مخزن کدها از طریق مرورگر وب.

اگر با خطای ۵۰۵ مواجه شدید، به event log viewer ویندوز مراجعه کرده و خطای مورد نظر را مشاهده نمائید. عموماً اشتباهات تایپی در این فایل مساله ساز می‌شوند که علت دقیق خطا در event viewer استاندارد ویندوز قابل مشاهده خواهد بود.

مرحله بعدی این تنظیمات، بحث اعتبار سنجی و محدود کردن کاربرانی است که می‌توانند به مخزن کد مورد نظر از طریق مرورگر وب دسترسی داشته باشند. برای این منظور تنظیمات زیر را بجای تنظیمات ساده قبلی وارد نمائید:

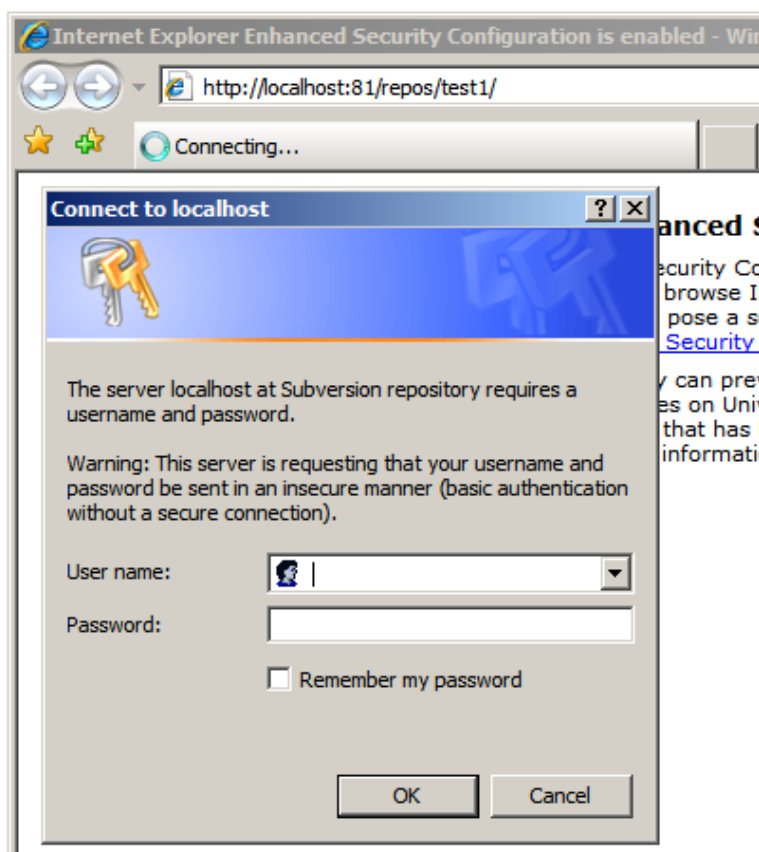
```
<Location /repos/>
  DAV svn
  SVNParentPath "C:\svn-repos"
  SVNListParentPath on
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile bin/apachesvnpasswd
  Require valid-user
</Location>
```

با توجه به عبارت Require valid-user ، تنها کاربرانی که در فایل apachesvnpasswd تعریف شوند مجاز به دسترسی خواهند بود. برای تعریف این کاربران و ایجاد این فایل مراحل زیر را طی کنید:

`C:\Program Files\Apache Software Foundation\Apache2.2\bin>htpasswd.exe -c apachesvnpasswd user1`

در اینجا با توجه به سوئیچ `-c`، فایل `apachesvnpasswd` در پوشه `bin` وب سرور آپاچی ایجاد شده و سپس برای تعریف کاربر `user1`، کلمه عبور آن نیز سؤال پرسیده خواهد شد. برای ایجاد کاربری جدید نیازی به استفاده از سوئیچ `-c` نمی‌باشد (استفاده از این سوئیچ سبب ایجاد مجدد فایل از نو خواهد شد).

مجدداً وب سرور آپاچی را `stop/start` کنید تا تنظیمات اخیر اعمال شوند. اکنون اگر URL قبلی را در مرورگر وارد کنید با شکل ۱۹ مواجه خواهید شد.



شکل ۱۹- پیاده سازی اعتباری سنجی برای دسترسی وب به مخزن کدها

علاوه بر این می‌توان دسترسی کاربران را نیز کاملاً تعریف نمود. در این حالت تنظیمات ما به شکل زیر درخواهد آمد:

```
<Location /repos/>
  DAV svn
  SVNParentPath "C:\svn-repos"
  SVNListParentPath on
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile bin/apachesvnpasswd
  AuthzSVNAccessFile bin/apachesvnauth
  Require valid-user
</Location>
```

برای تعریف فایل سطوح دسترسی، یک فایل متنی را در مسیر پوشه bin وب سرور آپاچی به نام apachesvnauth با محتویات زیر ایجاد نمائید:

```
[test1:/]
user1 = rw
```

در اینجا به کاربر user1 دسترسی خواندن و نوشتن به مخزن کد test1 داده شده است. برای مثال تنظیم زیر دسترسی تنها خواندنی به تمامی کاربران اعتبار سنجی شده اعطا می کند:

```
[/]
* = r
```

پس از انجام این تغییرات وب سرور آپاچی را stop/start نمائید.

### نکته: نصب SVN با استفاده از Visual SVN Server

برنامه رایگان دیگری به نام Visual SVN Server که از آدرس زیر قابل دریافت است، کار نصب وب سرور آپاچی، SVN، یک کنسول مدیریتی و سایر موارد را به صورت یک بسته نرم افزاری بسیار ساده انجام می دهد.

<http://www.visualsvn.com/server/download>

همچنین تمامی تنظیماتی که پیشتر در مورد ایجاد کاربران و تعیین سطوح دسترسی آنها نیز عنوان شد توسط این سرور رایگان به سادگی هر چه تمام تر و بدون نیاز به درگیر شدن با جزئیات آنها قابل انجام است. بعلاوه توسط آن امکان استفاده از کاربران تعریف شده در Active directory نیز به سادگی مهیا است (شکل ۲۰).

پس از نصب آن بر روی شاخه repositories کلیک راست کرده و گزینه create new repository را انتخاب کنید تا مخزن کد جدیدی را بتوان ایجاد نمود. پس از ایجاد مخزن جدید، بر روی شاخه مخزن ایجاد شده کلیک راست نمائید و گزینه خواص را انتخاب کنید تا بتوان سطوح دسترسی به مخزن را تنظیم نمود (بر روی هر یک از پوشه های مخزن، مانند trunk و غیره نیز می توان این سطوح را تنظیم کرد). صفحه مرورگری که توسط Visual SVN Server ارائه می شود نسبت به نمونه استاندارد ارائه شده توسط SVN نسبتاً شکیل تر است.

توسط Visual SVN Server دسترسی کاربرانی که اعتبار سنجی نشده باشند پشتیبانی نمی شود. در صورت نیاز برای اینکه به تمامی کاربران دسترسی های لازم را بدهید، باید از کاربری به نام everyone استفاده نمائید.

هرچند احتمالاً شما نیز از این محصول استفاده خواهید کرد، اما اکنون با توجه به توضیحات پیشین ارائه شده در این فصل می دانید که ساز و کار درونی برنامه هایی مانند Visual SVN Server چگونه است.

مواردی که در این فصل عنوان شدند شاید کمی طولانی به نظر برسند اما تنها یکبار باید انجام شوند و نرم افزارهایی مانند Visual SVN Server عمده مطالب عنوان شده در این فصل را در کمتر از یک دقیقه برای شما پیاده سازی خواهند کرد.

لازم به ذکر است سایت هایی نیز وجود دارند که بعنوان سرور SVN مجانی (جهت پروژه های شخصی) خدمات ارائه می دهند مانند:

<http://www.assembla.com/>

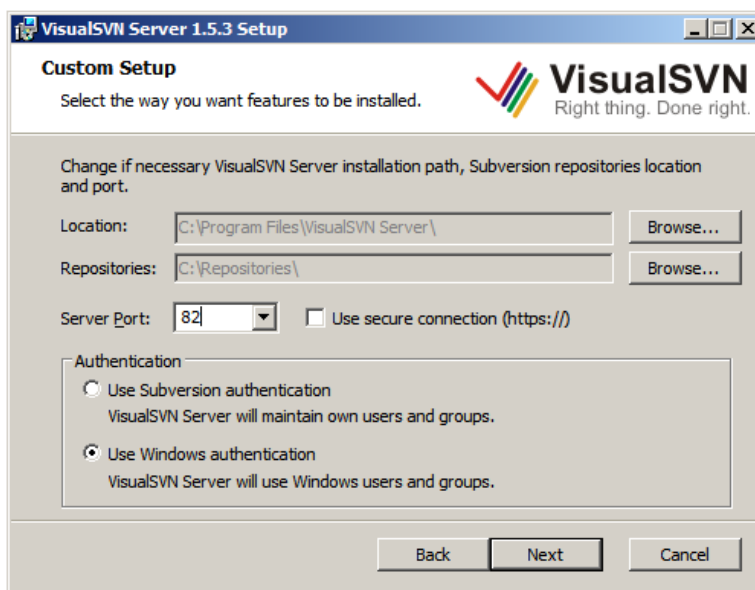
<http://unfuddle.com/>

<http://www.beanstalkapp.com/>

<https://gna.org/>

و یا LiveCD های لینوکسی نیز وجود دارند که برای این منظور طراحی شده‌اند. برای مثال:  
<http://buildix.thoughtworks.com/>

احتمالا با سایت‌های Sourceforge و یا CodePlex آشنایی دارید. اگر تمایل داشته باشید نمونه‌ای شبیه به آنها را بر اساس ASP.Net 2.0 با استفاده از امکانات SVN راه اندازی کنید، برنامه سورس باز SharpForge این امکان را به شما خواهد داد:  
<http://www.codeplex.com/SharpForge>



شکل ۲۰- برنامه نصب Visual SVN server.

### نکته: بهینه سازی کارایی سیستم پس از نصب TortoiseSVN

برنامه TortoiseSVN به صورت پیش فرض تمامی پوشه‌های موجود در تمامی درایوهای رایانه را جهت اعمال آیکون‌های مخصوص خود تحت کنترل قرار می‌دهد. بدیهی است که این مورد می‌تواند تاثیر منفی بر روی کارایی سیستم داشته باشد (هر چند بسیار ناچیز است).

برای محدود کردن این محدوده تحت کنترل، به قسمت settings برنامه مراجعه کرده و گزینه زیر را انتخاب کنید:

#### Look and Feel > Icon Overlays

در اینجا با تنظیم قسمت include paths ، محدوده تحت نظر قرارگیری برنامه را می‌توان محدود کرد. برای اینکه جستجوی TortoiseSVN در پوشه‌های مشخص شده بازگشتی یا عمقی (recursive) باشد، نیاز است در پایان مسیر ذکر شده یک ستاره نیز قرار داد. برای مثال \*c:\svn\.

## فصل ۲ - انجام کارهای متداول کنترلی با TortoiseSVN

### مقدمه:

نحوه استفاده عمومی از SVN به صورت زیر است:

- فضایی در مخزن کدها به یک پروژه اختصاص داده می‌شود.
- سپس اولین نگارش فایل‌های پروژه به مخزن کد وارد می‌شود (check in).
- در ادامه هر یک از اعضای تیم برنامه نویسی، یک نسخه کاری محلی را از سرور مخزن کد دریافت کرده و شروع به کار می‌کنند (check out).
- هر یک از اعضاء در حین کار، نگارش‌های تکمیل شده خود را با سرور هماهنگ می‌کنند (commit).
- در این بین سایر اعضاء می‌توانند با استفاده از SVN، همواره آخرین نگارش تغییرات را اختیار داشته باشند (update) یا تغییرات خود را با سایر اعضای تیم ادغام نمایند (merge).

در ادامه نحوه انجام این امور روزمره کاری را با استفاده از TortoiseSVN که یکی از کلاینت‌های SVN به‌شمار می‌آید، بررسی خواهیم کرد.

### بررسی ویژگی‌های TortoiseSVN و دلایل محبوبیت آن:

- محصولی است رایگان و سورس باز.
- استفاده از آن ساده است:
  - بدلیل یکپارچگی آن با windows explorer، تمامی دستورات اجرایی SVN را می‌توان مستقیماً توسط windows explorer فراخوانی کرد.
  - تنها دستوراتی را نمایش می‌دهد که قابل اجرا بر روی یک فایل یا پوشه هستند و دستوراتی را که مجاز به اجرای آنها نمی‌باشید، نمایش نخواهد داد.
  - نمایش وضعیت جاری فایل‌ها و پوشه‌های پروژه با استفاده از آیکون‌های ویژه.
  - رابط کاربری قوی و پیوسته در حال بهبود و توسعه بر اساس نظرات کاربران.
  - امکان انتقال و یا کپی فایل‌ها و پوشه‌های تحت کنترل را با استفاده از windows explorer فراهم کرده است.
- تمامی پروتکل‌های SVN را پشتیبانی می‌کند. این پروتکل‌ها شامل موارد زیر هستند:
  - http
  - https
  - svn://
  - svn+ssh://
  - file:///
  - svn+xxx://
- دارای صفحات commit بسیار حرفه‌ای با قابلیت‌های بررسی خطاهای نوشتاری، فرمت کردن کد و غیره است.
- امکان تهیه نمودار از انشعابات پروژه توسط آن.
- امکان تنظیمات ویژه تنها برای یک پروژه خاص نیز در آن مهیا است (سفارشی سازی خواص). برای مثال تنظیم حداقل تعداد حروف مورد نیاز در مورد توضیحات یک commit پیش از ارسال آن به سرور مخزن کد.
- امکان یکپارچه سازی با سیستم‌های ردگیری خطاها و باگ‌های برنامه (issue tracking systems).

- همراه آن یک سری از ابزارهای بسیار سودمند جهت کار با فایل‌ها و پوشه‌ها نیز ارائه می‌شود مانند TortoiseMerge ، TortoiseBlame ، TortoiseIDiff و SubWCRev . (در مورد این ابزارها در طی این فصل و فصل‌های آینده بیشتر توضیح داده خواهد شد)
- فایل‌های زبان آن برای بسیاری از کشورهای مختلف دنیا وجود دارد.
- بسیار پایدار است. (این مورد عموماً یکی از ویژگی‌های محصولات سورس باز است که با تعامل با تعداد بیشماری از کاربران در سراسر دنیا تهیه می‌شوند و پیوسته در حال بهبود هستند)
- پشتیبانی خوبی دارد. (اعم از راهنمای بسیار قوی تا mailing list ، forum مخصوص و غیره)

### دریافت آخرین سورس کدها از سرور مخزن کد

اگر شما برنامه نویسی هستید که قصد دارید به یک سرور مخزن کد متصل شوید، برای دریافت آخرین سورس کدها از مخزن کد (جهت ایجاد اولین کپی محلی)، ابتدا پوشه دلخواهی را ایجاد نموده، بر روی آن کلیک راست کرده و مسیر زیر را طی نمایید:

TortoiseSVN → Checkout

در صفحه باز شده آدرس مخزن کد را وارد کرده و بر روی دکمه OK کلیک نمایید (شکل‌های ۱۰ و ۱۱ فصل ۱). پس از مدتی (بسته به سرعت دریافت اطلاعات از شبکه ویا اینترنت) تمامی اطلاعات پروژه را دریافت خواهید کرد. اصطلاحاً به این عملیات تهیه یک نسخه کاری (Working copy) نیز گفته می‌شود.

در صفحه Checkout باز شده اگر Head Revision را انتخاب کنید، آخرین نگارش موجود در مخزن کد را دریافت خواهید کرد و اگر در قسمت Revision عددی را وارد نمایید، آن نگارش ویژه به شما ارائه خواهد شد.

پس از پایان کار اگر بر روی پوشه اصلی این پروژه کلیک راست نموده و گزینه خواص را انتخاب کنید، برگه جدیدی به نام Subversion به آن اضافه شده‌است که یک سری اطلاعات آماری را از وضعیت فعلی آن ارائه می‌دهد. عملیات checkout توسط دستور خط فرمان زیر نیز قابل اجرا است:

```
C:\> svn checkout file:///c:/svn/repo1 c:\project
```

### دریافت اطلاعات از یک Client و وارد کردن آن به سرور مخزن کد

اگر مخزن کدی را بر روی سرور ایجاد کرده‌اید و قصد دارید پروژه‌ای را از یک کلاینت به آن وارد نمایید بر روی پوشه آن کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → Import

در صفحه بازه شده آدرس کلاینت عموماً شبیه به لینک استاندارد زیر می‌باشد:

```
svn://(servername)/(NameOfRepository)/(ProjectSource)/trunk
```

این عملیات با دستور خط فرمان زیر هم قابل انجام است:

```
C:\> svn import c:\temp\project file:///c:/svn/repo1 -m "Initial Load"
```

### تحت کنترل قرار دادن یک فایل جدید

اگر فایل یا پوشه جدیدی را اخیراً ایجاد کرده‌اید، برای اینکه آنرا تحت کنترل مخزن کد قرار دهید، بر روی آن کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → Add

پس از آن باید آنرا با مخزن کد هماهنگ نمایید. برای این منظور مجدداً بر روی فایل یا پوشه کلیک راست نموده و مسیر زیر را طی کنید:

TortoiseSVN → Commit

همچنین اگر علاقمند نیستید که یک یا چند فایل و یا حتی زیر پوشه در پوشه مورد نظر شما تحت کنترل قرار گیرند، بر روی آنها کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → Add to ignore list

یا اگر از اضافه کردن فایلی که هم اکنون با استفاده از گزینه Add آنرا تحت کنترل قرار داده‌اید پشیمان شده‌اید، بر روی فایل کلیک راست نموده و گزینه زیر را انتخاب نمایید:

TortoiseSVN → Undo add

### تغییر نام یا حذف فایل

لازم به ذکر است که تغییر نام یک فایل یا حذف آن به صورت مستقیم از طریق windows explorer سبب از تحت کنترل خارج شدن آن فایل می‌گردد. بنابراین جهت تغییر نام و یا حذف یک فایل و ثبت این وقایع در مخزن کد از یکی از دو گزینه زیر پس از کلیک راست بر روی آن فایل یا پوشه استفاده نمایید:

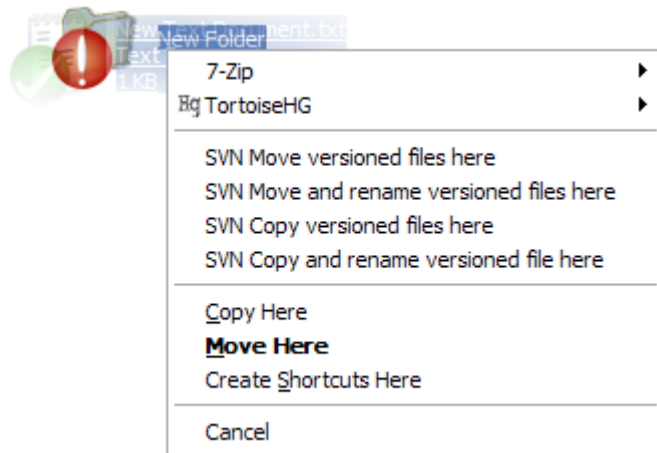
TortoiseSVN → Rename

TortoiseSVN → Delete

### انتقال فایل به مکانی دیگر

انتقال (move) یک فایل تحت کنترل نیز نباید به صورت متداول صورت گیرد. برای این منظور روی فایل کلیک راست نمایید، اکنون فایل را در همین حال به مکان مورد نظر کشیده و رها کنید (drag & drop). سپس از منوی ظاهر شده گزینه SVN move versioned files here را جهت اتمام کار انتخاب نمایید. اکنون به پوشه اصلی پروژه بازگشته، بر روی آن کلیک راست نمایید و سپس TortoiseSVN → Commit (شکل ۱).





شکل ۱- نحوه انتقال یا کپی یک فایل تحت کنترل Subversion با استفاده از کلیک راست، کشیدن و رها کردن.

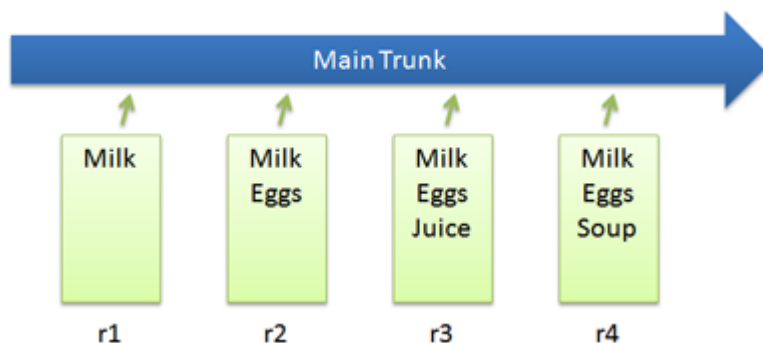
### ارسال آخرین تغییرات انجام شده بر روی سورس کدها به سرور مخزن کد

پس از دریافت آخرین کدها از سرور و یا افزودن پروژه به یک مخزن کد (که در فصل یک توضیح داده شد)، با اعمال تغییرات بر روی کدها، آیکون آنها در windows explorer به رنگ قرمز درخواهد آمد و این مورد بیانگر تغییرات انجام شده توسط شما بر روی این فایل‌ها و یا پوشه‌ها است (نسبت به نمونه همگام شده با سرور). برای همگام سازی سرور با تغییرات انجام شده و ارسال آخرین تغییرات به سرور، بر روی فایل یا پوشه‌های تغییر کرده کلیک راست کرده و سپس مسیر زیر را انتخاب کنید:

TortoiseSVN → Commit

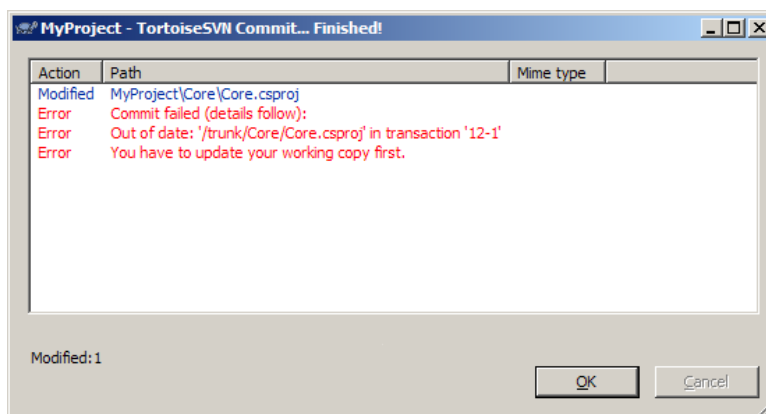
در صفحه باز شده (شکل ۱۳ فصل یک)، توضیح درخوری را وارد کرده، فایل‌هایی را که باید با سرور همگام شوند مرور کرده و سپس بر روی دکمه OK کلیک نمائید.

## Basic Checkins



شکل ۲- با هر بار هماهنگ سازی پروژه با مخزن کد، یک شماره بازنگری جدید دریافت خواهیم کرد.

باید در نظر داشت اگر در یک تیم برنامه نویسی کار می‌کنید همواره پیش از اجرای گزینه Commit از گزینه Update استفاده نمائید. زیرا اگر فایلی که شما بر روی آن مشغول به کار هستید توسط برنامه نویس دیگری تغییر یافته و با مخزن کد هماهنگ شده باشد، شما مجاز به Commit تغییرات خود نخواهید بود (شکل ۳)، زیرا نسخه شما یک نسخه قدیمی محسوب می‌شود.



شکل ۳- همیشه پیش از انتخاب گزینه Commit از گزینه Update استفاده کنید.

آخرین نسخه‌ای که در مخزن کدها قرار دارد، Head نامیده می‌شود و نسخه‌ای که شما هم اکنون مشغول به کار بر روی آن می‌باشید، Base نام دارد. SVN جهت بررسی قدیمی بودن نگارش فایل شما، این دو نگارش را با هم مقایسه می‌کند. اگر تعداد فایل‌های زیادی را در لیستی که قرار است با مخزن کد همگام شود مشاهده می‌کنید، می‌توان موارد اضافی را در لیست موارد صرفنظر شونده که در فصل قبل توضیح داده شد، اضافه نمود.

### به روز رسانی کپی محلی با آخرین تغییرات انجام شده توسط تیم برنامه نویسی

برای اینکه تفاوت‌های نمونه موجود محلی را با آخرین نمونه موجود بر روی سرور پیش از update مشاهده نمائیم، بر روی فایل یا پوشه مورد نظر کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → Check for Modifications

جهت دریافت تغییرات اعمال شده توسط سایر اعضای تیم برنامه نویسی بر روی یک فایل یا یک پوشه، بر روی آن کلیک راست کرده و مسیر زیر را طی کنید:

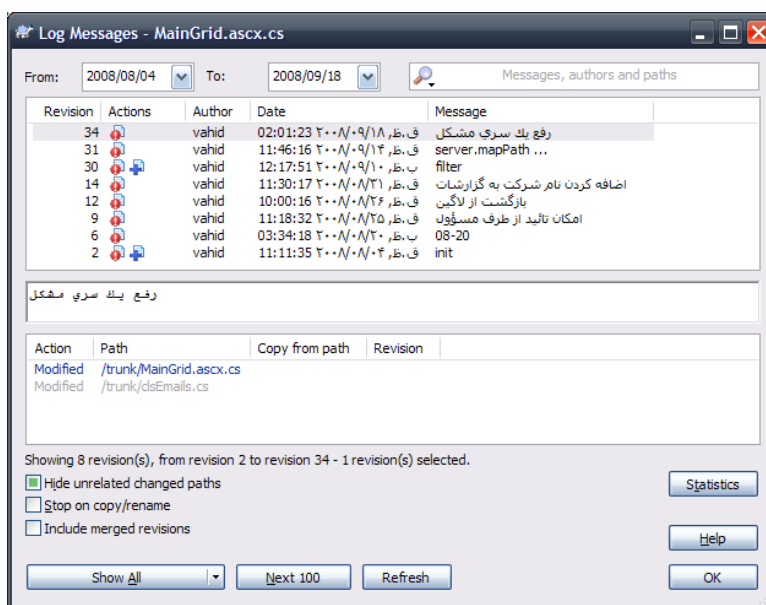
TortoiseSVN → Update

در این حالت دستور update آخرین تغییرات اعمال شده را از سرور دریافت کرده و با فایل و یا فایل‌های شما یکی می‌کند. تنها در صورتیکه تداخلی وجود داشته باشد، پیغام خطایی را دریافت خواهید کرد (در مورد تداخلات در ادامه بحث خواهد شد).

### مشاهده جزئیات تغییرات انجام شده بر روی یک فایل در نگارش‌های مختلف آن

برای مشاهده لیست تغییرات انجام شده بر روی یک فایل، بر روی آن کلیک راست کرده و مسیر زیر را طی نمائید:

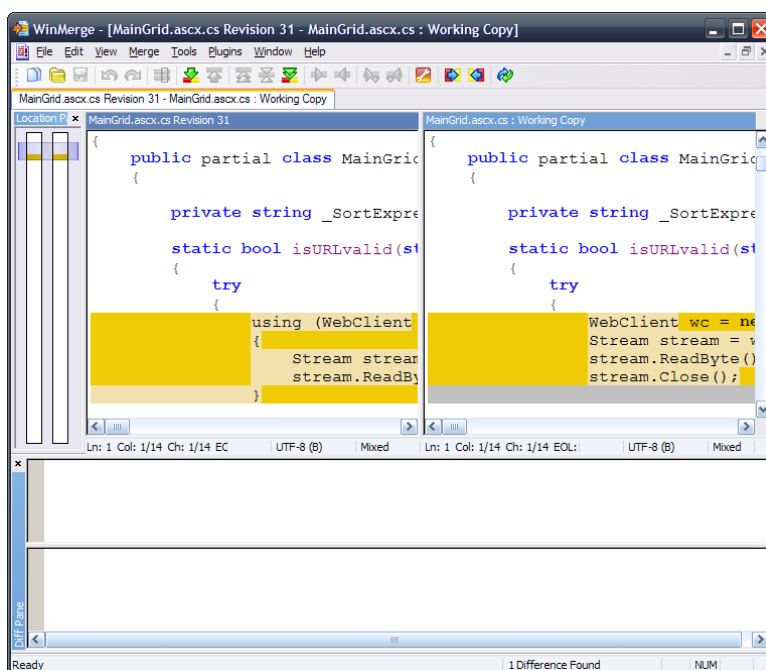
TortoiseSVN → Show log



شکل ۴- مشاهده لیست تغییرات انجام شده بر روی یک فایل

در صفحه ظاهر شده (شکل ۴)، لیست تمامی تغییرات همگام شده با سرور را مشاهده خواهید کرد. برای مقایسه هر یک از موارد لیست شده با آخرین نسخه کاری موجود، بر روی آن مورد کلیک راست کرده و گزینه **compare with working copy** را انتخاب نمایید. در صفحه‌ای که ظاهر خواهد شد (شکل ۵)، با استفاده از الگوریتم‌های Diff، ریز تغییرات را مشاهده خواهید نمود. در اینجا از ابزار Diff پیش فرض استفاده خواهد شد. اگر امکانات آن برای شما راضی کننده نیست می‌توانید از سایر ابزارهای خارجی دیگر نیز استفاده کنید. برای نمونه، برنامه رایگان WinMerge انتخاب بهتری است. این برنامه را از آدرس زیر می‌توانید دریافت کنید:

<http://www.winmerge.org/>

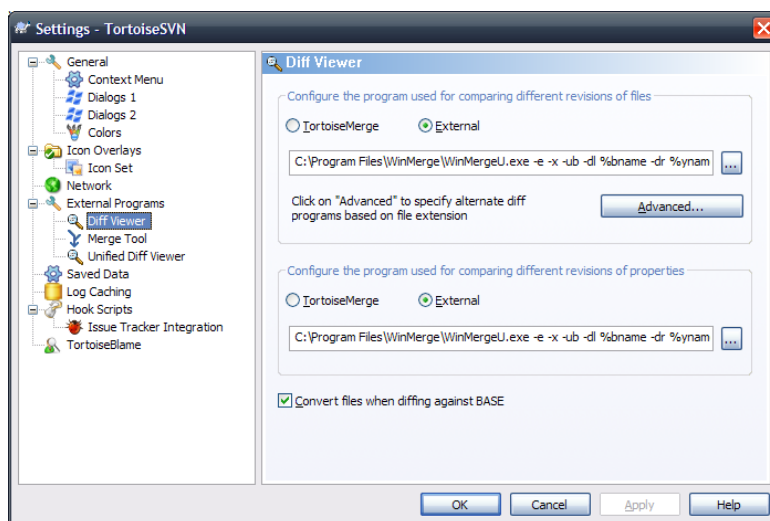


شکل ۵- مشاهده ریز تغییرات انجام شده یک نگارش در مقایسه با نگارش دیگر.

برنامه WinMerge هنگام نصب، خود را با TortoiseSVN یکپارچه می‌کند. این مورد در صفحه تنظیمات برنامه TortoiseSVN قابل مشاهده است (شکل ۶).

مثالی دیگر برنامه غیر رایگان Beyond Compare است که نسخه آزمایشی آن از آدرس زیر قابل دریافت است (در راهنمای موجود در سایت آن نحوه یکپارچه سازی آن با TortoiseSVN نیز ذکر شده است):

<http://www.scootersoftware.com>



شکل ۶- یکپارچه شدن برنامه WinMerge با TortoiseSVN.

#### نکته:

TortoiseSVN پس از نصب، امکان مقایسه دو نگارش مختلف فایل‌های MS-Word را نیز مهیا می‌کند. اینکار به صورت خودکار با استفاده از امکانات درونی MS-Word انجام خواهد شد و نیاز به تنظیم خاصی و یا برنامه جانبی اضافی ندارد. بنابراین از TortoiseSVN می‌توان جهت مدیریت اسناد MS-Word یک سازمان نیز استفاده کرد (همانند SharePoint مایکروسافت و کتابخانه اسناد آن با تمامی قابلیت‌های آن اعم از نگهداری نسخ مختلف، تنظیم سطوح دسترسی، مشاهده تغییرات، دسترسی تحت وب و غیره).

سایر فرمت‌هایی را که با روش‌های مشابه پشتیبانی می‌شوند، در پوشه زیر می‌توانید مشاهده نمائید:  
C:\Program Files\TortoiseSVN\Diff-Scripts

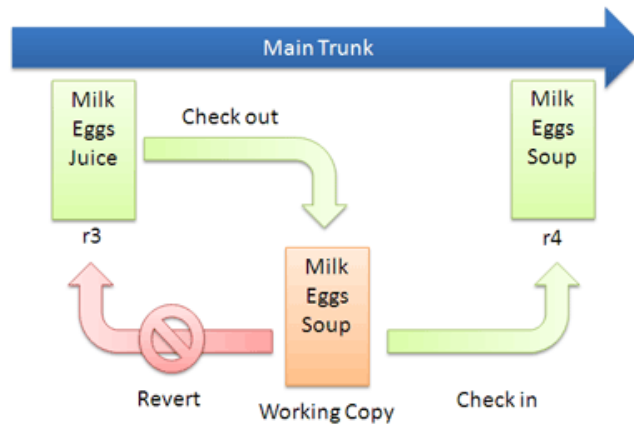
#### نحوه بازگشت به نگارش‌های قبلی

فرض کنید در حال کار بر روی یک پروژه هستید و ناگهان محیط مجتمع برنامه نویسی (IDE) شما از کار افتاده و crash می‌کند. پس از گشودن مجدد این IDE، با فرمی کاملاً تخریب شده مواجه می‌شوید. در این حالت باید چکار کرد؟ سریعترین راه برای بازیابی آخرین کپی قابل استفاده فایل تخریب شده چیست؟ اینجاست که ارزش SVN مشخص می‌شود. در این موارد تنها کافی است بر روی فایل کلیک راست کرده و مسیر زیر را انتخاب کنید:

TortoiseSVN → Revert

پس از انتخاب گزینه revert، فایل تغییر کرده با آخرین نمونه هماهنگ شده با مخزن کدها به روز رسانی خواهد شد.

## Checkout and Edit



شکل ۷- با استفاده از گزینه revert می‌توان به‌سادگی به نگارش‌های قبلی بازگشت.

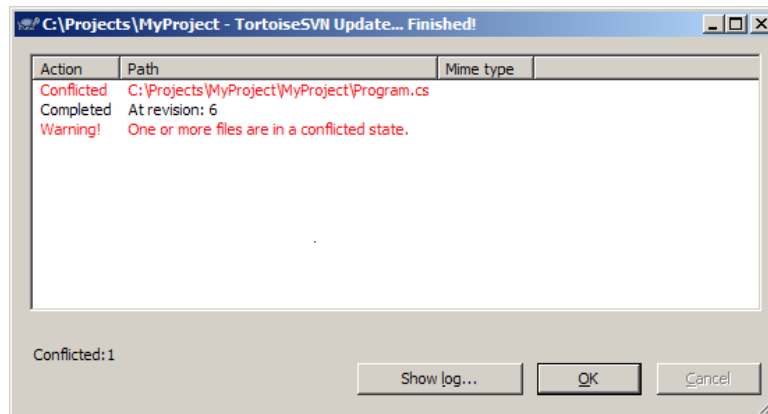
اگر نیاز داشتید تا یک فایل یا یک پوشه را برای مثال به سه نگارش قبل که از نظر شما پایدارتر است بازگردانید، تنها کافی است با انتخاب گزینه show log در TortoiseSVN، از لیست ظاهر شده نگارش مورد نظر را انتخاب کنید، سپس بر روی آن کلیک راست کرده و گزینه revert to this revision را انتخاب کنید. بدیهی است پس از انجام این بازگشت، نگارش‌های دیگر ارسال شده به سرور محفوظ بوده و هر زمان که نیاز بود می‌توان به همین روش به آنها بازگشت نمود. روش دیگر انجام اینکار استفاده از گزینه زیر است:

### TortoiseSVN → Update to Revision

همانند صفحه Checkout، در اینجا نیز اگر Head Revision را انتخاب کنید، آخرین نگارش موجود در مخزن کد را دریافت خواهید کرد و اگر در قسمت Revision عددی را وارد نمایید، به آن نگارش ویژه باز خواهید گشت.

## رفع تعارضات، تداخل‌ها یا تصادم‌ها (conflicts)

با توجه به اینکه تداخلات در یک تیم نرم افزاری محثی شایع است، این مورد را به تفصیل مورد بررسی قرار خواهیم داد. در یک تیم کاری ممکن است زمانیکه یک برنامه نویس مشغول تمام کردن وظایف محوله خویش است، برنامه نویس دیگری کار خویش را به پایان رسانده و تغییرات جدید را به مخزن کد ارسال کرده باشد (شکل ۸). بنابراین همانطور که پیش‌تر نیز ذکر شد باید ابتدا آخرین تغییرات هماهنگ شده با سرور را با استفاده از گزینه update دریافت کرده و سپس نسبت به ارسال تغییرات خود اقدام نمائیم. SVN عملیات یکی کردن تمامی تغییرات موجود را به خوبی مدیریت کرده و در این‌باره مشکلی وجود ندارد (تمامی قسمت‌های جدیدی که سایر برنامه نویس‌ها بر روی آن کار کرده و با سرور هماهنگ کرده‌اند به صورت خودکار به سرور کد موجود شما اضافه خواهد شد). اما مشکل زمانی رخ خواهد داد که یک یا چند خط از کدهای درحال به روز رسانی شما با نمونه موجود در سرور مخزن کد متفاوت باشد (شکل ۹). در این حالت یک Conflict یا تداخل به شما گزارش خواهد شد (شکل ۱۰)، همچنین در کنار آیکون فایل نیز یک علامت تعجب ظاهر خواهد شد.



شکل ۸- گزارش تداخل توسط SVN.

برای رفع تداخل گزارش شده، ابتدا باید با سایر اعضای تیم برنامه نویسی مشورت نموده و سپس نگرارش مورد تأیید را انتخاب نمود (شکل‌های ۱۱ و ۱۲).

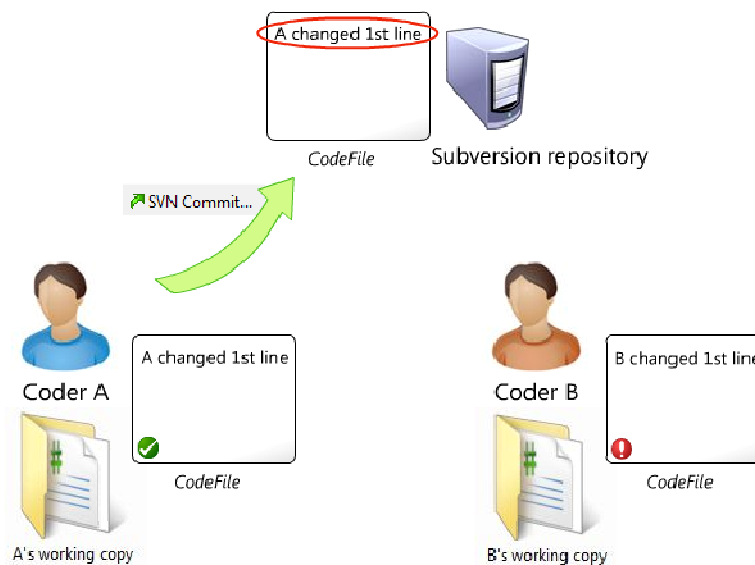
پس از مشخص شدن نگرارش مورد توافق، روی فایل کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → Edit conflicts

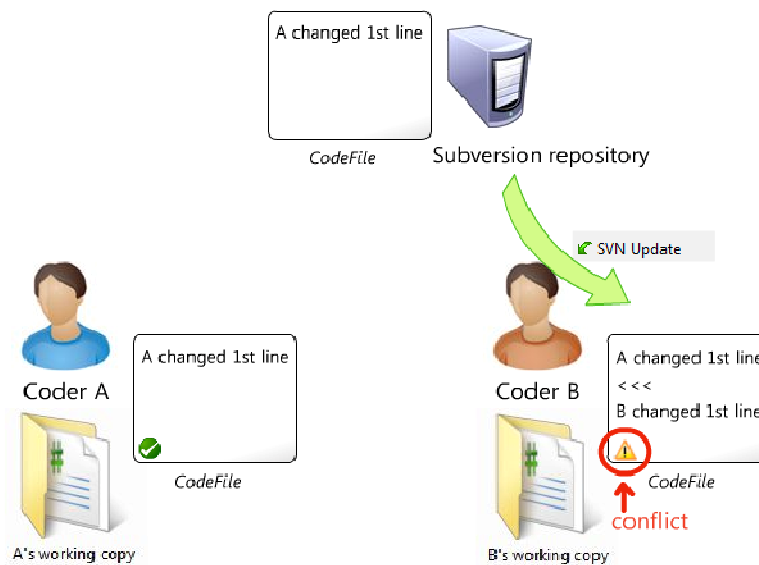
در صفحه باز شده (توسط برنامه TortoiseMerge)، روی سطر مورد نظر کلیک راست نموده و گزینه use this block را انتخاب کنید (گزینه‌های دیگری مانند انتخاب کل فایل و یا انتخاب تغییرات دیگران نیز در اینجا مهیا است). فایل را ذخیره کرده و از برنامه خارج شوید. اکنون مجدداً روی فایل کلیک راست نموده و گزینه زیر را جهت اعلام رفع تعارض به SVN انتخاب نمایید (شکل ۱۳):

TortoiseSVN → Resolved

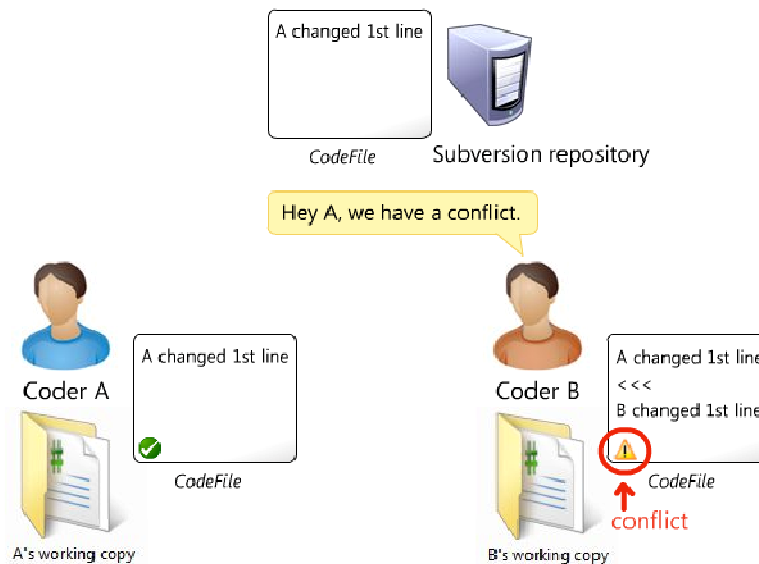
در پایان فایل نهایی را با استفاده از گزینه‌های Update و سپس Commit با سرور هماهنگ کنید (شکل‌های ۱۴ و ۱۶). به این صورت سورس کد نهایی توافق شده با سرور هماهنگ گردیده و تداخل برطرف خواهد شد (شکل ۱۶).



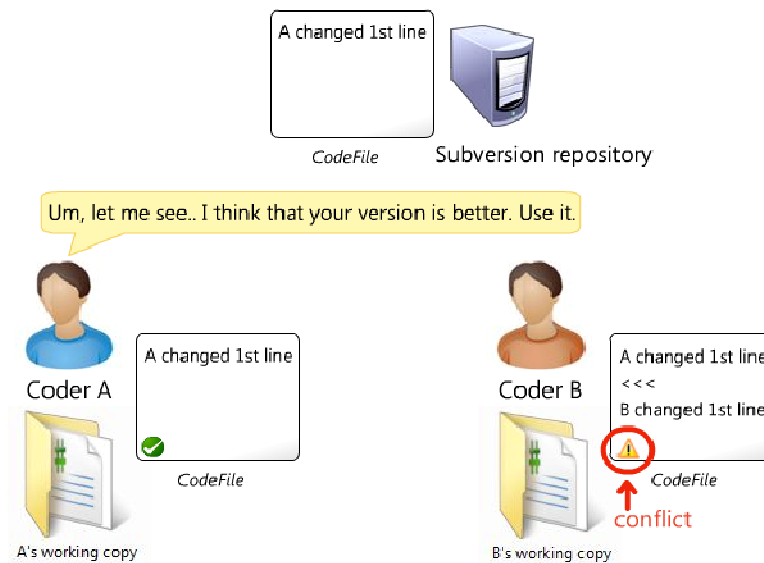
شکل ۹- در اینجا دو برنامه نویس سطر مشابهی از یک فایل را تغییر داده‌اند و برنامه نویسی A، نمونه نهایی خود را با سرور هماهنگ کرده است.



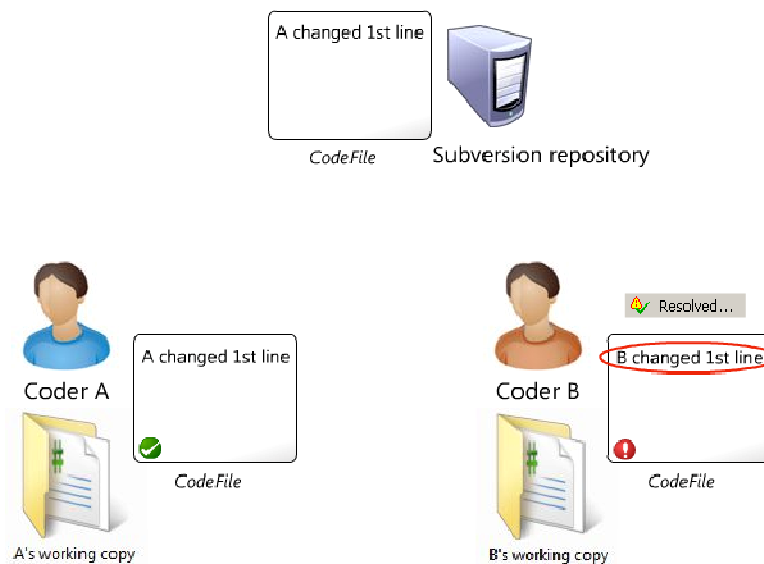
شکل ۱۰- برنامه نویس B قادر به اجرای کامل گزینه update نخواهد بود زیرا در سطر یک با برنامه نویس A تعارض وجود دارد.



شکل ۱۱- موارد گزارش شده به صورت تداخل را باید پس از هماهنگی با تیم برنامه نویسی برطرف نمود.

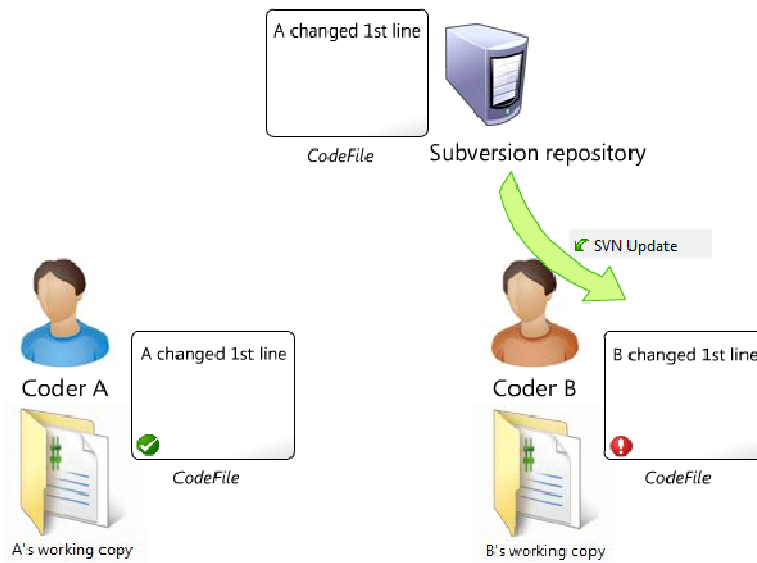


شکل ۱۲- در اینجا پس از مشورت با سایر اعضای تیم، نگارش مورد توافق انتخاب می‌گردد.

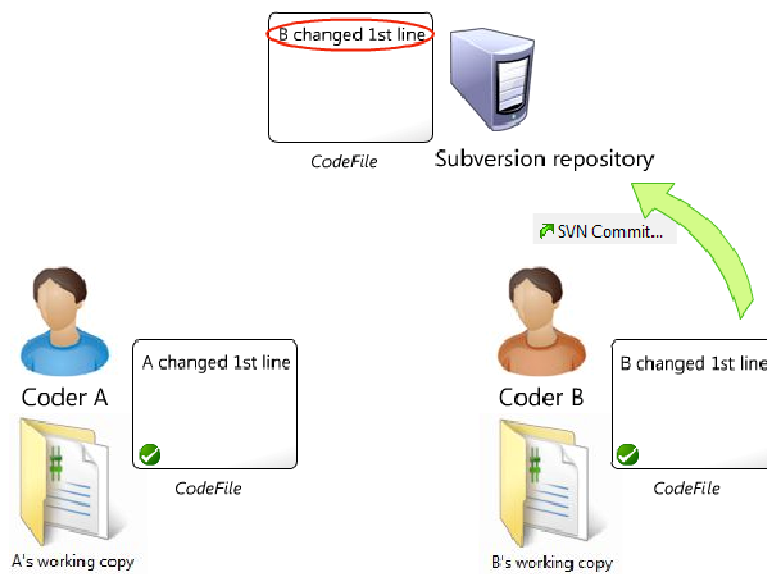


شکل ۱۳- اعلام رفع اختلاف و تعارض به SVN.

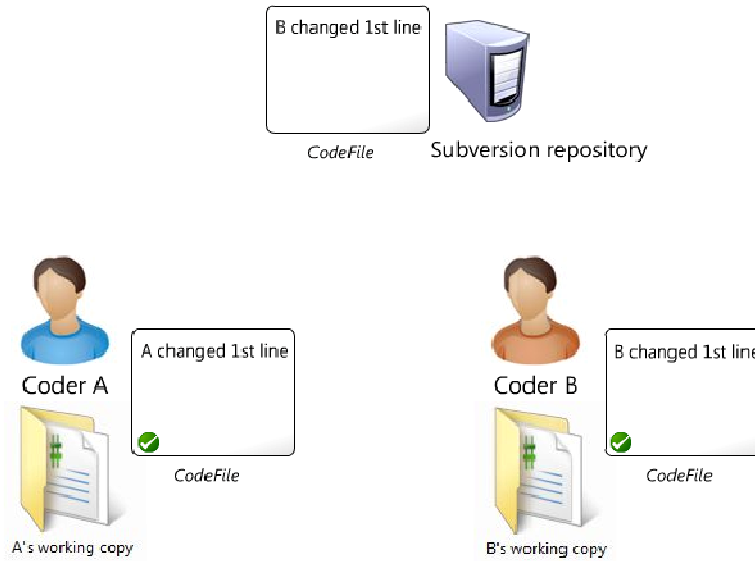




شکل ۱۴- به روز رسانی نسخه کاری با SVN .



شکل ۱۵- هماهنگ کردن کد مورد توافق با SVN .



شکل ۱۶- نتیجه نهایی رفع تداخل، هماهنگ شدن سطر یک تغییر یافته توسط برنامه نویس B با سرور مخزن کد است.

### دو نکته مهم برنامه نویسی تیمی

- همیشه پیش از commit از update استفاده کنید. (با دلیل این نکته در قسمت قبل آشنا شدیم)
- تنها نمونه کاری مطمئن و آزمایش شده را با trunk همگام کنید. اگر کدی که مشغول به کار بر روی آن هستید دارای ایراد است بهتر است یک انشعاب جدید و ایزوله برای آن ایجاد کرده و پس از پایان کار آنرا با خط اصلی کاری (trunk) ادغام نمائید. (در مورد انشعابات در فصلی جداگانه به تفصیل بحث خواهیم کرد)

### منع سایر کاربران از تغییر دادن یک فایل یا پوشه

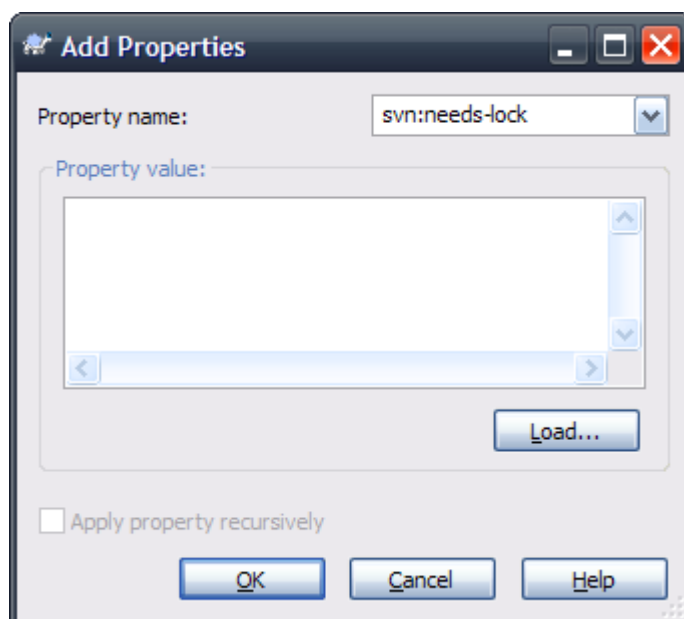
سناریوی فوق در مورد فایل‌های متنی کاملاً کارآ است. اما فرض کنید که در پروژه شما یک فایل تصویری بایناری نیز موجود است. اگر دو کاربر این فایل را تغییر دهند، امکان یکی کردن نتایج حاصله وجود ندارد و در این حالت یکی از آنها تغییرات خود را کاملاً از دست خواهد داد. در اینجا امکان قفل کردن یک فایل نیز فراهم شده است (همانند SourceSafe مایکروسافت). در این حالت تنها یک کاربر می‌تواند مخزن کد را به روز رسانی کند.

برای پیاده سازی عملیات قفل کردن یک فایل یا یک پوشه، باید خاصیت `svn:needs-lock` را به آنها اعمال کرد. برای این منظور روی فایل یا پوشه مورد نظر کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → Properties

در صفحه باز شده بر روی دکمه New کلیک نمائید تا بتوان خاصیت جدیدی را افزود (شکل ۱۷). در اینجا خاصیت `svn:needs-lock` را انتخاب نموده و بر روی دکمه OK کلیک کنید. سپس این تغییر را با سرور هماهنگ نمائید. بر روی فایل کلیک راست نموده و مسیر زیر را طی کنید:

TortoiseSVN → Commit



شکل ۱۷- افزودن خاصیت svn:needs-lock.

پس از این هماهنگ سازی، آیکون فایل نیز تغییر خواهد کرد و فایل به حالت فقط خواندنی درخواهد آمد. اکنون اگر سایر کاربران جهت دریافت آخرین نگارش موجود از گزینه Update → TortoiseSVN استفاده کنند، آنها نیز این فایل را با آیکون تغییر یافته و به صورت تنها خواندنی مشاهده خواهند کرد.

در این حالت برای تغییر فایل باید بر روی فایل کلیک راست کرده و گزینه زیر را انتخاب نمود (و یا شکل ۱۸):  
TortoiseSVN → Get lock



شکل ۱۸- محل قرارگیری گزینه Get lock اینبار کمی متفاوت است.

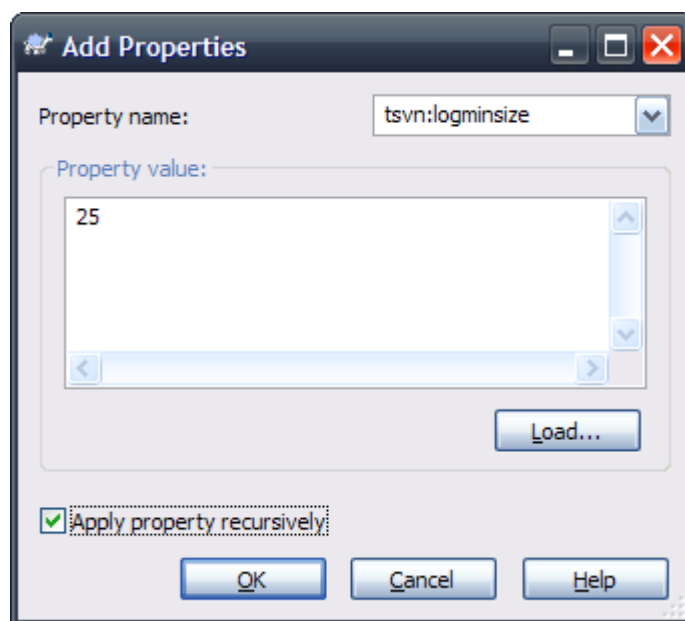
اکنون کاربر، مجاز به ویرایش فایل شده و پس از اعمال تغییرات خود می‌تواند فایل را با سرور هماهنگ نماید. زمانیکه یک کاربر از گزینه get lock استفاده کند هرگونه سعی سایر کاربران دیگر جهت استفاده از این گزینه با یک خطا متوقف خواهد شد. بنابراین در حالت استفاده از خاصیت svn:needs-lock، در هر زمان تنها یک کاربر می‌تواند با فایل مورد نظر کار کند و تا زمانیکه تغییرات کاربری که فایل را قفل کرده با سرور هماهنگ نشود یا آن کاربر از گزینه Release lock → TortoiseSVN استفاده نکند، کاربران دیگر موفق به قفل کردن فایل جهت اعمال تغییرات بر روی آن نخواهند شد. می‌توان از گزینه svn:needs-lock نیز استفاده نکرد. در این حالت بازهم امکان قفل کردن فایل مهیا است اما پس از اعمال تغییرات و هماهنگ سازی آن با سرور، تمامی کاربران همانند قبل و مانند روش متداول کار کردن بدون قفل که حالت پیش فرض SVN است، می‌توانند از فایل استفاده نمایند.

## اجباری کردن وارد نمودن توضیحات مربوط به هر هماهنگ سازی با سرور مخزن کد

برای این منظور برای روی پوشه اصلی پروژه کلیک راست کرده و سپس گزینه زیر را انتخاب کنید:

TortoiseSVN → Properties

در صفحه باز شده بر روی دکمه New کلیک نمائید تا بتوان خاصیت جدیدی را افزود (شکل ۱۹).



شکل ۱۹ - مقدار دهی خاصیت tsvn:logminimize.

خاصیت tsvn:logminimize را از لیست انتخاب کرده و عدد مناسبی را وارد کنید. گزینه Apply property recursively را نیز انتخاب نمائید تا تغییرات به تمامی فایل‌های پروژه اعمال گردد.

پس از این تغییر، تنها کاربرانی که حداقل ۲۵ کاراکتر توضیح در مورد تغییرات خود ارائه دهند، مجاز به هماهنگ کردن تغییرات خود با سرور مخزن کدها خواهند شد (دکمه OK هماهنگ سازی پس از ورود حداقل ۲۵ کاراکتر فعال می‌شود).

اگر می‌خواهید کاربر بدانند حداقل چه مقدار توضیحات را باید وارد کند، خاصیت زیر را نیز به روشی که گفته شده به عدد ۲۵ تنظیم کنید:

tsvn:logwidthmarker

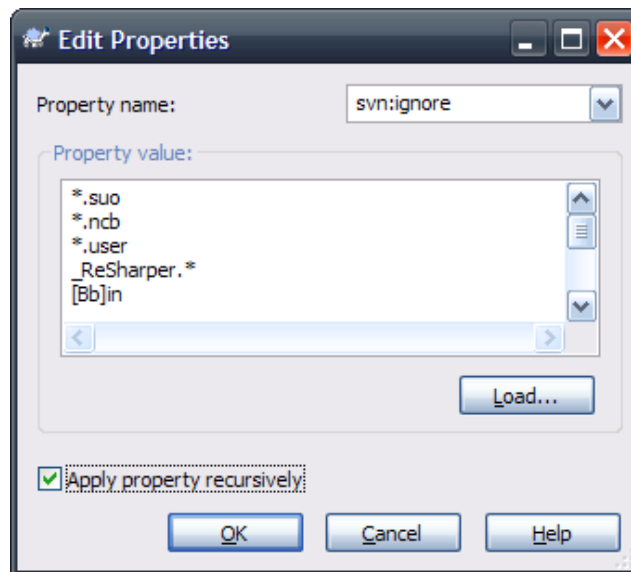
## صرفنظر کردن از یک سری از فایل‌ها، تنها در پوشه‌ای خاص

در فصل قبل در مورد لیست موارد صرفنظر شونده (ignore list) توضیح داده شد. این تنظیمات در سطح کل تمامی پروژه‌های تحت کنترل اعمال می‌شود. اما اگر تنها یک پوشه خاص را مد نظر دارید می‌توان خاصیت زیر را همانند سایر موارد مطرح شده در این فصل به آن پوشه افزود و مقدار دهی کرد (شکل ۲۰):

svn:ignore

و مقدار پیشنهادی آن برای یک پروژه VS.NET می‌تواند به این شکل باشد (هر سطر یک مقدار):

\*.suo  
\*.ncb  
\*.user  
\_ReSharper.\*  
[Bb]in  
obj  
[Dd]ebug  
[Rr]elease  
\*.aps  
\*.eto



شکل ۲۰- مقدار دهی خاصیت svn:ignore.

### چه کسی کدام سطر را تغییر داده است؟

گاهی از اوقات تنها نیاز است بدانیم چه افرادی بر روی یک فایل کار کرده‌اند و همچنین چه تغییراتی را ایجاد نموده‌اند. برای این منظور بر روی فایل کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → Blame

در صفحه باز شده می‌توان مشخص کرد که از چه نگارشی به بعد این گزارش برای شما حائز اهمیت است. مقدار پیش فرض آن از اولین نگارش می‌باشد.

### افزودن شماره بازبینی خودکار به فایل‌های پروژه

تعدادی واژه کلیدی از پیش تعریف شده در SVN وجود دارند که در صورت به‌کارگیری، پس از هماهنگ سازی فایل‌ها با مخزن کد، به عباراتی مشخص بسط داد خواهند شد. به آنها RCS keywords نیز گفته می‌شود. برای مثال برچسب \$Id\$ به عبارتی شبیه عبارت بعد بسط داده می‌شود:

\$Id: NewDocument.txt 32 2008-09-20 18:04:02Z vahid \$

پر کاربرد ترین واژه‌های کلیدی از این نوع به شرح زیر هستند:

- **\$Id\$:**  
پس از بسط شامل نام فایل، شماره بازبینی، تاریخ، زمان و نام نویسنده خواهد شد.
- **\$Date\$:**  
پس از بسط تبدیل به زمان و تاریخ هماهنگ سازی با سرور مخزن کد می‌گردد.
- **\$Rev\$:**  
پس از بسط شامل شماره بازبینی خواهد شد.
- **\$Revision\$:**  
نام دیگری است برای شماره بازبینی

اگر هم اکنون این موارد را تست کنید کار نخواهند کرد. برای فعال سازی آنها باید به ترتیب زیر عمل نمود:  
صفحه تنظیمات TortoiseSVN را باز کنید (TortoiseSVN → Settings). در برگه General بر روی دکمه Edit که جهت ویرایش فایل تنظیمات SVN بکار می‌رود، کلیک کنید. در این فایل، تنظیمات موجود را فعال کنید (به صورت زیر):

[miscellany]

enable-auto-props = yes

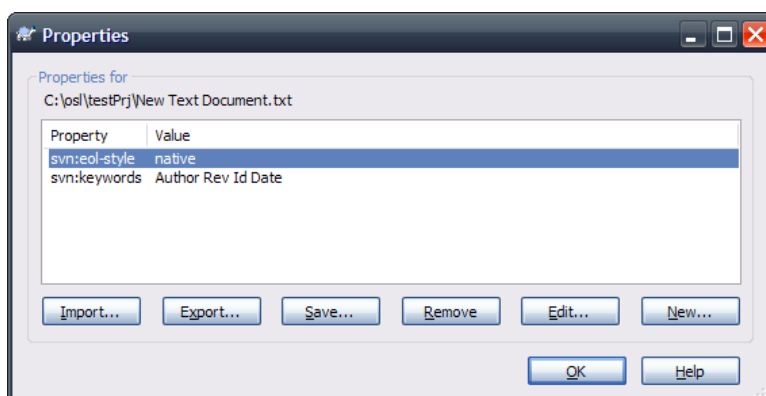
[auto-props]

```
*.c = svn:eol-style=native
*.cpp = svn:eol-style=native
*.h = svn:eol-style=native
*.dsp = svn:eol-style=CRLF
*.dsw = svn:eol-style=CRLF
*.sh = svn:eol-style=native;svn:executable
*.txt = svn:eol-style=native
*.png = svn:mime-type=image/png
*.jpg = svn:mime-type=image/jpeg
*.cs = svn:eol-style=native
```

پس از آن بر روی پوشه یا فایل مورد نظر (و تحت کنترل SVN) کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → Properties

در صفحه باز شده بر روی دکمه New کلیک نمایید تا بتوان خاصیت جدیدی را افزود. ابتدا خاصیت svn:eol-style را انتخاب کرده و مقدار آنرا native وارد نمایید (گزینه اعمال به صورت بازگشتی را نیز انتخاب کنید). پس از اعمال این خاصیت، مجدداً بر روی دکمه new کلیک کنید تا خاصیت svn:keywords را بتوان اضافه نمود. مقدار آنرا مساوی Author Rev Id Date قرار دهید (گزینه اعمال به صورت بازگشتی را نیز انتخاب کنید). پس از اعمال این موارد (شکل ۲۱)، برای تست واژه‌های کلیدی بازبینی، برای مثال \$Id\$ را به یکی از فایل‌ها اضافه کرده و آنرا commit کنید. اکنون فایل را گشوده و نتیجه تغییر را مشاهده نمایید.



شکل ۲۱- تنظیمات مربوط به RCS keywords.

اگر تمایل داشتید تا این واژه‌های کلیدی را به صورت عمومی به تمامی پروژه‌ها اضافه نمائید، کافی در فایل تنظیمات SVN که بیشتر در مورد فعال سازی خطوط آن توضیح داده شد، خطوط مورد نظر را به صورت زیر اصلاح نمائید، برای مثال:

**\*.c = svn:eol-style=native;svn:keywords=Author Rev Id Date**

به این صورت دیگر نیازی به افزودن خواص به فایل‌ها یا پوشه‌های پروژه‌ها نیست و این موارد به صورت عمومی اعمال می‌شوند. استفاده از این نوع شماره‌های خودکار در سورس کدهای برنامه‌ها به صورت توضیحات آنها بسیار متداول هستند. eol در تنظیمات فوق مخفف end of line است و همانطور که می‌دانید eol در ویندوز (carriage return + line feed) و لینوکس (line feed) متفاوت هستند. با انتخاب مقدار آن مساوی native، برنامه SVN بر اساس نوع سیستم عامل موجود رفتار خواهد کرد.

## SVN Cleanup

اگر در حین کار با مخزن کد، سیستم عامل crash کند یا پروسه به هر نحوی نیمه کاره قطع گردد، می‌توان با استفاده از دستور svn cleanup، وضعیت را حالت مطلوبی بازگشت داد. برای این منظور بر روی پوشه پروژه کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → clean up

به این صورت تمامی کارهای نیمه کاره به پایان رسیده و تمامی قفل‌های ایجاد شده نیز برطرف می‌شوند.

## SVN export

در ابتدای فصل با عملیات import به یک مخزن کد آشنا شدید. با استفاده از دستور svn export می‌توان یک کپی از نسخه کاری را بدون پوشه‌های مدیریتی آن (پوشه‌های svn یا \_svn) به دیگران ارائه داد و بدیهی است در این حالت امکان اجرای سایر دستورات مربوط به SVN میسر نخواهد بود. برای انجام اینکار بر روی پوشه پروژه کلیک راست کرده و گزینه زیر را انتخاب نمائید:

TortoiseSVN → export

## ایجاد و اعمال وصله‌ها (patches)

اگر به روند کاری پروژه‌های سورس باز دقت کنید، همیشه عده‌ای هستند که تغییراتی را جهت بهبود کارکرد و یا افزودن یک ویژگی جدید به پروژه تهیه و آنرا به صورت یک فایل وصله (patch) به تیم مدیریتی پروژه ارائه می‌دهند. تیم مدیریتی پس از بررسی وصله و پذیرش آن، آنرا به پروژه اعمال می‌کنند. این فایل وصله حاوی دستور العمل‌هایی است جهت اعمال تغییرات صورت گرفته بر روی یک یا چندین فایل پروژه. به همین جهت برای کار با این نوع تیم‌ها مرسوم نیست که کل پروژه را فشرده کرده و تمامی فایل‌ها را با محتویات کامل آن‌ها جهت بررسی تیم مدیریتی پروژه ارائه دهند. آنها (افرادی خارج از تیم اصلی برنامه که دسترسی مستقیمی به مخزن کد جهت اعمال تغییرات ندارند) تنها فایلی را که حاوی تغییرات انجام شده بر روی تعدادی از فایل‌های پروژه است (فایل وصله یا patch)، ارائه می‌نمایند.

برای ایجاد فایل وصله، بر روی پوشه پروژه کلیک راست کرده و گزینه زیر را انتخاب کنید:  
TortoiseSVN → Create patch...

در اینجا ابتدا فایل‌های مورد نیاز را که در تهیه فایل وصله نقش خواهند داشت انتخاب کرده و سپس بر روی دکمه OK کلیک کنید. در ادامه محل ذخیره سازی فایل وصله پرسیده خواهد شد.  
تیم مدیریتی برای اعمال این فایل وصله، باید بر روی پوشه پروژه کلیک راست کرده و گزینه زیر را انتخاب کنند:  
TortoiseSVN → Apply patch...

با انتخاب این گزینه برنامه TortoiseMerge ظاهر شده و به صورت خودکار مسیر فایل وصله جهت گشوده شدن در برنامه پرسیده خواهد شد. پس از گشودن فایل وصله، لیست فایل‌هایی که تغییر خواهند کرد نمایش داده می‌شود. با دوبار کلیک بر روی نام هر فایل، تفاوت‌های دو فایل قابل مشاهده خواهند بود. اگر علاقمند بودید که تمامی تغییرات را یکجا اعمال نمائید، تمامی فایل‌ها را انتخاب کرده و سپس بر روی لیست فایل‌ها کلیک راست نموده و گزینه patch all را انتخاب کنید. یا برای اعمال تغییرات تنها بر روی یک فایل انتخابی، پس از کلیک بر روی نام فایل در لیست فایل‌های پنجره file patches، گزینه patch selected را انتخاب کنید. همچنین اعمال کنترل ریزتری نیز در اینجا مهیا است. برای مثال خطوطی که پس از وصله شدن به فایل اضافه خواهند شد با علامت + در برنامه TortoiseMerge نمایش داده می‌شوند. اگر نیاز بود که سطری اعمال نگردد، بر روی علامت + دوبار کلیک نمائید تا حالت آن به علامت منفی تغییر یابد.

## استفاده از SVN برای کار با SourceForge و CodePlex

برای مدیریت یک پروژه سورس باز می‌توان از سایت‌هایی که برای این منظور طراحی شده‌اند استفاده کرد. برای مثال sourceforge.net و یا CodePlex.com. SourceForge از لحاظ پذیرش پروژه‌های مختلف تحت سیستم عامل‌ها و زبان‌های برنامه نویسی گوناگون بسیار متنوع است. CodePlex بیشتر به توسعه سورس باز برنامه‌هایی مبتنی بر دات نت مایکروسافت اختصاص دارد. خوشبختانه هر دو سایت امکان مدیریت پروژه نرم‌افزاری را توسط SVN جهت توسعه‌گران فراهم آورده‌اند که در ادامه قصد داریم به این موضوع بپردازیم.

## الف) استفاده از TortoiseSVN برای کار با SourceForge

- قبل از هر کاری مدیر پروژه باید شما را بعنوان یکی از اعضای گروه اضافه نماید تا مجوز هماهنگ سازی کدهای خود را با مخزن کد در SourceForge پیدا نمائید.



- سپس باید پوشه‌ای محلی را جهت پروژه ایجاد کرده و اولین Checkout را انجام دهیم. نکاتی که در اینجا وجود دارند همانند سایر اعمال متداول با TortoiseSVN است و نکته جدید خاصی وجود ندارد. در اینجا آدرس مخزن کد به صورت زیر خواهد بود (به <https> دقت نمائید):  
`https://{project unix name}.svn.sourceforge.net/svnroot/{project unix name}`  
و یا برای مثال آدرس trunk پروژه به صورت زیر است:  
`https://{project unix name}.svn.sourceforge.net/svnroot/{project unix name}/trunk`  
نکته‌ای را که باید بخاطر داشت این است که برای شروع کار Checkout حتما از trunk شروع نمائید در غیراینصورت با انبوهی از انشعابات و برچسب‌ها که در طول مدت زمان انجام پروژه ایجاد شده‌اند مواجه خواهید شد.
- در ادامه، TortoiseSVN از شما خواهد خواست که نام کاربری و کلمه عبور خود را جهت اتصال به مخزن کد وارد نمائید.
- مدتی صبر نمائید تا فایل‌ها از سرور به نسخه کاری شما منتقل شوند.
- از این مرحله به بعد روال کاری همانند روال کاری متداول با یک پروژه تحت کنترل مخزن کد است. همیشه ابتدا با استفاده از گزینه update آخرین تغییرات سایر اعضای گروه را دریافت کرده و سپس نسبت به commit تغییرات خود اقدام نمائید. همچنین دقت نمائید که سرور سورس فورج غیرویندوزی است و نسبت به کوچکی و بزرگی نام فایل‌ها حساس است.
- لازم به ذکر است برنامه‌نویس‌های علاقمندی که خارج از تیم مدیریتی پروژه قرار دارند نیز می‌توانند فایل‌های patch (وصله) خود را در قسمت patches → tracker سایت ارسال نمایند. همچنین این برنامه‌نویس‌ها می‌توانند دسترسی فقط خواندنی نیز به مخزن کد داشته باشند (بدون ارائه نام کاربری و کلمه عبور).

### ب) استفاده از SVN برای کار با CodePlex

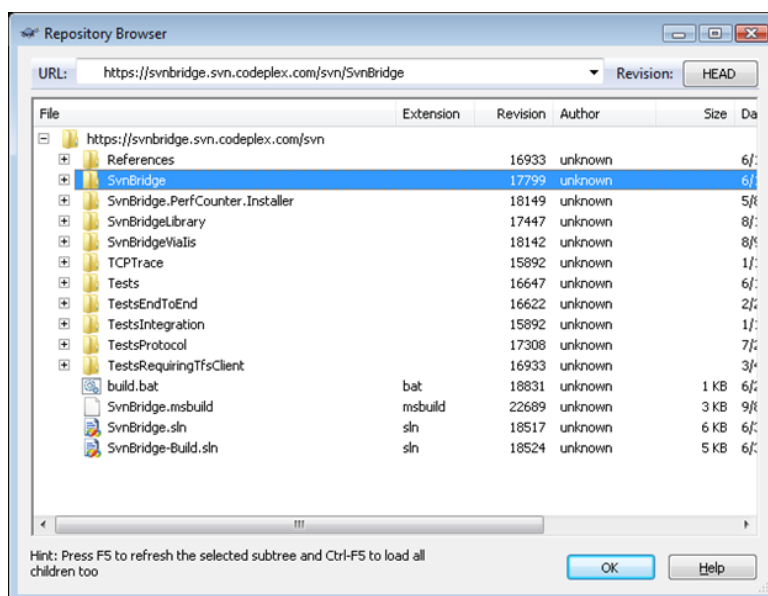
مدتی است مایکروسافت امکان استفاده از یک کلاینت SVN را جهت کار با CodePlex بدون استفاده از برنامه [codeplex.com/SvnBridge](https://codeplex.com/SvnBridge) فراهم کرده است. SvnBridge برنامه‌ای است که به صورت یک پروکسی بین کلاینت SVN و Team Foundation Server مایکروسافت عمل می‌کند. اما با اقداماتی که صورت گرفته است با استفاده از TortoiseSVN بدون نصب هیچگونه پروکسی خاصی با استفاده از آدرس زیر می‌توان با سایت CodePlex نیز ارتباط برقرار نمود:

<https://<projectname>.svn.codeplex.com/svn>

با کمک Repo-browser برنامه TortoiseSVN می‌توان دسترسی فقط خواندنی را نیز به مخزن کد داشت (در آدرس فوق تنها کافی است نام پروژه مورد نظر خود را وارد کنید (شکل ۲۲)). برای مثال آدرس‌های زیر را آزمایش نمائید:

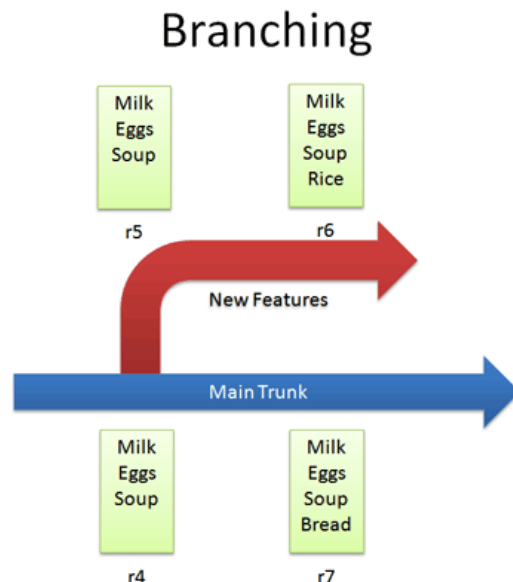
<https://svnbridge.svn.codeplex.com/svn>  
<https://aspnet.svn.codeplex.com/svn/MVC>  
<https://mvcsamples.svn.codeplex.com/svn>

مراحل بعدی کار تقریباً همانند کار با SourceForge است و باید ابتدا پوشه‌ای را ایجاد کرده و پروژه را از سرور مخزن کد Checkout نمائید و سپس مشغول به همکاری با تیم برنامه نویسی شوید.



شکل ۲۲- مرور یک پروژه در سایت CodePlex توسط Repo-browser.

## فصل ۳ – آشنایی با مفاهیم Branching & Tagging



شکل ۱- جهت بررسی و آزمایش ویژگی‌های پیشنهادی جدید در یک پروژه می‌توان انشعابی جدید را معرفی نمود.

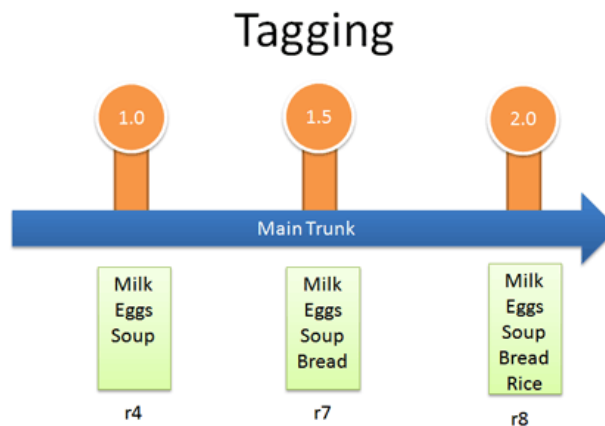
### مقدمه:

برچسب‌ها (tags) و انشعابات (branches) درحقیقت کپی‌هایی از پوشه‌های موجود پروژه هستند که در پوشه‌های ویژه‌ای قرار می‌گیرند تا سایر کاربران متوجه ویژه بودن آنها شوند.

عموماً از برچسب‌ها جهت مشخص کردن یک نگارش ویژه پروژه استفاده می‌شود. SVN پس از هر commit موفق، یک شماره بازنگری خودکار را به عملیات رخ داده انتساب می‌دهد. با استفاده از برچسب‌ها دیگر نیازی نخواهد بود تا این شماره‌های بازنگری را به‌خاطر بسیاری و بدانید که برای مثال سومین ارائه عمومی برنامه مطابق چندمین شماره بازنگری SVN است.

انشعابات اندکی پیچیده‌تر هستند. بجای ایجاد یک کپی و رها کردن آن به وضع خود (همانند بحث برچسب‌ها)، انشعابات بدین جهت ایجاد می‌شوند که قابل تغییر باشند. گاهی از اوقات می‌خواهید تغییرات آزمایشی خود را تحت نظارت SVN انجام دهید اما این تغییرات نباید وارد نسخه اصلی کاری موجود در سرور مخزن کد گردد و سایر برنامه‌نویس‌های تیم را دچار اشتباه سازد. عموماً نگارش‌های آزمایشی که جهت برطرف کردن خطاها با سرور هماهنگ می‌شوند از این دست هستند. پس از اطمینان حاصل کردن از صحت عملیات، می‌توان این انشعابات را با نگارش اصلی موجود در trunk ادغام کرد.

هدف از این انشعابات، تر و تازه نگه داشتن trunk پروژه (که سایر اعضای تیم هم اکنون مشغول به کار بر روی آن هستند) از هرگونه آزمایش و یا کدهای موقتی است.

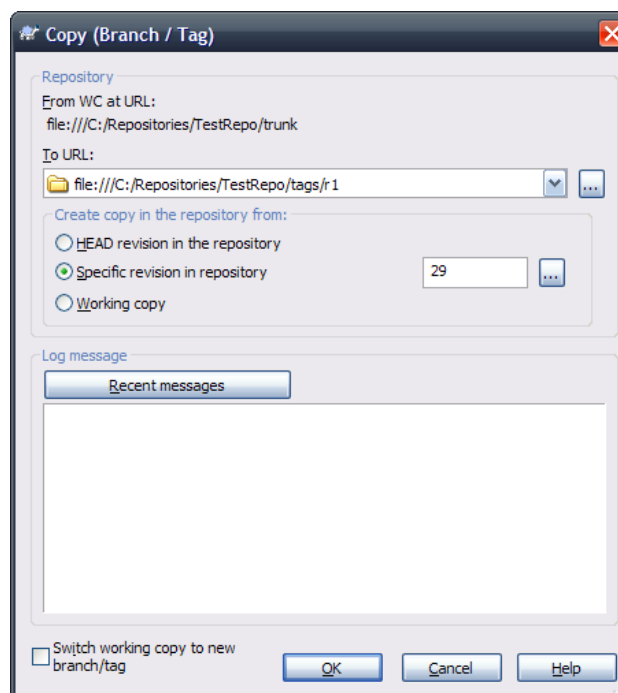


شکل ۲- برچسب دهی به شماره‌های خودکار بازنگری در SVN .

### ایجاد برچسب‌ها و انشعابات با استفاده از TortoiseSVN

برای ایجاد یک برچسب یا انشعاب بر روی فایل یا پوشه مورد نظر کلیک راست کرده و گزینه زیر را انتخاب کنید:  
TortoiseSVN → Branch/Tag

سپس بسته به اینکه مسیر To URL شکل ۳ را پوشه tags انتخاب کنید یا branches ، یک برچسب یا انشعاب جدید ایجاد می‌شود (برای مثال اینجا r1 در انتهای آدرس، جهت ایجاد این برچسب جدید اضافه شده است). همانطور که پیش‌تر نیز ذکر شد، برچسب یا انشعاب هر دو یک کپی از پوشه یا فایل موجود هستند اما مفهوم برچسب، یک کپی فقط خواندنی است و از انشعاب برای آزمایش ایده‌های جدید و یا رفع یک سری از باگ‌ها و سپس ادغام آن با شاخه اصلی پروژه استفاده می‌شود.  
برای تست صحت عملیات صورت گرفته می‌توان به repo-browser برنامه TortoiseSVN مراجعه کرد (شکل ۴).



شکل ۳- ایجاد یک برچسب جدید.

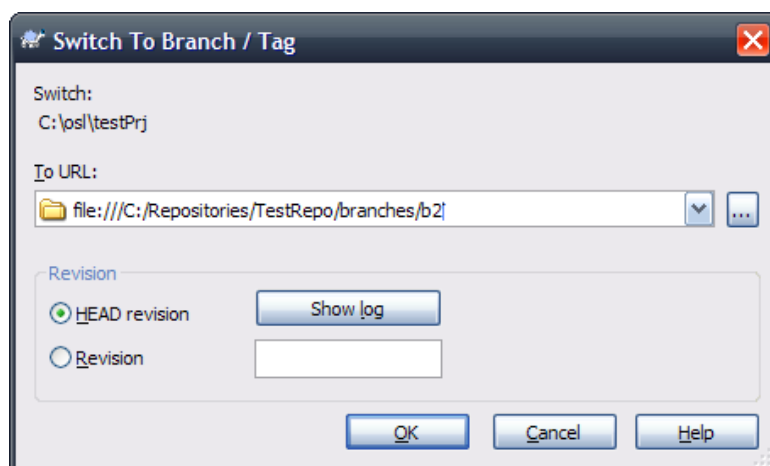


شکل ۴- نمایی از repo-browser پس از ایجاد اولین برچسب.

### سوئیچ کردن بین انشعابات

تا بدینجا موفق به ایجاد یک برچسب یا انشعاب جدید شده‌ایم. اما اگر تغییرات انجام شده را با سرور هماهنگ کنیم مجدداً به انشعاب اصلی یا trunk اضافه خواهند شد. بنابراین نیاز است تا به انشعاب ایجاد شده سوئیچ کرد. برای این منظور بر روی پوشه پروژه یا پوشه و فایل مورد نظر کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → switch



شکل ۵- سوئیچ کردن به یک انشعاب

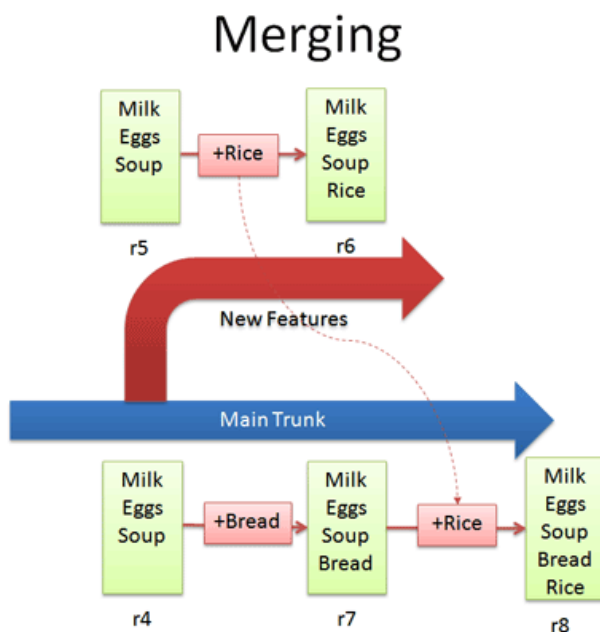
در صفحه باز شده (شکل ۵)، آدرس انشعاب مورد نظر را وارد (برای مثال اینجا b2 در انتهای آدرس، جهت ایجاد این انشعاب جدید اضافه شده است)، شماره بازنگری مناسبی را انتخاب کرده و بر روی دکمه OK کلیک کنید. از این پس هرگونه هماهنگ سازی با مخزن کد به این انشعاب منتقل خواهد شد (این مورد در صفحه commit در قسمت URL آن مشخص می‌شود). بدیهی است در هر زمانی می‌توان از یک انشعاب به انشعاب دیگری سوئیچ کرد. حالت دیگری که ممکن است از switch استفاده شود، تغییر مکان مخزن کد برای مثال به سروری دیگر باشد.

برای مشاهده نمودار انشعابات پروژه در زمان‌های مختلف، بر روی پوشه مورد نظر کلیک راست کرده و گزینه زیر را انتخاب کنید:

TortoiseSVN → Revision graph

### ادغام نتایج انشعابات با انشعاب اصلی پروژه

با توجه به اینکه انشعابات خطوط جدای کاری از خط اصلی پروژه محسوب می‌شوند، پس از پایان عملیات مربوطه، نیاز به ادغام نتایج آنها با خط اصلی کاری پروژه وجود خواهد داشت.



شکل ۶- ادغام نتایج یک انشعاب با خط اصلی کاری پروژه.

برای این منظور بر روی پوشه یا فایل مورد نظر کلیک راست کرده و گزینه زیر را انتخاب کنید:  
TortoiseSVN → Merge

سپس صفحه wizard مربوطه باز خواهد شد. این صفحه سه گزینه دارد که به بررسی آنها خواهیم پرداخت:

- Merge a range of revisions

در این حالت برای مثال می‌توان تغییرات انجام شده از بازنگری ۲ تا ۵ یک انشعاب را دریافت کرده و آنها را با انشعاب اصلی (یا انشعابی دیگر) ادغام کرد (شکل ۷). در صفحه مربوطه در قسمت From، آدرس انشعابی را وارد کنید که قرار است تغییرات موجود در آن با نسخه کاری شما ادغام شود. در قسمت Revision range to merge صفحه ادغام، لیست شماره‌های بازنگری را که باید با یکدیگر ادغام شده و به گزینه From منتقل شوند، وارد نمایید. اگر در این صفحه گزینه Reverse merge را انتخاب کنید، تمامی تغییرات پیشین بر اساس بازه وارد شده به حالت اول باز خواهند گشت (همان مبحث revert).

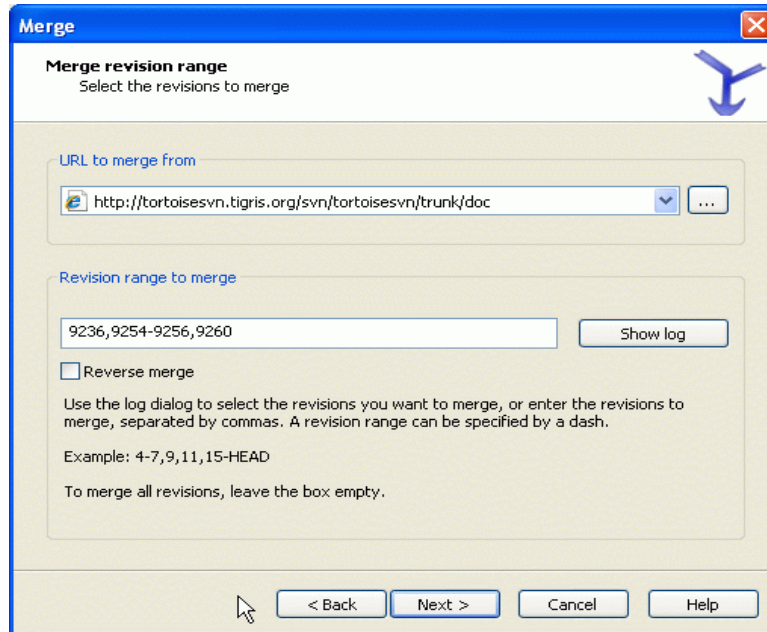
- Reintegrate a branch

فرض کنید انشعابی را جهت ایجاد یک ویژگی جدید در برنامه ایجاد کرده‌اید. در طی توسعه این انشعاب، هر روز آنرا با انشعاب اصلی به روز خواهید کرد تا آخرین تغییرات را نیز در این انشعاب ویژه داشته باشید. پس از پایان کار، ادغام و همگام سازی نهایی را انجام خواهید داد (شکل ۸). در اینجا با توجه به همگام بودن این انشعاب با انشعاب اصلی که در طی روزهای متوالی صورت گرفته، پس از ادغام، دیگری نیازی به آن نخواهد بود و قابل حذف است (زیر اکنون هر دو انشعاب یکسان هستند). در این حالت گزینه From URL، آدرس انشعابی است که باید ادغام گردد.

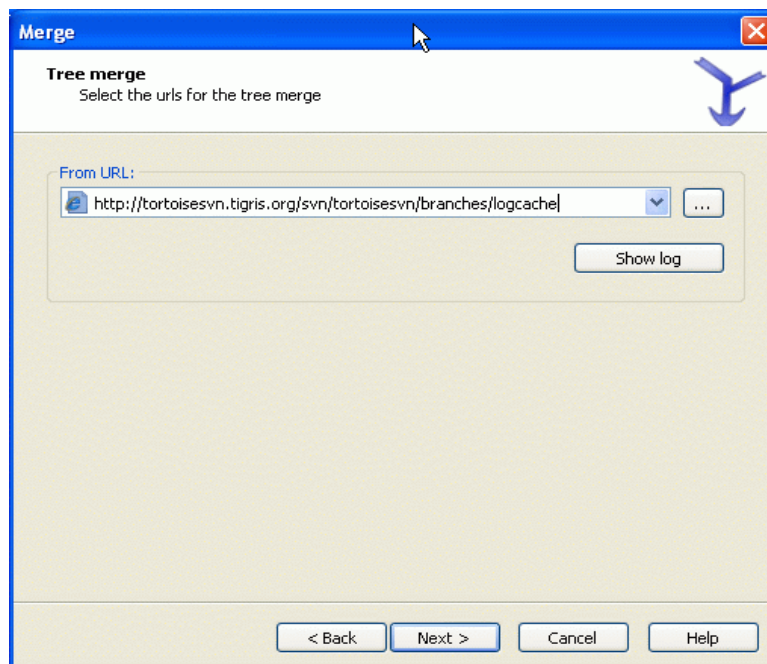
- Merge two different trees

این مورد حالت عمومی گزینه قبل است. از این حالت برای ادغام یک انشعاب و اعمال تغییرات آن به انشعاب اصلی استفاده می‌شود (شکل ۹). نکته مهمی که در اینجا باید به آن دقت داشت این است که در قسمت From صفحه مربوطه باید آدرس trunk وارد شود (زیرا trunk نقطه آغازین اعمال تغییرات انشعاب است). در قسمت To آدرس مربوط به انشعاب ادغام شونده را وارد کنید.

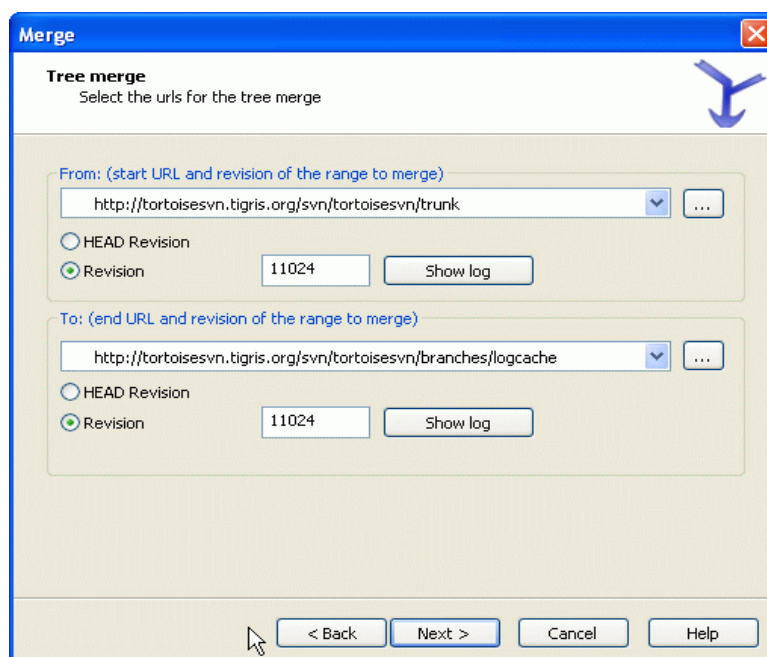
پس از انتخاب یکی از گزینه‌های فوق و تکمیل فرم مربوطه، صفحه بعدی wizard مربوط به تنظیمات ادغام است. پیش فرض‌های این صفحه عموماً صحیح بوده و هدف اصلی ادغام است.



شکل ۷- ادغام بازه‌ای از بازنگری‌ها

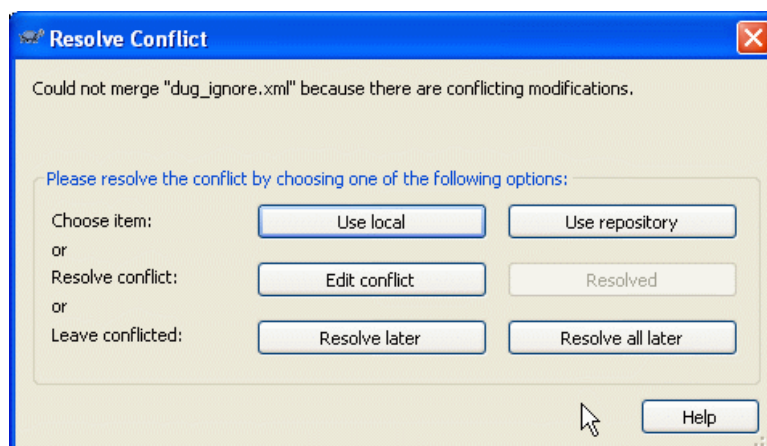


شکل ۸- یکپارچگی مجدد با یک انشعاب



شکل ۹- ادغام دو انشعاب

عملیات ادغام همیشه با موفقیت به پایان نخواهد رسید و ممکن است تداخل‌هایی نیز وجود داشته باشد. در صورت بروز تداخل شکل ۱۰ نمایان خواهد شد.



شکل ۱۰- بروز تداخل در اثر ادغام

در این حالت عملیات رفع تداخل همانند مباحث پیشین باید دنبال گردد.

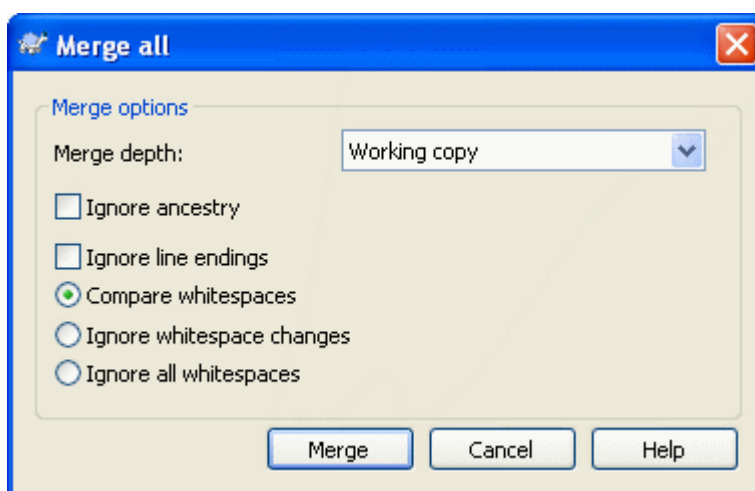
#### نکته:

فرض کنید انشعاب جاری شما، یکی از انشعابات فرعی پروژه است (توسط گزینه سوئیچ آنرا انتخاب کرده‌اید) و اکنون کار این مرحله به پایان رسیده است. برای ادغام آن با trunk گزینه ساده‌تری نیز وجود دارد. دکمه shift را نگه داشته و بر روی پوشه پروژه کلیک راست کنید. سپس گزینه زیر را انتخاب نمایید:

TortoiseSVN → Merge reintegrate...



سپس صفحه ساده زیر نمایان خواهد شد (شکل ۱۱) و در یک مرحله کار به پایان خواهد رسید.



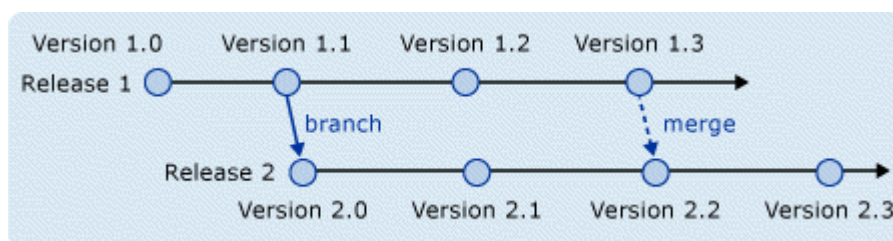
شکل ۱۱- ادغام سریع یک انشعاب با trunk.

### خط مشی‌های متداول در ایجاد انشعابات

ایجاد انشعابات هنگام کار در یک تیم، بازدهی اعضای تیم را بدلیل کار کردن در انشعابی ایزوله بالا می‌برد اما مسایل همراه با ادغام‌های آتی را به همراه خواهد داشت. به همین جهت تیم‌های مختلف خط مشی‌های مختلفی را جهت ایجاد انشعابات دنبال می‌کنند که در ادامه به آنها خواهیم پرداخت:

#### الف) ایجاد انشعاب به ازای هر ارائه عمومی برنامه

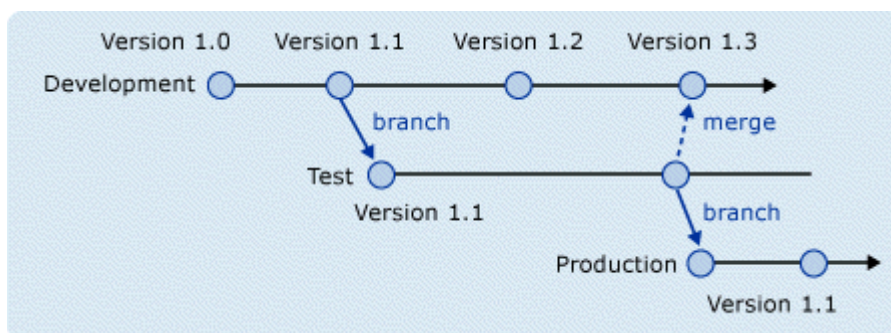
یکی از متداول‌ترین خط مشی‌های ایجاد انشعاب، ایجاد انشعاب به ازای هر ارائه عمومی برنامه است که در شکل ۱۲ نشان داده شده است. هر انشعاب شامل تمامی تلاش‌های یک تیم در هنگام ارائه عمومی برنامه در آن نگارش خاص است. این انشعاب پس از ایجاد خاتمه یافته تلقی شده و از آن جهت نقطه آغازین و به روز رسانی اولیه نگارشی جدید استفاده می‌شود. بنابراین بهتر است این نوع انشعابات را در فرهنگ SVN با یک Tag مشخص نمود.



شکل ۱۲- ایجاد انشعاب به ازای هر ارائه عمومی برنامه.

### ب) ایجاد انشعاب به ازای هر فاز پروژه

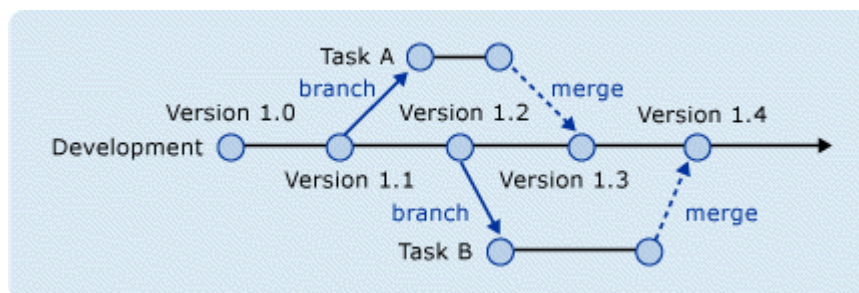
در این روش مطابق و به ازای فازهای تعریف شده پروژه، انشعابی جدید جهت توسعه، آزمایش و تکمیل آن مرحله ایجاد گردیده و پس از اتمام کار با خط تولید ادغام می‌گردد (شکل ۱۳). در حین توسعه در این انشعاب، برنامه نویس همواره خود را با خط اصلی تولید به روز نگه خواهد داشت.



شکل ۱۳- ایجاد انشعاب به ازای هر فاز پروژه.

### ج) ایجاد انشعاب به ازای هر فعالیت جدید

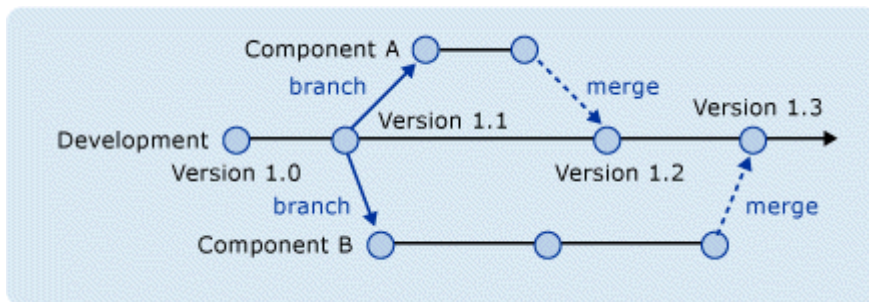
جهت جلوگیری از تداخل وظایف محوله (یا فعالیت‌ها) و بالا بردن بهره‌وری تیمی، می‌توان به ازای هر فعالیت جدید، انشعابی را ایجاد نموده و سپس نتایج آنرا با خط تولید ادغام نمود (شکل ۱۴). این انشعابات باید کوتاه مدت بوده و نتایج آن سریعاً با خط تولید ادغام شوند تا هزینه ادغام‌ها بیشتر از منافع این روش نگردند.



شکل ۱۴- ایجاد انشعاب به ازای هر فعالیت جدید.

### د) ایجاد انشعاب به ازای هر زیر سیستم جدید

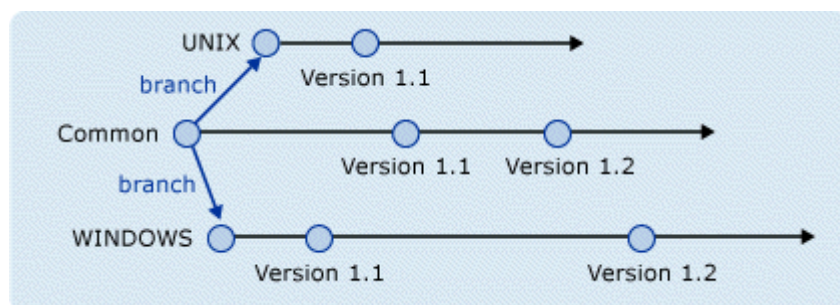
پس از طراحی سیستم، به ازای هر کامپوننت یا زیر سیستم تعریف شده می‌توان یک انشعاب جدید تعریف، کار را در هر انشعاب به صورت ایزوله به پایان رساند و سپس نتیجه را با خط تولید ادغام کرد (شکل ۱۵). در این حالت کنترل بهتری را می‌توان بر تک تک اجزای سیستم داشت.



شکل ۱۵- ایجاد انشعاب به ازای هر زیر سیستم جدید.

### ه) ایجاد انشعاب به ازای فناوری‌های مختلف

اگر سیستم شما قرار است تحت سکوهایی کاری مختلف کار کند، می‌توان به ازای هر کدام یک انشعاب جدید تعریف نمود و همچنین موارد مشترک را در انشعاب common توسعه داد (شکل ۱۶). در این حالت هیچگاه ادغام انشعابات مد نظر نیست.



شکل ۱۶- ایجاد انشعاب به ازای فناوری‌های مختلف.

## فصل ۴ - یکپارچگی SVN با محیط‌های توسعه مجتمع

### مقدمه:

هر چند TortoiseSVN استفاده از SVN را در ویندوز بسیار ساده کرده است اما نیاز به یکپارچگی محیط‌های توسعه مجتمع (IDE) با SVN جهت بالا بردن کارایی برنامه نویس‌ها بدون مراجعه مکرر آنها به windows explorer بیش از پیش احساس می‌شود. بدین منظور افزودنی‌هایی (Add-in) جهت VS.Net و سایر محیط‌های توسعه مجتمع توسعه یافته‌اند که در این فصل به معرفی آنها خواهیم پرداخت.

### الف) یکپارچگی SVN با Visual Studio.Net

#### ساده ترین راه حل موجود: Garry Broadsword's Add-on

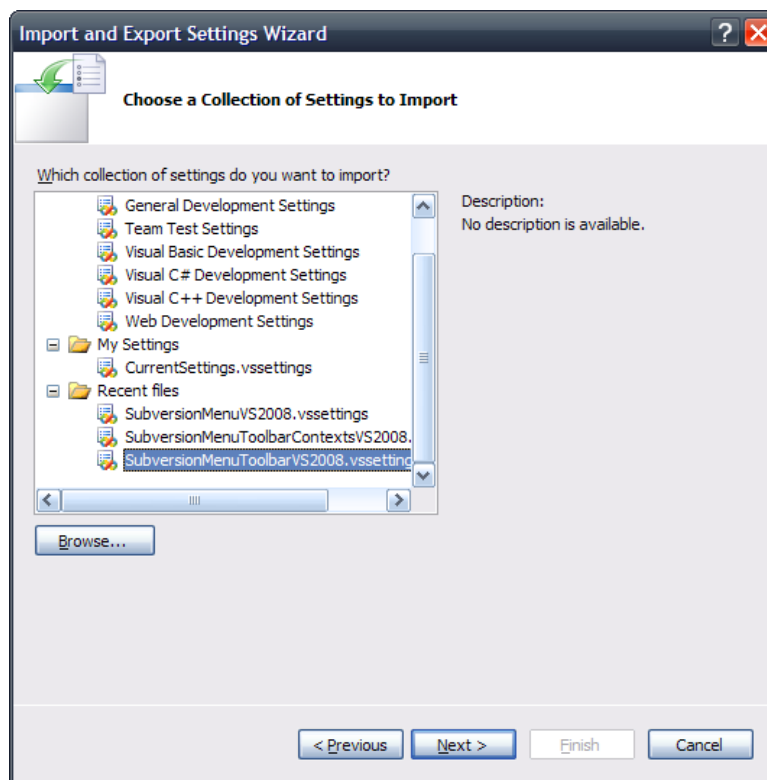
این افزودنی که در حقیقت محصور کننده‌ای (wrapper) برای TortoiseSVN به‌شمار می‌آید را از آدرس زیر می‌توانید تهیه کنید:  
<http://garrys-brain.blogspot.com/2007/07/tortoisesvn-and-visual-studio.html>

با استفاده از این افزودنی تقریباً تمامی توانایی‌های TortoiseSVN را در VS.Net خواهید داشت. این موارد شامل Commit ، Repository ، Resolve ، Edit Conflicts ، Modifications ، Revert ، Blame ، Diff ، History ، Update ، Add Solution ، Project History و Branch/Tag Settings می‌باشند. متأسفانه مبحث تغییر رنگ آیکون‌های فایل‌ها (که توسط TortoiseSVN در windows explorer نمونه آنرا دیده‌اید) به این صورت در VS.Net پشتیبانی نخواهند شد و اگر این مورد برای شما اهمیتی ندارد، این افزودنی رایگان، ساده‌ترین و سبک‌ترین افزودنی موجود (از لحاظ مصرف منابع سیستمی) جهت کار با SVN در VS.Net می‌باشد.  
برای نصب آن به صورت زیر عمل کنید:

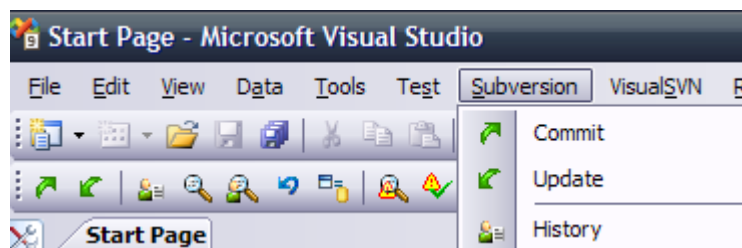
۱- فایل SubversionInstall.vbs را گشوده و بنابر نگارش VS.Net مورد استفاده شما، مقدار متغیر strVisualStudioVersionNumber را ویرایش کنید (به صورت پیش فرض برای VS.Net 2008 تنظیم شده است). همچنین اگر TortoiseSVN را در مکان پیش فرض آن نصب نکرده‌اید، مقدار متغیر strTortoiseSVNBin را نیز ویرایش نمایید. اکنون این اسکریپت را اجرا نمایید (روی آن دوبار کلیک کنید یا بر روی آن کلیک راست نموده، گزینه Open with را انتخاب کرده و سپس بر روی Microsoft windows based script host کلیک کنید).

۲- بسته به نگارش VS.Net خود، فایل‌های vssettings موجود را با استفاده از منوی tools در VS.Net و گزینه Import and Export Settings ، به VS.Net معرفی نمایید (شکل ۱).

نتیجه این عملیات را در شکل ۲ ملاحظه می‌کنید. از همین ایده در IDE های دیگر نیز استفاده شده است.



شکل ۱- معرفی فایل‌های vssettings به VS.Net.



شکل ۲- افزوده شدن منو و نوار ابزار Subversion به VS.Net.

### جامع‌ترین راه حل سورس باز موجود: AnkhSVN

AnkhSVN جامع‌ترین افزودنی سورس باز موجود جهت یکپارچگی VS.Net و SVN بشمار می‌رود. این افزودنی را از آدرس زیر می‌توانید دریافت نمایید:

<http://ankhsvn.open.collab.net/>

برای استفاده از آن نیازی به نصب TortoiseSVN نمی‌باشد و امکانات مشابهی را در VS.Net در اختیار شما قرار می‌دهد (بعلاوه امکان تغییر خودکار آیکون‌های فایل‌ها). پس از نصب آن به منوی View گزینه toolbars در VS.Net مراجعه کرده و موارد مربوط به Source control را انتخاب نمایید تا نوار ابزار آن قابل استفاده گردد.

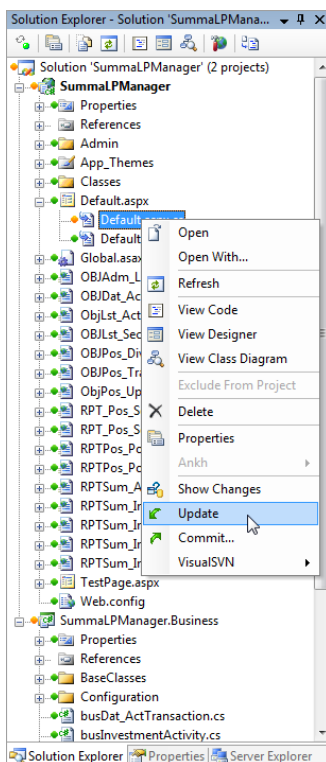
استفاده از این افزودنی بسیار ساده است. برای افزودن یک پروژه به SVN بر روی نام پروژه در لیست solution explorer کلیک راست کرده و گزینه Add solution to subversion را انتخاب کنید (در اینجا فرض بر این است که ایجاد مخزن کد را پیشتر انجام داده‌اید و در صفحه ظاهر شده URL آنرا باید وارد کرد). این افزودنی پس از افزودن یک فایل به پروژه، کار افزودن آن فایل را به

SVN نیز به صورت خودکار انجام می‌دهد و نیازی به دخالت برنامه نویس نخواهد بود و یا برای مثال پس از تغییر هر فایل می‌توان بر روی آن کلیک راست نمود و گزینه Commit را که اکنون به VS.Net افزوده شده است، انتخاب نمود. سایر موارد عنوان شده در فصل‌های پیشین نیز به همین ترتیب در دسترس خواهند بود.

### بهترین راه حل تجاری موجود: VisualSVN

از فصل‌های قبل با VisualSVN Server رایگان آشنا شدید. همان شرکت محصول دیگری را نیز جهت یکپارچگی SVN با VS.Net ارائه داده است که غیررایگان است اما مطابق خواهد بود پرکاربردترین افزودنی جهت کار با SVN در VS.Net به‌شمار می‌رود. این Add-in محصور کننده توانایی‌های TortoiseSVN است بنابراین جهت استفاده از آن حتما باید TortoiseSVN را پیشتر نصب کرده باشید. برخلاف اولین راه‌حل عنوان شده این فصل که آن نیز محصور کننده توانایی‌های TortoiseSVN بود، توسط VisualSVN امکان تغییر خودکار آیکون‌های فایل‌ها در VS.Net نیز مهیا است (همانند TortoiseSVN در windows explorer) و این مورد کمک شایانی را در مورد درک وضعیت فعلی فایل‌های پروژه از نظر هماهنگی با مخزن کد به برنامه نویس خواهد کرد (شکل ۳).

هر چند نگارش‌های اخیر این افزودنی امکانات بیشتری نسبت به یک محصور کننده صرف را ارائه داده‌اند. برای مثال گزینه add solution to subversion آن، به سادگی تمامی مراحل ایجاد مخزن، ایجاد سه پوشه استاندارد trunk، tags و Branches، مباحث check out و check in را به صورت خودکار انجام می‌دهد. بدیهی است پس از این عملیات یکبار باید عملیات commit را انجام داد تا پروژه با مخزن کد هماهنگ گردد. برای این منظور روی نام پروژه در VS.Net کلیک راست کرده و گزینه Commit را که پس از نصب این افزودنی اضافه شده است، انتخاب نمایید. صفحه آشنای Commit فایل‌ها توسط TortoiseSVN ظاهر شده و باقی قضا یا همانند مطالب عنوان شده در فصل‌های قبل است.



شکل ۳- نمایشی از تغییر آیکون‌های فایل‌ها در VS.Net توسط VisualSVN که بیانگر وضعیت هماهنگی آنها با مخزن کد است.

## ب) یکپارچگی SVN با Delphi

### استفاده از File Browser

ساده‌ترین راه حل استفاده از توانایی‌های TortoiseSVN در دلفی، استفاده از گزینه زیر است:

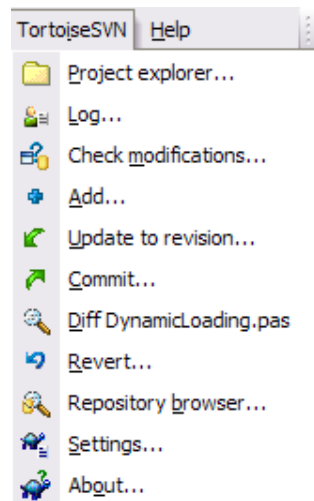
View menu → File Browser

به این صورت از درون IDE دسترسی کاملی به منوهای TortoiseSVN خواهید داشت و همچنین مبحث تغییر آیکون‌های فایل‌ها توسط TortoiseSVN نیز در اینجا کاملاً فعال است. تنها نکته‌ای که باید در مورد دلفی به‌خاطر سپرد، مبحث فایل‌های صرفنظر شونده است که پیشتر در مورد آنها بحث شد. در غیراینصورت هر بار با انبوهی از فایل‌های غیر ضروری که باید به مخزن کد وارد شوند روبرو خواهید شد.

### پروژه سورس باز DelphiAddinForTortoiseSVN

این افزودنی محیط توسعه مجتمع دلفی (شکل ۴)، همانند اولین افزودنی معرفی شده برای VS.Net عمل می‌کند و محصور کننده توانایی‌های TortoiseSVN است. برای دریافت سورس کد آن به آدرس زیر مراجعه نمایید:

<http://delphiaddinfortortoisesvn.tigris.org/>



شکل ۴- افزودنی TortoiseSVN برای محیط توسعه مجتمع دلفی.

برای نصب آن دو فایل `tsvnWizard.pas` و `icons.res` را از آدرس فوق دریافت کنید. یک پروژه `package` جدید را در دلفی آغاز کرده و این دو فایل را به آن اضافه نمایید. سپس بر روی پروژه در IDE کلیک راست نموده و گزینه `Install` را انتخاب کنید. متأسفانه مبحث تغییر رنگ آیکون‌های فایل‌ها (که توسط TortoiseSVN در windows explorer نمونه آنرا دیده‌اید) به این صورت در دلفی پشتیبانی نخواهند شد.

### پروژه سورس باز delphisvn

افزودنی دیگری به نام `delphisvn` نیز برای دلفی موجود است که آنرا می‌توان از آدرس زیر دریافت کرد:

<http://sourceforge.net/projects/delphisvn/>

این افزودنی نیازی به نصب TortoiseSVN نداشته و توانایی‌های آنرا بازنویسی کرده است.

### افزودنی JCL Version

کتابخانه معروف jedl یا jedi control library ، افزودنی را به نام JCL Version به IDE دلفی اضافه می‌کند که آن نیز محصور کننده توانایی‌های TortoiseSVN است. این کتابخانه از آدرس زیر قابل دریافت است:  
<http://homepages.codegear.com/jedi/jvcl/>

### استفاده از Tools menu در IDE

منوی استاندارد tools امکان افزودن آیتم‌های جدیدی را نیز برای شما فراهم کرده است. نحوه افزودن آیتم‌های مربوط به TortoiseSVN به این منو در آدرس زیر بر شمرده شده است:

[http://delphi.wikia.com/wiki/Adding\\_TortoiseSVN\\_to\\_the\\_Tools\\_menu](http://delphi.wikia.com/wiki/Adding_TortoiseSVN_to_the_Tools_menu)

### ج) یکپارچگی SVN با Eclipse

برای Eclipse نیز افزودنی سورس بازی در آدرس زیر قابل دریافت است:

<http://subclipse.tigris.org/>

پیش از نصب subclipse بهتر است اطمینان حاصل نمائید که اینکار پیشتر صورت نگرفته باشد. برای این منظور مسیر زیر را بررسی کنید:

Help → About Eclipse SDK → Plugins

برای مثال PHPEclipse این پلاگین را نیز به صورت خودکار نصب می‌کند.

### د) یکپارچگی SVN با NetBeans

پشتیبانی از SVN در NetBeans 5.5 به صورت پیش فرض نصب نشده است، اما از نگارش ۶ به بعد آن، این پشتیبانی به صورت کامل وجود دارد (توسط گزینه Versioning در منوهای برنامه) و نیازی به هیچگونه تنظیمات خاصی برای فعال کردن اولیه آن نیست.



## فصل ۵ – تهیه پشتیبان از مخزن‌های کدها

### مقدمه:

بدون تهیه پشتیبان‌های منظم از مخزن‌های کد، عملاً استفاده از این سیستم کاری بیهوده است. در این فصل نحوه تهیه پشتیبان از اطلاعات مخزن‌های کدها و بازیابی آنها را مورد بررسی قرار خواهیم داد.

### تهیه پشتیبان از مخزن کد:

#### الف) روش کپی کردن فایل‌ها

این روش سریعترین حالت ممکن تهیه پشتیبان است اما باید سرور SVN را متوقف کرد.

#### ب) تهیه فایل dump

با استفاده از برنامه استاندارد `svnadmin` و اجرای خط فرمان زیر می‌توان از یک مخزن کد پشتیبان تهیه کرد:

```
C:\>SVNAdmin dump C:\Repositories\TestRepo > C:\backup\dumps\dmp01.bak
```

در این دستور با توجه به اینکه پوشه `bin` برنامه SVN در `PATH` ویندوز قرار گرفته است (فصل یک)، نیازی به مشخص نمودن مسیر کامل آن نخواهد بود. سپس `dump` مخزن کد واقع شده در مسیر `C:\Repositories\TestRepo` در فایل `C:\backup\dumps\dmp01.bak` ذخیره خواهد شد (پسوند فایل اهمیتی ندارد). این روش از فایل‌های تنظیمات مخزن کد پشتیبان تهیه نمی‌کند.

 اگر هنگام اجرای این فرمان با خطای زیر مواجه شدید، بدین معنا است که باید آخرین نگارش SVN را از سایت آن دریافت کرده و نصب نمایید (همواره پس از به روز رسانی TortoiseSVN، برنامه SVN را نیز باید به روز نمود):

```
svnadmin: Expected format '3' of repository; found format '5'
```

اگر تمایل دارید که به این روش پشتیبان تهیه کنید، برنامه سورس باز زیر رابط کاربر مناسبی را در اختیار شما قرار می‌دهد:

<http://www.codeplex.com/svnbackupwidget>

#### ج) روش hotcopy

اگر شخصی یک یا چند فایل پروژه را قفل کرده باشد، استفاده از روش فوق هنگام بازیابی اطلاعات با مشکل مواجه خواهد شد. در این موارد از روش تهیه `hotcopy` باید استفاده شود (که در هر زمانی و تحت هر شرایطی قابل اجرا است و به معنای یک کپی امن از مخزن کد می‌باشد):

```
SVNAdmin hotcopy C:\Repositories\TestRepo C:\backup\hotdumps --clean-logs
```

در اینجا مسیر مقصد باید یک پوشه خالی باشد.

برای بررسی صحت عملیات تهیه پشتیبان، می‌توان دستور خط فرمان زیر را اجرا کرد:

```
svnadmin verify C:\backup\hotdumps
```

با استفاده از برنامه استاندارد windows task scheduler می‌توان عملیات تهیه پشتیبان روزانه را تحت یک زمان بندی مشخص انجام داده و سپس آنرا به صورت خودکار به رایانه‌ای دیگر در شبکه منتقل نمود. برای مثال یک فایل متنی با پسوند bat ایجاد کنید و محتویات آنرا به صورت زیر تنظیم نمایید:

### DailyBackup.bat

```
::Reset final backup storage:
rmdir /S /Q \\archive\svn_backup\daily\v5
mkdir \\archive\svn_backup\daily\v5
::Reset Temp backup storage
rmdir /S /Q C:\SVNBACKUP
mkdir C:\SVNBACKUP
::Initiate SVN backup. Use svadmin hotcopy --help for details
svadmin hotcopy C:\ARMS\C:\SVNBACKUP --clean-logs
::Copy temp backup storage to final backup storage
xcopy C:\SVNBACKUP \\archive\svn_backup\daily\v5 /E /I /Q /H /R /Y
```

در این batch file ، نام سروری که فایل‌ها پس از تهیه پشتیبان به آن منتقل خواهند شد، archive نام دارد. سایر مسیرها را نیز باید همانند مخزن کد خود و مسیرهای موجود در شبکه تنظیم نمایید.

برای تنظیم زمانبندی اجرای این batch file ، مسیر زیر را در ویندوز طی نمایید:

All programs → accessories → system tools → scheduled tasks

در صفحه باز شده، بر روی add scheduled tasks کلیک کرده و مطابق wizard آن، ابتدای فایل bat ایجاد شده را به آن معرفی کنید، سپس زمان اجرای فایل bat را انتخاب و در ادامه نام کاربری و کلمه عبور مناسبی را وارد نمایید. باید دقت داشت که این کاربر باید دسترسی کپی کردن فایل‌ها را به سرور مورد نظر در شبکه، داشته باشد. در غیراینصورت عملیات xcopy با موفقیت اجرا نخواهد شد.

### بازیابی اطلاعات از پشتیبان تهیه شده توسط روش dump

برای بازیابی اطلاعات از پشتیبان (یک فایل dump تهیه شده)، ابتدا باید مخزن کد جدیدی را ایجاد نمود (که در طی فصل‌های قبل با آن آشنا شده‌اید). سپس دستور زیر را در خط فرمان وارد نمایید:

```
svadmin load C:\SVN\MyProject < C:\tmp\MyProject.bak
```

برای بررسی صحت عملیات صورت گرفته می‌توان از repo-browser برنامه TortoiseSVN استفاده کرد.

### مقایسه سه روش پشتیبان گیری

روش	نیاز به متوقف سازی سرور دارد؟	از فایل‌های تنظیمات پشتیبان می‌گیرد؟	سرعت
File Copy	بله	بله	سریعترین
Dump/Load	خیر	خیر	کندترین
Hotcopy	خیر	بله	سریع

## فصل ۶ – آشنایی با قلاب‌ها در SVN

## مقدمه:

Subversion hooks یا قلاب‌های SVN ، امکان انتساب اعمالی (برنامه‌هایی) را به رخ داده‌های معین SVN می‌دهند. برای مثال ارسال ایمیل پس از هر commit به سایر اعضای تیم، به روز رسانی خودکار یک سایت پس از هماهنگ سازی با مخزن کد، به روز رسانی issue tracking system و امثال آن. برای مثال اگر مخزن کد ما در آدرس C:\Repositories\TestRepo\ قرار داشته باشد در اینجا پوشه Hooks نیز قابل مشاهده است (C:\Repositories\TestRepo\hooks).

نمونه‌ای از اسکریپت‌هایی را که به قلاب‌های SVN قابل اتصال هستند، در آدرس‌های زیر می‌توانید دریافت کنید:

<http://svn.collab.net/repos/svn/trunk/tools/hook-scripts/>  
<http://svn.collab.net/repos/svn/trunk/contrib/hook-scripts/>

## انواع قلاب‌های SVN :

- **start-commit**  
این رخداد پیش از شروع تراکنش commit ، فراخوانی می‌شود. یکی از کاربردهای آن جلوگیری از commit اعضای تیم است اگر شرایط ویژه‌ای محقق نشده باشد. اگر خروجی اسکریپت مورد استفاده در اینجا غیرصفر باشد عملیات متوقف می‌شود. آرگومان‌هایی را که این رخداد به برنامه یا اسکریپت قلاب ارسال می‌کند شامل مسیر مخزن کد و نام کاربر است.
- **pre-commit**  
این رخداد در هنگام شروع عملیات commit و پیش از اجرای واقعی آن فراخوانی می‌شود. یکی از کاربردهای آن بررسی استانداردهایی است که در یک تیم برنامه نویسی باید رعایت شود (برای مثال تمام برنامه نویس‌ها موظف به ارائه توضیحاتی در مورد commit شوند). اگر خروجی اسکریپت مورد استفاده در اینجا غیرصفر باشد عملیات متوقف می‌شود. آرگومان‌هایی را که این رخداد به برنامه یا اسکریپت قلاب ارسال می‌کند شامل مسیر مخزن کد و نام تراکنش است.
- **post-commit**  
این رخداد پس از اتمام تراکنش commit فراخوانی می‌شود. یکی از کاربردهای آن ارسال ایمیل به سایر اعضای تیم است. آرگومان‌هایی را که این رخداد به برنامه یا اسکریپت قلاب ارسال می‌کند شامل مسیر مخزن کد و شماره بازنگری است.
- **pre-revprop-change**  
این رخداد پیش از تغییر خاصیت بازنگری (revision property) فراخوانی می‌شود (در مورد این خاصیت در فصل‌های قبل توضیحات لازم ارائه شد). آرگومان‌هایی را که این رخداد به برنامه یا اسکریپت قلاب ارسال می‌کند شامل مسیر مخزن کد ، شماره بازنگری، نام کاربر و نام خاصیت است.
- **post-revprop-change**  
این رخداد پس از تغییر خاصیت بازنگری فراخوانی می‌شود. عموماً از آن برای اطلاع رسانی استفاده می‌شود. آرگومان‌هایی را که این رخداد به برنامه یا اسکریپت قلاب ارسال می‌کند شامل مسیر مخزن کد ، شماره بازنگری، نام کاربر و نام خاصیت است.
- **pre-lock**  
این رخداد پیش از ایجاد یک قفل فراخوانی می‌گردد. آرگومان‌هایی را که این رخداد به برنامه یا اسکریپت قلاب ارسال می‌کند شامل مسیر مخزن کد ، مسیری که در آن قفل ایجاد خواهد شد و نام کاربر مربوطه می‌باشد. اگر خروجی اسکریپت مورد استفاده در اینجا غیرصفر باشد عملیات متوقف می‌شود.
- **post-lock**

این رخداد پس از کامل شدن عملیات قفل کردن فایل‌ها فراخوانی می‌شود. آرگومان‌هایی را که این رخداد به برنامه یا اسکریپت قلاب ارسال می‌کند شامل مسیر مخزن کد و نام کاربر مربوطه می‌باشد.

• **post-unlock**

این رخداد پس از قفل‌گشایی فایل‌ها فراخوانی خواهد شد. آرگومان‌هایی را که این رخداد به برنامه یا اسکریپت قلاب ارسال می‌کند شامل مسیر مخزن کد و نام کاربر مربوطه می‌باشد.

• **pre-unlock**

این رخداد پیش از قفل‌گشایی فایل‌ها فراخوانی خواهد شد. آرگومان‌هایی را که این رخداد به برنامه یا اسکریپت قلاب ارسال می‌کند شامل مسیر مخزن کد، مسیری که در آن قفل باز خواهد شد و نام کاربر مربوطه می‌باشد. اگر خروجی اسکریپت مورد استفاده در اینجا غیرصفر باشد عملیات متوقف می‌شود.

### استفاده از رخداد post-commit برای مطلع‌سازی سایر اعضای تیم

پیش از بررسی محتویات فایل `post-commit.bat` که در ادامه ارائه خواهد شد، نیاز به برنامه‌ای داریم که بتواند از طریق خط فرمان ویندوز ایمیل ارسال کند. نمونه‌ای از این برنامه‌ها را از آدرس‌های زیر می‌توانید دریافت کنید:

<http://caspiandotconf.net/menu/Software/SendEmail/>  
<http://sourceforge.net/projects/blat/>

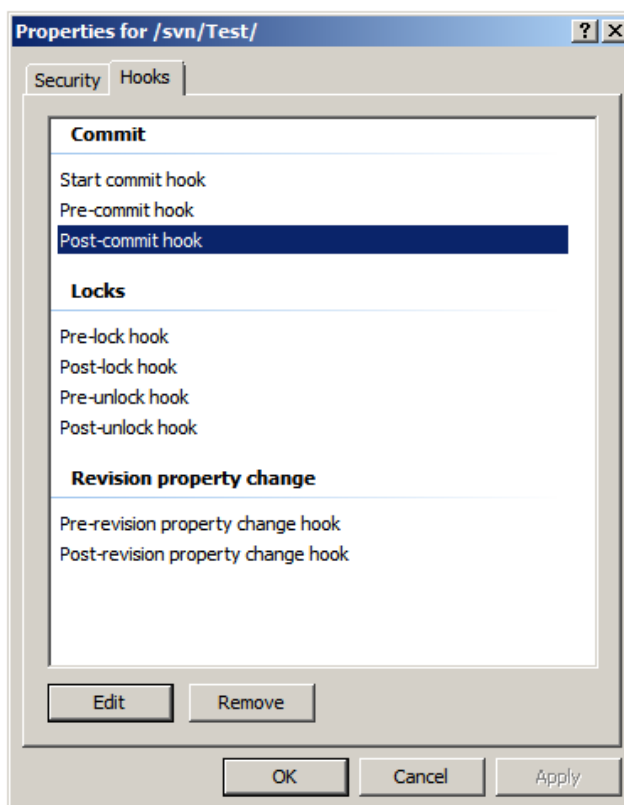
محتویات فایل `post-commit.bat`:

```
@ECHO OFF
REM *****
REM * this sets the arguments supplied by Subversion
REM *****
SET REPOS=%1
SET REV=%2
SET DIR=%REPOS%/hooks
SET PATH=%PATH%;%DIR%;C:\Utils
SET WORKING_COPY=c:\path\to\working\copy
SET SITENAME=My Code Base
SET SMTP_SERVER=smtp.yourdomain.com
SET EMAIL_TO=svnupdates@yourdomain.com
SET EMAIL_FROM=svnupdates@yourdomain.com
SET SUBJECT=SVN Update - %SITENAME% - rev %REV% - %REPOS%
svn update %WORKING_COPY%
ECHO The following changes were made to the code: > %DIR%/email.txt
ECHO. >> %DIR%/email.txt
REM *****
REM * dump the log of changes to the e-mail message
REM *****
svn log %WORKING_COPY% -v -r "%REV%" >> %DIR%/email.txt
REM *****
REM * dump the diff changes to the e-mail message
REM *****
svn diff %WORKING_COPY% -c "%REV%" --no-diff-deleted >> %DIR%/email.txt
sendEmail -s %SMTP_SERVER% -t %EMAIL_TO% -f %EMAIL_FROM% -u "%SUBJECT%" -o message-file=%DIR%/email.txt
```

توضیحاتی در مورد اسکریپت فوق:

- رخداد **post-commit** اولین آرگومانی را که به اسکریپت ما ارسال می‌کند مسیر پوشه **hooks** است و دومین آرگومان ارسال شده شماره خودکار بازنگری عملیات می‌باشد.
- برنامه خط فرمان **sendMail** در اینجا در مسیر **C:\utils** قرار گرفته و سپس به **PATH** ویندوز جهت سهولت کار اضافه شده است.
- مسیر **WORKING\_COPY** را باید اصلاح نمائید (همانطور که ملاحظه می‌کنید در صورت نیاز به محتویات **diff** می‌توان یک نسخه کاری را نیز بر روی سرور قرار داد).
- موارد مربوط به **smtp server** و تنظیمات ایمیل‌ها را نیز مطابق اطلاعات سرور خود ویرایش کنید.
- عملیات **diff** ممکن است حجم زیادی از اطلاعات را ایجاد نماید. اگر نیازی به آن ندارید می‌توانید از آن خط صرفنظر نمائید.
- بدلیل امنیتی، **svn** اسکریپت‌های قلاب را در محیطی ایزوله اجرا می‌کند. بنابراین در این محیط امکان دسترسی به متغیرهای محیطی مانند **PATH** و غیره نخواهد بود. به همین جهت در اسکریپت فوق متغیر **PATH** یکبار مقدار دهی شده است.
- آرگومان‌های **SET DIR=%1/hooks** هنگام اجرای اسکریپت توسط **svn** به آن انتقال داده خواهند شد (مسیر پوشه **hooks**). در غیراینصورت (عدم استفاده از این مسیرهای استاندارد) خروجی فایل‌های ما در مسیرهای پیش بینی شده قرار نخواهند گرفت.

برای اجرا شدن این اسکریپت تنها کافی است فایل **post-commit.bat** (نام این فایل باید مطابق یکی از رخدادهای تعریف شده فوق باشد، در غیراینصورت اجرا نخواهد شد)، را در پوشه **hooks** مخزن کد خود قرار دهید. راه دیگر استفاده از **VisualSVN server** است. بر روی مخزن کد خود در این برنامه کلیک راست کرده و خواص آنرا انتخاب کنید. در صفحه باز شده در برگه **Hooks** به سادگی می‌توان رخدادهای منتسب به اسکریپت‌های قلاب را مدیریت کرد (شکل ۱).



شکل ۱- امکان مدیریت ساده قلاب‌ها با استفاده از VisualSVN server.

## دریافت خروجی‌های svn هنگام اجرای یک اسکریپت قلاب

برای دیدن یک اسکریپت قلاب می‌توان از روش زیر استفاده کرد.

- نام فایل `post-commit.bat` را به `post-commit-run.bat` تغییر دهید.
- اکنون یک فایل `post-commit.bat` جدید را با محتویات زیر ایجاد کنید:  
`call %~dp0post-commit-run.bat %* > %1/hooks/post-commit.log 2>&1`
- با استفاده از این روش خروجی اجرای اسکریپت ما در فایل `post-commit.log` ذخیره خواهد شد که جهت دیدن عملیات بسیار مفید است.



**نکته:**

در ویندوز اسکریپت‌های قلاب باید از نوع فایل‌های اجرایی باشند برای مثال فایل‌هایی با پسوند `exe` یا `bat`. اگر اسکریپت شما از نوع `vbs` است برای اجرا آن یک فایل `bat` با محتوای زیر درست کنید:

```
SET PATH=C:\Windows;C:\Windows\system32;
cscript.exe //NoLogo C:\Repositories\TestRepo\hooks\sample.vbs
```

## استفاده از C# یا زبان‌های دیگر برنامه نویسی برای نوشتن قلاب‌ها

با استفاده از قابلیت‌های `C#` یا هر زبان برنامه نویسی دیگر نیز می‌توان قلاب‌هایی را برای `svn` نوشت. برای مثال:

```
using System;

namespace post_commit {
    class Class1 {
        [STAThread]
        static void Main(string[] args) {
            System.IO.StreamWriter sw = new
                System.IO.StreamWriter(@"f:\PostCommit.txt", true,
                    System.Text.Encoding.Default);
            sw.Write(System.DateTime.Now.ToString() + ":" +
                string.Join("|", args));
            sw.Close();
        }
    }
}
```

این برنامه از نوع `console` است که نام نهایی آن باید مطابق یکی از رخدادهای استاندارد قلاب‌ها باشد (تا پس از قرارگیری در پوشه `hooks` قابل استفاده شود).

بدیهی است این روش توانایی قابل توجهی را در اختیار شما در مقایسه با اسکریپت نویسی ساده قرار می‌دهد. البته همیشه باید این نکته را به‌خاطر داشت که سرعت اجرای قلاب‌ها باید زیاد بوده و هرچه سریعتر به پایان برسند، در غیراینصورت کاربران با مشکل مواجه خواهند شد (برای مثال دریافت `timeout` در حین کار با یک سرور راه دور).

نمونه‌ای دیگر از این تلاش‌ها برنامه `Subversion Notify` است:

<http://www.subversionnotify.com/>

## قلاب‌های سمت کلاینت

اگر به تنظیمات برنامه TortoiseSVN مراجعه کنید، قسمت Hook scripts آن قابل تنظیم است و قلاب‌هایی که در اینجا تنظیم شوند تنها بر روی کلاینت اجرا خواهند شد (برخلاف قلاب‌های سمت سروری که تاکنون مورد بررسی قرار گرفتند).

## فصل ۷ - مرور بهترین تجربه‌های کاری با SVN

### مقدمه:

برای کار با SVN یا هر سیستم کنترلی دیگر باید اندکی در شیوه‌های کاری خود تغییر ایجاد نمود تا بتوان از توانایی‌های این محصولات حداکثر استفاده را برد. باید در نظر داشت که هرچند SVN و امثال آن به دلایل بسیاری که در فصل‌های قبل در مورد آنها بحث شد برای یک برنامه نویسی مفید هستند، اما ابتدایی‌ترین دلیل توسعه آنها تسهیل روابط کاری با سایر اعضای یک تیم برنامه نویسی است. در ادامه این فصل درباره بهترین تجربه‌های کاری استفاده از SVN در یک تیم برنامه نویسی بحث خواهیم کرد.

### انتخاب آرایش (layout) یک مخزن کد

انتخاب آرایش یک مخزن کد برای کمک نمودن به اعضای تیم در جهت یافتن ساده‌تر نگارش‌های مختلف پروژه بسیار حائز اهمیت است. برای یک پروژه آرایش trunk/branches/tags بسیار عالی جواب می‌دهد و ریشه مخزن کد شما به صورت زیر خواهد بود:

```
branches
tags
trunk
```

این آرایش کاملاً استاندارد بوده و هر شخصی در سراسر دنیا که پیشتر تجربه کار با SVN را داشته باشد، با اولین بررسی قادر به تشخیص ساختار مخزن کد و استفاده از آن خواهد بود. اما در حالت مدیریت چندین پروژه، استاندارد مشخصی وجود ندارد. در این حالت عموماً یکی از دو آرایش زیر مورد استفاده قرار می‌گیرد که در ادامه به مزایا و معایب هریک اشاره خواهد شد:

```
branches
  project1
  project2
tags
  project1
  project2
trunk
  project1
  project2
```

و یا آرایش بعد:

```
project1
  branches
  tags
  trunk
project2
  branches
  tags
  trunk
```



اولین آرایش برای حالتی مناسب است که پروژه‌ها واحدهای مشترکی را با یکدیگر داشته باشند و یا کلا یک واحد اصلی بزرگتر را تشکیل دهند. مزیت آن استاندارد بودن سطح اصلی مخزن کد است و مشکل آن جابجا کردن فایل‌ها و پوشه‌های پروژه‌ها است. برای مثال اگر در آینده قصد تفکیک آنها را داشته باشید کار مشکلی را پیش رو خواهید داشت. همچنین مباحث اعتبار سنجی ریزتری را با توجه به یکی بودن ریشه اصلی مخزن کد نیز مشکل می‌توان اعمال کرد.

دومین آرایش برای حالتی مناسب است که پروژه‌ها از یکدیگر مجزا بوده و تنها تحت یک مخزن کد نگهداری می‌شوند. در این حالت هر پروژه آرایشی استاندارد داشته و همچنین اعمال سطوح دسترسی مختلف نیز به آنها ساده‌تر است.

### از قفل کردن فایل‌های قابل ادغام خودداری کنید

بسیاری از سازمان‌ها در حال کنار گذاشتن استفاده از سیستم‌های کنترل سورس کدی که برای کار کردن با فایل‌ها در آنها باید حتما فایل‌ها را قفل نمود، هستند (برای مثال SourceSafe). قفل کردن فایل‌ها مشکلات عدیده‌ای را برای سایر اعضای تیم به همراه خواهد داشت، برای مثال قفل کردن طولانی مدت یک فایل و سپس فراموش نمودن رها کردن آن قفل. در یک تیم ممکن است دو نفر بر روی دو قسمت مجزای یک فایل کار کنند و قفل کردن یک فایل در اینجا صرفاً به مشکلات خواهد افزود و با امکانات merge خودکار در SVN نیازی به قفل کردن فایل اصلی نیست. همچنین با امکاناتی که در فصل‌های قبل در مورد قلاب‌ها ارائه شد، می‌توان با ارسال ایمیل، بلافاصله سایر اعضای تیم را در جریان تغییرات قرار داد.

تنها فایل‌های گرافیکی و امثال آنرا که نمی‌توان با استفاده از ابزار merge به صورت خودکار یکی کرد، باید قفل کرد.

### همگام سازی متناوب با مخزن کد

هدف نهایی یک سیستم کنترل سورس کد، commit آخرین تغییرات جهت نگهداری سوابق آنها و همچنین در دسترس قرار دادن آن موارد جهت سایر اعضای تیم می‌باشد. هرچقدر تغییرات را دیرتر با مخزن کد هماهنگ کنید، مشکلات بیشتری را برای اعضای تیم پدید خواهید آورد. با ارسال متناوب کدهای خود به مخزن، در صورت بروز هرگونه مشکل سخت افزاری و امثال آن، نگران از دست دادن کدهای محلی خود نخواهید بود چون عموماً سخت افزارهای بهتری را برای رایانه‌های سرویس دهنده در نظر می‌گیرند. همچنین همواره سیستم تهیه پشتیبان بر روی سرورها منظم‌تر از سیستم‌های تهیه پشتیبان‌های محلی است و نظارت بیشتری بر آنها صورت می‌گیرد. بعلاوه ارسال متناوب کدها به مخزن از احتمال ایجاد تداخل‌ها با سایر اعضای تیم جلوگیری می‌کند و امکان هماهنگ سازی بیشتر آنها را با تغییرات جدید فراهم می‌آورد.

هرچند ممکن است این تصور وجود داشته باشد که ارسال یک کد باگ دار ممکن است بر روی سایر هم تیمی‌ها اثر منفی بگذارد و بنابراین بهتر است کمی دیرتر نسبت به ارسال و اعمال تغییرات اقدام کرد، اما بر اساس قوانین مورفی همواره یک باگ وجود دارد که شما تنها پس از commit به آن پی خواهید برد! (اگر قرار باشد چیزی خراب شود، می‌شود)

به همین جهت تغییرات خود را سریعتر با مخزن کد هماهنگ سازید، زیرا رفع خطای مربوط به تغییراتی با وسعت کم، ساده‌تر است و درک آن نیز برای سایر اعضای تیم دشوار نخواهد بود.

و یا اگر کدی که مشغول به کار بر روی آن هستید بیشتر حالت آزمایشی دارد بهتر است یک انشعاب جدید برای آن ایجاد نموده و سایر اعضای تیم را درگیر خطاهای کدهای خود در انشعاب اصلی کاری ننمائید.

### هنگام commit از توضیحات معنا داری استفاده کنید

هماهنگ سازی متناوب و سریع تغییرات کوچک در کدها با مخزن کد، یکی از مواردی است که به سایر هم تیمی‌ها در درک کدهای شما کمک خواهد کرد. مورد دیگر توضیحاتی است که هنگام هماهنگ سازی آخرین تغییرات خود با مخزن کد ارائه می‌دهید. بعلاوه، این مورد هنگام تهیه گزارش از تغییرات انجام شده در یک مخزن کد در طول زمان نیز بسیار مفید می‌باشد.

## انشعابات و ادغام سازی

هرچند SVN نسبت به CVS از لحاظ ایجاد انشعابات سریع و کم حجم، بسیار پیشرفت داشته است، اما این بدان معنا نیست که هر زمان باید یک انشعاب جدید را ایجاد کرده و در آن مشغول به کار شد. همانطور که عنوان شد هدف نهایی از یک مخزن کد `commit` اطلاعات جدید به آن است اما اگر این جمله را بخواهیم اصلاح کنیم باید گفت `commit` اطلاعات جدید به `trunk` و شاخه اصلی کاری است. بنابراین تا حد امکان از ایجاد انشعابات خودداری کنید مگر مواردی که نیاز به تست و آزمایش دارند و بهتر است آنها را از خط اصلی کاری دور نگه داشت و گاهی از اوقات حتی مشخص نیست این تلاش‌ها آیا واقعا در `trunk` استفاده خواهند شد یا خیر. یا برای مثال به ازای هر نگارش عمومی برنامه بهتر است یک انشعاب نیز ایجاد شود تا تمامی آزمایشات لازم بر روی این نگارش خارج از خط اصلی کاری انجام شود. پس از اطمینان حاصل کردن از کیفیت کار حاصل شده در انشعاب ارائه عمومی، می‌شود به آن برچسب زد (ایجاد `tag`). برای مثال گزارشی از برچسب زدن به ارائه‌های عمومی برنامه:

tags

```
7 months ago 0.9.0
8 months ago 0.8.7
9 months ago 0.8.6
9 months ago 0.8.5
```

همانطور که پیش‌تر نیز عنوان شد، پس از ایجاد یک `tag`، جهت ارائه برنامه (اگر سورس باز است) می‌توان با استفاده از دستور `export`، یک کپی سریع بدون پوشه‌های مدیریتی `svn` ایجاد کرد.

## فصل ۸ – یکپارچگی SVN با سایر ابزارها

### مقدمه:

SVN بدلیل ارائه دو ویژگی به سادگی با محصولات مختلف قابل یکپارچه شدن است. امکان اجرای دستورات در خط فرمان و همچنین API بسیار قوی جهت کار با توانایی های مختلف این برنامه سبب شده است که ابزارهای زیادی جهت سهولت کار با SVN طراحی شوند. در این فصل بیشتر به رابط های کاربری تحت وب طراحی شده برای SVN خواهیم پرداخت. هرچند ابزار استاندارد `mod_dav_svn` که پیشتر با آن آشنا شدیم امکان مرور تحت وب مخزن کد را می دهد اما این ابزار برای کاربران حرفه ای به هیچ عنوان راضی کننده نیست زیرا امکان مشاهده نحوه تغییرات، جزئیات آنها و همچنین مشاهده نگارش های قبلی و امثال آنرا فراهم نمی کند. تلاش های بسیاری در این زمینه صورت گرفته است، برای مثال رابط های کاربری تحت وب زیر را می توان نامبرد:

ViewVC → <http://www.viewvc.org/>  
SVN::Web → <http://search.cpan.org/~nikc/SVN-Web/>

اما در سال های اخیر سیستم یکپارچه مدیریت پروژه Trac محبوبیت بسیاری یافته است که در این فصل به نحوه استفاده از آن خواهیم پرداخت. این سیستم از آدرس زیر قابل دریافت است:

<http://trac.edgewall.com/>

### Trac چیست؟

Trac محصولی است سورس باز و به عنوان سیستم بهبود یافته Wiki و ردیابی خطاها شناخته می شود که در سال ۲۰۰۶ موفق به دریافت جایزه زیر شد:

#### UK Linux & Open Source Award for Best Linux OSS Developer Tool

از Wiki در Trac برای تهیه مستندات پروژه، راهنمای سیستم و امثال آن استفاده می شود و با توجه به ماهیت آن هیچ ساختار از پیش تعیین شده ای به تیم دیکته نشده و نحوه تهیه و مدیریت آن به صلاح دید اعضای تیم خواهد بود. همچنین یک Wiki به صورت خودکار تحت یک سیستم کنترل نگارش قرار دارد و نحوه تغییرات و نگارش های مختلف صفحات در آن کاملاً قابل بررسی و کنترل است. در سیستم Trac ماژول دیگری به نام ماژول بلیط (Trac ticket system) نیز وجود دارد که از آن جهت ساده تر نمودن مراحل مختلف کنترل پروژه اعم از ردیابی وظایف محوله، ارائه درخواست های جدید، گزارش خطاها و پشتیبانی برنامه، استفاده می شود. همچنین امکان انتساب هر بلیط به افراد تیم، واگذار کردن وظایف محوله به دیگران و همچنین گزارش گیری از این اتفاقات نیز در این سیستم میسر است. یکی از افزونه های ماژول بلیط، `roadmap` (کارراهه) می باشد که در آن امکان مشاهده زمان های ارزیابی (milestones) آتی وجود داشته و مشخص می کند که چه بلیط هایی به هر کدام انتساب داده شده است. `Roadmap` چشم انداز پروژه را نمایش داده و توسط آن می توان دریافت که به کجا خواهیم رسید و بلیط ها نحوه رسیدن به اهداف پروژه را بیان می کنند. بعلاوه ماژول `Timeline` در Trac امکان گزارش گیری از وقایع انجام شده در پروژه را مهیا می کند. توسط این ماژول امکان گزارش گیری از تغییرات Wiki، بلیط ها، تغییرات سورس کدها و زمان های ارزیابی امکان پذیر است. این گزارش به صورت یک تغذیه خبری RSS نیز قابل دسترسی می باشد. علاوه بر این موارد، Trac امکان مرور یک مخزن کد را نیز فراهم کرده و علاوه بر نمایش محتویات آن، تغییرات انجام شده و گزارشی از وقایع ثبت شده در سیستم را نیز ارائه می دهد. همچنین امکان مشاهده `diff` آنلاین نگارش های مختلف یک فایل نیز در آن مهیا است.

## نصب و راه اندازی Trac در ویندوز

### الف) روش سنتی نصب Trac و بررسی جزئیات مربوطه:

برای نصب Trac در ابتدا نیاز به نصب SVN و وب سرور Apache می‌باشد که در طی فصل‌های قبیل این موارد بررسی شدند و در این فصل از پرداختن مجدد به آنها خودداری خواهد شد (تمام موارد نصب ماژول‌های SVN در وب سرور Apache باید رعایت گردد). پس از نصب مواردی که در فصل یک به آنها اشاره شد باید خط زیر را در فایل `Apache2.2\conf\httpd.conf` فعال کرد (علامت # ابتدای آنرا حذف کنید)

```
AddHandler cgi-script .cgi
```

پس از این تغییر باید یکبار Apache را از نو راه اندازی کنید (stop/start).

سپس باید Python را بر روی ویندوز نصب کرد. به آدرس زیر مراجعه نموده و گزینه Python windows installer را انتخاب کنید:

<http://www.python.org/download/>

نصب Python (شکل ۱) نکته خاصی نداشته و پیش فرض‌های آن را بپذیرید.



شکل ۱- نصب Python بر روی ویندوز

اکنون باید مسیر پوشه python را به PATH ویندوز اضافه کرد (روش انجام این عملیات در فصل یک بررسی گردید). دو مسیر `C:\Python25\Scripts` و `C:\Python25` بهتر است به PATH ویندوز اضافه شوند.

سپس نیاز به نصب **setuptools** می‌باشد که از آدرس زیر قابل دریافت است:

<http://pypi.python.org/pypi/setuptools/>

دقت نمائید نسخه مرتبط با python خود را باید از آدرس فوق دریافت کرده و نصب کنید. در صورت عدم نصب این برنامه، در ادامه مراحل نصب موفق به اجرای برنامه trac-admin نخواهید شد.

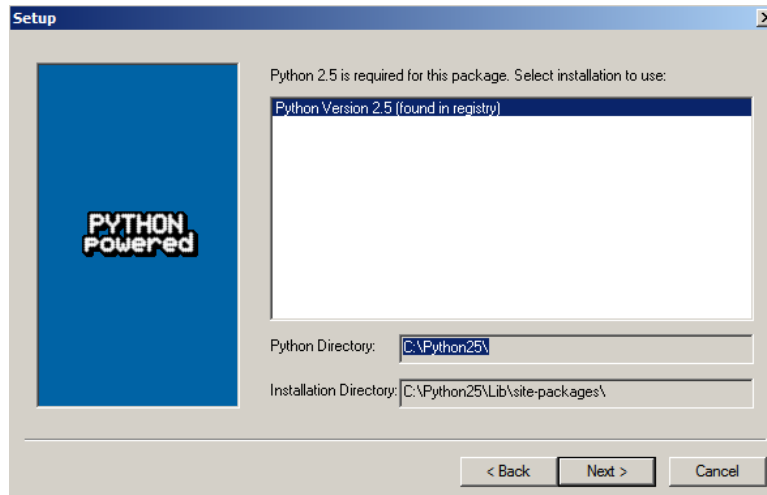
در ادامه باید python را به وب سرور Apache معرفی کرد. برای این منظور به آدرس زیر مراجعه نموده:

<http://www.apache.org/dist/httpd/modpython/win/>

و پس از دریافت آخرین نگارش mod\_python، آنرا نصب نمائید (دقت نمائید نگارش این ماژول باید با نگارش python نصب شده بر روی سیستم شما همخوانی داشته باشد).

مرحله بعد، انقیاد Python به SVN است (Python Subversion Bindings). از آدرس زیر آخرین نگارش svn-python-XYZ.win32-pyXYZ.exe را دریافت کنید:

<http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91>



شکل ۲- نصب svn-python.

نکته‌ای را که باید در اینجا دقت داشت این است که امکان وجود چندین نگارش مختلف python بر روی سیستم وجود دارد. هنگام نصب svn-python (شکل ۲)، آخرین نگارش نصب شده را از لیست انتخاب کنید. همچنین همواره باید نگارش فایل‌های انقیاد با نگارش SVN شما یکسان باشد.

مرحله بعد، انقیاد python به SQLite است. از SQLite جهت مدیریت بانک اطلاعاتی سیستم در اینجا استفاده خواهد شد. آخرین نگارش **pysqlite** را از یکی از آدرس‌های زیر دریافت کنید:

<http://www.inithd.org/pub/software/pysqlite/releases/>

<http://oss.itsystementwicklung.de/trac/pysqlite/>

نصب آن ساده است و پس از انتخاب نگارش python مورد نظر (آخرین نگارش نصب شده)، چند بار بر روی دکمه next کلیک نمائید تا کار به پایان برسد.

لازم به ذکر است در اینجا امکان استفاده از سایر بانک‌های اطلاعاتی مانند MySQL نیز فراهم است:

<http://sourceforge.net/projects/mysql-python>

که نیاز به نصب و راه اندازی و تنظیمات مخصوص به خود را دارد. SQLite عملاً نیازی به هیچگونه تنظیم خاصی برای نصب و راه اندازی ندارد، حجم کمی داشته و بسیار سریع است.

سپس **ClearSilver** را که یک HTML template system است باید نصب کرد. برای دریافت `clearsilver-0.9.14.win32-py2.4.exe` به آدرس زیر مراجعه نمایید:  
<http://www.clearsilver.net/downloads/win32/>

متأسفانه این نگارش تنها با python2.4 سازگار است. به همین منظور فایل `clearsilver_python-0.9.12-win32.zip` را از آدرس فوق دریافت نموده و پس از استخراج محتویات آن، فایل `neo_cgi.pyd` را در پوشه `Lib\site-packages` در محل نصب python خود کپی کنید.

در ادامه **genshi** را باید نصب نمود. این برنامه بعنوان XML & HTML parser عمل می کند.  
<http://genshi.edgewall.org/wiki/Download>  
نسخه سازگار با نگارش python خود را دریافت کرده و سپس آنرا نصب نمایید.

اکنون می توان **Trac** را نصب کرد. برای دریافت آخرین نگارش آن به آدرس زیر مراجعه کنید:  
<http://trac.edgewall.org/wiki/TracDownload>  
نصب آن همانند موارد ذکر شده با انتخاب نسخه python شروع شده و با چندبار کلیک کردن بر روی دکمه `next` به پایان می رسد.

سپس باید بانک اطلاعاتی `trac` را به ازای هر مخزن کد تنظیم نمود. فرض کنید تنظیمات زیر بر روی سیستم ما موجود هستند:  
مخزن کد `TracTest` در مسیر `C:\Repositories\TracTest`  
پوشه ذخیره سازی تنظیمات `Trac` مخصوص مخزن کد `TracTest` در مسیر `C:\Repositories\TracConfig\TracTest`

برای تنظیم نمودن بانک اطلاعاتی `trac` دستور زیر را در خط فرمان اجرا کنید:  
`C:\>C:\Python25\Scripts\trac-admin.exe C:\Repositories\TracConfig\TracTest initenv`

در اینجا یک سری سؤال از شما پرسیده خواهند شد که به شرح زیر هستند:

```
Project Name [My Project]>
Database connection string [sqlite:db/trac.db]>
Repository type [svn]>
Path to repository [/path/to/repos]> C:\Repositories\TracTest
```

در ابتدا نام دلخواه پروژه پرسیده می شود. می توانید مقدار پیش فرض را با فشردن دکمه `Enter` قبول نمایید. این مورد را نیز برای رشته اتصالی به بانک اطلاعاتی و نوع مخزن کد تکرار نمایید (فشردن دکمه `Enter`). در پایان مسیر صحیح مخزن کد مورد استفاده را وارد کنید (حتماً باید مسیر کامل را وارد نمایید).

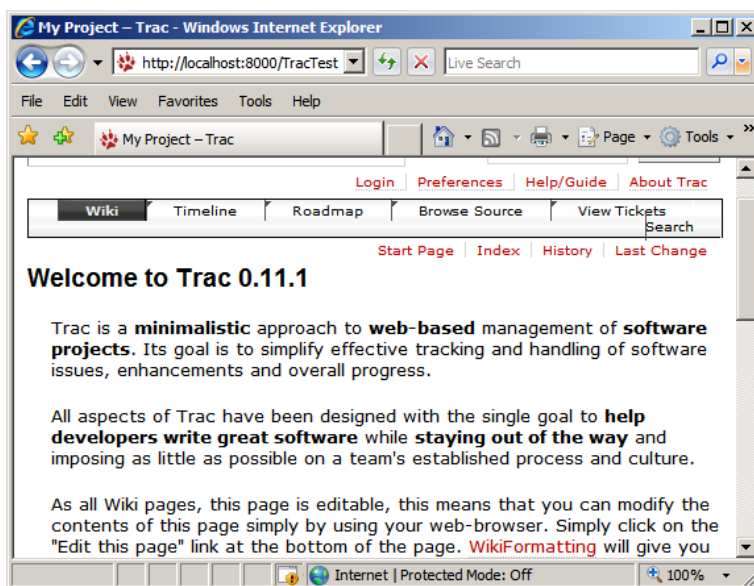
پس از پایان عملیات فایل `C:\Repositories\TracConfig\TracTest\conf\trac.ini` حاوی تنظیمات مربوط به `Trac` خواهد بود. برای آزمایش عملیات دستور زیر را در خط فرمان اجرا کنید:  
`C:\>C:\Python25\Scripts\tracd --port 8000 C:\Repositories\TracConfig\TracTest`

بدیهی است اگر `C:\Python25\Scripts` به `PATH` ویندوز اضافه شده باشد نیازی به ذکر آن به صورت فوق نخواهد بود.

اکنون در مرورگر خود آدرس زیر را وارد کنید:

<http://localhost:8000/>

لیست پروژه‌های تحت کنترل ظاهر شده و با کلیک بر روی یکی از آنها به صفحه اصلی Trac وارد خواهید شد (شکل ۳).



شکل ۳- نصب اولیه Trac بر روی ویندوز

امکان نصب `tracd` به صورت سرویس ویندوز نیز مهیا است. برای این منظور باید دستور زیر را اجرا کرد:  
`InstSrv tracd "C:\Program Files\Windows Resource Kits\Tools\srvany.exe"`

پس از اجرای این دستور، `registry editor` ویندوز را گشوده و مدخل زیر را پیدا کنید  
`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\tracd`

کلید جدیدی را به نام `Parameters` به آن اضافه نموده و دو مقدار رشته‌ای `Application` و `AppParameters` را تعریف کنید. `Application` باید به مسیر `python` نصب شده اشاره کند و `AppParameters` باید شامل مسیر کامل برنامه `tracd` و پارامترهای آن باشد.

پس از نصب و آزمایش اولیه، باید `Trac` را به وب سرور `Apache` متصل نمود تا هر بار نیاز به اجرای دستی برنامه `tracd` نباشد (یا نیازی به نصب آن به صورت سرویس ویندوز نباشد).

ابتدا سطر زیر را به فایل `Apache2.2\conf\httpd.conf` اضافه نمائید.

```
LoadModule python_module modules/mod_python.so
```

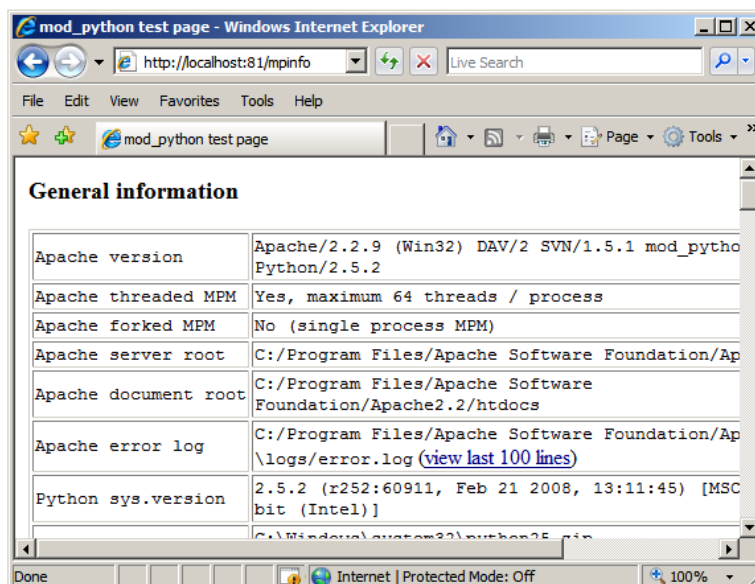
سپس به انتهای فایل فوق خطوط زیر را اضافه کنید:

```
<location /mpinfo>  
    SetHandler mod_python  
    PythonInterpreter main_interpreter  
    PythonHandler mod_python.testhandler  
</location>
```

اکنون یکبار وب سرور `Apache` را از نو راه اندازی کرده (`stop/start`) و مسیر زیر را در مرورگر خود وارد کنید:

<http://localhost:81/mpinfo>

در صورت موفقیت آمیز بودن عملیات باید شکل ۴ ظاهر شود. اگر خطایی ظاهر شد، جزئیات آنرا با استفاده از Event viewer استاندارد ویندوز می‌توان ملاحظه کرد.



شکل ۴- اطلاعات عمومی وب سرور آپاچی

سپس خطوط زیر را به انتهای فایل Apache2.2\conf\httpd.conf اضافه کنید:

```
<Location /trac>
    SetHandler mod_python
    PythonInterpreter main_interpreter
    PythonHandler trac.web.modpython_frontend
    PythonOption TracEnv "C:/Repositories/TracConfig/TracTest/"
    PythonOption TracUriRoot /trac
</Location>
```

برنامه Trac اکنون باید از طریق آدرس زیر در دسترس باشد:

<http://localhost:81/trac>

## (ب) روش دوم نصب Trac

تیم VisualSVN مراحل فوق را ساده کرده و امکان یکپارچه کردن Trac و VisualSVN Server را فراهم آورده است. برای دریافت بسته نصب آن به آدرس زیر مراجعه کنید:

<http://www.visualsvn.com/server/trac/>

توصیه می‌شود Trac را از این راه نصب کنید (زیرا تمامی مازول‌های موجود در این بسته با یکدیگر کاملاً همخوانی داشته و در حین اجرای Trac با پیام‌های خطای گوناگونی مواجه نخواهید شد). نحوه نصب قدم به قدم در آدرس فوق بر اساس بسته ارائه شده نیز ذکر شده است که چند تنظیم بسیار ساده می‌باشند. بعلاوه امکان یکپارچگی با حالت windows authentication را نیز در این حالت خواهید داشت که مزیت امنیتی بسیار بزرگی است.



## مدیریت Wiki در Trac

راهنمای کاملی در برنامه Trac جهت کار با Wiki و قالب‌های پشتیبانی شده در آن وجود دارد که قصد تکرار آنرا در این فصل نخواهیم داشت. پس از ورود به Trac ، بر روی لینک Help/Guide در بالای صفحه کلیک کرده و در صفحه باز شده در قسمت Table of contents بر روی لینک Wiki Formatting کلیک نمایید. در اینجا با لیست کاملی از قالب‌های ساخت Wiki آشنا خواهید شد. برای مثال برای ایجاد یک جدول در Wiki می‌توان از فرمت زیر استفاده کرد:

```
||Cell 1||Cell 2||Cell 3||
||Cell 4||Cell 5||Cell 6||
```

که خروجی نهایی صفحه در این حالت به صورت زیر خواهد بود:

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

لازم به ذکر است هنگامیکه که در پایین هر صفحه بر روی دکمه Edit this page کلیک کنید، ادیتور باز شده به کمک نوار ابزار بالای آن پرکاربردترین قالب‌های Wiki را بدون نیاز به حفظ کردن آنها در اختیار شما قرار می‌دهد (شکل ۵). اگر این نوار ابزار را در برنامه IE ویندوز سرور مشاهده نمی‌کنید، آدرس سایت را باید توسط گزینه Security در قسمت Trusted sites وارد نمایید.

### Editing WikiStart



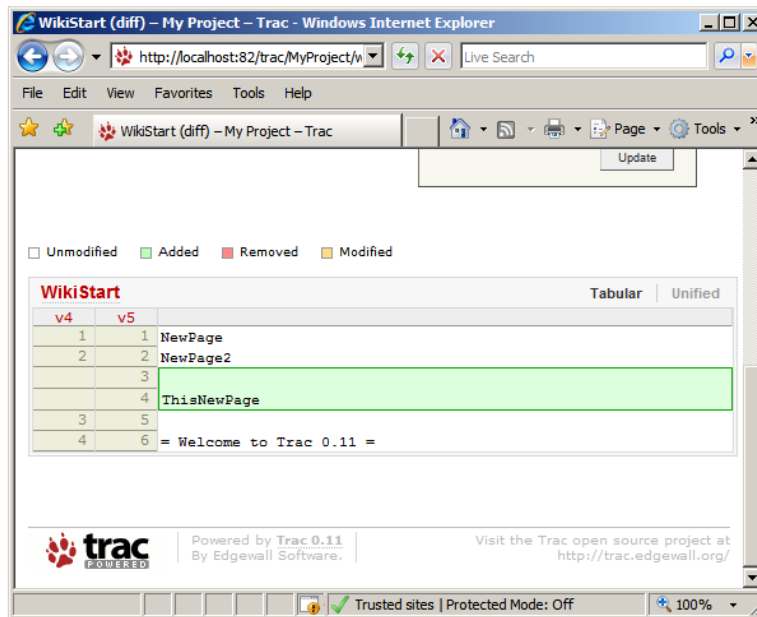
شکل ۵- ادیتور Wiki در برنامه Trac

یکی از مهم‌ترین مواردی که در یک Wiki با آن برخورد خواهیم داشت، ایجاد لینک به یک صفحه جدید و ایجاد صفحه‌ای کاملاً جدید است. اگر با اصول نامگذاری توابع در زبان‌های برنامه نویسی آشنا باشید احتمالاً نام CamelCase را شنیده‌اید. در حالت CamelCase ، فاصله بین کلمات حذف و اولین کاراکتر آنها با حرف بزرگ نوشته می‌شود برای مثال ThisPage . در هنگام ایجاد صفحات Wiki ، برنامه Trac به صورت خودکار موارد CamelCase را تشخیص داده و به صورت لینک‌هایی خاکستری با یک علامت سؤال نمایش می‌دهد. با کلیک بر روی آنها به صفحه جدیدی رهنمون خواهیم شد که با کلیک بر روی دکمه Create this page ، امکان ایجاد محتوای صفحه جدید به سادگی فراهم خواهد شد.

اگر نمی‌خواهید عبارات CamelCase به صورت خودکار تبدیل به لینک شوند پیش از آنها یک علامت تعجب قرار دهید. این علامت تنها هنگام ویرایش صفحه نمایش داده می‌شود.

برای مشاهده لیست صفحات ایجاد شده در برنامه Trac ، در منوی اصلی بالای صفحه، بر روی لینک Index کلیک نمایید. همچنین برای مشاهده لیست تغییرات انجام شده بر روی هر صفحه و مشاهده نگارش‌های مختلف آنها می‌توان بر روی دو لینک History و Last change در منوی اصلی بالای هر صفحه کلیک کرد (شکل ۶).

در صفحه Diff دو نگارش یک صفحه Wiki ، لینک‌های Previous change و Next Change جهت مشاهده تغییرات نگارش‌های قبلی و بعدی نیز مهیا هستند.



شکل ۶- مشاهده ریز تغییرات انجام شده بر روی یک صفحه Wiki سایت.

## ردیابی وظایف در Trac

احتمالا با انجمن‌های بحث و گفتگو در اینترنت آشنایی دارید (forums). گروهی از برنامه‌نویس‌ها برای مدیریت پروژه خود از این نوع سیستم‌ها استفاده می‌کنند. برای مثال حداقل چهار بخش مختلف تازه‌ها، گزارش خطاها، راهنمای عمومی کاربران و بخش پیشنهاد ویژگی‌های جدید را در این انجمن‌ها می‌توان دید. Trac نیز به صورت یکپارچه چنین ویژگی‌هایی را با دیدگاه‌های ویژه مدیریت پروژه ارائه می‌دهد. از Wiki به صورت بسیار پیشرفته‌تری می‌توان جهت ارائه راهنمای سیستم به کاربران استفاده کرد. اشخاص می‌توانند توسط یک بلیط جدید مطلب جدیدی را گزارش دهند. اینبار به صورت ساخت یافته‌تر حالت‌های پیش فرض خطا، پیشنهادات و وظایف را می‌توان مشاهده کرد که اینها نیز قابلیت تغییر و سفارشی سازی را نیز دارا هستند. هر مطلب ارسال شده را می‌توان به یکی از اعضای تیم انتساب داد و همچنین افرادی که علاقمند باشند می‌توانند نظرات خود را پیرامون مطلب ارسالی بیان کنند. در ادامه این موارد را بررسی خواهیم کرد.

## الف) سیستم بلیط

در Trac با کلیک بر روی لینک **New ticket** می‌توان بلیط جدیدی را ایجاد نمود (شکل ۷). در این صفحه ابتدا کاربر، عنوانی را وارد می‌نماید. از این عنوان در نمایش لیست بلیطها در برنامه استفاده می‌شود (لینک **View Tickets** در منوی بالای صفحه برنامه Trac). سپس کاربر نوع بلیط را انتخاب کرده و توضیحات مربوطه را وارد می‌کند. امکان الصاق فایل، انتساب آن به افراد پروژه، رونوشت (Cc) و غیره نیز در این صفحه پیش بینی شده است. بعلاوه امکان نظر دهی سایر کاربران نیز در این صفحه پس از ارسال میسر است (همانند یک انجمن). پس از ثبت، شماره‌ای خودکار به بلیطها انتساب داده می‌شود و اگر زمانی نیاز به ارجاع به یکی از بلیطها در Wiki وجود داشت تنها کافی است شماره بلیط را به صورت #1 وارد کرد (و یا ticket:1). در این مثال به صورت خودکار لینکی به بلیط شماره یک ایجاد خواهد شد.

شکل ۷- ایجاد یک بلیط جدید در Trac

در صفحه ثبت بلیط اگر دقت کرده باشید یک سری فیلد از پیش تعریف شده مانند میزان اهمیت، قسمت ویژه‌ای از پروژه (زیر سیستم) که در اینجا مورد بحث است (component) و غیره را می‌توان مشاهده نمود. برای تغییر این پیش فرض‌ها حداقل دو راه وجود دارد. اجرای `trac-admin` از طریق خط فرمان و یا ویرایش بانک اطلاعاتی برنامه به صورت مستقیم. این برنامه در مسیر زیر قرار دارد (اگر از Trac نصب شده توسط VisualSVN server که پیشتر توضیح داده شد، استفاده می‌کنید):

```
C:\>"C:\Program Files\VisualSVN Server\trac\python\python.exe" "C:\Program Files\VisualSVN Server\trac\python\scripts\trac-admin-script.py" trac-admin
```

اکنون به محیط خط فرمان برنامه `trac-admin` وارد شده اید:

```
Welcome to trac-admin 0.11
Interactive Trac administration console.
Copyright (c) 2003-2008 Edgewall Software
Type: '?' or 'help' for help on commands.
Trac [C:\trac-admin]>
```

در اینجا برای مثال دستور `ticket_type list`، لیست بلیط‌ها را نمایش می‌دهد و یا با استفاده از دستور زیر می‌توان مقداری موجود را به مقداری جدید تبدیل کرد:

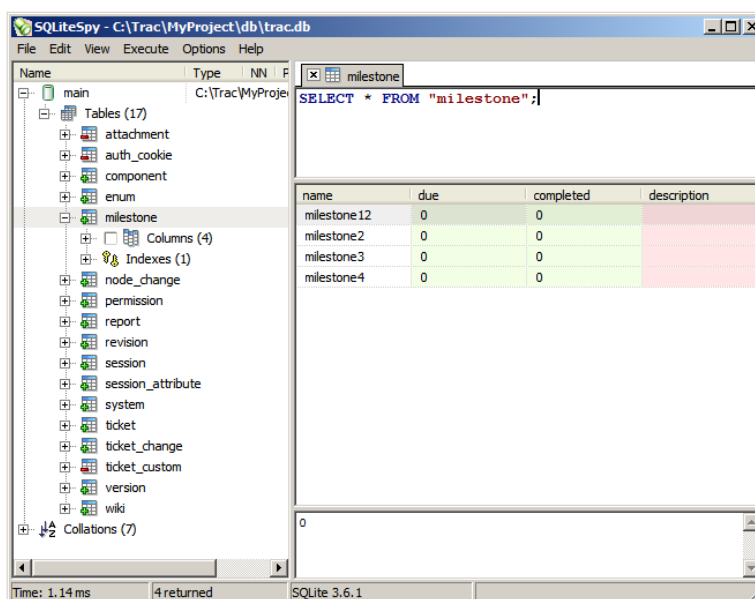
```
ticket_type change <value> <newvalue>
```

لیستی از دستورات قابل اجرا را با تایپ `help` در خط فرمان برنامه مدیریتی Trac می‌توان بدست آورد.

اگر علاقمند باشید که بانک اطلاعاتی SQLite برنامه را مستقیماً و بصری ویرایش کنید (یا خصوصاً اطلاعات فارسی را به سادگی وارد نمائید)، ابتدا برنامه رایگان SQLiteSpy را از آدرس زیر دریافت کنید:

<http://www.yunqa.de/delphi/doku.php/products/sqlitespy/index>

بانک اطلاعاتی سایت Trac برای نمونه در مسیر C:\Trac\MyProject\db قرار دارد. این بانک را توسط برنامه باز نمائید. همانطور که در تصویر نیز مشخص است، رکوردهای جدول مختلف این بانک اطلاعاتی را به سادگی می‌توان توسط این برنامه ویرایش کرده و تغییر داد (بر روی هر مقدار دکمه F2 را فشرده و مقدار موجود را ویرایش کنید).



شکل ۸- برنامه رایگان و بسیار سریع SQLiteSpy جهت مدیریت یک بانک اطلاعات SQLite.

علاوه بر این روش‌ها یک Web Admin Plugin نیز برای Trac موجود است که بتوان از طریق خود برنامه و محیط وب این تغییرات را اعمال نمود.

<http://trac.edgewall.org/wiki/WebAdmin>

سایتی نیز به این نوع فعالیت‌ها و توسعه افزودنی‌های مخصوص Trac در آدرس زیر مشغول به فعالیت است:

<http://trac-hacks.org/>

 نکته:

امکان دریافت ایمیل اطلاع رسانی پس از ایجاد هر بلیط و یا انجام تغییرات بر روی آنها نیز میسر است. برای این منظور باید فایل `tarc.ini` را گشوده و قسمت `notification` آنرا فعال کرد:

```
[notification]
smtp_enabled = true
smtp_server = mail.mycompany.com
smtp_from = trac@mycompany.com
smtp_replyto = myproj@mycompany.com
smtp_always_cc = projectmanager@mycompany.com, qa@mycompany.com
always_notify_owner = true
always_notify_reported = true
always_notify_updater = true
```

برای ارسال ایمیل به سایر کاربران برنامه، آنها باید قسمت مربوط به تنظیمات کاربری را تکمیل نموده و ایمیل خود را وارد کنند. برای این منظور باید به لینک Settings در منوی اصلی Trac مراجعه نمود.

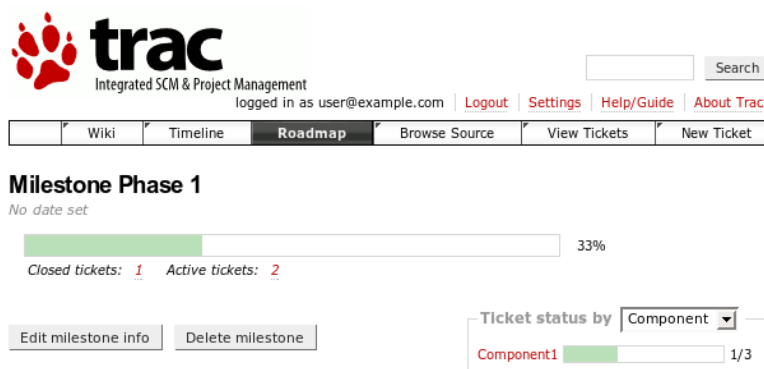
علاوه بر آن در انتهای صفحه Timeline، امکان دریافت لینک تغذیه خبری سایت (RSS feed) نیز مهیا است. به این صورت بدون فعال سازی قسمت ارسال ایمیل‌های اطلاع رسانی نیز می‌توان از اخبار تغییرات سایت مطلع شد. برای مثال می‌توان از خبرخوان رایگان و بسیار حرفه‌ای FeedDemon استفاده کرد:

<http://www.newsgator.com/Individuals/FeedDemon/Default.aspx>

و یا Outlook 2007 نیز امکانات یک خبرخوان استاندارد را به صورت یکپارچه به شما ارائه می‌دهد (هر چند از لحاظ سفارشی سازی و امکانات ویژه با FeedDemon قابل مقایسه نیست). همچنین آخرین نگارش‌های Firefox و IE نیز مجهز به خبرخوان هستند.

## ب) زمان‌های ارزیابی (milestones)

Milestones امکان گروه بندی بلیط‌ها و بررسی آن‌ها را در Trac فراهم می‌کنند. بر اساس تعداد بلیط‌های بررسی و رفع شده در یک milestone، میزان پیشرفت آن مشخص می‌شود (شکل ۹). نحوه ایجاد milestones و یا ویرایش نمونه‌های از پیش تعریف شده را در قسمت قبل بررسی کردیم. همچنین می‌دانیم که هنگام تعریف و ایجاد یک بلیط جدید در برنامه، milestone مربوطه را نیز می‌توان انتخاب و مشخص کرد. برای مشاهده milestones می‌توان از طریق لینک roadmap (کارراهه) در منوی اصلی برنامه اقدام کرد. از طریق این قسمت، milestone های خاتمه نیافته و درصد پیشرفت آنها قابل بررسی است.



شکل ۹- milestones در Trac.

برای مشاهده ریز فعالیت‌های روزانه انجام شده می‌توان به قسمت Timeline منوی اصلی برنامه مراجعه کرد. امکان دریافت تقویم فعالیت‌ها در انتهای صفحه Roadmap به صورت فرمت استاندارد iCalendar نیز مهیا است. برای مثال برنامه Outlook این فرمت استاندارد را پشتیبانی می‌کند.

## ج) مرور مخزن کد

مهم‌ترین دلیلی که جهت پرداختن به Trac در این فصل از کتاب می‌توان برشمرد، رابط دسترسی تحت وب بسیار پیشرفته‌تر آن برای کار با یک مخزن کد SVN نسبت به ابزار استاندارد mod\_dav\_svn می‌باشد (شکل ۱۰).

trac  
Integrated SCM & Project Management  
logged in as user@example.com | Logout | Settings | Help/Guide | About Trac

Wiki | Timeline | Roadmap | **Browse Source** | View Tickets | New Ticket  
Last Change | Revision Log

root

View revision:

Name ▲	Size	Rev	Age	Last Change
branches		1	2 weeks	user@example.com: Initial repository structure
tags		1	2 weeks	user@example.com: Initial repository structure
trunk		3	1 hour	user@example.com: Added the ability to say hello to an inputted r

Note: See [TracBrowser](#) for help on using the browser.

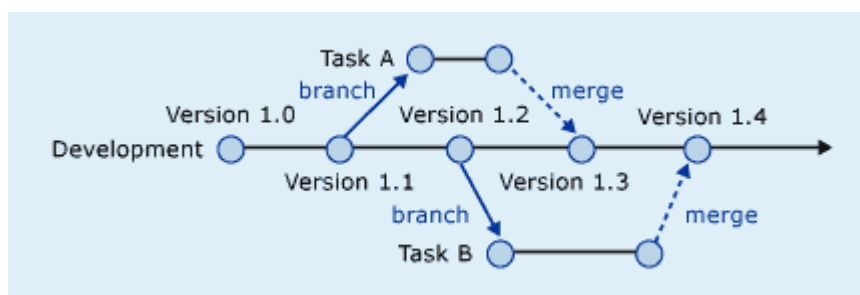
View changes...

شکل ۱۰- مرور مخزن کد در Trac.

اگر مراحل نصب Trac را بخاطر داشته باشید، یکی از مواردی که هنگام نصب باید مشخص شود مسیر مخزن کد است. پس از نصب، در قسمت `browse source` منوی Trac، می‌توان از طریق وب به مخزن کد دسترسی داشت، تغییرات را مشاهده نمود، توضیحات ارائه شده کاربران را بررسی کرده و حتی عملیات `diff` را به صورت آنلاین و در مرورگر وب خود مشاهده نمود. با استفاده از Trac عملیات متداول استفاده از SVN به صورت زیر درخواهد آمد:

- ۱- یک بلیط جدید را ایجاد کنید.
- ۲- `milestone` (بازه زمانی انجام یا زمان ارزیابی) و برنامه نویسی آنرا مشخص کنید.
- ۳- انشعابی (`branch`) را جهت کار بر روی این بلیط ایجاد نمایید.
- ۴- تغییرات را انجام داده و به انشعاب اعمال کنید.
- ۵- پس از پایان کار، تغییرات را با `trunk` ادغام کنید.
- ۶- بلیط را خاتمه دهید.

این روش در مورد رفع باگ‌ها و یا پیشنهادهای بهبود برنامه یکسان است. در مورد بهبودهای پیشنهادی بهتر است یک صفحه `Wiki` را به آن اختصاص داده، موارد را در آن مطرح کنید و با توجه به قابل ویرایش بودن آن توسط سایر اعضای تیم، نظرات آنها را جویا شوید. سپس به ازای هر مورد یک بلیط جدید ایجاد کرده و لینک آنها را در `Wiki` منعکس کنید (روش برای مثال `ticket:1` که در مورد آن توضیح داده شد). این بلیط‌ها با نوع پیشنهاد مشخص می‌شوند (اگر این نوع به صورت پیش فرض وجود ندارد، آنرا به بانک اطلاعاتی برنامه اضافه کنید). با توجه به اینکه در این سیستم هر بلیط باید دارای یک انشعاب مجزا باشد، هنگام `commit` تغییرات به مخزن کد، در توضیحات مربوطه شماره بلیط را نیز ذکر نمایید. برای مثال انشعاب `feature-29` را مطابق بلیط شماره ۲۹ ایجاد کرده و سپس توضیحات `commit` تغییرات انجام شده باید برای مثال `Addresses ticket #29 by implementing spell checking` باشد. در این حالت دقیقاً می‌دانیم که چه کاری قرار است انجام شود (توسط ایجاد یک بلیط جدید)، موعد آن چه زمانی است (`milestone`)، چه شخصی باید بر روی آن کار کند (انتساب بلیط به شخص) و همچنین با توجه به ایزوله شدن کار در یک انشعاب مجزا از `trunk`، تداخلی در خط اصلی تولید پدید نخواهد آمد (شکل ۱۱). علاوه توسط Trac و محیط تحت وب آن می‌توان نظرات دیگران را درباره یک بلیط جویا شد و به بحث و تبادل نظر پرداخت. پس از پایان کار و اعمال تغییرات و ادغام نهایی، می‌توان بلیط را بست (قسمت `action` در صفحه بلیط و گزینه `resolve as`). درصد پیشرفت کار را نیز همانطور که پیشتر عنوان شد می‌توان از قسمت `roadmap` و مشاهده `milestone` مربوطه، بررسی کرد.



شکل ۱۱- پیاده سازی ایده ایجاد انشعاب به ازای هر بلیط جدید در Trac.

روال کاری برخورد با گزارشات خطاها اندکی متفاوت است (در Trac به آنها **defects** هم گفته می‌شود). اینجا نیز همواره گزارش یک خطا (حتی اگر توسط یکی از اعضای تیم صورت گیرد) باید با ایجاد یک بلیط جدید آغاز گردد. سپس باید پیش از اقدام بررسی شود آیا این خطا پیشتر گزارش شده است؟ آیا گزارش خطا معتبر است؟ آیا تیم برنامه نویسی می‌تواند آنرا مجدداً تولید کند؟ شدت آن در چه حدی است؟ اگر گزارش خطا تکراری است در قسمت **action** مربوط به یک بلیط می‌توان آنرا به صورت مورد تکراری (**duplicate**) معرفی کرده و کار را خاتمه داد (هر چند بهتر است شماره بلیط معادل را نیز در اینجا ذکر کرد). اگر گزارش خطا معتبر نبود یا قابل تکرار تشخیص داده نشد می‌توان در قسمت **action** گزینه **wontfix** را انتخاب کرده و به کار این مورد نیز خاتمه داد. سایر موارد انتساب بلیط به یک برنامه نویس یا تعیین **milestone** مربوطه همانند قبل در اینجا نیز صورت می‌گیرد. ایجاد انشعاب جهت رفع خطاها با ایجاد انشعاب جهت معرفی یک ویژگی جدید اندکی متفاوت است. در هنگام ایجاد انشعاب جهت کار بر روی یک ویژگی جدید همواره باید انشعاب بر اساس آخرین نگارش موجود در مخزن کد صورت گیرد، اما در حالت بررسی یک خطا باید انشعاب از نگارش مربوط به آن خطا ایجاد گردد. شاید به نظر رسد که پیدا کردن نگارش مربوطه اندکی مشکل است، اما اگر مباحث ایجاد برچسب‌ها (**tags**) در سیستم به درستی انجام شده باشد، کاربر هنگام گزارش خطا و ایجاد یک بلیط جدید، نگارش مربوطه را نیز انتخاب خواهد کرد (که این نگارش باید معادل باشد با یکی از برچسب‌های ایجاد شده). جهت ایجاد انشعاب نیز بهتر است شماره بلیط در نام انشعاب و همچنین موارد **commit** شده نیز لحاظ گردد (برای مثال ایجاد انشعاب **bug-51**).

### آشنایی با خواص **bugtraq** در SVN

امکان به روز رسانی Trac و تعامل با آن از طریق SVN با بکارگیری مباحث **Hooks** که پیشتر درباره آنها بحث شد نیز میسر است. در سایت Trac اسکرپتی جهت این امر توسعه یافته است که از آدرس زیر قابل دریافت است:

<http://trac.edgewall.org/browser/trunk/contrib/trac-post-commit-hook>

پس از تنظیم و قرار دادن آن در پوشه **Hooks** مربوط به SVN (همانطور که از فصل مربوطه به خاطر دارید باید یک فایل **bat** برای اینکار تهیه نمود، مسیر کامل **python** برای اجرای اسکرپیت کاملاً قید شود و همانند راهنمای ابتدای این اسکرپیت، **REPOS**، **REV** و **TRAC\_ENV** نیز باید تنظیم و مقدار دهی شوند)، پس از هر **commit** یک سری از واژه‌های مخصوص پیغام **commit** تشخیص داده شده و به صورت خودکار به Trac اعمال خواهند شد. برای مثال **fixes**، **fixes**، **fixes** به همراه شماره بلیط مربوطه، به صورت خودکار سبب بسته شده آن بلیط خواهند شد. برای مثال **Close #1** یا **Close #2 and #3** این اصلاحیه **fixes** و یا کلمات **addresses**، **re**، **references**، **refs** و **see** سبب افزوده شدن یک پیغام به بلیط مربوطه می‌شوند.

اگر علاقمند باشید هنگام **commit** اطلاعات به مخزن کد، در صفحه **commit** مربوط به برنامه **TortoiseSVN**، فیلد شماره بلیط متناظر نیز ذکر شود، باید به صورت زیر عمل کرد:

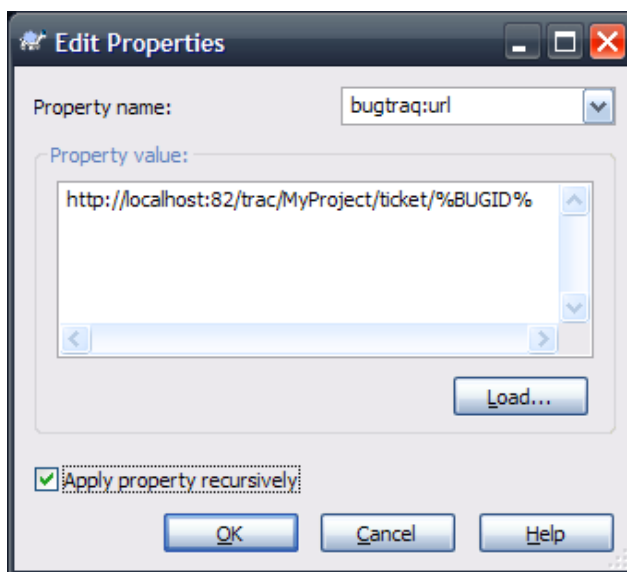
بر روی پوشه پروژه کلیک راست کرده و سپس گزینه زیر را انتخاب کنید:

TortoiseSVN → Properties

در صفحه باز شده بر روی دکمه **New** کلیک نمائید تا بتوان خاصیت جدیدی را افزود. خاصیت `bugtraq:message` را انتخاب کرده و مقدار آنرا برای مثال مساوی عبارت زیر قرار دهید:

`http://localhost:82/trac/MyProject/ticket/%BUGID%`

سپس گزینه اعمال به صورت بازگشتی را نیز انتخاب نموده و بر روی دکمه **OK** کلیک کنید. اکنون مجدداً بر روی دکمه **new** کلیک کرده و خاصیت `bugtraq:url` را انتخاب و مقدار آنرا نیز همانند `URL` فوق وارد نمائید (شکل ۱۲).

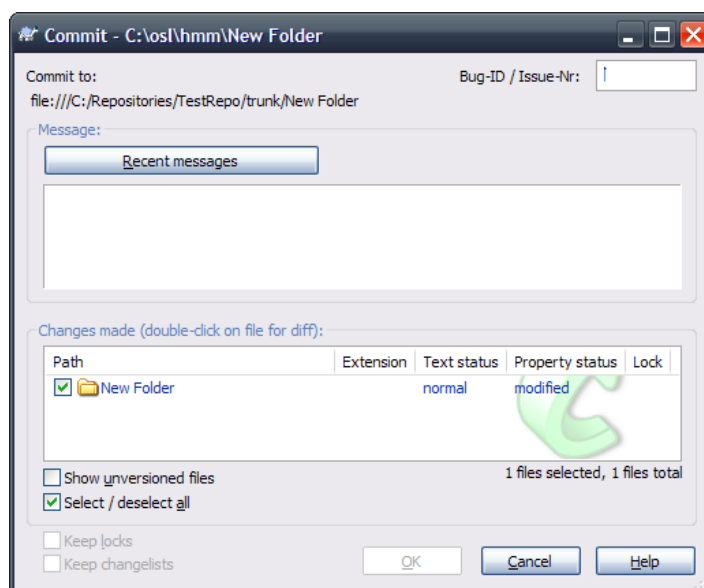


شکل ۱۲- اضافه کردن خاصیت `bugtraq:url` توسط TortoiseSVN.

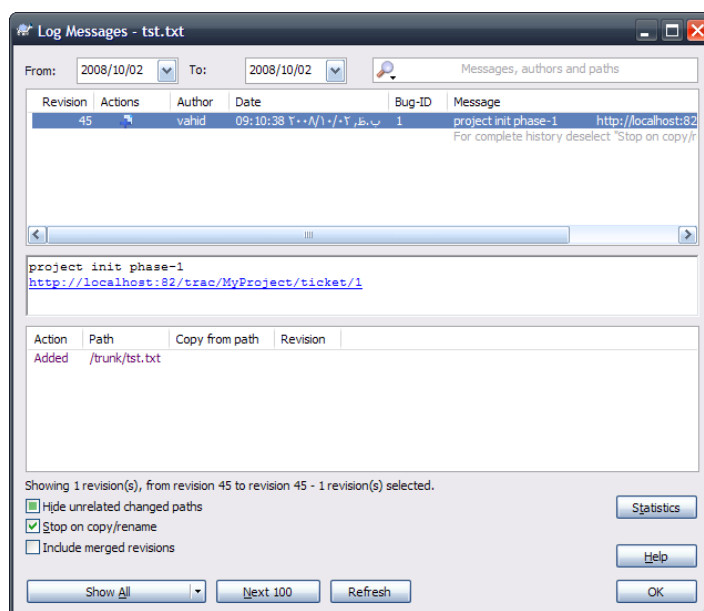
این `URL` در حقیقت به آدرس بلیط ایجاد شده در سیستم `Trac` ما اشاره می‌کند. پس از اعمال این خواص، هنگام `commit`، یک جعبه متنی (`TextBox`) با برچسب `Bug-ID/Issue-Nr` نیز ظاهر خواهد شد (شکل ۱۳). پس از `Commit` و ذکر شماره بلیط مربوطه، اگر به قسمت `show log` در `TortoiseSVN` مراجعه کنیم، یک ستون جدید تحت عنوان `Bug-ID` به گزارش مربوطه اضافه شده است. با انتخاب هر یک از ردیف‌های `log` ظاهر شده، آدرس قابل کلیک متناظر نیز ظاهر خواهد شد که در اینجا `%BUGID%` به صورت خودکار با عدد بلیط مربوطه مقدار دهی می‌شود (شکل ۱۴).

اگر نیاز به تغییر متن برچسب پیش فرض `bug-id/Issue-Nr` دارید، باید خاصیت `bugtraq:label` را به عبارت دلخواه خود مقدار دهی کنید (`TortoiseSVN → Properties`) و اگر می‌خواهید در صورت خالی بودن جعبه متنی `bug-id/Issue-Nr` هنگام `commit` اختطاری به کاربر نمایش داده شود خاصیت `bugtraq:warnifnoissue` را باید مقدار دهی نمود.





شکل ۱۳- افزودن جعبه متنی Bug-ID/Issue-Nr به صفحه Commit در TortoiseSVN.



شکل ۱۴- افزودن شدن ستون Bug-ID به گزارشات log در TortoiseSVN.

## فصل ۹ - یکپارچگی مداوم و پیوسته (Continuous Integration)

### مقدمه:

یکپارچگی مداوم یکی از مفاهیم اصلی کار تیمی و همچنین یکی از اصول متدولوژی Extreme Programming نیز می‌باشد. در یک تیم در طول روز کدهای بسیاری با مخزن کد هماهنگ خواهند شد. با استفاده از روش‌های یکپارچگی مداوم، پس از هر هماهنگ سازی با سرور مخزن کد، عملیات کامپایل پروژه و همچنین انجام تست‌های مربوطه به صورت خودکار صورت گرفته و نتایج موفقیت یا شکست آن به برنامه نویس‌ها یا تیم مدیریتی پروژه ارسال می‌گردد. به این صورت زمان نهایی برای یکپارچگی پروژه و تحویل آن به حداقل خواهد رسید. در این فصل برای راه اندازی یک Continuous Integration Server از برنامه CruiseControl.NET استفاده خواهیم کرد. CruiseControl.NET نگارش ویندوزی (C#) برنامه جاوای CruiseControl است که تحت اکثر سیستم عامل‌های موجود نگارش‌های ویژه آن قابل دسترسی است. این برنامه سورس باز از آدرس زیر قابل دریافت است:

<http://ccnet.thoughtworks.com/>  
<http://cruisecontrol.sourceforge.net/>

همچنین کلاینت‌ها برای دریافت وقایع مربوطه می‌توانند از برنامه CCTray استفاده نمایند. این برنامه را از آدرس زیر می‌توانید دریافت کنید:

<http://confluence.public.thoughtworks.org/display/CCNET/CCTray>

علاوه بر محصول بسیار متداول فوق، برنامه teamcity نیز می‌تواند برای این مقصود مورد استفاده قرار گیرد. نگارش حرفه‌ای این برنامه رایگان است. آدرس دریافت برنامه:

<http://www.jetbrains.com/teamcity>

محصولات سورس باز دیگری نیز در این ارتباط موجود هستند که به اندازه CruiseControl.NET متداول نبوده و صرفاً به معرفی لینک‌های دریافت آنها اکتفا خواهد شد:

Draco.NET <http://draconet.sourceforge.net/>  
Hippo.NET <http://hipponet.sourceforge.net/>

### نصب و راه اندازی CruiseControl.NET

فایل‌های نصب CruiseControl.NET و CCTray را از سایت thoughtworks دریافت نمائید (این آدرس‌ها به لینک زیر ختم خواهند شد).

<http://sourceforge.net/projects/ccnet/>

لازم به ذکر است CruiseControl.NET جهت نصب بر روی سروری که IIS (Internet Information Services) بر روی آن نصب شده است (برای راه اندازی Web Dashboard) و CCTray برای نصب بر روی کلاینت‌های متصل به CruiseControl.NET طراحی شده‌اند. این برنامه‌ها حداقل به Microsoft.NET Framework Version 2.0 جهت اجرا نیاز دارند.

نصب مقدماتی هر دو مورد نکته خاصی ندارد و با چندبار کلیک کردن بر روی دکمه Next و پذیرفتن پیش فرض‌های آن‌ها انجام خواهند شد.

پس از نصب مقدماتی، مهمترین قسمت کار با CruiseControl.NET تنظیمات فایل `ccnet.config` آن می‌باشد. اگر برنامه را در مسیر پیش فرض آن نصب کرده باشید، این فایل متنی در آدرس زیر قابل مشاهده خواهد بود:

`C:\Program Files\CruiseControl.NET\server\ccnet.config`

در ادامه چندین حالت مختلف تنظیمات این فایل را بررسی خواهیم کرد:

### الف) تنظیمات ccnet برای پروژه‌های دلفی

برخلاف اسم این برنامه که به `Net` ختم می‌شود، توانایی‌های آن محدود به برنامه‌های دات نت مایکروسافت نیست و برای مثال می‌توان از آن برای کنترل مداوم پروژه‌های دلفی نیز استفاده کرد. حتی اگر با دلفی برنامه نویسی نمی‌کنید، مطالب این قسمت را مطالعه نمایید زیرا اصول آن برای سایر محیط‌های توسعه نیز مشترک است. خطوط زیر محتوای تنظیم شده فایل `ccnet.config` برای این منظور هستند:

```
<cruisecontrol>
  <!-- This is your CruiseControl.NET Server Configuration file. Add
your projects below! -->
  <project name="Project1">
    <webURL>http://localhost/ccnet/</webURL>
    <sourcecontrol type="svn">
      <trunkUrl>file:///C:/SVNRepos/Project1/trunk</trunkUrl>
      <workingDirectory>c:\Sandbox\Project1</workingDirectory>
      <executable>c:\program
files\subversion\bin\svn.exe</executable>
      <username>user1</username>
      <password>pass1</password>
      <autoGetSource>>true</autoGetSource>
      <tagOnSuccess>>false</tagOnSuccess>
      <tagBaseUrl>file://projects/project1/tags</tagBaseUrl>
      <timeout>12000000</timeout>
    </sourcecontrol>
    <triggers>
      <intervalTrigger seconds="60" />
    </triggers>
    <tasks>
      <exec>
        <executable>"c:\program
files\Borland\BDS\4.0\bin\dcc32.exe"</executable>
        <baseDirectory>C:\Sandbox\Project1</baseDirectory>
        <buildArgs>-q project1.dpr</buildArgs>
        <buildTimeoutSeconds>10</buildTimeoutSeconds>
      </exec>
    </tasks>
  </project>
</cruisecontrol>
```

در این فایل موارد زیر را مشاهده خواهید کرد:

توضیحات	عبارت
نام دلخواه پروژه‌ای که بر روی آن مشغول به کار هستیم.	project name
آدرس web dashboard برنامه	webURL
تنظیمات sourcecontrol	
نوع برنامه کنترل سورس کد مورد استفاده. در اینجا از SVN استفاده شده است. (برای مثال ccnet با SourceSafe مایکروسافت و یا CVS نیز می‌تواند یکپارچه شود)	sourcecontrol type
آدرس مخزن trunk (فرض بر این است که پروژه خود را تحت کنترل مخزن کد SVN قرار داده‌اید و اکنون آدرس trunk آنرا در این قسمت باید وارد کنید).	trunkUrl
مکان فیزیکی پروژه شما که اولین checkout در آنجا صورت گرفته است.	workingDirectory
مسیر فایل اجرایی برنامه SVN (svn.exe) بر روی سیستم.	Executable
نام کاربری جهت اتصال به مخزن کد	Username
کلمه عبور کاربر فوق جهت اتصال به مخزن کد	Password
آیا برای کامپایل پروژه آخرین سورس کدها را به صورت خودکار از SVN دریافت کند؟	autoGetSource
آیا به ازای هر کامپایل موفقیت آمیز یک برچسب جدید در SVN ایجاد شود؟	tagOnSuccess
آدرس ریشه اصلی برچسب‌ها در مخزن کد شما	tagBaseUrl
حداکثر زمان مجاز تعیین شده جهت انجام عملیات بر اساس میلی ثانیه	Timeout
تنظیمات triggers	
تعیین کننده فواصل زمانی معینی است برحسب ثانیه که ccnet به SVN مراجعه کرده و آخرین تغییرات هماهنگ شده را دریافت می‌کند.	intervalTrigger
تنظیمات tasks	
مسیر کامپایلر مورد استفاده	Executable
مسیر فیزیکی پروژه	baseDirectory
آرگومان‌های کامپایل کردن پروژه	buildArgs
حداکثر زمان مجاز تعیین شده جهت انجام عملیات بر حسب ثانیه	buildTimeoutSeconds

پس از ذخیره سازی تغییرات فوق، برنامه CruiseControl.NET را اجرا کنید تا فایل ccnet.config خوانده شده و سرویس مربوطه اجرا گردد.

Start → All programs → CruiseControl.NET

اکنون مخزن کد شما تحت کنترل CruiseControl.NET قرار گرفته است. برای مثال تغییراتی را اعمال کرده و آنرا با سرور هماهنگ نمایید (commit). پس از لحظه‌ای، CruiseControl.NET متوجه تغییرات شده و پروژه را کامپایل می‌کند. در اینجا نتایج را به روش‌های مختلفی می‌توان مشاهده کرد. برای مثال پیگیری اطلاعات نمایش داده شده در کنسول برنامه و یا مراجعه به web dashboard آن در آدرس بعد (این وب سایت ASP.Net هنگام نصب CruiseControl.NET ایجاد شده است):

<http://localhost/ccnet>

در ویندوز سرور ۲۰۰۳ به صورت پیش فرض قادر به اجرای برنامه‌های ASP.Net توسط IIS 6 نخواهید بود و باید آنرا در قسمت webservice extensions کنسول مدیریتی IIS فعال کنید و یا می‌توانید از دستور خط فرمان زیر نیز استفاده نمایید:

```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis -i -enable
```

در ویندوز سرور ۲۰۰۸ و IIS 7، اولین تلاش جهت اجرای سایت ccnet با خطای زیر متوقف خواهد شد:

HTTP Error 500.23 - Internal Server Error. An ASP.NET setting has been detected that does not apply in Integrated managed pipeline mode.

برای رفع این خطا دستور خط فرمان زیر را اجرا کنید تا تنظیمات لازم بر روی فایل web.config سایت انجام شود:

```
c:\windows\system32\inetsrv\Appcmd migrate config "default web site/ccnet"
```

پس از اجرای این دستور، تگ‌های زیر به صورت خودکار به فایل web.config افزوده خواهند شد:

```
<system.webServer>
  <handlers>
    <add name="*.xml_*" path="*.xml" verb="*"
type="ThoughtWorks.CruiseControl.WebDashboard.MVC.ASPNET.HttpHandler,ThoughtWorks.C
ruiseControl.WebDashboard" precondition="integratedMode, runtimeVersionv2.0" />
    <add name="*.aspx_*" path="*.aspx" verb="*"
type="ThoughtWorks.CruiseControl.WebDashboard.MVC.ASPNET.HttpHandler,ThoughtWorks.C
ruiseControl.WebDashboard" precondition="integratedMode, runtimeVersionv2.0" />
  </handlers>
  <validation validateIntegratedModeConfiguration="false" />
</system.webServer>
```

اکنون سایت با IIS 7 نیز سازگار خواهد بود و به راحتی اجرا خواهد شد.

جهت مشاهده علت این تغییرات به آدرس زیر مراجعه کنید:

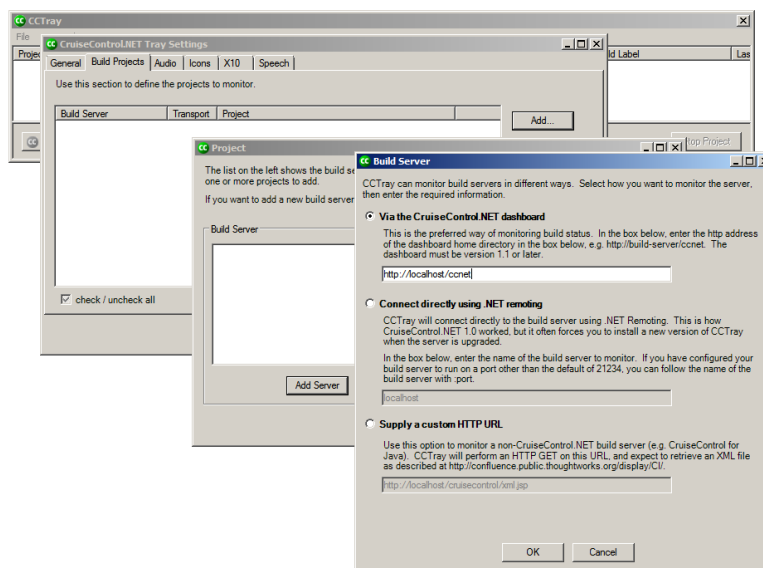
<http://learn.iis.net/page.aspx/381/aspnet-20-breaking-changes-on-iis-70/>

راه دیگر جهت مطلع شدن از وقایع رخ داده، استفاده از برنامه سمت کلاینت CCTray است. پس از نصب و اجرای آن، اکنون برنامه کنار ساعت ویندوز نمایان خواهد شد. پس از دوبار کلیک بر روی آن، در صفحه باز شده مسیر زیر را جهت معرفی سرور باید پیمود (شکل ۱):

File → Settings → Build projects → Add → Add server

در اینجا یکی از سه روش را جهت معرفی سرور ccnet به برنامه انتخاب نمایید. پس از کلیک بر روی دکمه OK لیستی از پروژه‌های تحت کنترل نمایش داده خواهد شد که پس از انتخاب یکی از آنها، برنامه کار مطلع سازی خودکار خویش را شروع خواهد نمود. علاوه بر CCTray، یک Gadget مخصوص ویندوز ویستا نیز برای اینکار در آدرس زیر قابل دریافت است:

<http://code.google.com/p/ccnet-monitor-sidebar-gadget/>



شکل ۱- معرفی کردن سرور ccnet به برنامه cctray.

## ب) تنظیمات ccnet برای پروژه‌های VS.Net

قسمت عمده‌ای از تنظیمات و توضیحات در اینجا نیز همانند قسمت الف است. تنها تفاوت موجود در تنظیمات این فایل برای کار با پروژه‌های VS.Net، قسمت tasks است که در ادامه بررسی خواهد شد.

```
<tasks>
  <msbuild>
<executable>C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\MSBuild.exe</executable>
    <workingDirectory>C:\Sandbox\Project1</workingDirectory>
    <projectFile>TestSystem.sln</projectFile>
    <buildArgs>/noconsolelogger /p:Configuration=Debug /v:diag</buildArgs>
    <targets>Build</targets>
    <timeout>15</timeout>
<logger>ThoughtWorks.CruiseControl.MsBuild.XmlLogger,ThoughtWorks.CruiseControl.MsBuild.dll</logger>
  </msbuild>
</tasks>
```

در تنظیمات قبل بجای کامپایلر دلفی از MSBuild مایکروسافت که همراه Net framework نصب می‌شود استفاده شده است. بنابراین نیازی به نصب VS.Net بر روی سرور نخواهد بود.

همانطور که در تنظیمات فوق ملاحظه می‌کنید ارجاعی به MsBuild.XmlLogger نیز وجود دارد. این XMLLogger را از آدرس زیر می‌توانید دریافت کنید:

<http://ccnetlive.thoughtworks.com/MSBuildXmlLogger-Builds/>

سپس فایل dll آنرا در پوشه اصلی پروژه خود کپی کنید.

در ادامه باید فایل `dashboard.config` را جهت نمایش بهتر خروجی MSBuild تنظیم کنیم. این فایل در مسیر نصب `web dashboard` برنامه قرار دارد.

```
<xslFileNames>
  <xslFile>xsl\compile-msbuild.xsl</xslFile>
</xslFileNames>
```

سطر فوق را به تگ `xslFileNames` اضافه نمائید. در همین فایل سطر زیر را نیز اضافه نمائید:

```
<xslReportBuildPlugin description="MSBuild Output"
actionName="MSBuildOutputBuildPlugin" xslFileName="xsl\msbuild.xsl" />
```

به این صورت خروجی دلپذیرتری را با MSBuild در `web dashboard` می‌توان تجربه کرد.

اگر به هر دلیلی VS.Net را بر روی سرور نصب کرده‌اید، می‌توان قسمت `tasks` را به صورت زیر نیز تنظیم نمود:

```
<tasks>
  <devenv>
    <solutionfile>C:\Sandbox\Project1</solutionfile>
    <configuration>Debug</configuration>
    <buildtype>Build</buildtype>
    <executable>C:\Program Files\Microsoft Visual Studio
8\Common7\IDE\devenv.com</executable>
    <buildTimeoutSeconds>60</buildTimeoutSeconds>
  </devenv>
</tasks>
```

## نکته‌ای درباره Web Application projects

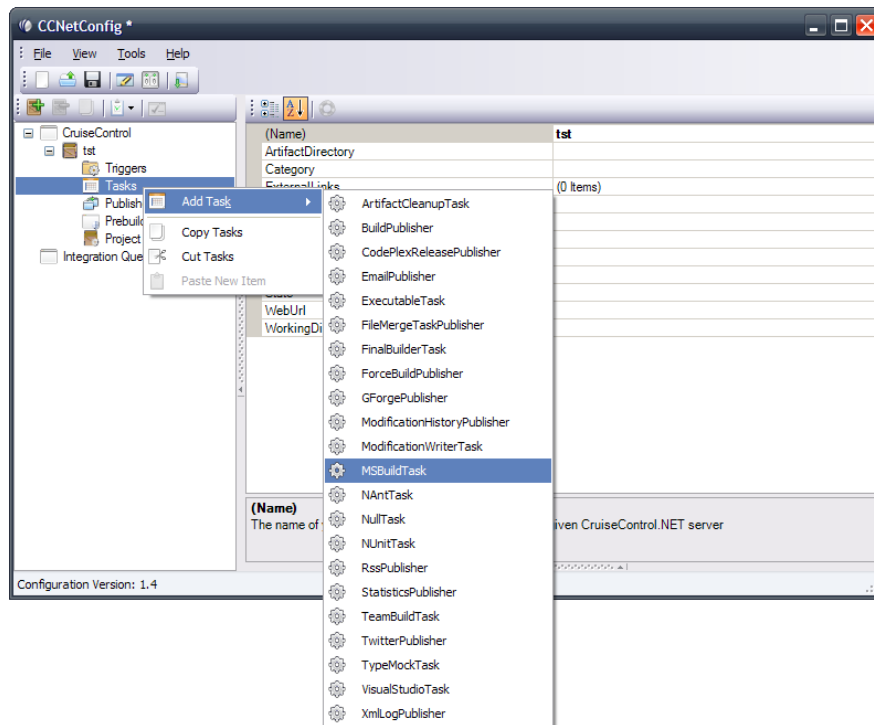
اگر سرویس پک ۱ ویژوال استودیو ۲۰۰۵ را نصب کرده باشید و یا از ویژوال استودیو ۲۰۰۸ استفاده می‌کنید، به طور قطع دریافته‌اید که مجدداً `Web Application projects` همانند ویژوال استودیو ۲۰۰۳ به مجموعه اضافه شده است. هنگام تحت کنترل قرار دادن اینگونه پروژه‌ها توسط `ccnet` بر روی سروری که `VS.Net` بر روی آن نصب نیست با مشکل مواجه خواهید شد. برای حل این مشکل فایل `Microsoft.WebApplication.targets` را از یکی از کلاینت‌ها از مسیر زیر دریافت

```
C:\Program Files\MSBuild\Microsoft\VisualStudio\v8.0\WebApplications\
```

و در مسیری مشابه مسیر فوق در سرور کپی نمائید (بدیهی است مسیر را باید به صورت دستی ایجاد کنید). همچنین اگر از `MS-Ajax` در پروژه خود استفاده کرده‌اید، آنرا نیز باید بر روی سرور نصب نمائید.

با استفاده از پروژه سورس باز زیر می‌توان تنظیمات ارائه شده در این فصل را به سادگی هرچه تمامتر و در اسرع وقت انجام داد (شکل ۲):

<http://www.codeplex.com/ccnetconfig>



شکل ۲- نمایی از برنامه CCNetConfig.



## فصل ۱۰ - دسترسی به SVN با برنامه نویسی

### مقدمه:

به تمامی امکانات مهیا در SVN با استفاده از API آن که به زبان C نوشته شده است نیز می‌توان دسترسی داشت. برای مثال TortoiseSVN از این API برای دسترسی به SVN استفاده می‌کند.

به آخرین مستندات API برنامه SVN به زبان C در آدرس زیر می‌توان دسترسی داشت:  
<http://svn.collab.net/svn-doxygen/>

برای استفاده از این API در سایر زبان‌های برنامه نویسی کتابخانه‌های محصورکننده‌ای (wrapper) نیز توسعه یافته‌اند که از آن جمله می‌توان به SubversionSharp (کتابخانه مبتنی بر دات نت) و SVNKit (کتابخانه مبتنی بر جاوا) اشاره کرد. SubversionSharp از آدرس زیر قابل دریافت است:

<http://www.softec.st/en/OpenSource/ClrProjects/SubversionSharp/>

برای تهیه یک نسخه کاری از سورس کد کتابخانه فوق، علاوه بر تمامی روش‌هایی که تاکنون گفته شد، از دستور خط فرمان زیر نیز می‌توان استفاده کرد:

```
svn co http://svn.softec.st/clr/trunk/SubversionSharp "C:\newPath"
```

نگارش دات نت ۲ این کتابخانه از آدرس‌های زیر قابل دریافت است:

<http://sharpsvn.open.collab.net/>

<http://www.pumacode.org/projects/svndotnet/>

همچنین نمونه مبتنی بر جاوا این نوع کتابخانه‌ها از سایت <http://svnkit.com> قابل دریافت است. و یا اگر از فصل‌های قبل به‌خاطر داشته باشید افزودنی‌های AnkhSVN و DelphiSVN نیز مستقیماً با SVN از طریق API آن ارتباط برقرار می‌کنند. بعلاوه امکان برقراری ارتباط با SVN از طریق اشیاء COM نیز توسط کتابخانه SVNCOM که از آدرس زیر قابل دریافت است نیز میسر است:

[http://www.pushok.com/soft\\_svncom.php](http://www.pushok.com/soft_svncom.php)

در ادامه با ارائه چند برنامه کوتاه، نحوه برقراری ارتباط با SVN را از طریق برنامه نویسی مرور خواهیم کرد.

**مثال ۱:** برنامه C# جهت بدست آوردن لیست فایل‌های یک نسخه کاری که با مخزن کد هماهنگ نشده‌اند.

در این مثال فرض بر این است که ارجاعی از کتابخانه SharpSVN (قابل دریافت از <http://sharpsvn.open.collab.net>) را به برنامه Console خود افزوده‌اید.

```
using System;
using System.Collections.ObjectModel;
using SharpSvn;

namespace SVNTest
{
    class Program
    {
        static void Main()
        {
            using (SvnClient client = new SvnClient())
            {
                Collection<SvnStatusEventArgs> changedFiles;
                client.GetStatus(@"C:\osl\testPrj", out
                                changedFiles);

                if (changedFiles != null)
                    foreach (SvnStatusEventArgs changedFile in
                             changedFiles)
                    {
                        switch (changedFile.LocalContentStatus)
                        {
                            case SvnStatus.Missing:
                                Console.WriteLine("Missing: " +
                                                  changedFile.Path);
                                break;
                            case SvnStatus.NotVersioned:
                                Console.WriteLine("NotVersioned: " +
                                                  changedFile.Path);
                                break;
                        }
                    }
            }
        }
    }
}
```

در قطعه کد فوق یک شیء کلاینت SVN ایجاد شده و سپس وضعیت فایل‌های موجود در نسخه کپی محلی دریافت شده است.

**مثال ۲:** تابع C# دریافت و نمایش لیست لاگ‌های ارسالی هنگام commit یک نسخه کاری

```
static void getLogs()
{
    using (SvnClient client = new SvnClient())
    {
        Collection<SvnLogEventArgs> logItems;
        client.GetLog(@"C:\osl\testPrj", out logItems);
        if (logItems != null)
            foreach (SvnLogEventArgs msg in logItems)
                Console.WriteLine("{0} : {1}", msg.Author, msg.LogMessage);
    }
}
```

این مثال می‌تواند ایده‌ای باشد برای نوشتن یک RSS feed generator. نویسنده، تاریخ و متن هر پیغام مشخص است. سپس با استفاده از کتابخانه جامع Argotic Syndication Framework می‌توان خروجی استاندارد را تهیه نمود. این کتابخانه از آدرس زیر قابل دریافت است:

<http://www.codeplex.com/Argotic>

اگر در مثال فوق قصد داشتیم مستقیماً به مخزن کد متصل شویم، سطر دریافت اطلاعات لاگ‌ها برای نمونه به شکل زیر تغییر می‌کرد:

```
client.GetLog(  
    new Uri("file:///C:/Repositories/TestRepo/trunk"),  
    out logItems);
```

**مثال ۳:** تابع C# جهت دریافت و نمایش آخرین شماره بازنگری مخزن کد

```
static void getInfo()  
{  
    using (SvnClient client = new SvnClient())  
    {  
        SvnInfoEventArgs info;  
        client.GetInfo(  
            SvnTarget.FromString(  
                "file:///C:/Repositories/TestRepo/trunk"),  
            out info);  
        if (info != null) Console.WriteLine("{0}",  
            info.LastChangeRevision);  
    }  
}
```

اگر مخزن کد، تحت اختیار و مدیریت شما نباشد عموماً مجوز نصب اسکریپت‌های قلاب را بدلیل امنیتی به شما نخواهند داد. کد فوق می‌تواند ایده‌ای باشد برای برنامه‌ای که در فواصل زمانی مشخص مخزن کد را بررسی کرده و آخرین شماره بازنگری انجام شده را در مقایسه با آخرین شماره گزارش شده، به شما اعلام می‌کند (برای مثال به تیم برنامه نویسی ایمیل می‌زند).

- 1- Collins-Sussman B., Fitzpatrick B., Pilato M., "Version Control with Subversion", web site, <http://svnbook.red-bean.com/>.
- 2- Smart J., "Java Power Tools", O'Reilly, 2008.
- 3- Mason M., "Pragmatic Version Control using Subversion", 2nd Edition, The Pragmatic Programmers LLC, 2006.
- 4- Nagel W., "Subversion Version Control: Using the Subversion Version Control System in Development Projects", Prentice Hall PTR, 2005.
- 5- SAM-BODDEN B., JUDD C., "Enterprise Java Development on a Budget: Leveraging Java Open Source Technologies", APress, 2004.
- 6- Berlin D., Rooney G., "Practical Subversion", APress, 2006.
- 7- Murphy D., "Managing Software Development with Trac and Subversion", PACKT Publishing, 2007.
- 8- HOLMES M., "Expert .NET Delivery Using NAnt and CruiseControl.NET", APress, 2005.