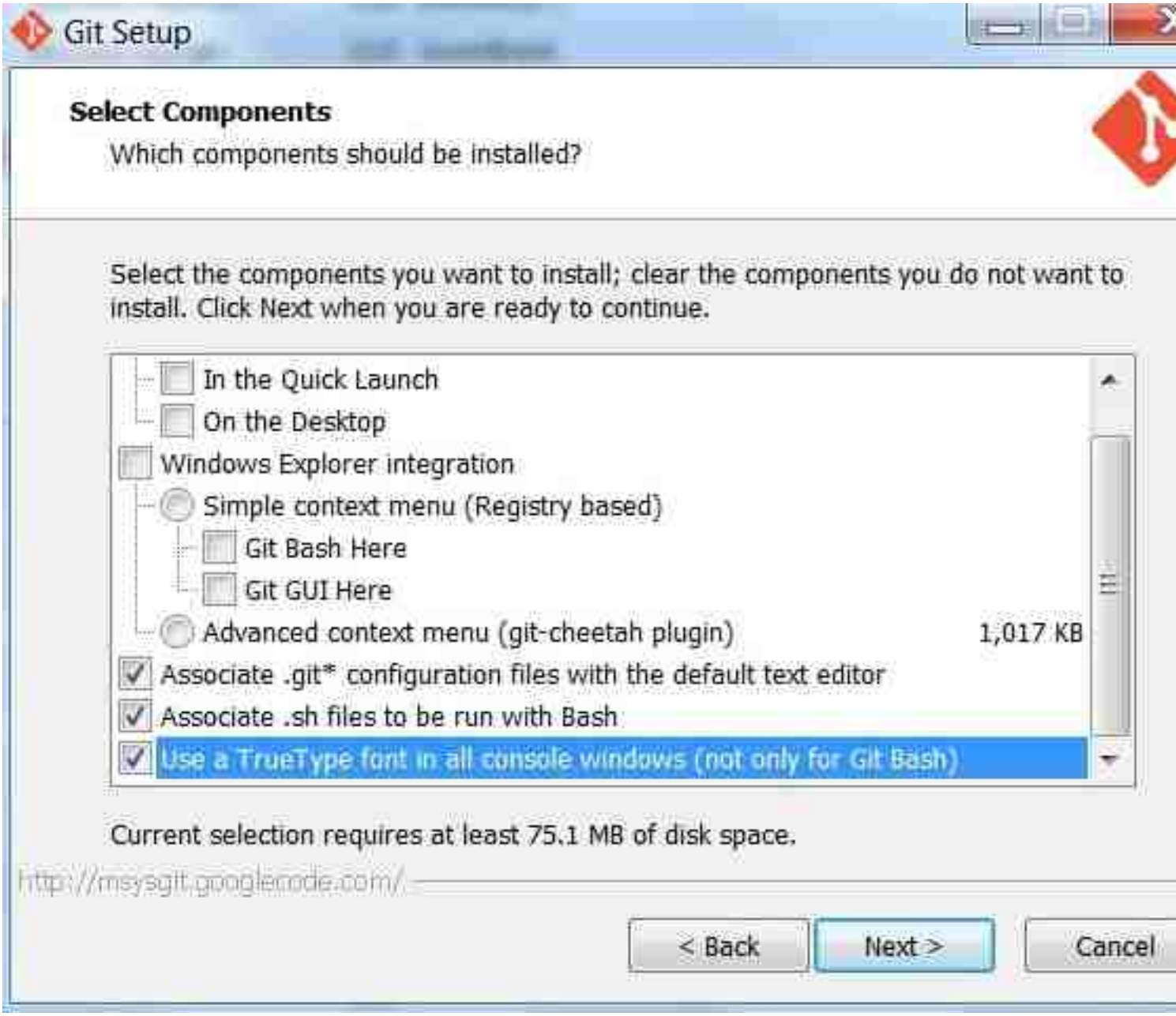
عنوان: **#4 آموزش سیستم مدیریت کد Git : نصب و پیکربندی**
نویسنده: حسام امامی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۸:۳۷
آدرس: www.dotnettips.info
گروه‌ها: Git

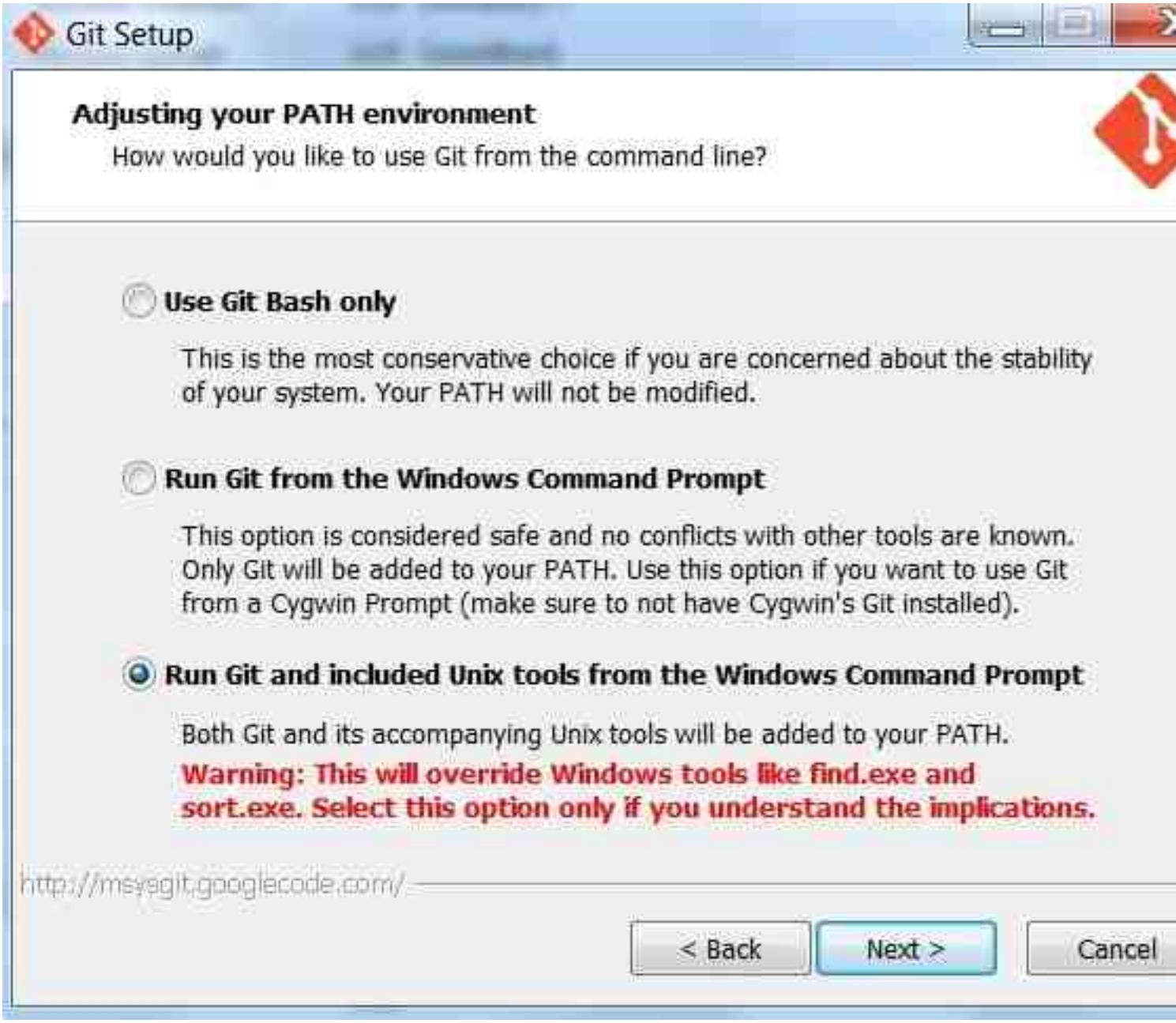
برای نصب Git ابتدا به msysgit رفته و مطابق شکل زیر بر روی گزینه دانلود کلیک کنید. سپس در صفحه باز شده آخرین نسخه Git را دانلود نموده و فایل مربوطه را اجرا کنید:

شروع نصب:

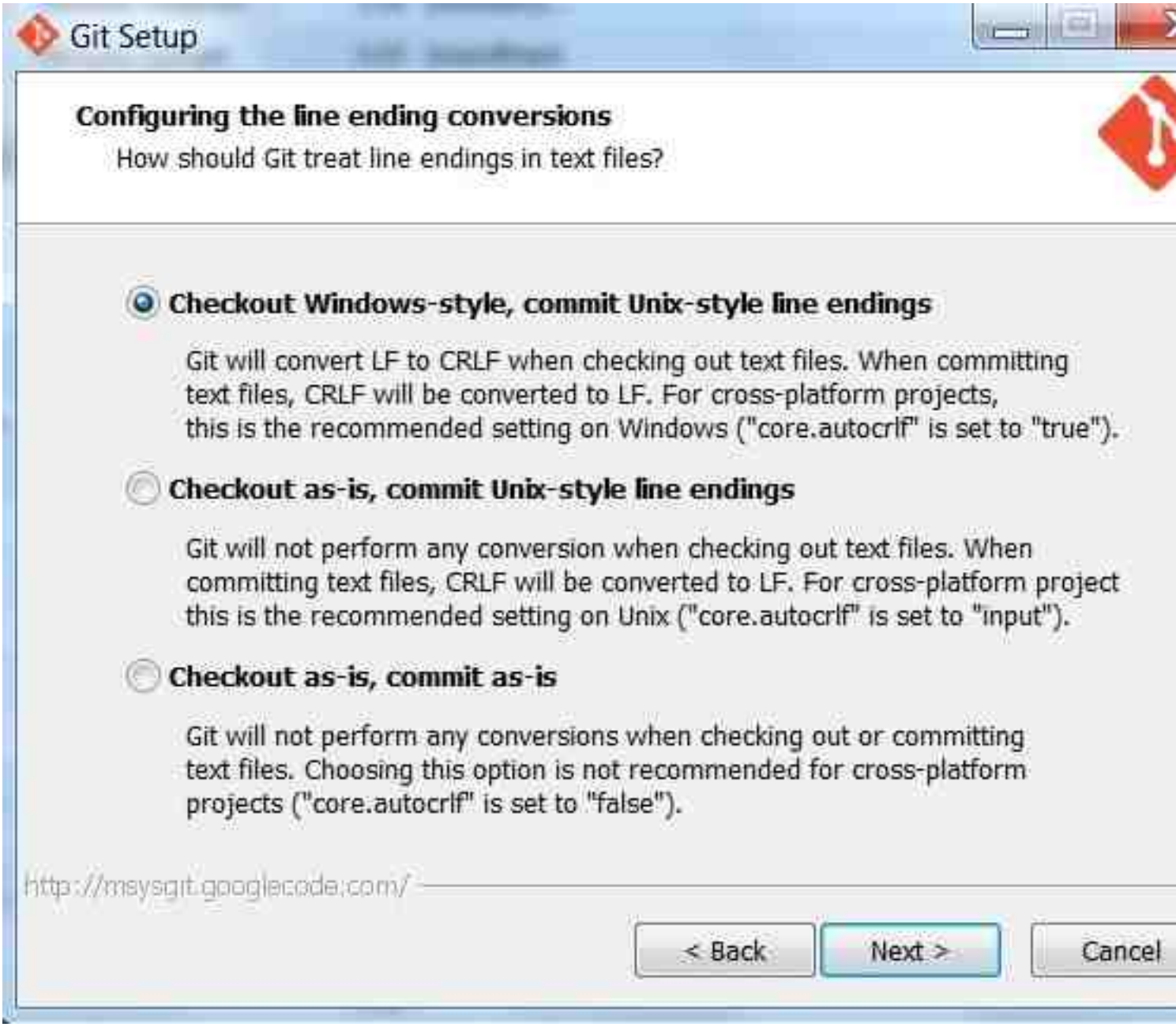
در این مرحله بخش Windows Explorer Integration اهمیت دارد. در صورت انتخاب این بخش، بعد از نصب، Git Bash و Git GUI به منوی راست کلیک شما اضافه می‌شود. به این ترتیب با سرعت بیشتری می‌توانید به Git در یک پوشه خاص دسترسی داشته باشید.



در این مرحله از شما خواسته می‌شود تعیین کنید که آیا فقط می‌خواهید از طریق Git Bash با Git کار کنید یا با اضافه کردن فایل اجرایی Git به متغیرهای محلی ویندوز از طریق Command Prompt ویندوز نیز می‌خواهید به Git دسترسی داشته باشید. گزینه سوم هم Git و هم برخی از ابزارهای یونیکسی را به متغیرهای محلی اضافه می‌کند که سبب می‌شود شما یک خط فرمان قدرتمندتر در ویندوز داشته باشید. اما این کار ممکن است در برخی از برنامه‌های پیش فرض اختلال ایجاد کند بنابراین در انتخاب این گزینه احتیاط کنید.



در این مرحله کاراکتری را که نشان دهنده انتهای خط است تعیین می‌کنید. این کاراکتر در ویندوز و یونیکس متفاوت است. بنابراین Git از شما می‌خواهد که برای حفظ سازگاری در محیط هایی که چند سیستمی هستند، آن‌را تعیین کنید.
گزینه اول به صورت فرمت یونیکس ذخیره و به شکل ویندوز بازیابی می‌شود (مناسب برای محیط ویندوز).
گزینه دوم ذخیره به فرمت یونیکسی است و مناسب محیط‌های یونیکس است.
و آخرین گزینه فایل را بدون تغییر ذخیره و بازیابی می‌کند (از این گزینه نیز می‌توان هم برای Unix و هم windows استفاده کرد).
بعد از این مرحله نصب آغاز می‌شود.



نکته: شما می‌توانید جهت دسترسی به یک محیط گرافیکی قوی از gitextensions استفاده کنید. با دانلود این فایل، هم خود Git و هم GUI هایی برای کارهای مختلف، نظیر مشاهده تفاوت‌های دو فایل یا نمایش گرافیکی شاخه‌ها به سیستم شما اضافه می‌شود.

پیکربندی Git:

برای پیکربندی Git شما باید یک فایل config ایجاد کنید و با استفاده از دستوراتی که در ادامه می‌آید این تنظیمات را انجام دهید. البته پیکربندی Git از طریق ابزارهای گرافیکی که در محله قبل نصب کردید نیز امکان‌پذیر است.

Git دارای سه نوع دسترسی برای پیکره‌بندی است:

سیستمی: این تنظیمات بر روی کل سیستمی که git برای روی آن نصب شده اعمال می‌شود. فایل gitconfig در مسیر program files/Git/etc/gitconfig قرار دارد و برای تغییر آن باید از سوئیچ system-- استفاده نمود.

در سطح کاربر: فایل config.در مسیر [username]/users/ برای این منظور است و تغییر این تنظیمات تنها بر روی همین کاربر اعمال می‌شود برای درسترسی به این فایل باید از سوئیچ global-- استفاده کرد.

در سطح Repository: برای هر پوشه repository این فایل موجود است و اگر از دستور config بدون هیچ سوئیچی استفاده

کنیم تغییرات بر روی این فایل اعمال می‌شود.

نکته: معمولا فایل پیکربندی git در سطح سیستم را تغییر نمی‌دهند.

دستورات پیکربندی:
همان‌طور که گفته شد هر Commit حاوی اطلاعات فردی است که آن‌را انجام داده است. این اطلاعات را می‌توان به صورت زیر تنظیم کرد:
نام کاربر:

```
git config --global user.name "Hessam"
```

ایمیل کاربر:

```
git config --global user.email "hessam@localhost.com"
```

با استفاده از دستور زیر می‌توان تنظیماتی را که تا کنون انجام شده ببینیم:

```
git config --global --list
```

همچنین می‌توان ویرایشگر متن پیش فرضی برای git تعیین کرد. از این ویرایشگر می‌توان به عنوان مثال بعد از فرخوانی دستور commit استفاده نمود تا دلیل commit مشخص شود. در صورت تعیین این ویرایشگر، git آن‌را خودکار باز می‌کند:

```
git config --global core.editor notepad
```

من در اینجا notepad را انتخاب کردم توجه کنید که مسیر ویرایشگر باید در متغیرهای محلی ویندوز باشد.
و در نهایت جهت نمایش بهتر پیام‌های git می‌توانیم تنظیم کنیم که آن‌ها را با رنگ‌های متفاوتی نمایش دهد:

```
git config --global color.ui auto
```

البته تنظیمات بیشتری را می‌توان در اینجا انجام داد، مانند تعیین برنامه پیش فرض برای نمایش اختلاف فایل‌ها و یا برنامه پیش فرض برای حل کردن مشکل conflict و غیره که این تنظیمات در همان بخش‌ها گفته خواهد شد.

در قسمت بعد دستورات اولیه کار با git به صورت محلی گفته خواهد شد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۹:۵

یک نکته. اگر به گوگل کد دسترسی ندارید، از این آدرس هم می‌توانید فایل‌ها را دریافت کنید.

نویسنده: پژمان
تاریخ: ۱۳۹۱/۰۶/۰۱ ۲۳:۲۲

ممنون. راهنمای نصب بسیار واضح و مفیدی بود.

نویسنده: اژدری
تاریخ: ۱۳۹۱/۰۶/۱۳ ۱۰:۴

با سلام و عرض خسته نباشید به همه‌ی دوستان و همکاران

سوالی که بنده داشتم این بود که چرا و به چه علتی با وجود ابزاری مثل visual studio team foundation server 2010 , 2012 باید با ابزاری مثل git کار کرد و البته با توجه به اینکه دوستان این سایت یا وبلاگ عموما در حوضه دات نت هستند این سوال مهم‌تر هم میشه ، در مورد مطالب در خصوص git باید بگم طرز کار کردن با این ابزار بسیار پیچیده‌تر و غیر اصولی‌تر از tfs هست ، مثلا اینکه خود فایل رو پس از تغییر نگهداری میکنه یک نقطه ضعفه ولی نویسنده مطلب از اون به عنوان نقطه قوت یاد کرده ، اگر فایل به صورت مجموعه تغییرات ذخیره بشه هم حجم اطلاعات ذخیره شده کاهش پیدا میکنه و هم منبع نگهداری سورس‌ها میتونه مثل ماشین زمان ما رو به جلو و عقب ببره و محدودیتی نخواهد داشت ، در هر حال با توجه به محصول مایکروسافت بودن tfs و رایگان بودن git فکر کنم حتی مقایسه این دو حتی درست هم نباشه.

با تشکر از تمامی زحمات شما دوستان عزیز

نویسنده: حسام امامی
تاریخ: ۱۳۹۱/۰۶/۱۳ ۱۱:۳۱

اگر شما به سایت‌های مدیریت کدی نظیر github مراجعه کنید و تعداد کاربران و یا پروژه‌های قرار گرفته بر روی آن‌ها را در نظر بگیرید متوجه محبوبیت سیستم مدیریت کد git خواهید شد در مورد تفاوت‌های سیستم‌های CVS و DVCS در مقاله اول توضیحاتی داده شد و در مقاله بعد درباره نحوه ذخیره سازی اطلاعات که باعث افزایش سرعت چشمگیر در عملیات check-in و check-out می‌شود
در ضمن در git و در همه سیستم‌های مدیریت کد امکان دستیابی به کدهای قبل وجود دارد و به طور کلی این یکی از اهداف سیستم‌های مدریت کد است.
خود من هم یک برنامه نویس دات نت هستم اما دلیلی ندارد که مجبور باشیم هر آنچه که مایکروسافت ساخته را استفاده کنیم
من با هر دو سیستم TFS و Git کار کردم و به شخصه استفاده و راه اندازی آن را از TFS ساده‌تر می‌بینم چون تنها یکی از کاربردهای TFS مدیریت کد است بنابراین شما به طور نسبی با سیستم پیچیده‌تری سرو کار خواهید داشت.
اما در نهایت نیاز شما به معماری مورد استفاده در مدیریت کدهای خود تعیین کننده است
اگر یک سیستم مدیریت کد توزیع شده لازم دارید بهترین انتخاب git است
موفق باشید

نویسنده: امید
تاریخ: ۱۳۹۱/۱۲/۰۹ ۱۰:۵۵

سلام
من اولین بار هست که میخوام از کنترل ورژن‌ها استفاده کنم

اگه gitextensions رو نصب کنم نیازی به نصب msysgit نیست؟
آیا استفاده از gitextensions برای اولین تجربه و شروع کار با git و کلا کنترل ورژن انتخاب درستی هست؟ یا بهتره از msysgit استفاده کرد؟

نویسنده: ندا صابری
تاریخ: ۱۳۹۲/۱۲/۲۶ ۱۴:۲۸

سلام ممنون از مطلب خوبتون.
من تازه VS2013 نصب کردم و گزینه هایی برای کار با Git دیدم که اینجا در موردش توضیح داده شده. میخواستم بدونم VS2013 خودش Git داره؟ لازم نیست دیگه msysgit رو نصب کنم؟

نویسنده: سعید قره داغی
تاریخ: ۱۳۹۳/۰۵/۰۱ ۱۶:۳۰

با سلام و عرض ادب اگه سرور مون لینوکس باشه ولی یوزرها ویندوزی باشن دیگه احتیاجی به CopSSH نیست؟
اصلا این CopSSH برای چی استفاده می‌کنن؟
بر اساس این لینک
http://git-scm.com/book/en/Git-on-the-Server-The-Protocols

خودش گفته که از Http پشتیبانی میکنن پس جه دلیلی به SSH هست تو ویندوز معادل COPSSH ریگان نرم افزاری وجود نداره؟
یه سوال دیگه اگه کلاینت‌ها ویندوزی باشن شما صلاح میدونین سرور گیت ، لینوکس باشه یا ویندوزی و کدوم راحت‌تر و بهتره ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۱ ۱۷:۲۱

- OpenSSH کار مدیریت و اجرای دستورات کاربران راه دور سرور Git را انجام می‌دهد.
- در لینوکس OpenSSH هست. کار CopSSH (که دیگر رایگان نیست) ساده سازی نصب OpenSSH بر روی ویندوز است. البته OpenSSH را در ویندوز بدون نیاز به این ابزارهای جانبی، توسط cygwin می‌شود نصب کرد (اصل کار و درستش به این صورت است). شبیه CopSSH، مثلا sshwindows هم هست ولی بهتره وقت بگذارید روی cygwin.
- اگر ویندوزی می‌خواهید کار کنید و سرور Git راه اندازی کنید، از Bonobo Git Server استفاده کنید. راهنمای نصب
- همچنین Bitvise SSH Server هم برای ویندوز تهیه شده و از آن هم می‌شود جهت نصب سرور Git استفاده کرد.
- لیست کاملتر نصاب‌های سرور Git روی ویندوز