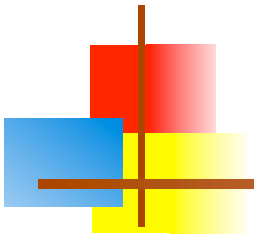
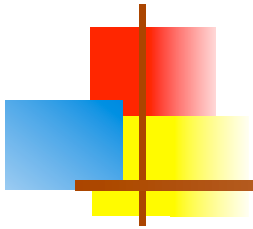


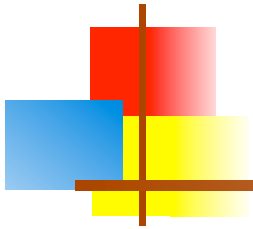
Git

A distributed version control system



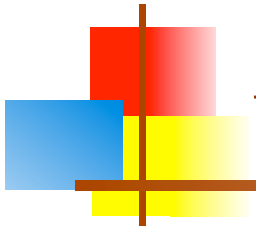
Version Control Systems

- Version control (or revision control, or source control) is all about managing multiple versions of documents, programs, web sites, etc.
  - Almost all “real” projects use some kind of version control
  - Essential for team projects, but also very useful for individual projects
- Some well-known version control systems are CVS, Subversion, Mercurial, and Git
  - CVS and Subversion use a “central” repository; users “check out” files, work on them, and “check them in”
  - Mercurial and Git treat all repositories as equal
- Distributed systems like Mercurial and Git are newer and are gradually replacing centralized systems like CVS and Subversion



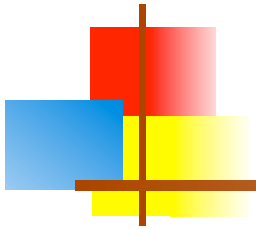
Why Version Control?

- For working by yourself:
  - Gives you a “time machine” for going back to earlier versions
  - Gives you great support for different versions (standalone, web app, etc.) of the same basic project
- For working with others:
  - Greatly simplifies concurrent work, merging changes
- For getting an internship or job:
  - Any company with a clue uses some kind of version control
  - Companies without a clue are bad places to work



Why Git?

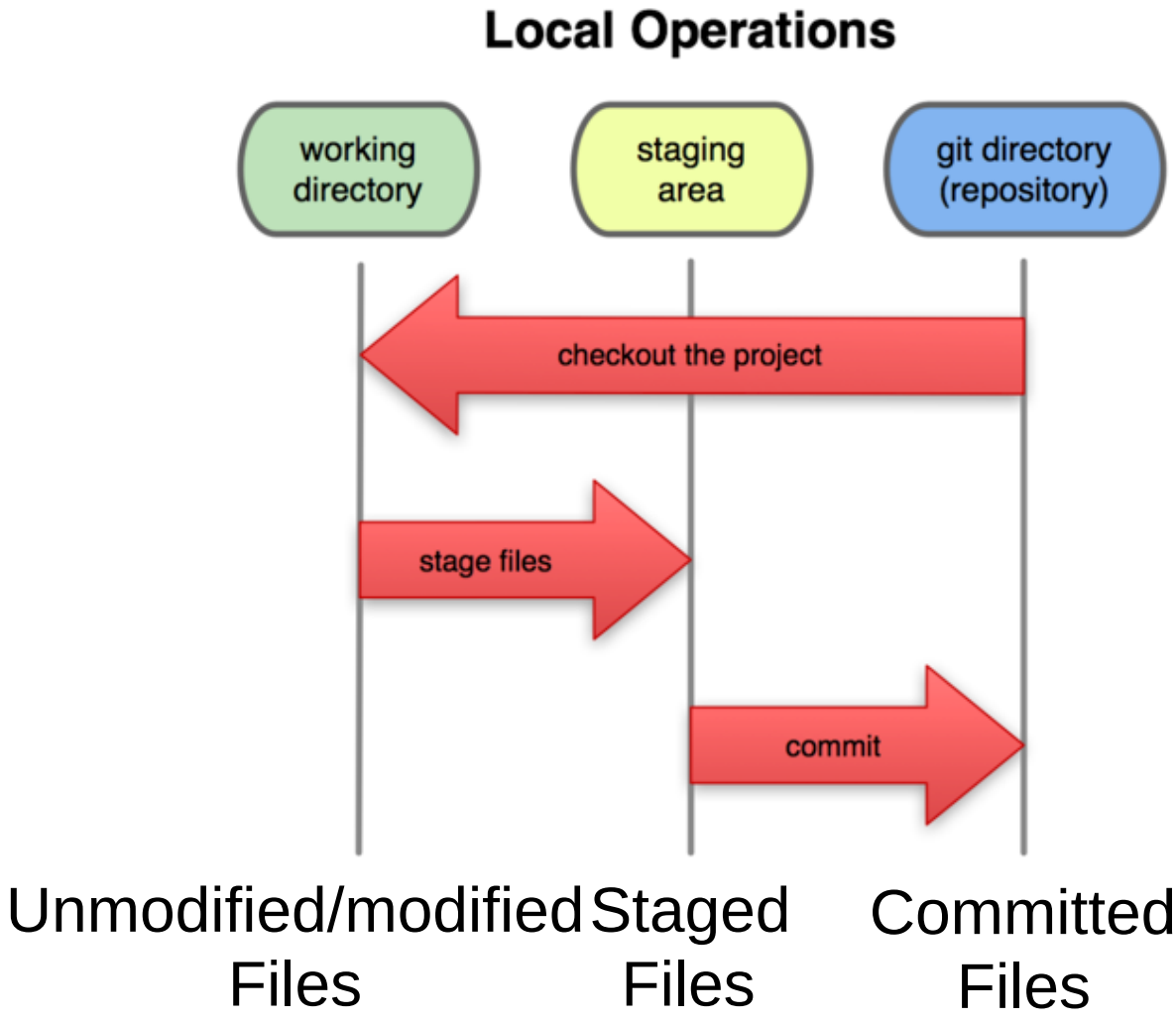
- Git has many advantages over earlier systems such as CVS and Subversion
  - More efficient, better workflow, etc.
  - See the literature for an extensive list of reasons
  - Of course, there are always those who disagree
- Best competitor: Mercurial
  - Same concepts, slightly simpler to use
  - Much less popular than Git



Git History

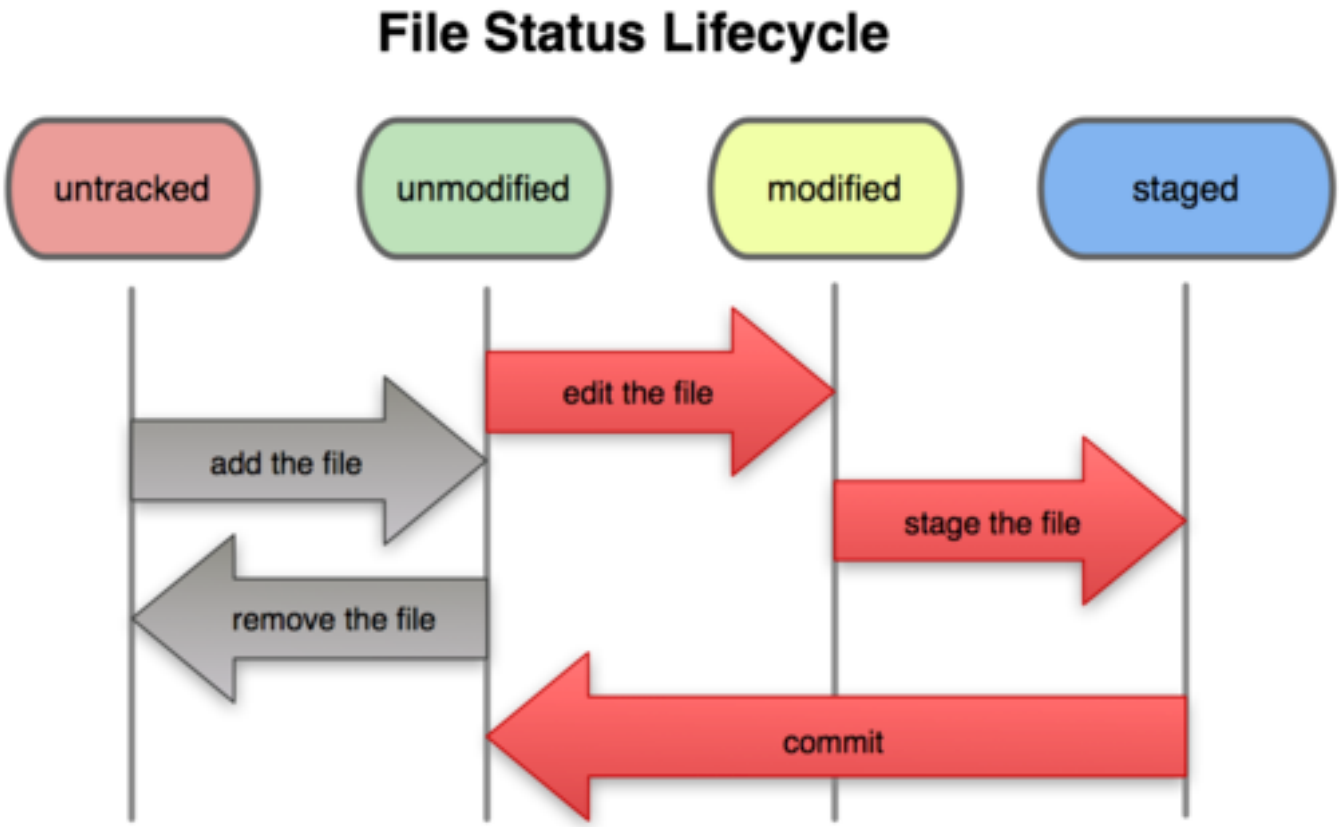
- Came out of Linux development community
- Linus Torvalds, 2005
- Initial goals:
  - Speed
  - Support for non-linear development (thousands of parallel branches)
  - Fully distributed
  - Able to handle large projects like Linux efficiently

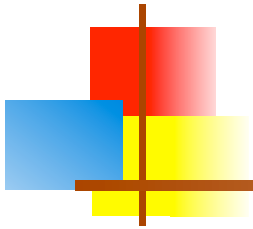
A Local Git Project Has Three Areas



Note: working directory sometimes called the “working tree”, staging area sometimes called the “index”.

Git File Lifecycle





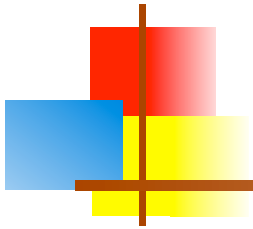
Basic Workflow

Basic Git workflow:

1. **Modify** files in your working directory.
2. **Stage** files, adding snapshots of them to your staging area.
3. Do a **commit**, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.

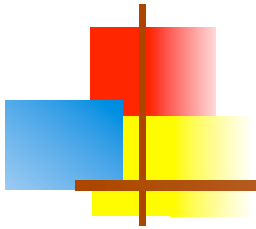
Notes:

- If a particular version of a file is in the **Git directory**, it's considered **committed**.
- If it's modified but has been added to the **staging area**, it is **staged**.
- If it was **changed** since it was checked out but has not been staged, it is **modified**.



Git Commands

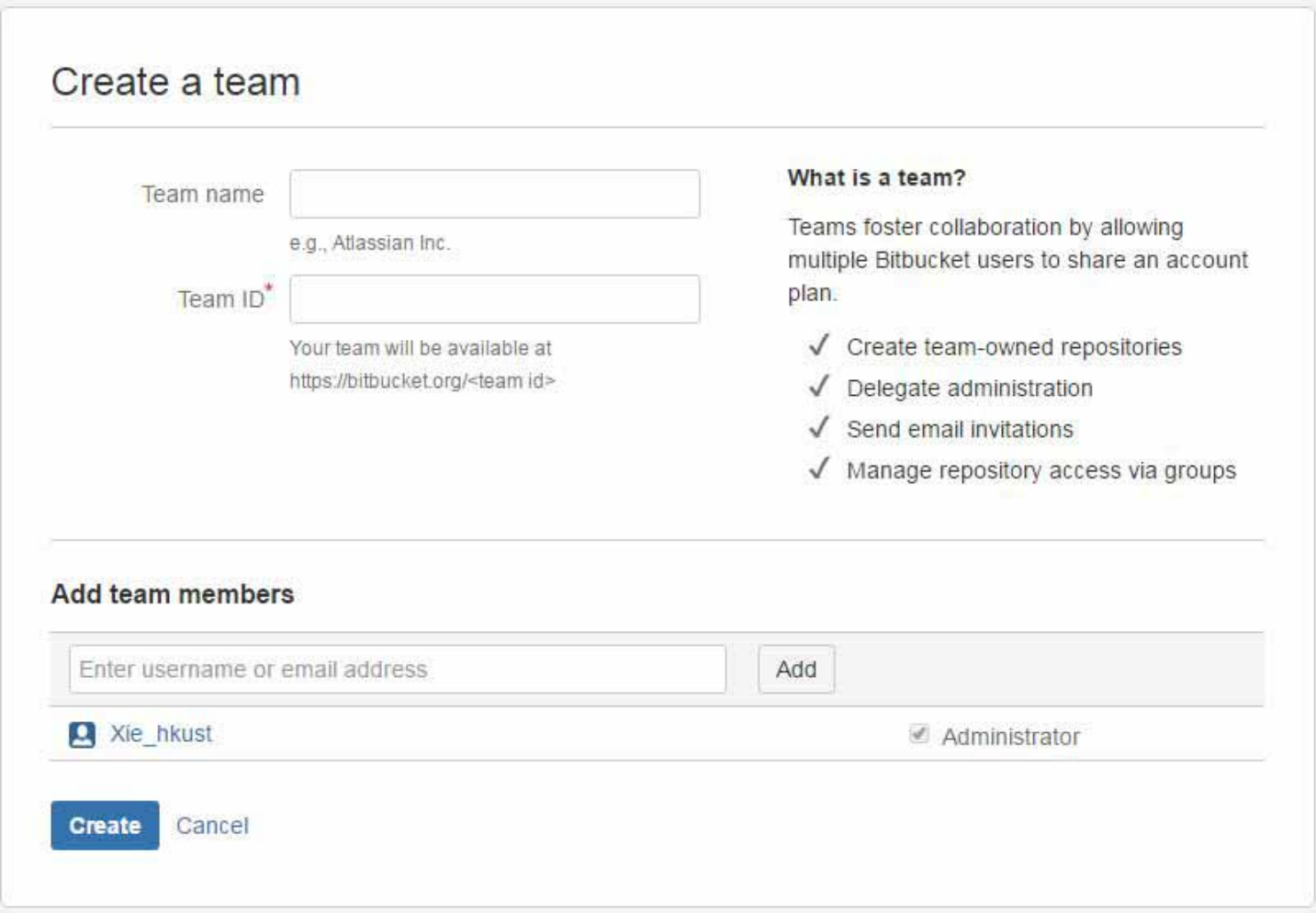
command	description
`git clone` ***`url [dir]`***	copy a git repository so you can add to it
`git add` ***`files`***	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help` ***`[command]`***	get help info about a particular command
`git pull`	fetch from a remote repository and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init, reset, branch, checkout, merge, log, tag`	



Online Tools

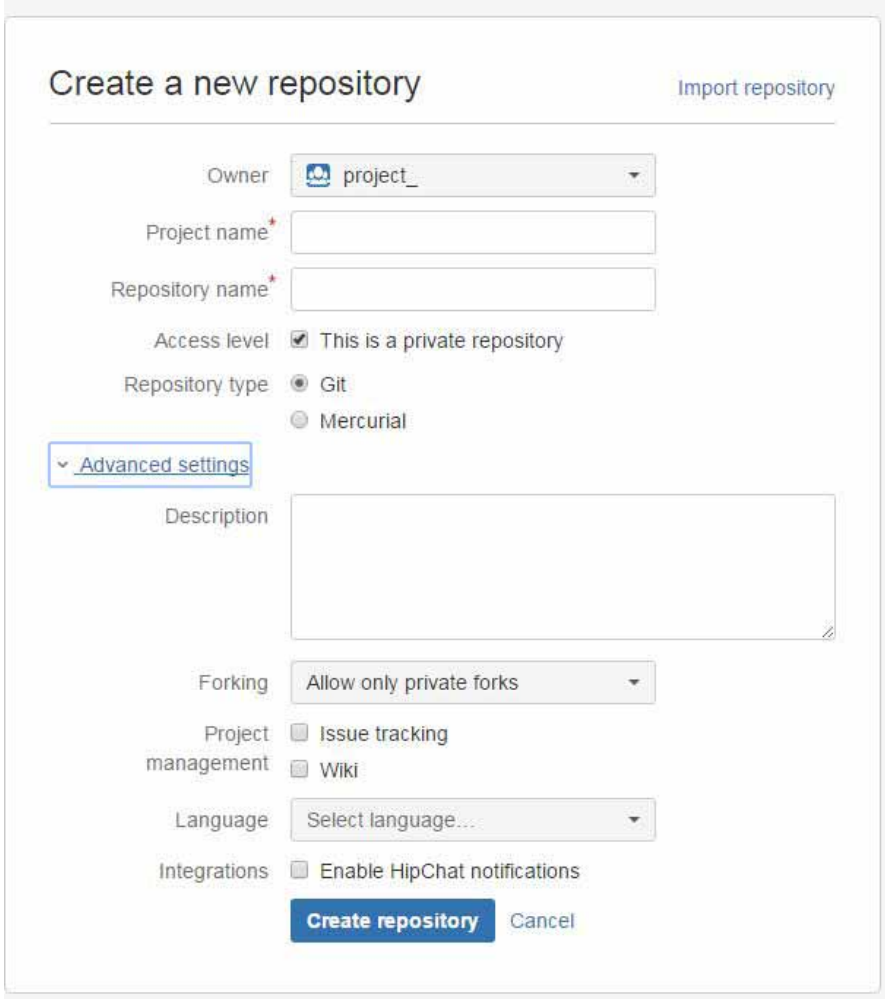
- Github
  - Public repository
  - Student account can have private account
- Bitbucket
  - Private repository
  - Only 5 team members
  - Student account can have more collaborators

Visual Studio with Bitbucket



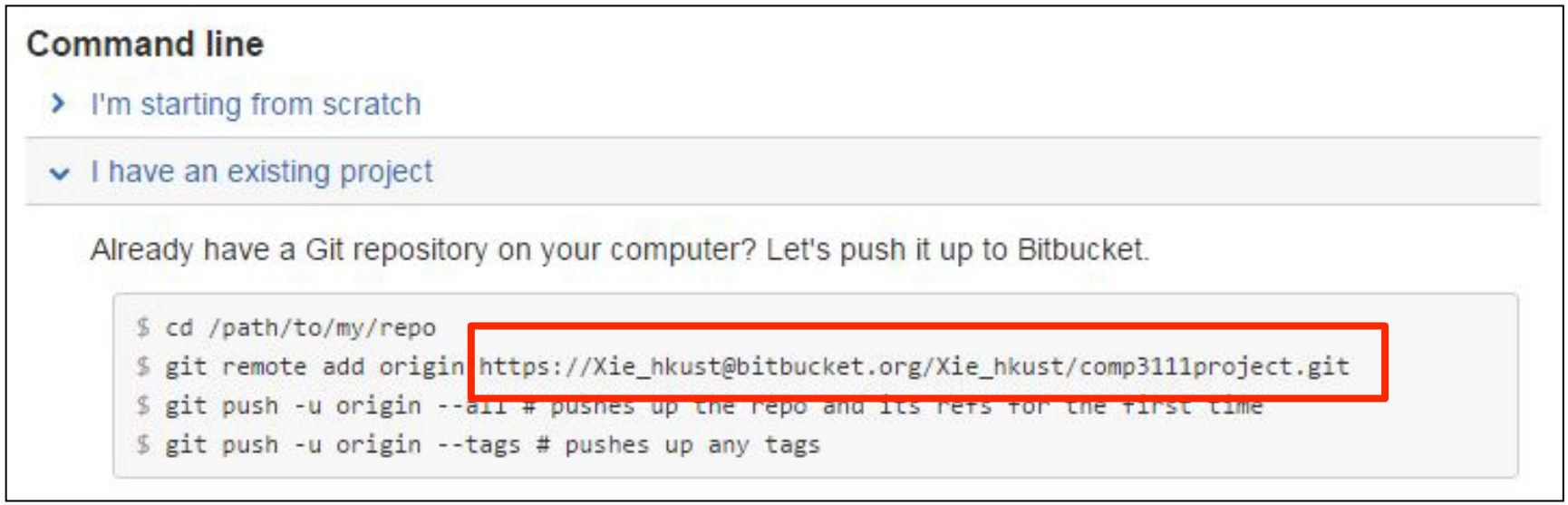
Create a team in Bitbucket

Visual Studio with Bitbucket



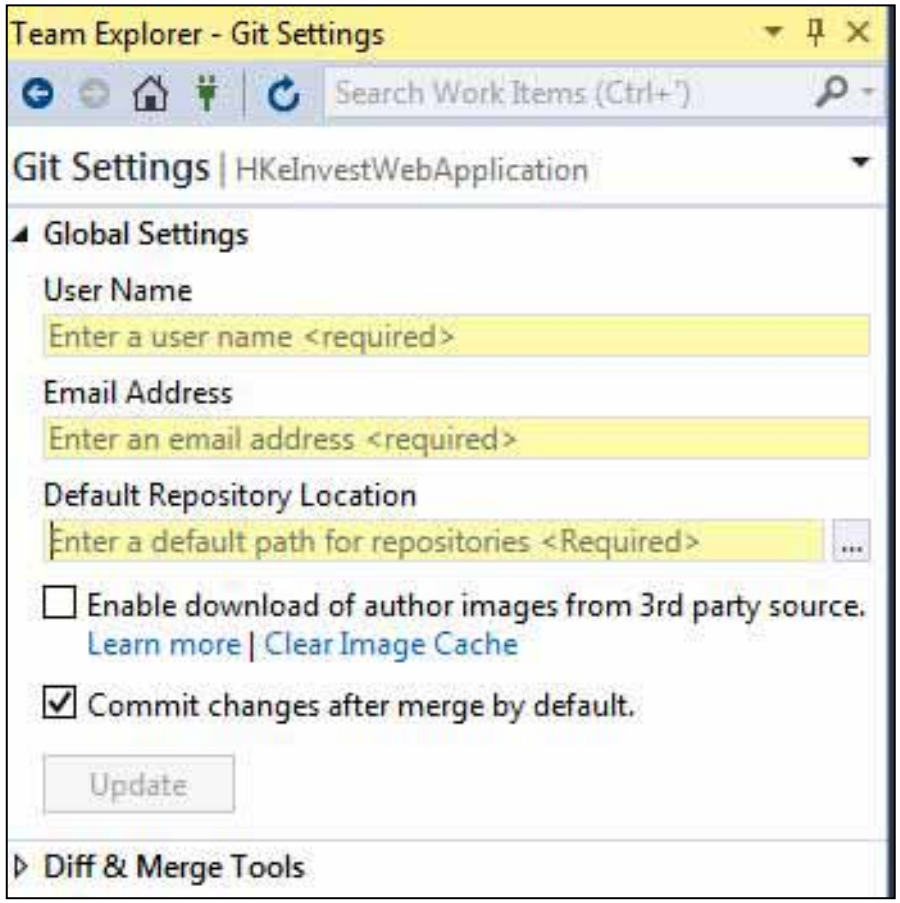
Create a repository in Bitbucket

Visual Studio with Bitbucket



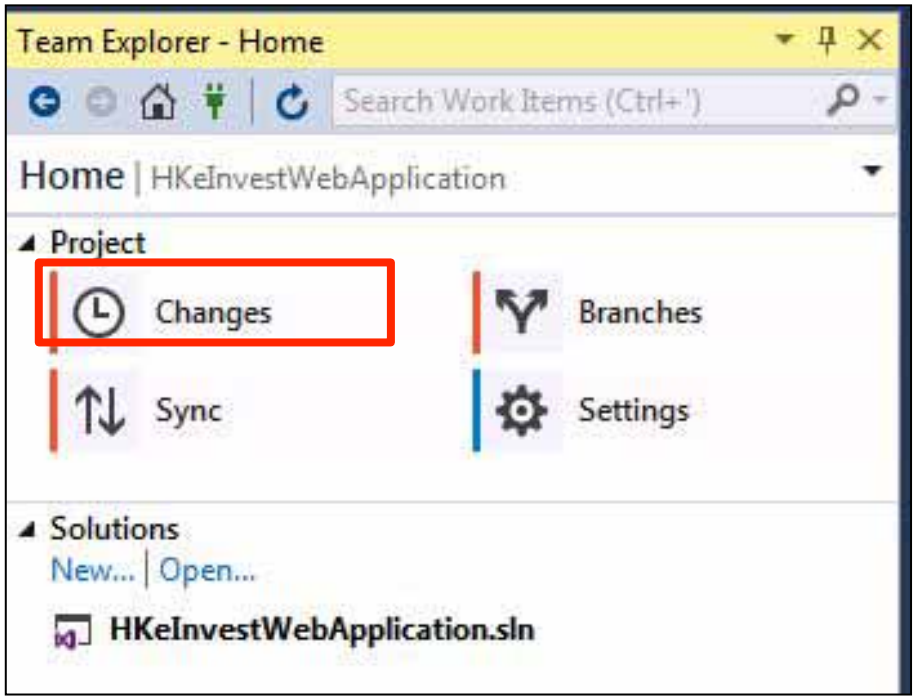
Get the repository address

Visual Studio with Bitbucket



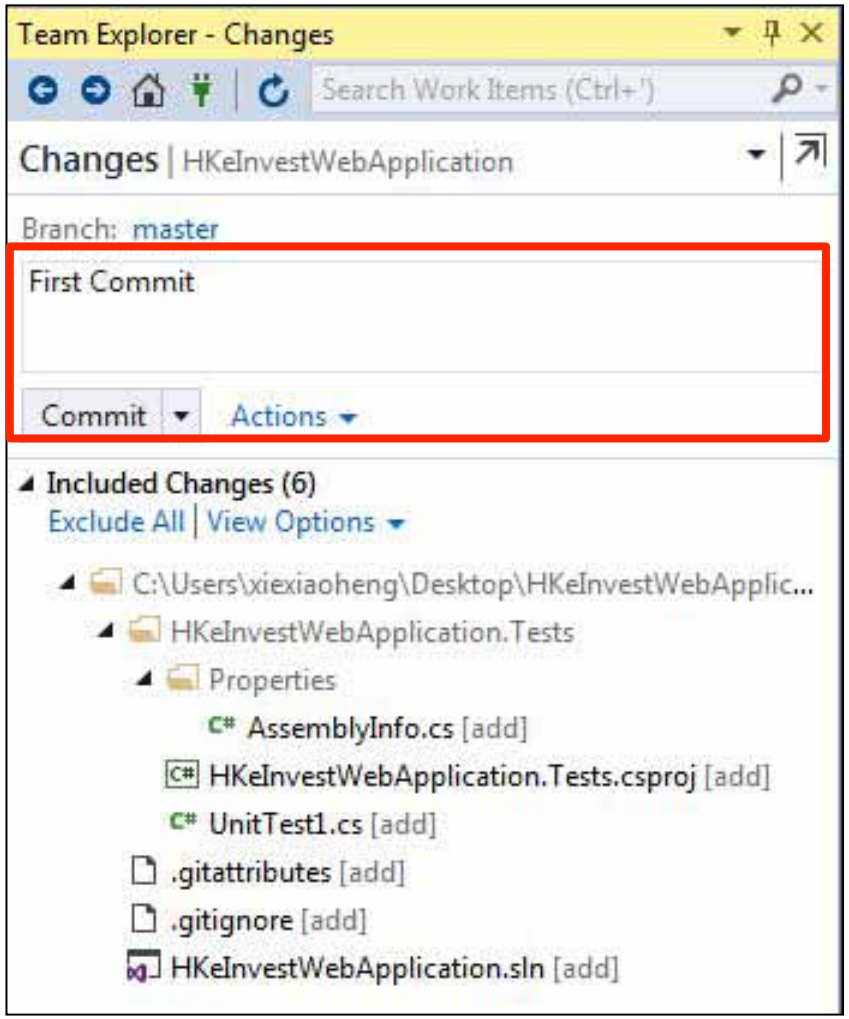
Introduce yourself to Git in Visual studio

Visual Studio with Bitbucket



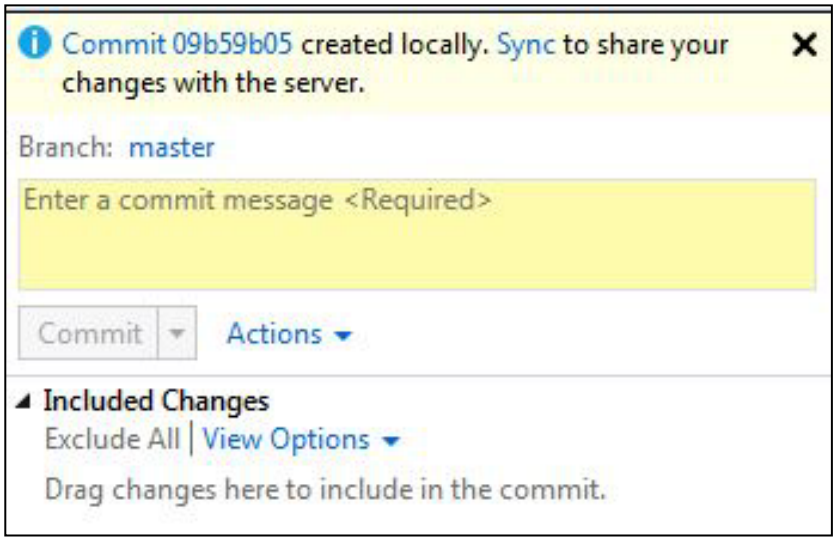
For a solution without source control, right-click solution and select Add solution to source control, and select Git.
In the Team Explorer window, choose **Changes** to add first commit.

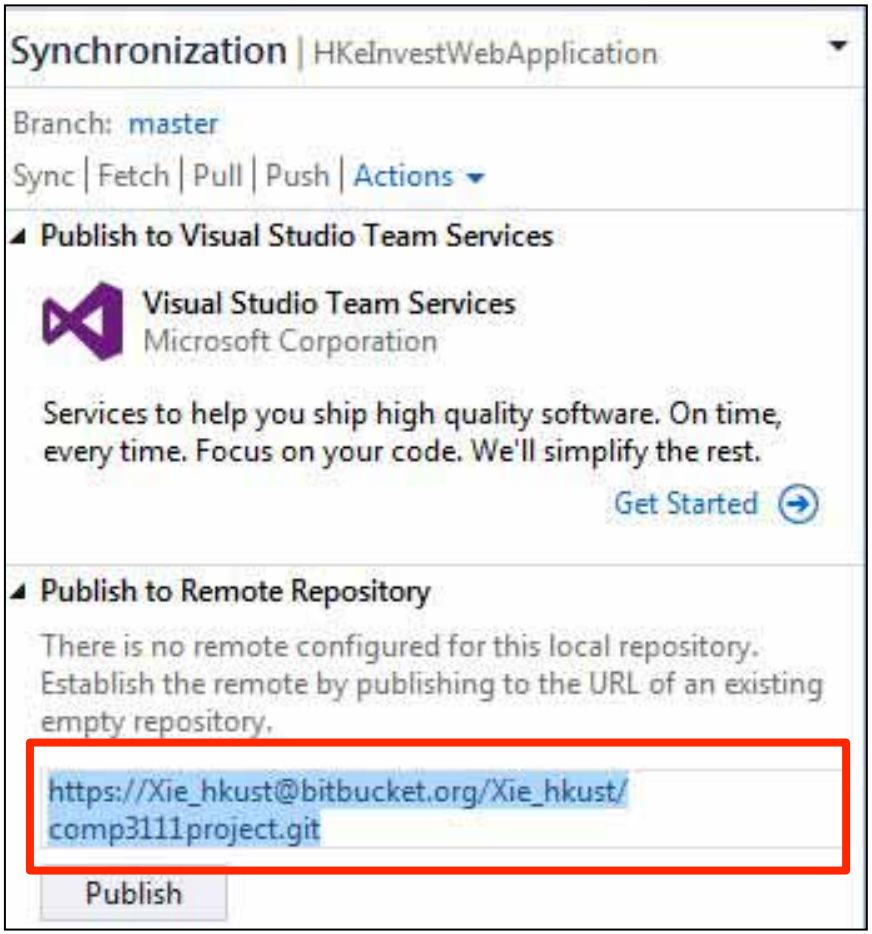
Visual Studio with Bitbucket



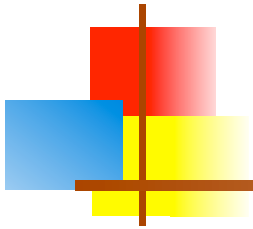
Enter the initial commit message and select **Commit**.

Visual Studio with Bitbucket





Choose **Sync** and on the synchronization window enter the Bitbucket repository URL, click **Publish**



Git Tutorials/Resources

- Using Bitbucket in Visual Studio 2013 for Beginners
  https://www.youtube.com/watch?v=WWh1VrgbQC0
- Getting started with GIT, Visual Studio, and BitBucket
  http://www.codeproject.com/Tips/900204/Getting-started-with-GIT-Visual-Studio-and-BitBuck
- Free Git on Bitbucket with Visual Studio
  http://mjsmithdev.com/2015/10/05/free-git-on-bitbucket-with-visual-studio/