Git 101

加国华人微软技术俱乐部

2g
2022-08

AGENDA

- history of version(or source) control
- what is git
- git repo providers
- start with GUI
- local vs remote
- git workflow
- some best practices

History of version(or source) control

- In first generation tools, concurrent development was handled solely with locks. Only one person could be working on a file at a time.
- The second generation tools are a fair bit more permissive about simultaneous modifications, with one notable restriction. Users must merge the current revisions into their work before they are allowed to commit.
- The third generation tools allow merge and commit to be separated.

History of version(or source) control

Generation	Networking	Operations	Concurrency	Examples
First	None	One file at a time	Locks	RCS, SCCS
Second	Centralized	Multi-file	Merge before commit	CVS, SourceSafe, Subversion, Team Foundation Server
Third	Distributed	Changesets	Commit before merge	Bazaar, Git, Mercurial

what is git

- created by Linus Torvalds in 2005
- distributed version control systems
- on every computer is a full-fledged repository with complete history and full version tracking abilities

git repo providers

- git != GitHub

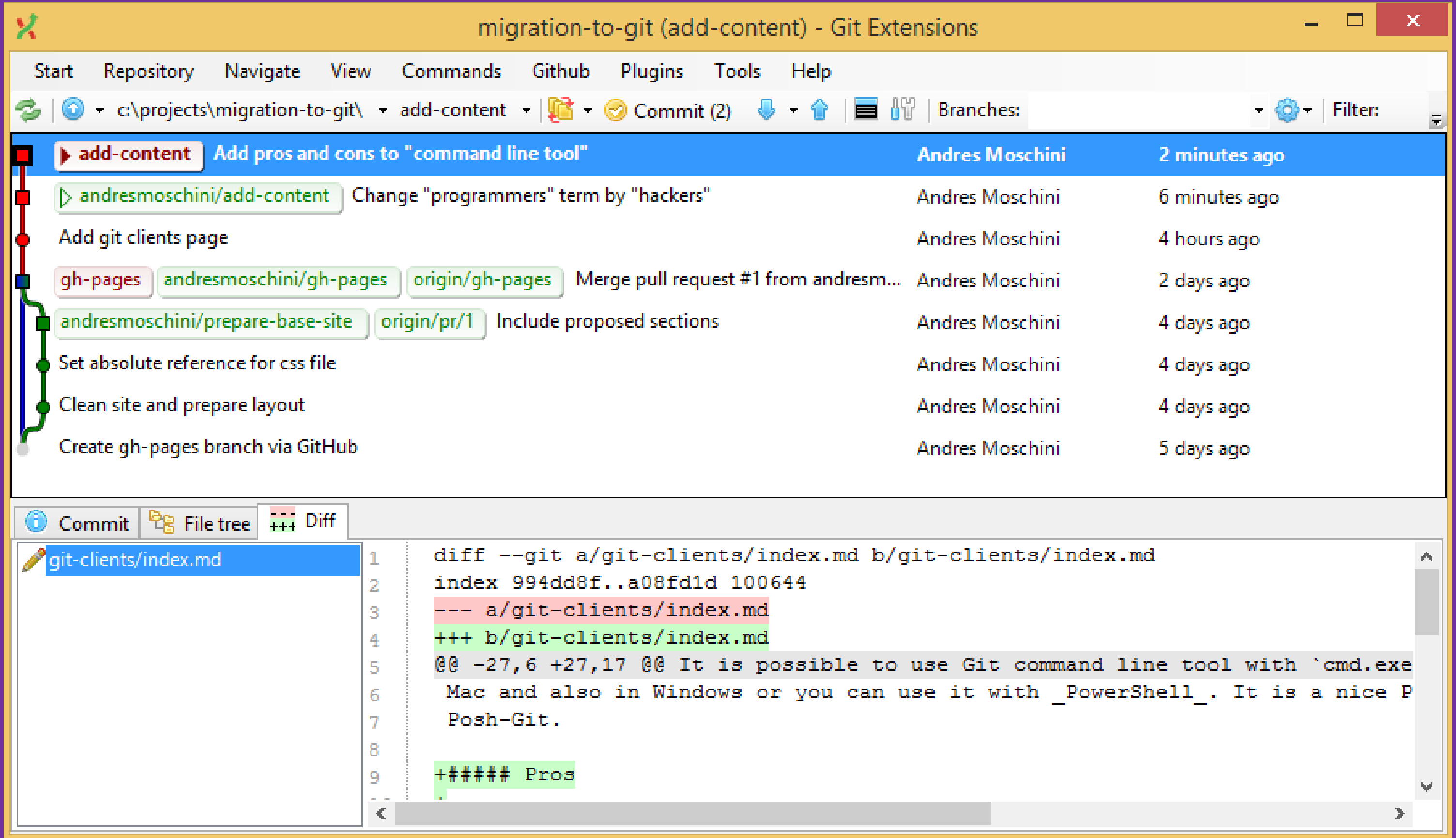
Provider	Framework is open-source?	Support for other SCM	Open-source repositories	Space (GB)	Free private repositories
GitHub	No	SVN	Yes	Unlimited	Yes Free
GitLab.com	Yes	No	Yes	Unlimited	Unlimited projects, unlimited collaborators
bitbucket.org	No	Mercurial	Yes	Unlimited	Unlimited projects, 5 collaborators
Codebase	No	Mercurial/SVN	Public access available	0.05	1 project (unlimited repos), 2 collaborators
SourceForge.net	Yes	Hg, SVN	Yes	Unlimited	No
Azure DevOps	No	TFVC	No	Unlimited	Unlimited, 5 users

start with GUI

- Git Extensions
- GitHub Desktop
- Visual Studio 2015/2017/2019/2022

Git Extensions

- https://github.com/gitextensions/gitextensions

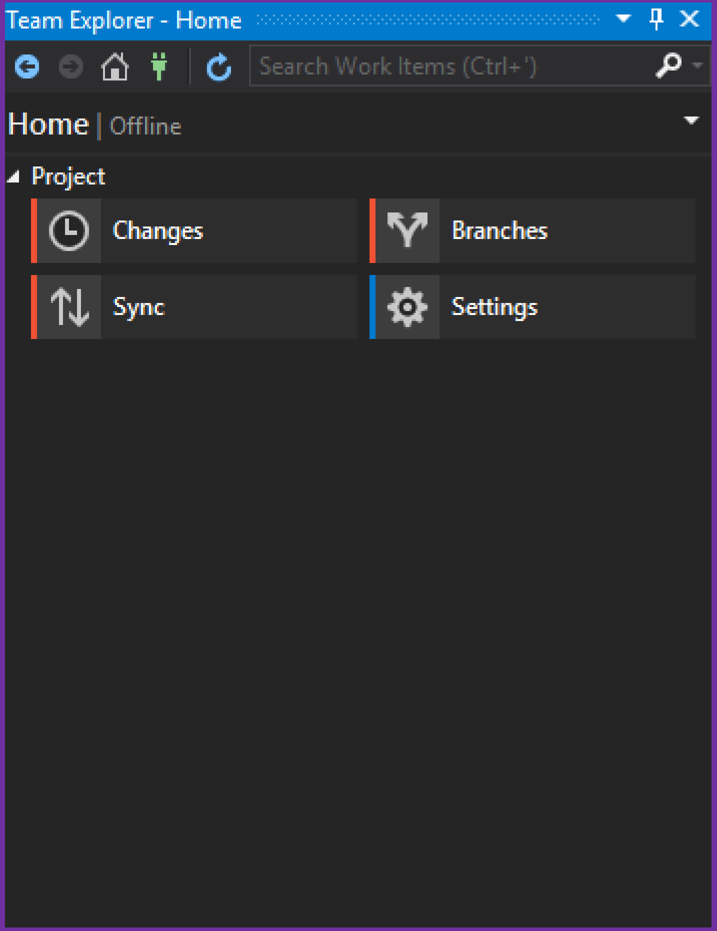


add-content	Add pros and cons to "command line tool"	Andres Moschini	2 minutes ago
andresmoschini/add-content	Change "programmers" term by "hackers"	Andres Moschini	6 minutes ago
	Add git clients page	Andres Moschini	4 hours ago
gh-pages andresmoschini/gh-pages origin/gh-pages	Merge pull request #1 from andresm...	Andres Moschini	2 days ago
andresmoschini/prepare-base-site origin/pr/1	Include proposed sections	Andres Moschini	4 days ago
	Set absolute reference for css file	Andres Moschini	4 days ago
	Clean site and prepare layout	Andres Moschini	4 days ago
	Create gh-pages branch via GitHub	Andres Moschini	5 days ago

```
diff --git a/git-clients/index.md b/git-clients/index.md
index 994dd8f..a08fd1d 100644
--- a/git-clients/index.md
+++ b/git-clients/index.md
@@ -27,6 +27,17 @@ It is possible to use Git command line tool with `cmd.exe
 Mac and also in Windows or you can use it with _PowerShell_. It is a nice P
 Posh-Git.

+##### Pros
```

Visual Studio 2015/2017



Team Explorer - Home

Search Work Items (Ctrl+')

Home | Offline

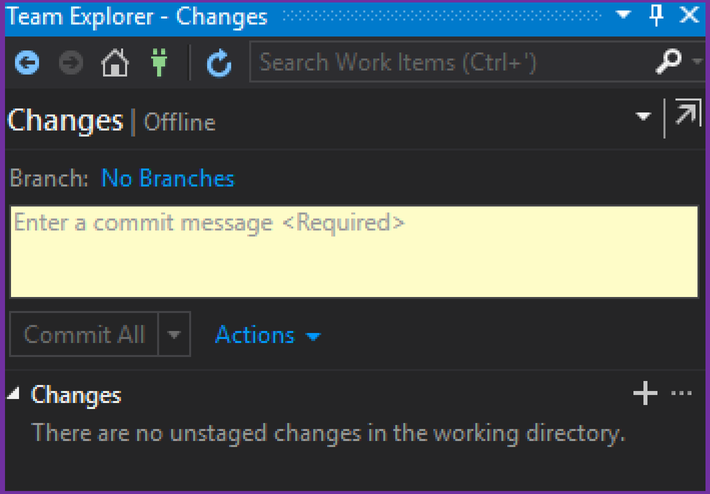
Project

Changes

Branches

Sync

Settings



Team Explorer - Changes

Search Work Items (Ctrl+')

Changes | Offline

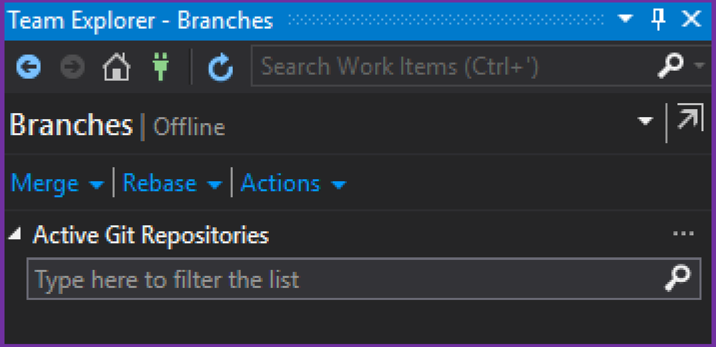
Branch: No Branches

Enter a commit message <Required>

Commit All | Actions

Changes

There are no unstaged changes in the working directory.



Team Explorer - Branches

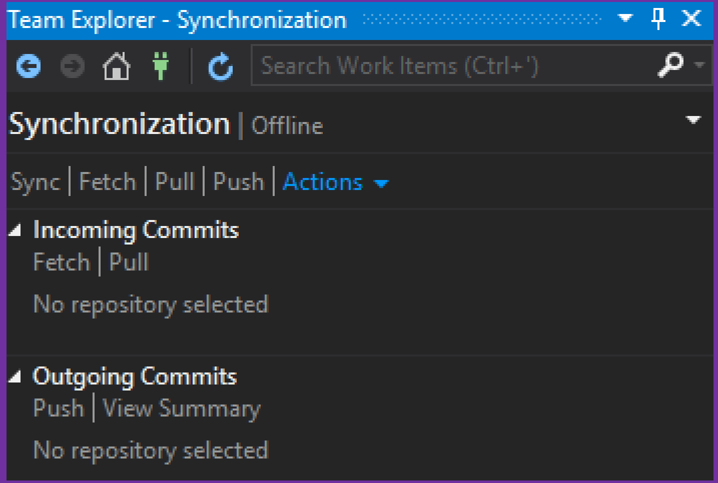
Search Work Items (Ctrl+')

Branches | Offline

Merge | Rebase | Actions

Active Git Repositories

Type here to filter the list



Team Explorer - Synchronization

Search Work Items (Ctrl+')

Synchronization | Offline

Sync | Fetch | Pull | Push | Actions

Incoming Commits

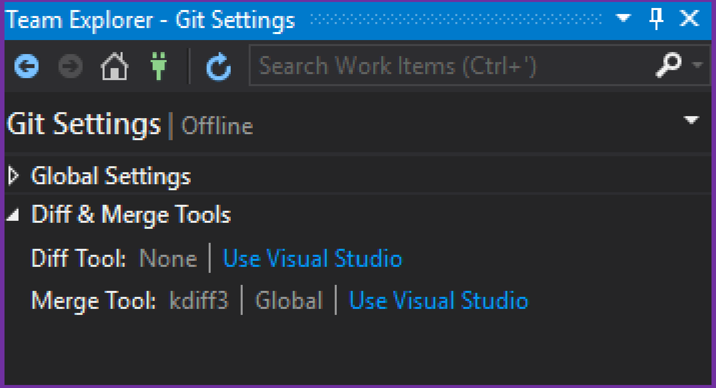
Fetch | Pull

No repository selected

Outgoing Commits

Push | View Summary

No repository selected



Team Explorer - Git Settings

Search Work Items (Ctrl+')

Git Settings | Offline

Global Settings

Diff & Merge Tools

Diff Tool: None | Use Visual Studio

Merge Tool: kdiff3 | Global | Use Visual Studio

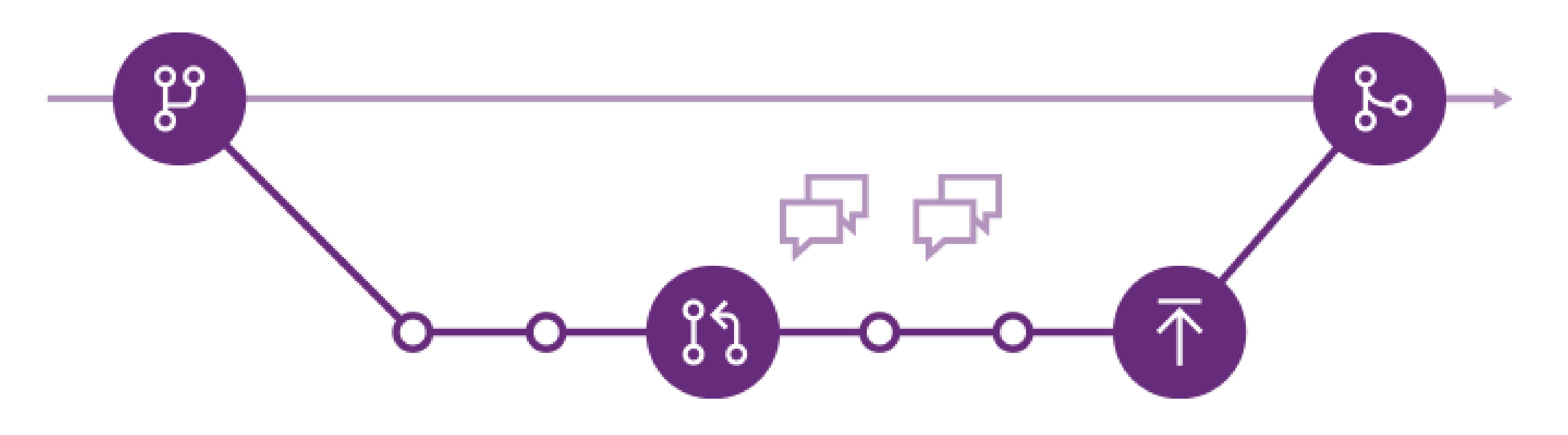
local vs remote

- create and work with local repo
  - you don’t need to install any source control server side software
- create and use remote repo
  - GitHub
  - VSTS

git workflow

- 1. Create a branch for the changes you plan to make and give it a name, such as fix-bug-3214 or cool-feature-x.
- 2. Commit changes to your branch. People often have multiple commits for a bug fix or feature.
- 3. Push your branch to the remote repository.
- 4. Create a pull request so other people can review your changes. To incorporate feedback, you might need to make more commits and push more changes.
- 5. Complete your pull request and resolve any merge conflicts from changes other people made after you created your branch.

git workflow



Some best practices

- Branch naming convention
  - feature-{alias/dev/qa}-{main}-{detail}
  - update-{alias/dev/qa}-{main}-{detail}
  - bugfix-{alias/dev/qa}-{main}-{detail}
  - hotfix-{alias/dev/qa}-{main}-{detail}
  - lab-{alias/dev/qa}-{main}-{detail}
  - wip-{alias/dev/qa}-{main}-{detail}
- Pull Request / Code review

THANK YOU!

Q&A