# Getting Started with Git

Jeremy Clark www.jeremybytes.com

# What is Git?

- Distributed Source Control
- Works Off-Line
  - Full functionality on the local machine
- Supports Remote Repositories
  - GitHub is the most popular
- Full Source Control Features
  - Change Tracking
  - Branching / Merging
  - History
  - And more...

# Centralized vs. Distributed

### Centralized Source Control

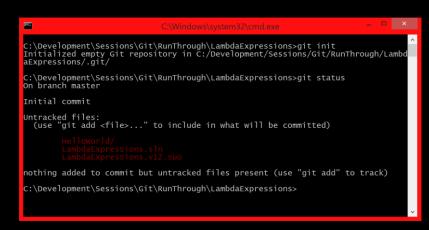
- TFS, Visual Source Safe, others
- Central server is required
- Server has the "true" copy
  - Checkout / locked files
- Server has full history
- Limited off-line features

### Distributed Source Control

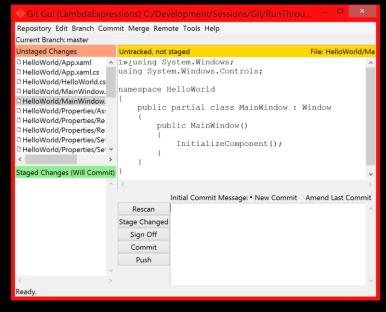
- Git, Mercurial, others (Git has won)
- Server is optional
- Each user has a "true" copy
  - No locked files
- Each user has full history
- Full features available off-line

# Interacting with Git

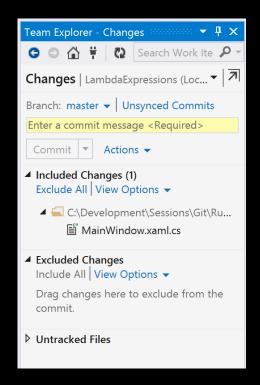
### Command Line



### Git GUI



### Visual Studio 2013

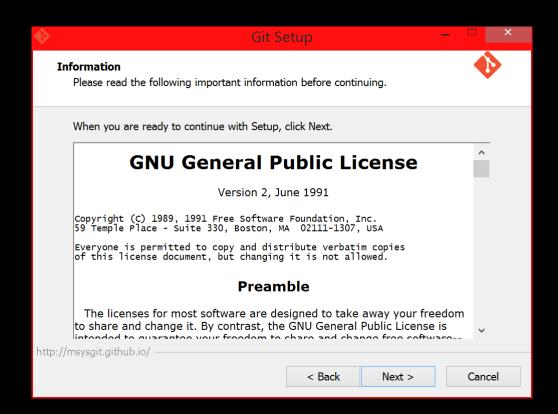


# Welcome to the Git Setup Wizard This will install Git version 1.9.5-preview20150319 on your computer. It is recommended that you close all other applications before continuing. Click Next to continue, or Cancel to exit Setup.

Next >

Cancel

# Installation



### **Select Components** Which components should be installed? Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue. Additional icons ✓ In the Quick Launch On the Desktop ✓ Windows Explorer integration Simple context menu (Registry based) ✓ Git Bash Here ✓ Git GUI Here Advanced context menu (git-cheetah plugin) 1.1 MB ✓ Associate .git\* configuration files with the default text editor ✓ Associate .sh files to be run with Bash Use a TrueType font in all console windows (not only for Git Bash) Current selection requires at least 99.9 MB of disk space. http://msysgit.github.io/ < Back Next > Cancel

# Installation

- Git Bash Here
  - Adds right-click menu to open Git command environment
  - bash is a Unix shell
  - \*I don't use this
- Git GUI Here
  - Adds right-click menu to open Git graphical environment
  - \*I don't use this

### Adjusting your PATH environment How would you like to use Git from the command line? Use Git from Git Bash only This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash. Use Git from the Windows Command Prompt This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt. Use Git and optional Unix tools from the Windows Command Prompt Both Git and the optional Unix tools will be added to your PATH. Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications. http://msysgit.github.io/ < Back Next > Cancel

# Installation

- Use Git from Windows Command Prompt
  - Add "git" to the path
  - Allows git to be used from the command prompt
  - \*I use this

### Configuring the line ending conversions How should Git treat line endings in text files? Checkout Windows-style, commit Unix-style line endings Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true"). Checkout as-is, commit Unix-style line endings Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input"). Checkout as-is, commit as-is Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false"). http://msysgit.github.io/ < Back Next > Cancel

# <u>Installation</u>

- Line Endings
  - Windows-style = CRLF
    - Carriage return & line feed
  - Unix-style = LF
    - Line feed
- This will impact code editors that don't understand Unix-style.
- This will impact "diff" tools that do file comparisons.

# Key Commands

- init
  - Create a repository
- add / rm
  - Include / exclude files from the repository (tracking)
- .gitignore
  - File types to ignore
- status
  - View unsaved changes
- commit
  - Save changes

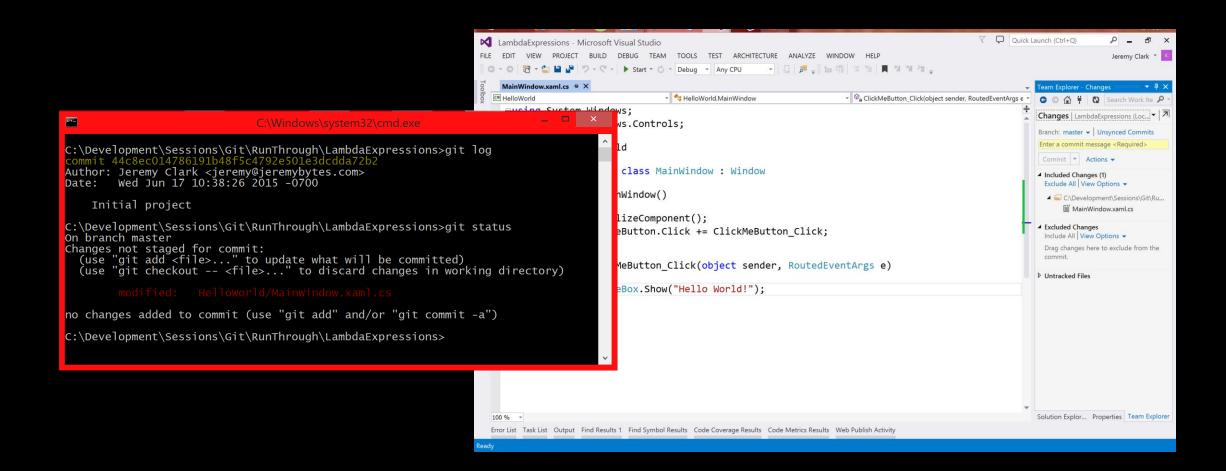
# Branching

- branch
  - Create a new branch
- checkout
  - Change to a different branch
- merge
  - Update branch with changes from another branch
- stash
  - Temporary holding area (like "shelve")
  - Can create a new branch from a stash

# Remote Repositories

- remote
  - Configure a remote location
- push
  - Move changes from local repository to remote repository
- pull
  - Move changes from remote repository to local repository

# Demo



# Links

- Code School <u>https://try.github.io</u>
- Git Documentation <u>https://git-scm.com/docs</u>
- MSDN Visual Studio Documentation <u>https://msdn.microsoft.com/en-us/library/hh850437.aspx</u>
- Git .gitignore <u>https://github.com/github/gitignore</u>

# Thank You!

## Jeremy Clark

- http://www.jeremybytes.com
- jeremy@jeremybytes.com
- @jeremybytes