# 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

#### C:\Windows\system32\cmd.exe

C:\Development\Sessions\Git\RunThrough\LambdaExpressions>git init Initialized empty Git repository in C:/Development/Sessions/Git/RunThrough/Lambd aExpressions/.git/

C:\Development\Sessions\Git\RunThrough\LambdaExpressions>git status On branch master

Initial commit

8×-

Untracked files: (use "git add <file>..." to include in what will be committed)

> HelloWorld/ LambdaExpressions.sln LambdaExpressions.v12.su

nothing added to commit but untracked files present (use "git add" to track)

C:\Development\Sessions\Git\RunThrough\LambdaExpressions>

### Git GUI

🚯 Git Gui (LambdaExpre		elopment/Sessions/Git/RunThro	u – 🗆 🗙	
Repository Edit Branch Commit Merge Remote Tools Help Current Branch: master				
Unstaged Changes			File: HelloWorld/Ma	
HelloWorld/App.xaml  HelloWorld/App.xaml.cs  HelloWorld/HelloWorld.cs  HelloWorld/HelloWorld.cs  HelloWorld/Properties/As:  HelloWorld/Properties/As:  HelloWorld/Properties/Re  HelloWorld/Properties/Se  HelloWorld/Properties/Se  HelloWorld/Properties/Se  HelloWorld/Properties/Se  HelloWorld/Properties/Se  Staged Changes (Will Commit	using Syste namespace F { public { puk { } }	<pre>/stem.Windows; em.Windows.Controls; delloWorld partial class MainWindow : plic MainWindow() InitializeComponent();</pre>	∧ Window ∨	
· · · · · · · · · · · · · · · · · · ·	<		· · · · · · · · · · · · · · · · · · ·	
	Rescan Stage Changed Sign Off Commit Push	Initial Commit Message: • New Commit	் Amend Last Commit	
< >			~	
Ready.				

### Visual Studio 2013

Team Explorer - Changes 👐 🕶 🔻 🗙			
🕒 🗇 🛱 🕴 🔯 Search Work Ite 🔎 🗝			
Changes   LambdaExpressions (Loc 🕶   🔊			
Branch: master 🗸   Unsynced Commits			
Enter a commit message <required></required>			
Commit 🔻 Actions 🗸			
✓ Included Changes (1) Exclude All   View Options			
🔺 🚄 C:\Development\Sessions\Git\Ru			
lainWindow.xaml.cs			
✓ Excluded Changes Include All View Options			
Drag changes here to exclude from the commit.			
▷ Untracked Files			

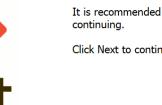
#### 🚽 Git Setup

#### 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.



gi

Next > Cancel

### Git Setup

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

#### **GNU General Public License**

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

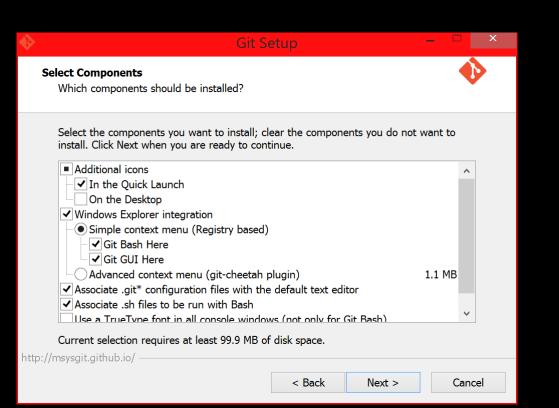
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software-

http://msysgit.github.io/

Information

< Back Next >

Cancel



- 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

#### it Setup

#### 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

 Use Git from Windows Command Prompt

- Add "git" to the path
- Allows git to be used from the command prompt
- \*I use this

#### Git Setup

**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").



- 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

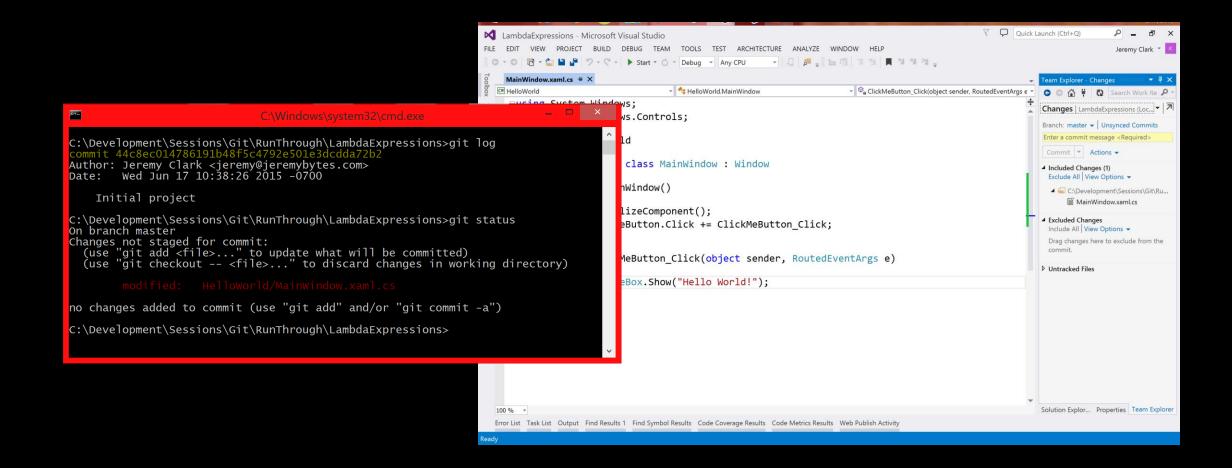
## Branching

- 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





- 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

- <u>http://www.jeremybytes.com</u>
- jeremy@jeremybytes.com
- @jeremybytes