



TOPICS

Version Control

在VS Code中使用版本控制

 (https://github.com/Microsoft/vscode-docs/blob/master/docs/editor/versioncontrol.md)

Visual Studio Code具有集成的源代码控制，并且包括内置的Git (https://git-scm.com/)支持。通过VS Code Marketplace上的扩展 (/docs/editor/extension-gallery)，可以使用许多其他源代码控制提供程序。




git历史

Donjayamanne

260万

查看git日志，文件历史记录，比较分支或C...




Azure存储库

ms-vssts

110万

连接到Azure Repos并与Git和团队合作...




汞

麦罗

56.2千

集成汞源控制



Perforce for VS代码

滑稽

47.2千

与VS Code的SCM功能进行Perforce集成

(https://marketplace.visualstudio.com/items?itemName=donjayamanne.githistory)

(https://marketplace.visualstudio.com/items?itemName=ms-vssts.team)

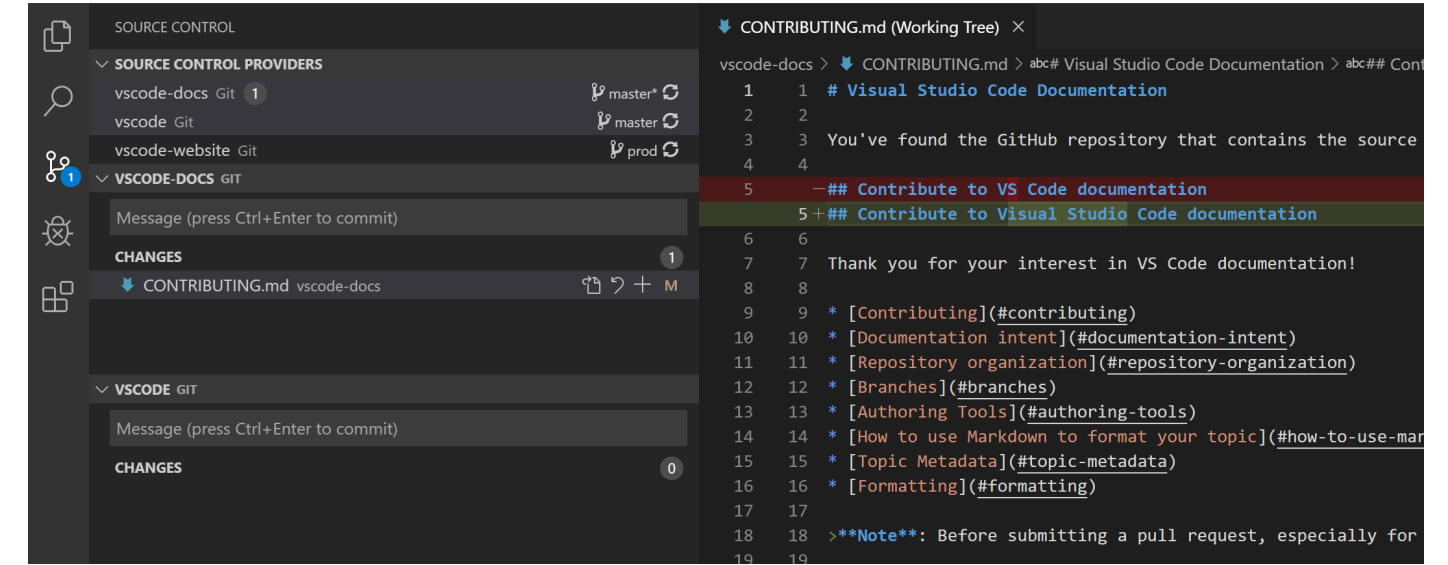
(https://marketplace.visualstudio.com/items?itemName=mrcrowl.hg)

(https://marketplace.visualstudio.com/items?itemName=slevesque.perforce)

提示： 点击扩展图块以阅读市场中的描述和评论。

单片机供应商

VS Code支持同时处理多个源代码控制提供程序。例如，您可以在TFS本地工作区旁边打开多个Git存储库，并在项目中无缝工作。**源代码管理视图**（ Ctrl + Shift + G ）的 **SOURCE CONTROL PROVIDERS**列表显示检测到的提供程序和存储库，您可以通过选择特定的提供程序来确定更改的显示范围。







SCM提供商扩展

如果要安装其他SCM提供程序，则可以在“扩展”视图（ Ctrl + Shift + X ）中的**scm提供程序**扩展类别中进行搜索。开始输入“ @ca”，您将看到有关扩展类别的建议，例如调试器和短毛绒。选择以查看可用的SCM提供程序。 @category:"scm providers"

EXTENSIONS: MARKETPLACE

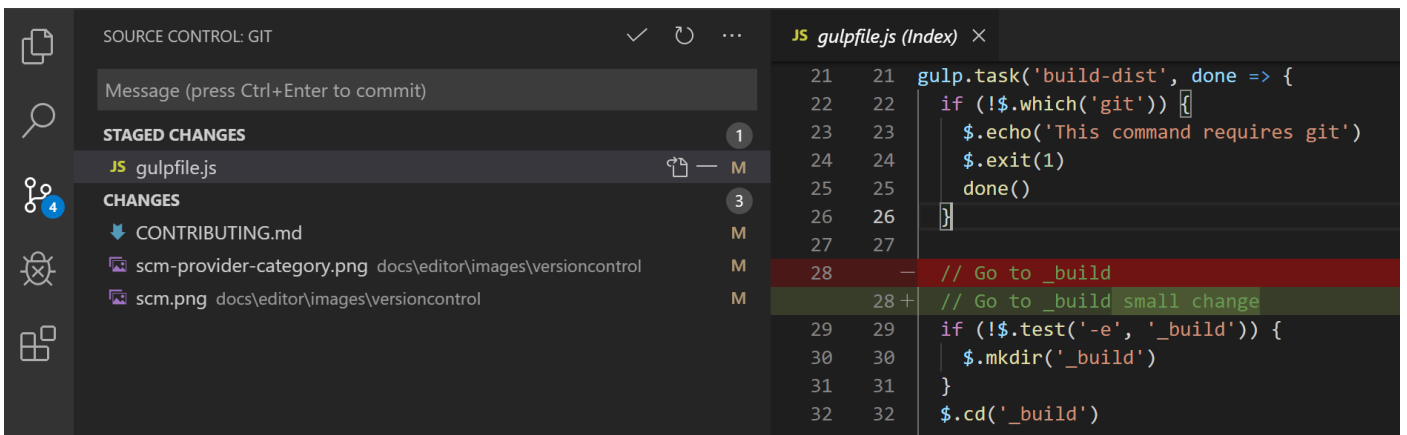
@category:"scm providers" |

	Azure Repos 1.149.2 Connect to Azure Repos and work with Git and Team Foundation ... Microsoft	2.1M ★ 2 Install
	SVN 1.54.4 Integrated Subversion source control Chris Johnston	2.3M ★ 4.5 Install
	TFS 0.7.2 TFS Visual Studio Code integration. Ivan Gabriele	84K ★ 3 Install
	Hg 1.3.0 Integrated Mercurial source control mrcrowl	89K ★ 4.5 Install

Git支持

VS Code附带了一个Git源代码管理器（SCM）扩展。大多数源代码管理UI和 workflows 在其他SCM扩展中是通用的，因此，阅读有关Git支持的信息将帮助您了解如何使用其他提供程序。

注意：如果您是Git的新手，那么git-scm (<https://git-scm.com/documentation>)网站是开始阅读流行的在线书籍 (<https://git-scm.com/book>)，入门视频 (<https://git-scm.com/video/what-is-git>)和备忘单 (<https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf>)的好地方。VS Code文档假定您已经熟悉Git。



注意：VS Code将利用计算机的Git安装，因此在获得这些功能之前，需要首先安装Git (<https://git-scm.com/download>)。确保至少安装了version 2.0.0。

提示：VS Code可与任何Git存储库一起使用。如果您还没有私人托管的Git提供程序，则Azure DevOps Services (<https://azure.microsoft.com/services/devops>)是一个不错的选择。您可以在Azure DevOps入门上进行 (<https://go.microsoft.com/fwlink/?LinkID=307137&campaign=o~msft~code~vc>)注册。

左侧的“源代码管理”图标将始终指示您当前在存储库中进行了多少更改的概述。单击它将会为您显示当前存储库更改的详细信息：**CHANGES**，**STAGED CHANGES**和**MERGE CHANGES**。

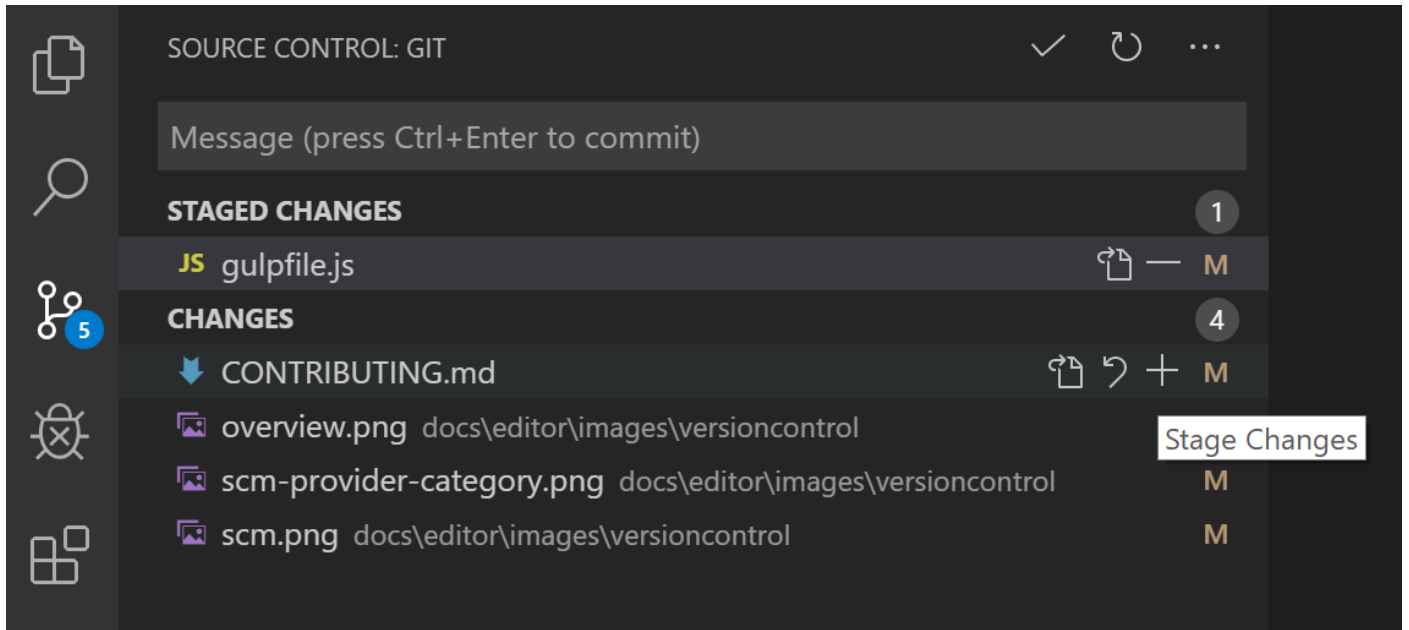
单击每个项目将向您详细显示每个文件中的文本更改。请注意，对于未分段的更改，右侧的编辑器仍可让您编辑文件：随时使用！

您还可以在VS Code的左下角找到存储库状态的指示符：当前分支，脏指示符以及当前分支的传入和传出提交的数量。您可以通过单击状态指示器并从列表中选择Git引用来检出存储库中的任何分支。

提示：您可以在Git存储库的子目录中打开VS Code。VS Code的Git服务仍将照常运行，显示存储库中的所有更改，但是作用域目录之外的文件更改将用工具提示阴影化，指示它们位于当前工作空间之外。

承诺

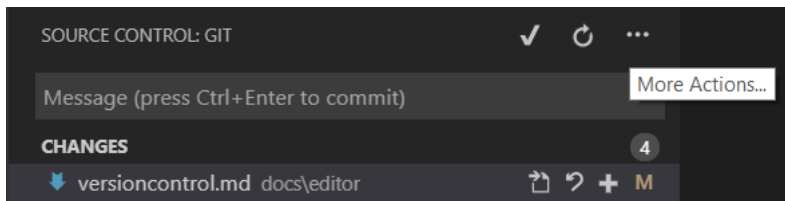
可以通过文件中的上下文操作或通过拖放来执行**暂存** (git add) 和**未暂存** (git reset) 。



您可以在更改上方输入提交消息，然后按 **Ctrl + Enter** (macOS: **⌘ + Enter**) 进行更改。如果有任何阶段性的更改，则仅将那些更改提交，否则将提交所有更改。

我们发现这是一个很棒的工作流程。例如，在先前的屏幕截图中，仅对的分段更改 `gulpfile.js` 将包含在提交中。连续的提交动作可以在以后的提交中提交对的更改 `gulpfile.js`，的删除 `yarn.lock` 以及对的更改 `tests.js`。

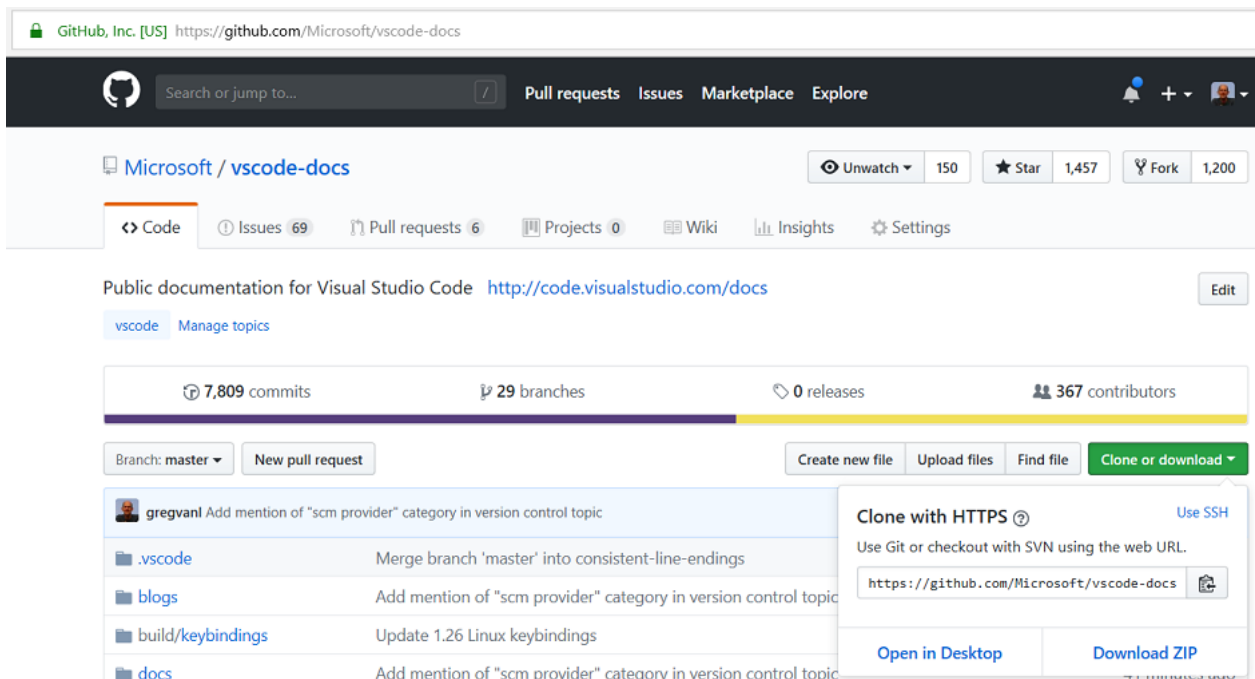
可以在Git视图顶部的“**更多操作**”菜单中找到更具体的“**提交**”操作。 ...



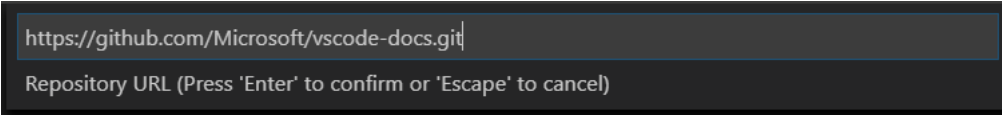
克隆存储库

您可以使用**命令面板** (**Ctrl + Shift + P**) 中的Git: **Clone**命令来克隆Git存储库。系统将要求您提供远程存储库的URL (例如GitHub上 (<https://github.com/>)的URL) 以及放置本地存储库的父目录。 (<https://github.com/>)

对于GitHub存储库，您可以从GitHub **Clone**或**下载**对话框中找到URL。



然后，您可以将该URL粘贴到Git: **Clone**提示中。

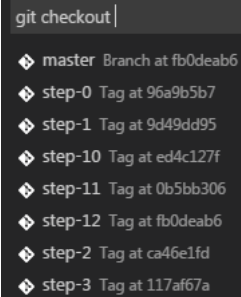


```
https://github.com/Microsoft/vscode-docs.git
Repository URL (Press 'Enter' to confirm or 'Escape' to cancel)
```

分支机构和标签

您可以在VS代码中直接通过Git: **创建分支**和Git: **在命令面板** (Ctrl + Shift + P) 中**签出**命令来创建和签出分支。

如果您运行Git: **Checkout to**, 您将看到一个下拉列表, 其中包含当前存储库中的所有分支或标签。



```
git checkout |
  master Branch at fb0deab6
  step-0 Tag at 96a9b5b7
  step-1 Tag at 9d49dd95
  step-10 Tag at ed4c127f
  step-11 Tag at 0b5bb306
  step-12 Tag at fb0deab6
  step-2 Tag at ca46e1fd
  step-3 Tag at 117af67a
```

该**混帐**: **建立分公司**命令, 可以快速创建一个新的分支。只要提供新分支的名称, VS Code就会创建该分支并切换到该分支。

遥控器

假设您的存储库已连接到某个远程站点, 并且您签出的分支具有到该远程站点的上游链接 (<https://git-scm.com/book/ch3-5.html>), 则VS Code为您提供了有用的操作来**推动**, **拉动**和**同步**该分支 (后者将运行pull命令, 然后执行一个**推**命令)。您可以在“**更多操作**”... 菜单中找到这些操作。

VS Code能够定期从您的遥控器获取更改。这使VS Code能够显示本地存储库在远程对象前面或后面的更改。从VS Code 1.19开始, 默认情况下禁用此功能, 您可以使用该 `git.autofetch` 设置 (/docs/getstarted/settings)启用它。

提示: 您应该设置一个凭证帮助程序, (<https://help.github.com/articles/caching-your-github-password-in-git/>)以避免每次VS Code与您的Git遥控器通话时都被要求提供凭证。如果您不这样做, 则可能要考虑通过该 `git.autofetch` 设置 (/docs/getstarted/settings)禁用自动获取, 以减少收到的提示数量。

Git状态栏操作

当前签出的分支配置了上游分支时, 在分支指示符旁边的状态栏中有一个“**同步更改**”操作。**同步更改**会将远程更改下拉到您的本地存储库, 然后将本地提交推送到上游分支。



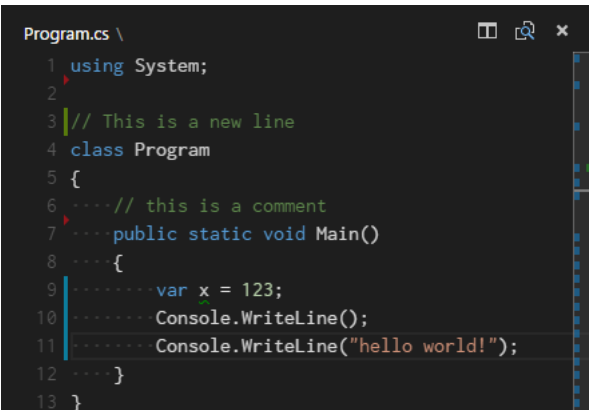
如果未配置上游分支, 并且Git存储库已设置了远程服务器, 则将启用“**发布**”操作。这将使您将当前分支发布到远程。



天沟指示器

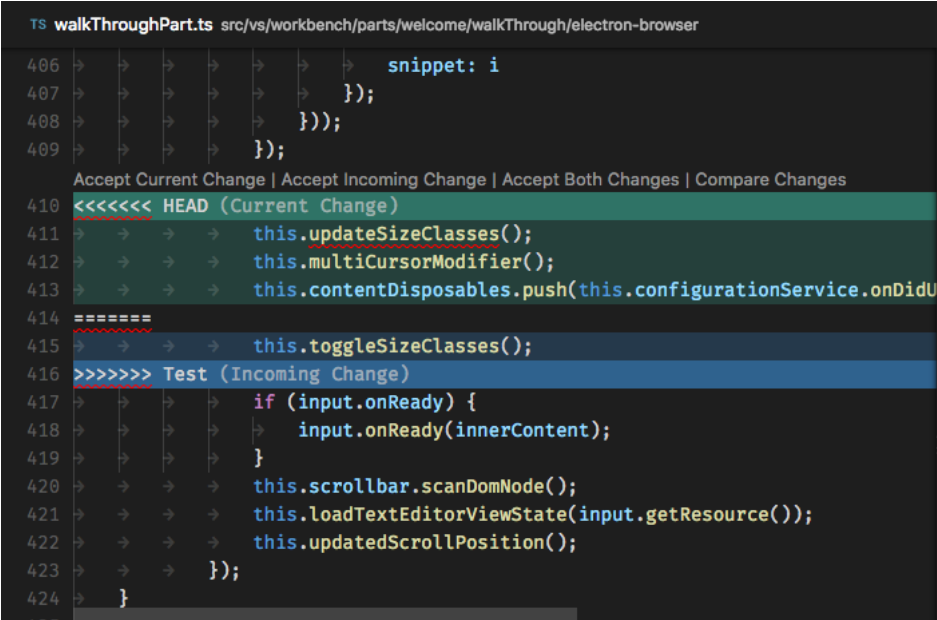
如果打开的文件夹是Git存储库并开始进行更改, 则VS Code将在装订线和概览标尺中添加有用的注释。

- 红色三角形表示已删除行的位置
- 绿色条表示新添加的行
- 蓝色条表示已修改的行



```
Program.cs \
1 using System;
2
3 // This is a new line
4 class Program
5 {
6     // this is a comment
7     public static void Main()
8     {
9         var x = 123;
10        Console.WriteLine();
11        Console.WriteLine("hello world!");
12    }
13 }
```

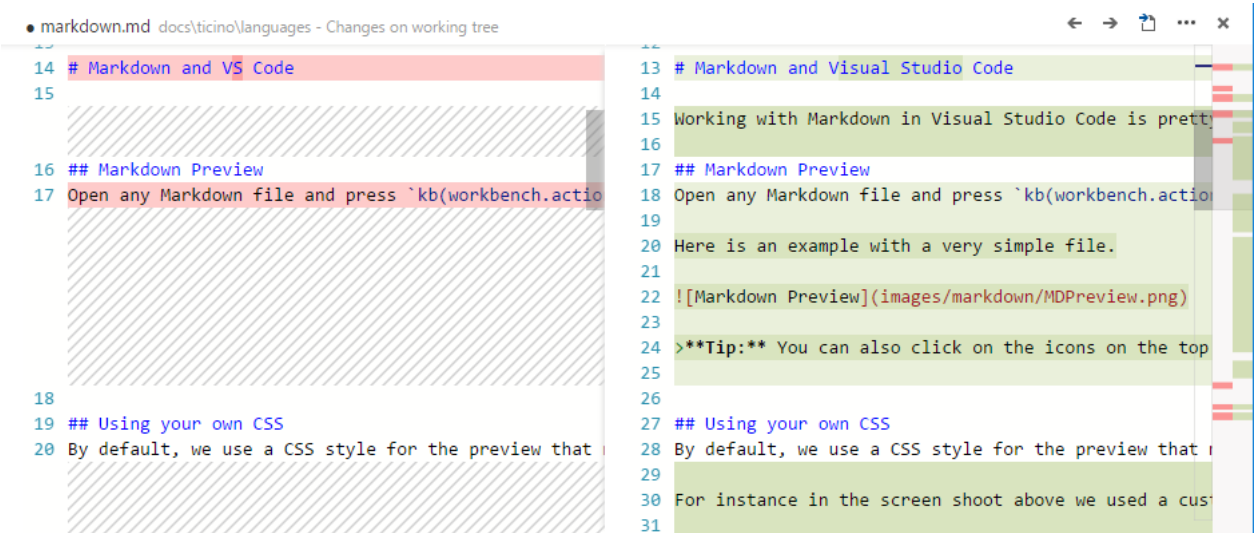
合并冲突



合并冲突由VS Code识别。突出显示了差异，并且有内联操作来接受一个或两个更改。解决冲突后，暂存有冲突的文件，以便您提交那些更改。

查看差异

我们的Git工具支持在VS Code中查看差异。



提示：您可以通过以下方式区分任意两个文件：首先右键单击资源管理器或“打开编辑器”列表中的文件，然后选择“选择比较”，然后右键单击要比较的第二个文件，然后选择“与file_name_you_chose比较”。或者，从键盘上按Ctrl + Shift + P并选择“文件：比较活动文件与”，将显示最近文件的列表。

差异编辑器查看窗格

差异编辑器中有一个检查窗格，以统一的补丁程序格式显示更改。您可以通过转到下一个差异（F7）和转到上一个差异（Shift + F7）在更改之间导航。可以使用箭头键浏览，然后按Enter键将在Diff编辑器和所选行中跳回。

```

JS diff.js (Index) x
1/2: @@ -7,10 +7,13 @@
7      7  Version: 0.5
8      8  */
9      9
10     10  +// Here are some inserted lines
11     11  +// with some extra comments
12     12  +...
13     13  (function (global, undefined) {
14     14  - "use strict";
15     15  - undefinedVariable = {};
16     16  - undefinedVariable.prop = 5;
17     17  - var definedVariable = {};
18     18  - definedVariable.prop = 5;
19     19  - function initializeProperties(target, members) {
20     20  -     var keys = Object.keys(members);
  
```

注意：这种经验对屏幕阅读器用户特别有用。

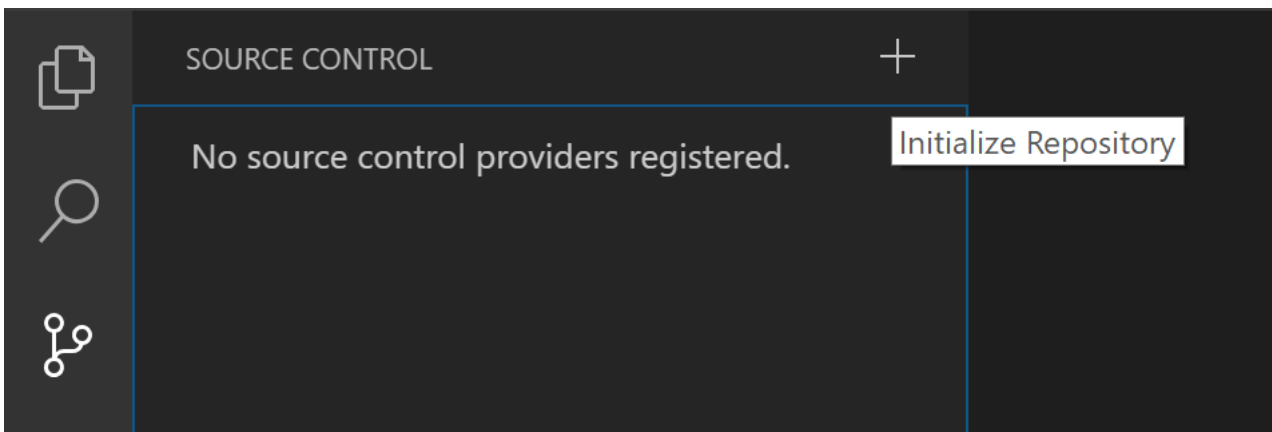
Git输出窗口

您总是可以在后台浏览我们正在使用的Git命令。如果发生奇怪的事情或只是好奇，这将很有帮助。:)

要打开Git输出窗口，请运行View > Output并下拉列表中选择Git。

初始化存储库

如果您的工作区不在Git源代码控制下，则可以使用Initialize Repository命令轻松创建一个Git存储库。当VS Code未检测到现有的Git存储库时，您将看到未注册任何源代码控制提供程序。标题栏上将显示“源代码管理”视图中的“消息”和“初始化存储库”命令。您还可以从命令面板（Ctrl + Shift + P）运行Git：初始化存储库命令。



运行Initialize Repository将创建必要的Git存储库元数据文件，并将您的工作空间文件显示为随时准备就绪的未跟踪更改。

VS Code作为Git编辑器

从命令行启动VS Code时，可以传递 --wait 参数以使启动命令等待，直到您关闭了新的VS Code实例。当您VS Code配置为Git外部编辑器时，这很有用，因此Git将等待直到关闭启动的VS Code实例。

这样做的步骤如下：

1. 确保您可以从命令行运行 `code --help` 并获得帮助。
 - 如果看不到帮助，请按照下列步骤操作：
 - macOS：选择Shell Command：在Command Palette 中的路径中安装“Code”命令。
 - Windows：确保在安装过程中选择了“添加到PATH”。
 - Linux：请确保您通过我们的deb或rpm软件包安装了Code。
2. 从命令行运行 `git config --global core.editor "code --wait"`

现在，您可以运行 `git config --global -e VS Code` 并将其用作配置Git的编辑器。

VS Code作为Git差异工具

将以下内容添加到您的Git配置中，以使用VS Code作为差异工具：

```

[diff]
  tool = default-difftool
[difftool "default-difftool"]
  cmd = code --wait --diff $LOCAL $REMOTE
  
```

这将利用 --diff 您可以传递给VS Code 的选项来并排比较2个文件。

总结一下，下面是一些可以使用VS Code作为编辑器的示例：

- `git rebase HEAD~3 -i` 使用VS Code进行交互式变基
- `git commit` 使用VS Code提交消息
- `git add -p` 然后是 `e` 进行交互式添加
- `git difftool <commit>^ <commit>` 使用VS Code作为差异编辑器进行更改

处理请求请求

Visual Studio Code还通过VS Code Marketplace上可用的扩展 (/docs/editor/extension-gallery)支持拉取请求工作流。通过拉取请求扩展，您可以直接在VS Code中查看，注释和验证源代码。



(<https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github>)

提示： 点击扩展图块以阅读市场中的描述和评论。

下一步

- 介绍性视频-Git版本控制 (/docs/introvideos/versioncontrol) -一种介绍性视频，概述VS Code Git支持。
- 基本编辑 (/docs/editor/codebasics) -了解强大的VS Code编辑器。
- 代码导航 (/docs/editor/editingevolved) -快速浏览源代码。
- 调试 (/docs/editor/debugging) -这就是VS Code真正发挥作用的地方
- 任务 (/docs/editor/tasks) -使用Gulp, Grunt和Jake运行任务。显示错误和警告
- 源代码管理API- (/api/extension-guides/scm-provider)如果要将其源代码管理提供程序集成到VS Code中，请参阅我们的源代码管理API。

常见问题

我初始化了仓库，但 ... 菜单中的操作全部变灰了

要推，拉和同步，您需要设置Git原点。您可以从存储库主机获取所需的URL。拥有该URL后，您需要通过运行几个命令行操作将其添加到Git设置中。例如：

```
> git remote add origin https://github.com/<repo owner>/<repo name>.git
> git push -u origin master
```

我的团队正在使用Team Foundation版本控制 (TFVC) 而不是Git。我该怎么办？

使用Azure Repos (<https://marketplace.visualstudio.com/items?itemName=ms-vsts.team>)扩展，它将点亮TFVC支持。

为什么“拉”，“推”和“同步”操作永远不会完成？

这通常意味着Git中没有配置凭据管理，并且由于某种原因您没有收到凭据提示。

您始终可以设置凭据帮助程序 (<https://help.github.com/articles/caching-your-github-password-in-git/>)，以便从远程服务器进行推送和推送，而无需每次都使用VS Code提示输入凭据。

如何通过需要多因素身份验证的Azure DevOps组织登录Git？

现在有Git凭证帮助者 (<https://devblogs.microsoft.com/devops/git-credential-manager-for-mac-and-linux>)可以协助进行多因素身份验证。您可以从下载这些混帐凭证管理Mac和Linux (<https://github.com/Microsoft/Git-Credential-Manager-for-Mac-and-Linux>)和Git的凭据管理器为Windows (<https://github.com/Microsoft/Git-Credential-Manager-for-Windows>)。

我的计算机上安装了GitHub Desktop，但VS Code忽略了它

VS Code仅支持官方Git发行版 (<https://git-scm.com/>)进行Git集成。

每当运行VS Code时，我都会不断收到Git身份验证对话框

VS Code会自动从服务器获取更改，以便向您呈现传入更改的摘要。Git身份验证对话框独立于VS Code本身，并且是当前Git凭据帮助器的一部分。

避免出现这些提示的一种方法是设置一个可记住您凭据的凭据助手 (<https://help.github.com/articles/caching-your-github-password-in-git/>)。

另一个选项是通过更改以下设置来禁用自动获取功能： `"git.autofetch": false`。