# GitHub / SourceTree

Our code repository of choice is GitHub (github.com). For this project, we are storing code under the abi-cip organization (<https://github.com/orgs/abi-cip/dashboard)>.

It is not required to use SourceTree to interact with the Git repository but it makes feature branching very simple via the Git Flow functionality and includes a user-friendly front-end.

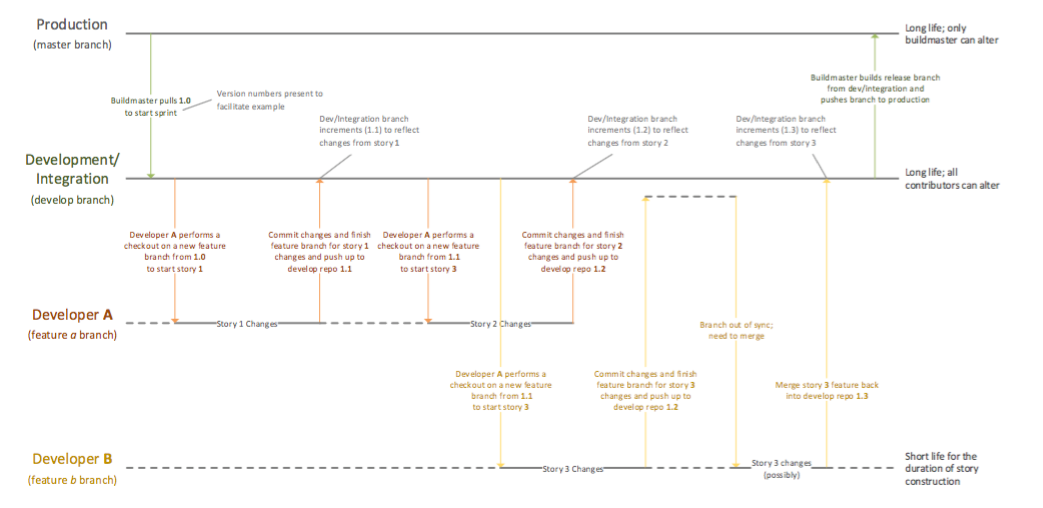
# Overview Diagram

In general, we aim to have a **master** branch which reflects the code that is in production. It will only be updated by the designated buildmaster. It will be updated by promotions from the Development/Integration (**dev**) branch.

The **dev** branch can be altered by any contributors and will reflect the code that is in the UAT environment. Ideally, this code base will be monitored and changes made here will automatically be integrated into UAT by Jenkins.

Individual developers will create local **feature** branches from the most current **dev** branch. These will have a very short life span, existing for the life of the feature development only. Once the feature development is complete, the code will be integrated into the remote **dev** branch and the feature branch will be deleted.

The only two branches on the GitHub server should be **master** and **dev**.



# 

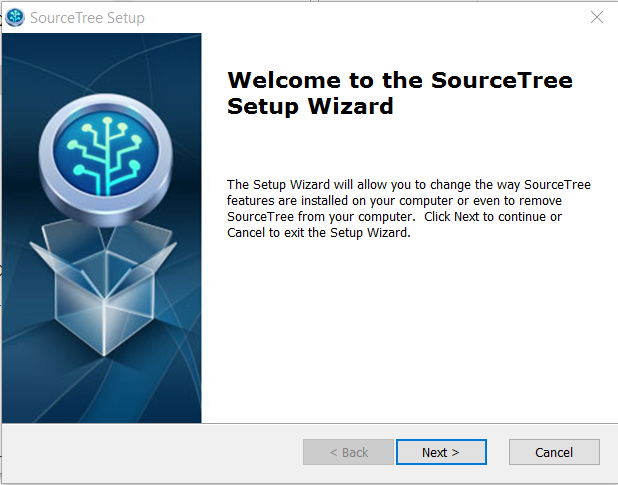
# Get access to the ABI-CIP repository

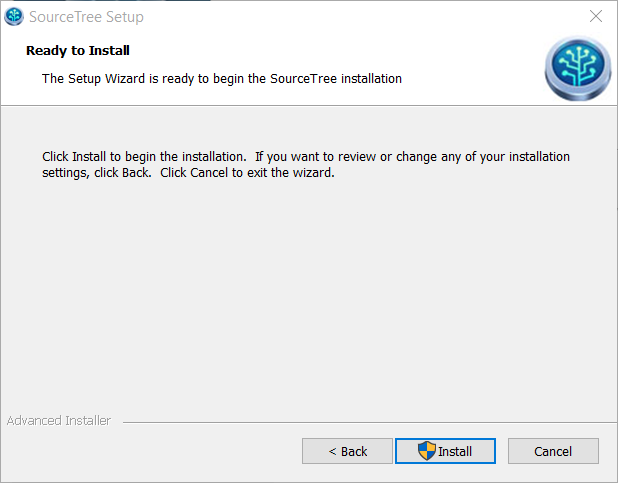
Please send email to ABI-CIP Github repository owner ([byron.lai@gettectonic.com](mailto:byron.lai@gettectonic.com)) with username to be granted access to the relevant repository. Upon receiving and confirming the Github email, you can proceed to next step.

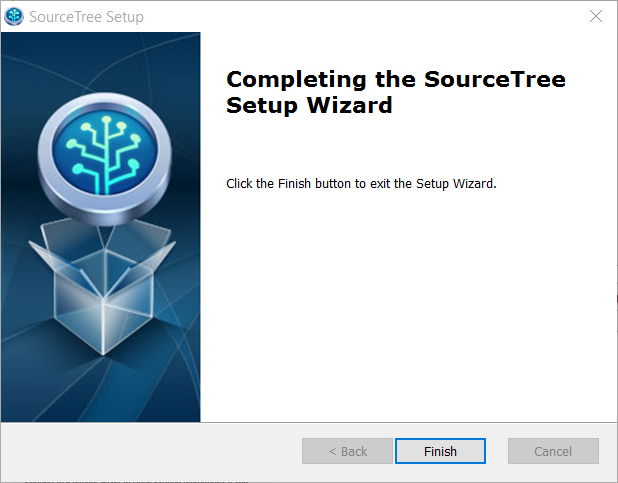
# SourceTree Installation

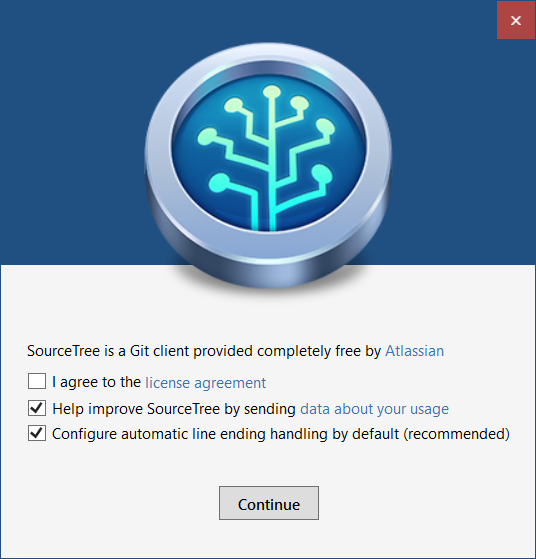
Use the following steps to install SourceTree.

1. Click the button for downloading SourceTree from the [SourceTree website](http://www.sourcetreeapp.com/). Depending on whether you are using a Windows or Mac, the installer is a DMG or EXE file.
2. Double-click the DMG or EXE file to open it.
3. Install SourceTree as you would any other installation. For details on how to go through this installation, expand the following sections.
   1. Install SourceTree for Windows
   2. Install SourceTree for Mac

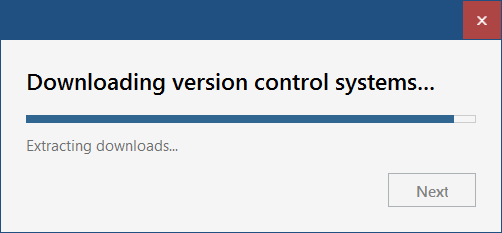


1. Click the **Next** button after Git is installed.
2. 
3. When it is ready to install, click Install.

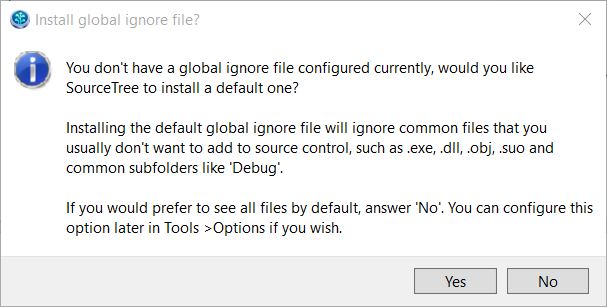


After Installation is complete, open Source Tree.

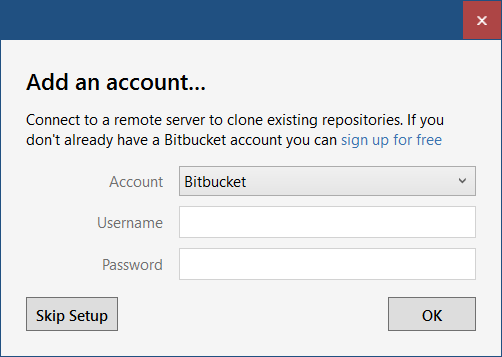
Click Continue, after checking “I agree to the license agreement”.



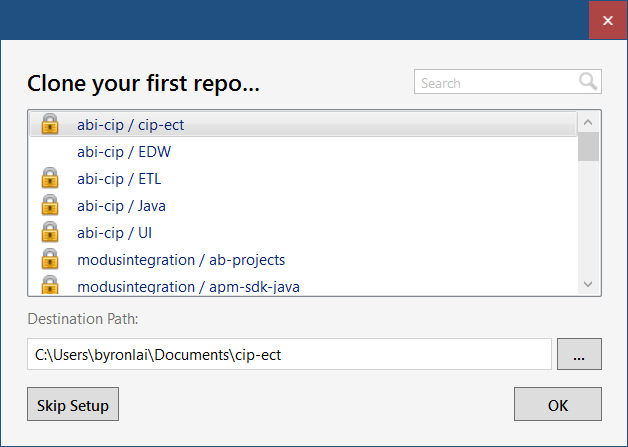
After downloading version control, click YES to accept default ignore file.



On the Add an account screen, change to add a GitHub account and supply Username / Password to logon



After successfully logon, select the correct repository to Clone. If abi-cip repositories are displayed, please confirm that you have requested and activated the ABI-CIP repository confirmation.



Symantec Endpoint Protection interferes with SourceTree, so it needs to be disabled when you are using the app. The easiest way to accomplish this is through a batch script:

* 1. Open **Notepad**.
  2. Copy and paste this text into Notepad:

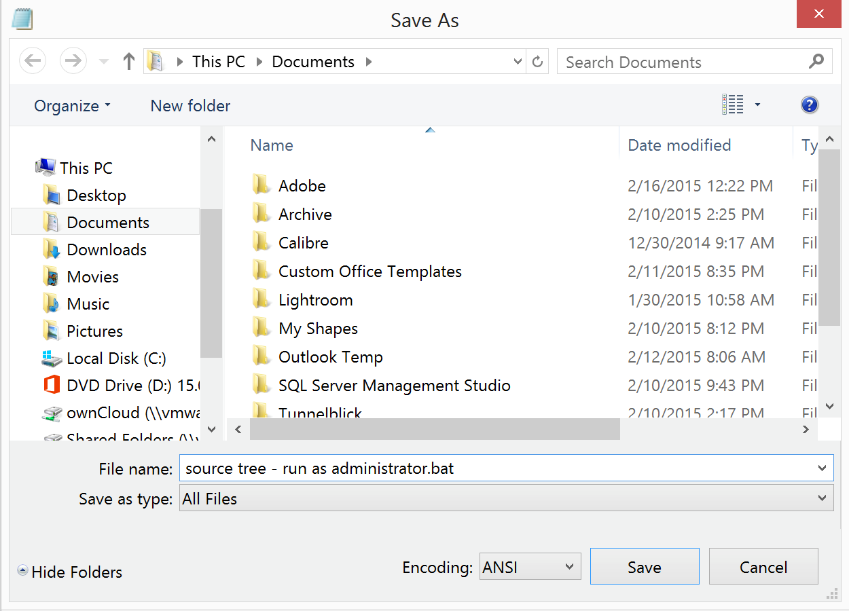
cd "C:\Program Files (x86)\Symantec\Symantec Endpoint Protection"

smc -stop

cd "C:\Program Files (x86)\Atlassian\SourceTree"

sourcetree.exe

* 1. Save the file locally and use it to start SourceTree. Note: It needs to be saved as a batch file, so make sure **Save as type** is set to *All Files* and the file name has a *.bat* extension.

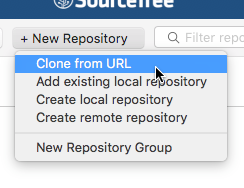


# SourceTree Project Setup

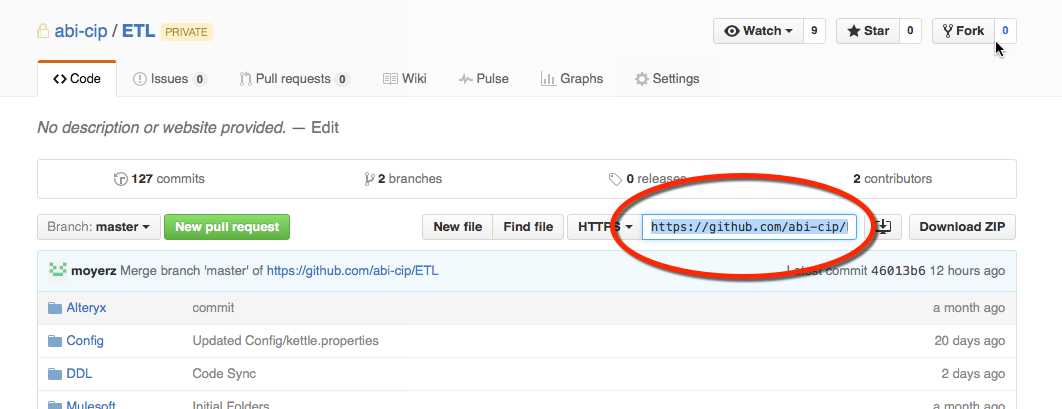
1. When SourceTree is open, click the **+New Repository** button to connect to the repository.



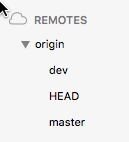
1. In the menu, choose **Clone from URL**:



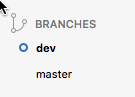
1. Retrieve the URL for the desired repository from the repository page of the GitHub site.



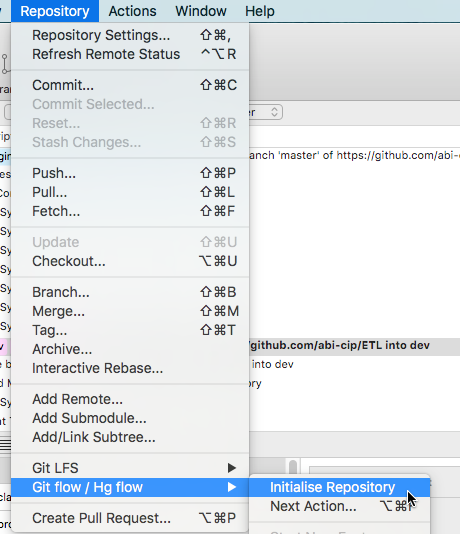
1. Under the **Clone a Repository** tab, specify:
   1. Source URL: [https://github.com/<your](https://github.com/%3Cyour) git directory>/<your project>
   2. Destination Path: <your local git directory>/<your project>
   3. Click the Clone button
2. Checkout the dev (development integration) branch.
   1. Expand **Remotes** in the project tree and right click **dev**



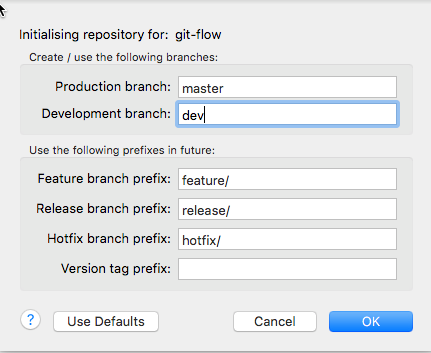
1. Select **Checkout…** from the menu
2. Click **OK** on the Checkout New Branch window.
3. You will now have a dev branch under the Branches section of the project tree.



1. Set up your workflow functionality.
   1. Ensure you are on the dev branch of the project tree.
   2. Under Repository, select **Git Flow / Hg flow** -> **Initialize Repository**



* 1. Change the Development branch from “develop” to dev and click OK on the Initialize repository for Git Flow window.

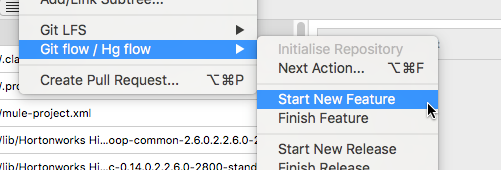


# Building a Feature (User Story)

1. To start work on your new feature:

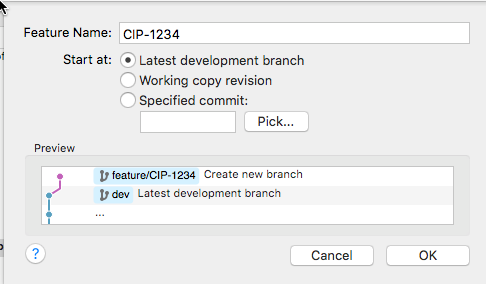
Note: If your local develop branch is out of sync with the remote (server), you must execute a Pull request (click the **Pull** button) first.

* 1. On the repository menu, choose Git flow / Hg flow -> Start New Feature.



* 1. Name your Feature branch with the Jira ID (i.e. CIP-1234). Then click **OK**.

This action will build a feature branch and checkout (clone) the current develop branch (dev integration) into your feature. It will also switch your local repository view from dev to your feature meaning that your local files are now feature files. This is all obfuscated from you by Git, even when you look at the project directory in and Explorer window.



1. Make changes to your code.

Your changes will show in the WORKSPACE section of the project tree. The SourceTree GUI will represent your changes as a count overlaying the File status text in the tree.

1. You may **Commit** your working changes to your feature branch at any time and any number of times.
   1. Select the Unstaged Files you wish to commit.
   2. Enter a commit comment (a sentence or two describing the change). These comments will be visible to all users and should include the Jira ticket number as well (e.g. CIP-1234 Removed the duplicate data in the staging set).
   3. Press the **Commit** button.

This is a healthy practice and helps you keep your feature in sync. You will not be able to finish your feature with uncommitted working changes. All changes need to be committed or backed out first.

1. To finish work on your feature:
   1. Get back into the **Git Flow** menu
   2. Click the **Finish Feature** button on the Choose Next Flow Action window.
   3. Click the **OK** button.

This action will merge your feature branch back into your local dev branch. Your feature branch will be removed and your local perspective will switch back to the dev branch.

* 1. Push your changes to the GitHub server. (Note that step 4c simply merged your local feature changes into your local dev branch). The **Push** button on the ribbon bar will show the number of changes in an overlaid circle; click that button. Then click the **OK** button.