Gitit user's manual

Bryanna Denney
Mike Salata
Steven Sennebogen

Introduction:
Git is a version control tool. Essentially, git will allow you to save multiple versions of your project, with more added features. Simply, git makes working on projects easier.

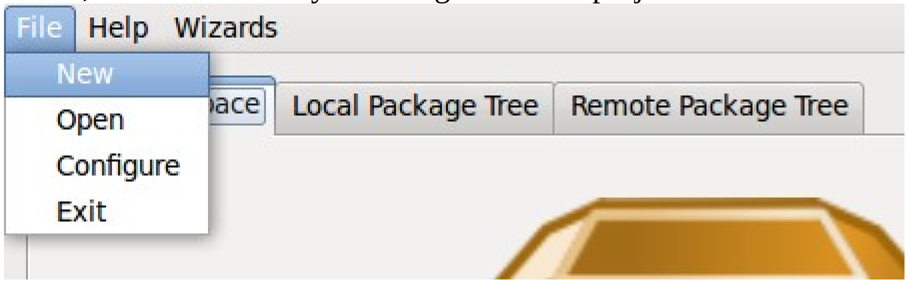
Have you ever started out on a feature in your code that changed many parts of the project, only to realize that you were headed on the wrong path and you need to “undo” the last 2 hours? Well, what git allows you to do is reverse your changes to the previous version that you saved.

Have you ever tried to work on a group without any version control? Git allows you to easily share your code with others. Git also provides a way to merge multiple changes to the same document, allowing more time to actually code and less time to integrate everyone's changes.

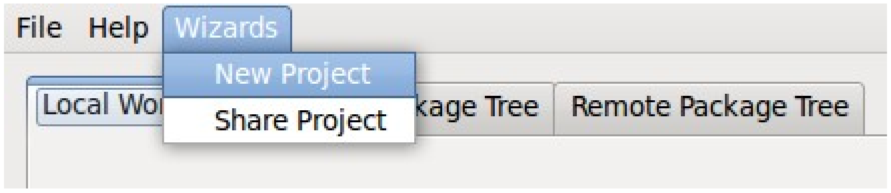
Have you ever lost parts of your project, or even worse all of your project, because it became corrupted or your machine broke? Git gives you the interface to keep multiple copies of your project on different machines. You can easily add multiple remote location to keep all of your projects.

There are more features that one can learn, but there are two ways to use git. There is the command line version and the graphical interface. Our goal is to present another way to use git through a new graphical interface that is more intuitive. We have named our project Gitit.

<u>Adding Git to a Project on Your Local Machine</u>:
If, on your local machine, you have a project that you would like to add git to, you will be creating what is called a git repository. This is a project that is using git. Having git connected to a project will mean that there are hidden folders within your project that will store the versions that you save of your project.
In Gitit, there are two ways to add git to a new project: File->New

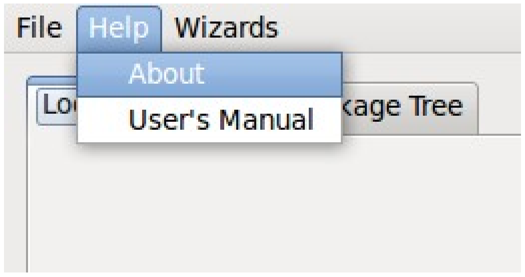


and Wizards->New Project.



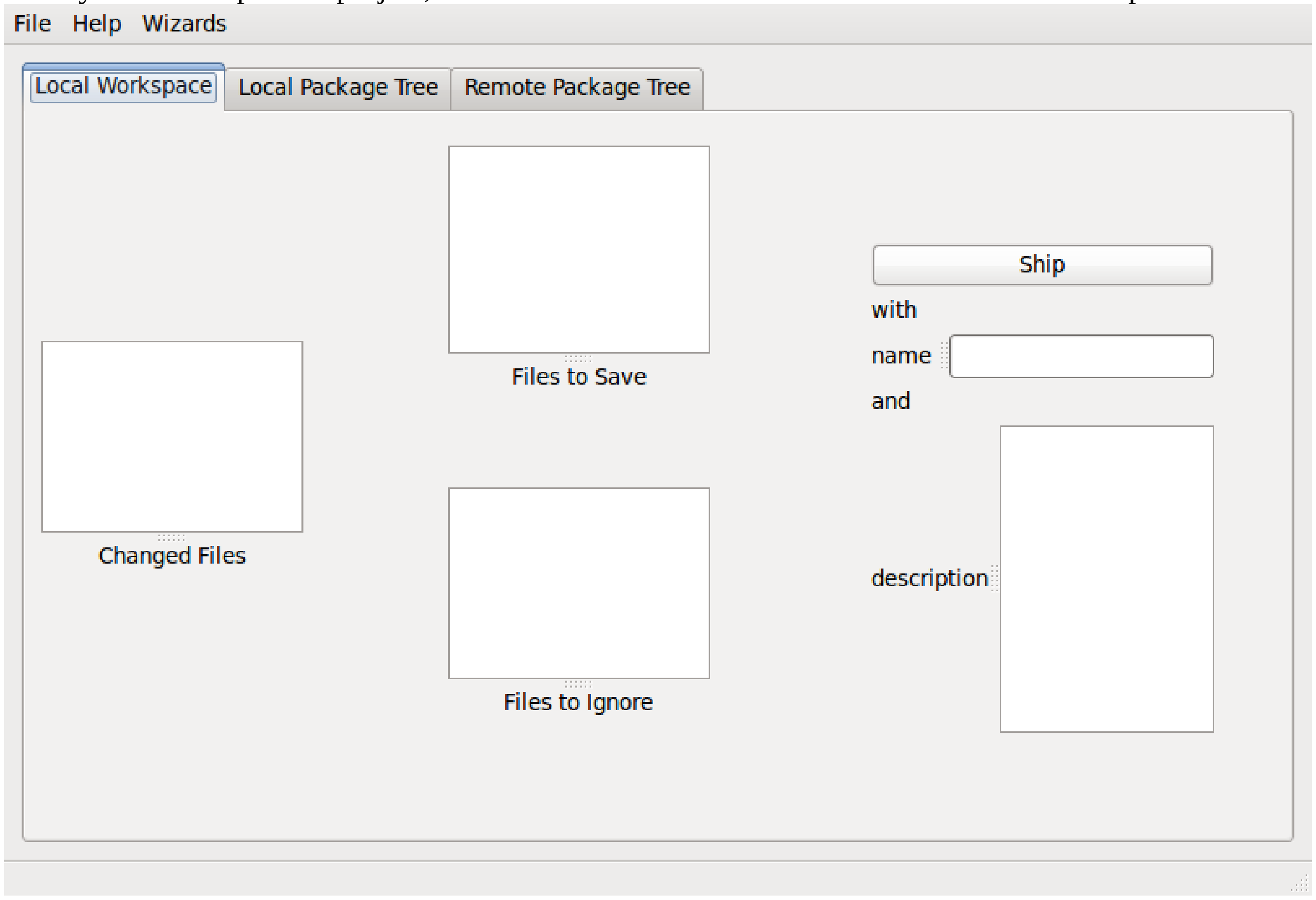
If you are new to git, We recommend the wizard.
You will need to specify that you are doing a local git repository.
Next, you need to pick a directory for git to changed into a repository. Everything inside that directory will be added to git including all files within subdirectories. Make sure that everything you want for the project is within it.
After submitting the required information, Gitit will create the git repository in the directory you specified.

<u>About (The Authors and other information)</u>:
To see the details about this project, you can go to Help->About:



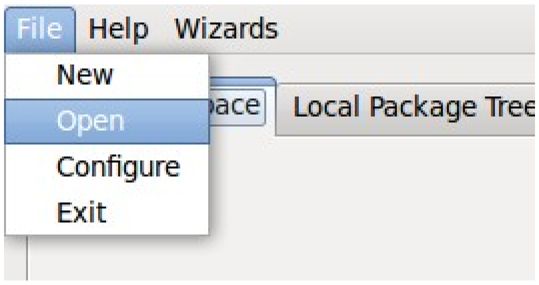
It will open a dialog box explaining the details of the project, including which class it was programmed for and the names and emails of the authors.

__Saving a version (Making a commit)__:
After you add or open the project, Gitit should load some information to the Local Workspace tab.



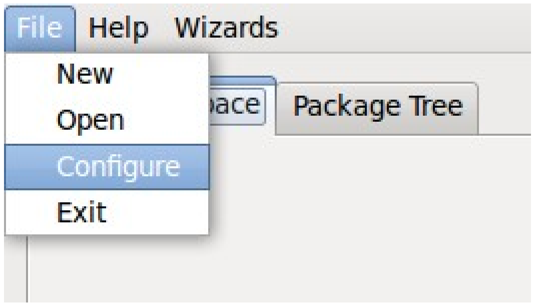
All of your files within the project should be located in the “Changed Files” box. Click the arrows according to what you want to do with the file. If you want to save the file in this version, go ahead and click the arrow leading to “Files to Save”. If you want to ignore the file, go ahead and click the arrow leading to “Files to Ignore”. Gitit will remember those files each time you load your project.
If you make a mistake, there are arrows leading from “Files to Save” to “Files to Ignore” and vice versa.
In order to save a version of your project, all of your files will need to be either saved or ignored.
Once finished, you can now start saving a version of your project, called making a commit. A commit is a saved version of your project. Make sure to name your commit and make a description. These are vital to do because you will be able to look at your commits and may need to revert to one of them. How hard would it be if none of them were named or had a description?
Once you hit “Ship”, Gitit will create the commit for you.

Opening a Git Repository:
If you have already created a git repository and just need to open it to create a commit, then you will use File->Open:



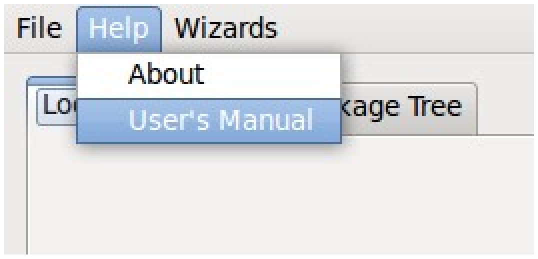
Gitit should then load the information to the Local Workspace tab and you can save a version (make a commit). See the Saving a version (Making a commit) section.

Configure Gitit:
When you make a commit, a couple of things are saved in addition to the project files. For every commit, the time, name of the person who made the commit, and the email of the person who made the commit. You will need to go to File-> Configure:



A dialog box will open asking for your name and email that you would like to be displayed on your commits. You only need to do this once per project or every time you would like to change. As well, changing your displayed email and name will not change past commits, only future ones.

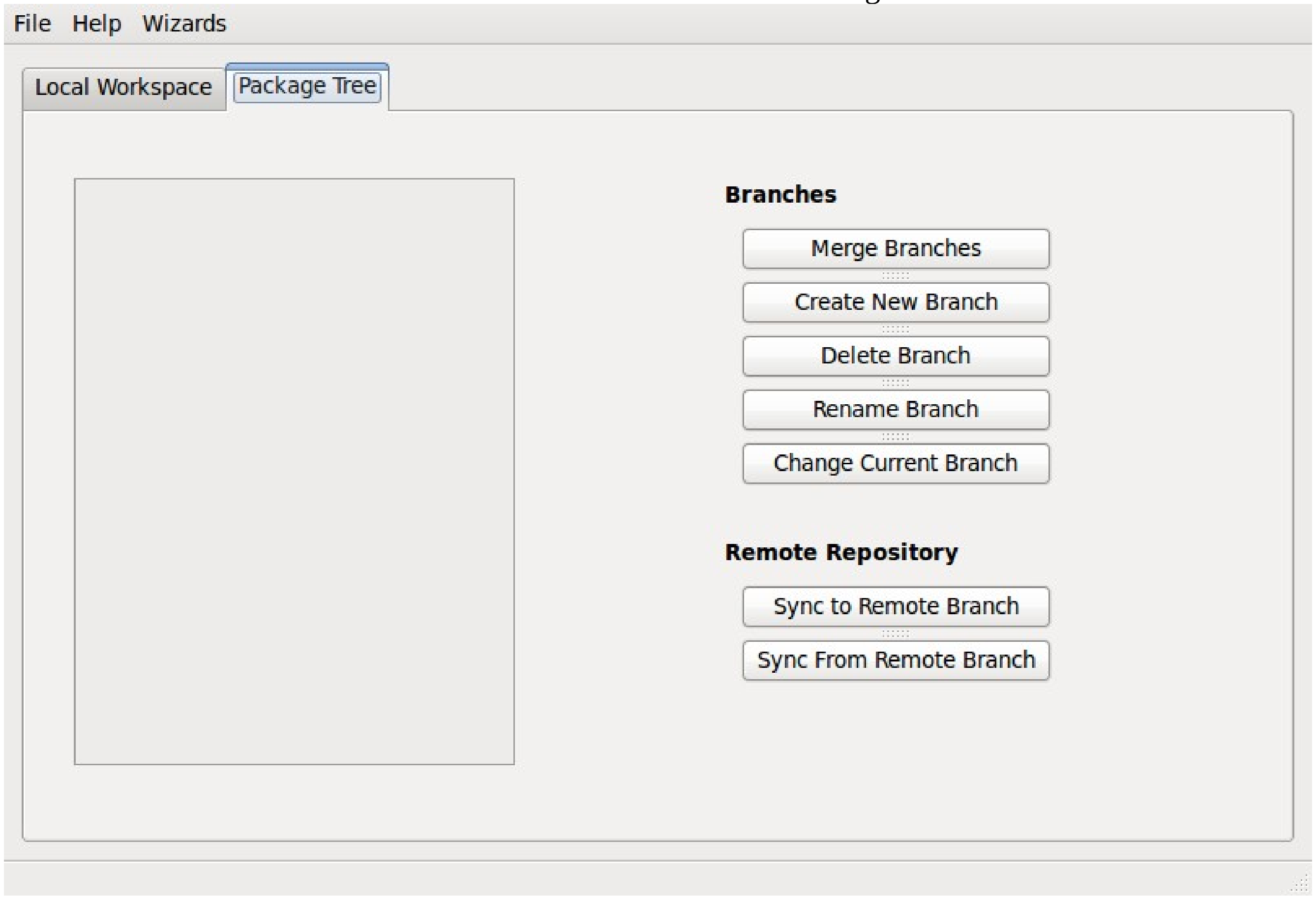
Getting Help:
This user's manual will be available within Gitit with Help-> User's Manual:



Good references to see are http://gitref.org and http://www.kernel.org/pub/software/scm/git/docs/gittutorial.html . The first is the Git reference, which have tutorials. The second is the official git tutorial.

Branches:

The buttons used to interact with branches are located on the “Package Tree” tab:



Branches are different topics/directions of your project. Let's say that you would like to have a working version at all times. You would have a branch that represents the working version of your project and another branch for features that you are currently developing. Let's say that you are adding a new feature, but aren't sure if it is going to be implemented in the final project. You could have a branch for that feature, but regularly work in another branch.

When you are working in a branch, your local directory will change to reflect the latest commit in that branch. You can switch easily in between branches.

There are several ways to work with branches:

Merge Branches:

You will merge branches when you want the content of two branches combined. You will need to select which two branches you want to merge.

Create New Branch:

You will be able to create new branches. You will need to provide a name of the new branch.

Delete Branch:

You will be able to delete branches. You will need to provide which branch you wish to delete. If you have commits that are only on that branch, those WILL be deleted. This is irreversible.

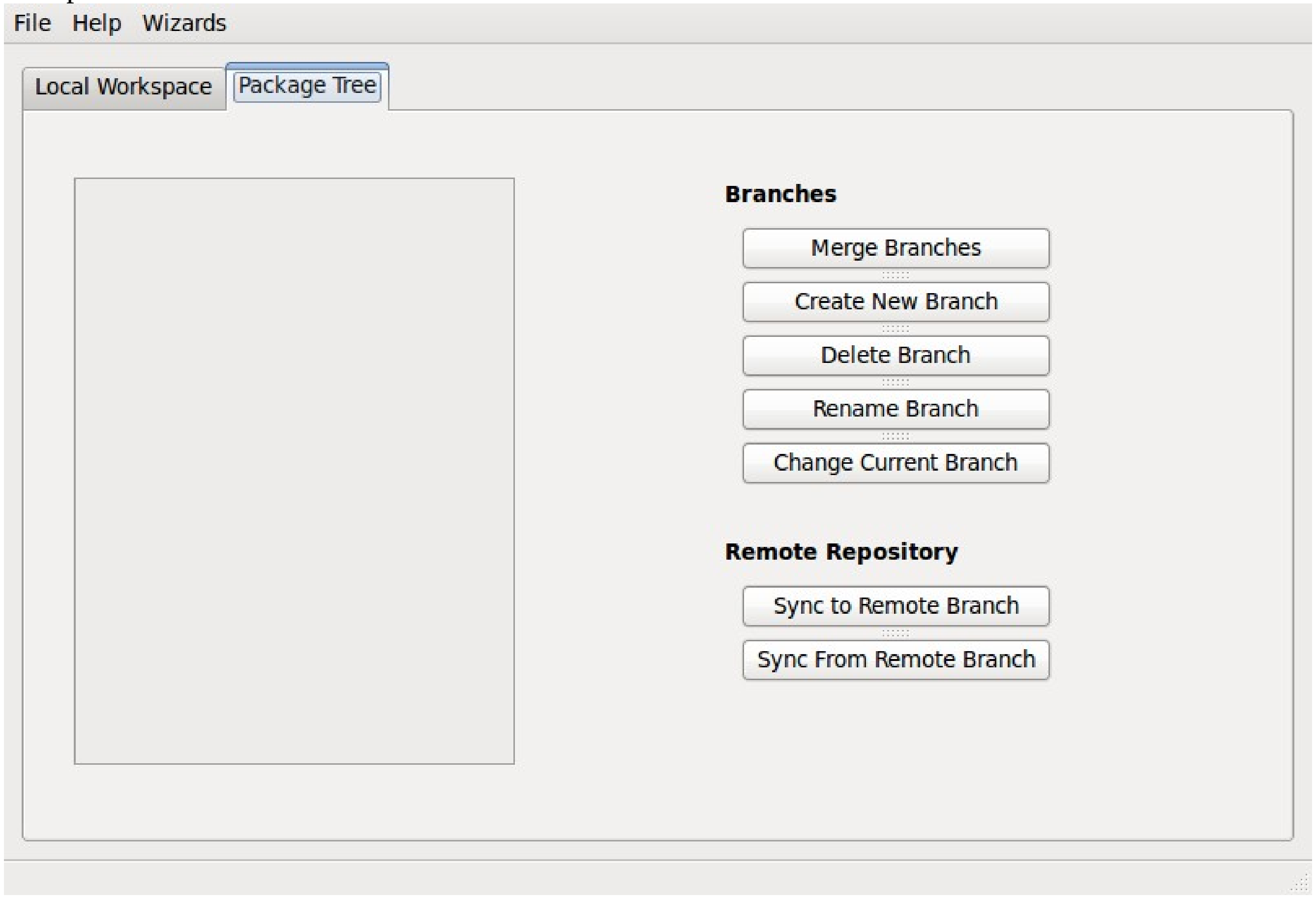
Rename Branch:

You will be able to rename branches. You will need to provide which branch you wish to rename and the new name of the branch.

Change Current Branch:

If you need to switch which branch your local directory represents, you can change your current branch. You will need to provide the name of the new branch you would like to switch to.

<u>Remote Repositories</u>:
Remote repositories are repositories that do not correspond to a directory. They usually reside on a server. You can get assistance in creating a remote repository using one of two wizards. They can be found in the "Wizards" file menu.
You will be able to sync to and sync from remote branches. "Sync to" indicates that you are sending up to the latest commit from one branch to the remote repository. "Sync from" indicates that you are receiving from the remote repository and you will need to specify to which branch you are receiving the update.


File
Help
Wizards
Local Workspace
Package Tree
Branches
Merge Branches
Create New Branch
Delete Branch
Rename Branch
Change Current Branch
Remote Repository
Sync to Remote Branch
Sync From Remote Branch