How to contribute changes to Ergo

Create Opensource Account

If you haven't already done so, obtain an account on the opensource repository for the Ergo project. To do this, do the following:

1. Go to http://opensource.ncsa.illinois.edu and signup for an account.

Setup Development Environment

After you have an opensource account, follow the steps here to setup your development environment. Please follow all links to make sure your environment is completely setup so your code contributions are formatted properly and include the appropriate license/class headers.

Development Workflow with GIT commands

After you have setup your development environment, you are now ready to start contributing code to Ergo. Bug fixes and new Features should be done in their own branch and the steps below will guide you through the process. If you created a fork of the Ergo repositories when you setup your environment, all of these steps are still relevant except when you create a pull request you will create a pull request for your fork instead of your feature branch.

Checkout Master

From inside Eclipse, do the following:

```
Team -> Switch To -> master
Team -> Pull
```

or

To checkout from a terminal:

```
git checkout master
git pull
git status
```

Create a branch and use JIRA issue as the branch name (e.g. ERGO-1)

If the issue does not exist in Jira, then first open a new issue using the "Create Issue" button here. Provide as much detail for the issue as you can. The issue can be modified later if any selection needs to be changed. The issue ID that gets assigned (e.g. ERGO-1) will be the name used for the git branch containing your new feature or bug fix. When contributing new features to Ergo, try to keep your changes to something that can be accomplished within one week or consider breaking up the feature into smaller tasks to avoid large commits and the potential hassle of resolving merge conflicts since master is under development.

To create a new branch inside Eclipse, do the following:

```
Team -> Switch To -> New Branch
Source: Remote Tracking > origin/master
Branch name: ERGO-1 (or whichever ID the issue you are working on was assigned by Jira)
Pull strategy: Rebase
```

Make sure **Checkout new branch** is selected and click Finish.

or

To create a new branch from a terminal:

```
git checkout -b <branch name> origin/master
```

Where branch name would be **ERGO-1** or whichever ID the issue you are working on was assigned by Jira.

Work on your issue

Since you are working on a branch from the master repository, you should commit and push regularly to your branch to save your work. The following are some useful commands for keeping your branch up to date.

From inside Eclipse:

```
# commit your changes to your local repository
Team -> Commit

# push your changes (commits), only needs to be done once
Team -> Push to Upstream

# update your git repository using the following command (this will not update any branches)
Team -> Fetch from Upstream

# update your branch with respect to the remote master (this does NOT do a fetch from remote!)
# In eclipse you can do use Team -> rebase and rebase with origin/master
Team -> Rebase and under Remote Tracking select origin/master
```

From a terminal:

```
# commit your changes to your local repository, use the Issue ID in commit message
git add *
git commit -m "ERGO-18 added missing license headers"

# push your changes (commits), only needs to be done once
git push origin <branch name>
# if you already pushed once using the above command you can just use
git push

# update your git repository using the following command (this will not update any branches)
git fetch

# update your branch with respect to the remote master (this does NOT do a fetch from remote!)
git rebase origin/master
```

When your work is done, do the following commands

From inside Eclipse:

```
# STEP 1: update your master from remote by rebase
# make sure all worked and no conflicts
Team -> Fetch from Upstream
Team -> Rebase and under Remote Tracking select origin/master

# STEP 2: push to the branch on remote
Team -> Push to Upstream

# STEP 3: goto stash website and issue a pull request
https://opensource.ncsa.illinois.edu/stash/projects/ERGO
```

From a terminal:

```
# STEP 1: update your master from remote by rebase
# make sure all worked and no conflicts
git fetch
git rebase origin/master

# STEP 2: push to the branch on remote
# if you have done a rebase and you had already pushed you might have to force (-f) the push
git push origin <branch name>

# STEP 3: goto stash website and issue a pull request
https://opensource.ncsa.illinois.edu/stash/projects/ERGO
```

For the pull request, please indicate Jong Lee and Chris Navarro as reviewers.

Update Jira

After making the pull request, update the JIRA issue and make the following statement in a comment:

```
1.  I authored 100% of the content I'm contributing
2.  I have the rights to contribute the content to Ergo
3.  I contribute the content under Ergo's License (Mozilla Public License v. 2.0)
```

At this point, the review process will begin. It will be an iterative process in which you interact with the reviewers who will comment and possibly request changes to your code. When changes are made to the branch, the reviewers will be notified so they process can continue until all outstanding issues are satisfied. After the review process is complete, the branch will be merged into master by one of the reviewers.

Cleanup

Once the branch is merged (it means your pull request was approved and merged to master on remote), you can cleanup. Make sure it is merged and you no longer need it before doing the following:

Inside Eclipse:

```
Team -> Switch to -> master

Under Branches -> Local, select the branch you created and select Delete Branch
```

From a Terminal:

```
git checkout master

# delete branch on local
git branch --delete <branchname>

# delete branch on remote
git push --delete origin <branchname>
```