

# Contributing to SystemML

There are many ways to become involved with SystemML:

- Mailing Lists
  - Development Mailing List
  - Issues Mailing List
  - Commits Mailing List
- Issue Tracker
- SystemML on GitHub
- Documentation

## Mailing Lists

### Development Mailing List

Perhaps the easiest way to obtain help and contribute to SystemML is to join the SystemML Development mailing list ([dev@systemml.incubator.apache.org](mailto:dev@systemml.incubator.apache.org)). You can subscribe to this list by sending an email to [dev-subscribe@systemml.incubator.apache.org](mailto:dev-subscribe@systemml.incubator.apache.org) (<mailto:dev-subscribe@systemml.incubator.apache.org>). You can unsubscribe from this list by sending an email to [dev-unsubscribe@systemml.incubator.apache.org](mailto:dev-unsubscribe@systemml.incubator.apache.org) (<mailto:dev-unsubscribe@systemml.incubator.apache.org>). The dev mailing list archive can be found here ([http://mail-archives.apache.org/mod\\_mbox/incubator-systemml-dev/](http://mail-archives.apache.org/mod_mbox/incubator-systemml-dev/)).

### Issues Mailing List

The issues mailing list archive can be found here ([http://mail-archives.apache.org/mod\\_mbox/incubator-systemml-issues/](http://mail-archives.apache.org/mod_mbox/incubator-systemml-issues/)). To subscribe to the issues list, send an email to [issues-subscribe@systemml.incubator.apache.org](mailto:issues-subscribe@systemml.incubator.apache.org) (<mailto:issues-subscribe@systemml.incubator.apache.org>). To unsubscribe from the issues list, send an email to [issues-unsubscribe@systemml.incubator.apache.org](mailto:issues-unsubscribe@systemml.incubator.apache.org) (<mailto:issues-unsubscribe@systemml.incubator.apache.org>).

# Commits Mailing List

The commits mailing list archive can be found here ([http://mail-archives.apache.org/mod\\_mbox/incubator-systemml-commits/](http://mail-archives.apache.org/mod_mbox/incubator-systemml-commits/)). To subscribe to the issues list, send an email to [commits-subscribe@systemml.incubator.apache.org](mailto:commits-subscribe@systemml.incubator.apache.org) (<mailto:commits-subscribe@systemml.incubator.apache.org>). To unsubscribe from the issues list, send an email to [commits-unsubscribe@systemml.incubator.apache.org](mailto:commits-unsubscribe@systemml.incubator.apache.org) (<mailto:commits-unsubscribe@systemml.incubator.apache.org>).

## Issue Tracker

Have you found a bug in SystemML? Have you thought of a way to improve SystemML? Are you interested in working on SystemML itself? If so, the SystemML JIRA Issue Tracker (<https://issues.apache.org/jira/browse/SYSTEMML>) is the place to go.

## SystemML on GitHub

Have you found an issue on the SystemML JIRA Issue Tracker (<https://issues.apache.org/jira/browse/SYSTEMML>) that you are interested in working on? If so, add a comment to the issue asking to be assigned the issue. If you don't hear back in a timely fashion, please contact us on the dev mailing list and we will be happy to help you.

Once you have an issue to work on, how do you go about doing your work? The first thing you need is a GitHub account. Once you have a GitHub account, go to the Apache SystemML GitHub site at <https://github.com/apache/incubator-systemml> (<https://github.com/apache/incubator-systemml>) and click the Fork button to fork a personal remote copy of the SystemML repository to your GitHub account.

The next step is to clone your SystemML fork to your local machine.

```
$ git clone https://github.com/YOUR_GITHUB_NAME/incubator-systemml.git
```

Following this, it's a good idea to set your git user name and email address. In addition, you may want to set the `push.default` property to `simple`. You only need to execute these commands once.

```
$ git config --global user.name "Your Name"
$ git config --global user.email "yourname@yourhost.com"
$ git config --global push.default simple
```

Next, reference the main SystemML repository as a remote repository. By convention, you can call this `upstream`. You only need to add the remote `upstream` repository once.

```
$ git remote add upstream https://github.com/apache/incubator-systemml.git
```

After this, you should have an origin repository, which references your personal forked SystemML repository on GitHub, and the upstream repository, which references the main SystemML repository on GitHub.

```
$ git remote -v
origin    https://github.com/YOUR_GITHUB_NAME/incubator-systemml.git (fetch)
origin    https://github.com/YOUR_GITHUB_NAME/incubator-systemml.git (push)
upstream  https://github.com/apache/incubator-systemml.git (fetch)
upstream  https://github.com/apache/incubator-systemml.git (push)
```

The main code branch by convention is the master branch. You can check out the master branch using the checkout command:

```
git checkout master
```

To update this branch with the latest official code, you can pull from the upstream master branch. A pull essentially does a fetch (retrieves code) and a merge (merges latest remote changes into your local branch):

```
git pull upstream master
```

It's recommended that you create a new, separate branch for your work based on the current master branch. Give this branch a descriptive name. For example, if you were assigned the issue SYSTEMML-101, you could use the checkout -b command to create a new branch based on the master branch and check out this branch:

```
git checkout -b SYSTEMML-101-my_cool_new_feature
```

At this point, you are ready to do your work on this branch.

If you updates involve code, you should run the complete test suite to verify that your updates have not had unexpected side-effects in the project. You can do this via the Maven verify command:

```
mvn clean verify
```

Your commit messages should follow standard git formatting conventions. If your commit is in regards to a particular JIRA issue, please include a reference to the JIRA issue, such as in the following:

```
git commit -m "[SYSTEMML-101] My cool new feature"
```

When ready, push your changes on this branch to your remote GitHub fork:

```
$ git push
fatal: The current branch SYSTEMML-101-my_cool_new_feature has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin SYSTEMML-101-my_cool_new_feature

$ git push --set-upstream origin SYSTEMML-101-my_cool_new_feature
```

At this stage, you can go to your GitHub web page and file a Pull Request for the work that you did on this branch. A Pull Request is a request for project committers (who have write access to Apache SystemML) to review your code and integrate your code into the project. Typically, you will see a green button to allow you to file a Pull Request.

Once your Pull Request is opened at SystemML Pull Requests (<https://github.com/apache/incubator-systemml/pulls>), typically Jenkins will automatically build the project to see if all tests pass when run for your particular branch. These automatic builds can be seen here (<https://sparktc.ibmcloud.com/jenkins/job/SystemML-PullRequestBuilder/>).

A conversation typically will proceed with regards to your Pull Request. Project committers and potentially others will give you useful feedback and potentially request that some changes be made to your code. In response, you can make the requested updates or explain why you feel that they make sense as they are. If you make additional updates, you can commit the changes and then push the changes to your remote branch. These updates will automatically appear in the pull request.

When your changes are accepted (a committer will write “Looks good to me”, “LGTM”, or something similar), a committer will attempt to incorporate your changes into the SystemML project. Typically this is done by squashing all of your commits into a single commit and then rebasing your changes into the master branch. Rebasing gives a linear commit history to the project.

If the merge is complicated, it is possible that a committer may ask you to resolve any merge conflicts in your pull request. If any difficulties are experienced, a project committer will be more than happy to assist in the integration of your work into the project.

After the Pull Request is closed, a comment can be added to the original JIRA issue referencing the Pull Request, and the issue can be resolved and closed.

## Documentation

Documentation is one useful way to become involved with SystemML. SystemML online documentation is generated from markdown using Jekyll. For more information, please see GitHub’s Using Jekyll with Pages (<https://help.github.com/articles/using-jekyll-with-pages/>).

After installing Jekyll, Jekyll can be run from the docs folder via:

```
bundle exec jekyll serve
```

This allows you to work on the documentation locally at <http://127.0.0.1:4000>.

You can allow others to preview your documentation updates on GitHub by pushing the docs subtree of your branch to your remote gh-pages branch.

```
git subtree push --prefix docs origin gh-pages
```

For instance, if you have filed a Pull Request for a documentation update on a regular branch, you could additionally push the docs subtree to the remote gh-pages branch. In the Pull Request conversation, you could include a link to the documentation that was automatically generated when you pushed to gh-pages. The URL is [http://<YOUR\\_NAME>.github.io/incubator-systemml/](http://<YOUR_NAME>.github.io/incubator-systemml/).

If you experience issues pushing the docs subtree to the gh-pages branch because you've previously pushed from a different branch, one simple solution is to delete the remote gh-pages branch and perform the subtree command again.

```
git push origin --delete gh-pages
```