

Contributing to HAWQ

- [HAWQ Mailing lists](#)
- [HAWQ Repositories](#)
- [HAWQ issue tracking - Jira](#)
- [HAWQ security reports](#)
- [Reporting New Security Problems with Apache HAWQ](#)
 - [Apache HAWQ Security Vulnerabilities](#)
 - [Errors and Omissions](#)
- [Coding conventions](#)
- [Contribution process \(for contributors\)](#)
- [Code review](#)
 - [Code review criteria](#)
- [Commit process \(for committers\)](#)

This guide gives the best way for contributing to Apache HAWQ. Contributing to HAWQ includes not only writing some code, it also includes helping new users on mailing list, testing releases and document improvement.

HAWQ Mailing lists

If you plan to contribute to HAWQ on an ongoing basis it is best to be subscribed to at least developer mailing list. As with any Apache Project all discussions are expected to happen there. Remember "if it didn't happen on the mailing list – it didn't happen". Follow [this link](#) to find out how to subscribe to various mailing lists.

HAWQ Repositories

- Apache HAWQ git repository (<https://git.apache.org/repos/asf/hawq.git>) : Apache internal git repo.
- Github Apache HAWQ mirror (<https://github.com/apache/hawq.git>) : This repo is synced automatically from Apache HAWQ git repo.

HAWQ issue tracking - Jira

Apache JIRA is used for issue tracking. All changes are discussed and reviewed in JIRA before commit. If you do not already have an Apache JIRA account, sign up [here](#). Note that the user name should have no white spaces or other special characters that complicate auto-completion within JIRA comments etc.

Please use a single JIRA account only (don't create multiple with different email addresses) to retain the issue history. Please use a permanent email address, for an existing account it can be changed in the profile. If you absolutely have to change your user name, contact INFRA.

HAWQ has 1 JIRA project:

1. [HAWQ](#) for [hawq](#) and [hawq-site](#)

Before working on changes for any of the repositories, please locate an existing JIRA ticket or submit a new one. Before a ticket can be assigned, you need to be listed as contributor in the JIRA project. One of the project's team members with JIRA administration access will be able to assist. Please request to be added through an email on the project's dev@ mailing list.

HAWQ security reports

Reporting New Security Problems with Apache HAWQ

The Apache Software Foundation takes a very active stance in eliminating security problems and denial of service attacks against Apache HAWQ.

We strongly encourage folks to report such problems to our private security mailing list first, before disclosing them in a public forum.

Please note that the security mailing list should only be used for reporting undisclosed security vulnerabilities in Apache HAWQ and managing the process of fixing such vulnerabilities. We cannot accept regular bug reports or other queries at this address. All mail sent to this address that does not relate to an undisclosed security problem in the Apache HAWQ source code will be ignored.

Questions about:

- how to configure HAWQ securely
- if a vulnerability applies to your particular application
- obtaining further information on a published vulnerability
- availability of patches and/or new releases

should be addressed to the users mailing list. Please see the HAWQ Mailing lists section for details of how to subscribe.

The ASF Security team maintains a page with a description of how vulnerabilities are handled, check their [Web page](#) for more information.

The private security mailing address is: security@apache.org.

Apache HAWQ Security Vulnerabilities

This page lists all security vulnerabilities fixed in released versions of Apache HAWQ.

Errors and Omissions

Please report any errors or omissions to the HAWQ Mailing lists section.

Coding conventions

HAWQ follows the PostgreSQL coding conventions: <http://www.postgresql.org/docs/9.1/static/source.html>

Contribution process (for contributors)

The overall contribution process is to work on apache repo (<https://git.apache.org/repos/asf/hawq.git>) and fork your own repo from the HAWQ github mirror repo for code review purpose.

Before forking your own repo, you need to have a github account. Then go to the web page <https://github.com/apache/hawq>, click "Fork" button. Now you get a forked repo (for example, <https://github.com/changleicn/hawq>).



Use the following steps to get the code to your local computer and do feature development.

```
# Clone the Apache repository.
git clone https://git.apache.org/repos/asf/hawq.git

# Enter the code directory
cd hawq

# Add your own forked github repo as "myfork" (used only for code review purpose)
git remote add myfork https://github.com/changleicn/hawq


# Merge the update of the upstream into your local master
git checkout master && git pull origin master

# Create a feature branch "feature-awesome" to work on
git checkout -b feature-awesome master

# Make changes and commit to local. Please format commit message as follows and feel free to add additional
# comments.
# <jira ticket number>. <Commit Message>
# Example: HAWQ-1002. Add awesome feature
git commit -a -m "HAWQ-1002. Add awesome feature"

# Push it to your own github fork for code review
git push --set-upstream myfork feature-awesome
```

Now you can go to your own github fork to create pull requests for code review (<https://github.com/changleicn/hawq>). Choose the master branch of <https://github.com/apache/hawq> as the base for code review. After your code is merged to Apache Repo by HAWQ committers, please close your pull request.

 base fork: **apache/incubator-hawq** ▾ base: **master** ▾ ... head fork: **changleicn/incubator-hawq** ▾ compare: **feature-awesome** ▾

Code review

We expect that pull requests which have been submitted on github will be reviewed by committers, but will be visible to the community for comments as well. Contributors should learn the code review criteria in order to make the changes accepted.

Code review criteria

- Features should have a lot of potential use cases and benefit a lot of users
- Fixes should fix the root cause of the problem
- Easily tested and have tests associated with the commits
- Changes have been discussed on JIRA and have a JIRA linked to your change(Pull Request)
- Code follows the coding conventions
- Do not introduce potential performance regressions

Commit process (for committers)

The overall commit process is to review the pull request. If the change passes code review, get the patch and commit it to Apache git repo.

```
# Create a working directory
mkdir working
cd working

# Clone the apache repository
git clone https://git.apache.org/repos/asf/hawq.git

# Add remote github mirror repository
git remote add github https://github.com/apache/hawq.git

# Get the pull request, using PR#2 as example, you can replace 2 with any other PR #.
git fetch github pull/2/head:feature_awesome

# Sometimes, the commits in the patch is based on old master
# So it needs to be rebased on current master
# For some cases, the committer should squash the commits in the branch into one commit
# And the original AUTHOR information should be kept to give credits to the contributor
git checkout feature_awesome
git rebase origin master

# Merge the patch (an alternative way to do this is to use "git cherry-pick")
git checkout master
git rebase feature_awesome

# Build
make

# Run tests (more in future before committing to "master" branch)
# Before run the following tests, you need to set up and start HDFS & HAWQ
# More setup details are at: https://cwiki.apache.org/confluence/display/HAWQ/Build+and+Install).
make installcheck-good

# In case of some contributor often forget to close pull request,
# Append commit message to automatically close pull request after code merged. e.g. PR number is 12:
run `git commit --amend` and append "(close #12)" in a new line to commit message.

# Push to apache repo (note: all commits needs an Apache JIRA)
git push origin master

# Delete the feature branch
# And close the JIRA & add comment in the pull request to ask the contributor to close the pull request
git branch -d feature_awesome
```