Contributing to Eagle

This guide documents the best way to make various types of contribution to Apache Eagle, including what is required before submitting a code change.

Contributing to Eagle doesn't just mean writing code. Helping new users on the mailing list, testing releases, and improving documentation are also welcome. In fact, proposing significant code changes usually requires first gaining experience and credibility within the community by helping in other ways. This is also a guide to becoming an effective contributor.

- Before Making Contribution
- Contributing by Helping Other Users
- Contributing by Reviewing Changes
- Contributing Documentation Changes
- Contributing Bug Reports
- Contributing Code Changes
  - JIRA
  - Pull Request
  - Review Process

Before Making Contribution

Before contributing, you should know that Apache Eagle is using:

- Mailing list for General Discussion
  - User: user@eagle.apache.org
  - Development: dev@eagle.apache.org
  - Issues: issues@eagle.apache.org
- Github for Code Collaboration: https://github.com/apache/eagle
- JIRA for Issue and Bug Tracking: https://issues.apache.org/jira/browse/EAGLE
- Website and Wiki for Documentation:
  - Website: https://eagle.apache.org/
  - Wiki: https://cwiki.apache.org/confluence/display/EAG/Eagle

Contributing by Helping Other Users

A great way to contribute to Eagle is to help answer user questions on the user@eagle.apache.org mailing list. There are always many new Eagle users; taking a few minutes to help answer a question is a very valuable community service.

Contributors should subscribe to this list and follow it in order to keep up to date on what's happening in Eagle. Answering questions is an excellent and visible way to help the community, which also demonstrates your expertise.

Contributing by Reviewing Changes

Changes to Eagle source code are proposed, reviewed and committed via Github (described later) at https://github.com/apache/eagle/pulls. Anyone can view and comment on active changes here. Reviewing others' changes is a good way to learn how the change process works and gain exposure to activity in various parts of the code. You can help by reviewing the changes and asking questions or pointing out issues -- as simple as typos or small issues of style.

Contributing Documentation Changes

Apache Eagle documentations are mainly host on:

- Site: https://eagle.apache.org/
- Wiki: https://cwiki.apache.org/confluence/display/EAG/Eagle

You could make contribution

- To modify the built-in documentation, edit the Markdown source files in Eagle's repository under https://github.com/eaglemonitoring/eaglemonitoring.github.io
- To modify or comment eagle wiki at https://cwiki.apache.org/confluence/display/EAG/Eagle
- To modify eagle site at https://eagle.apache.org/
  - Eagle site source files are hosted in apache SVN repo. https://svn.apache.org/repos/asf/eagle/site/
  - To view site content in local, use "jekyll serve"
  - Edit file and commit using SVN commands
- To have us add a link to an external tutorial you wrote, simply email the developer mailing list.

Contributing Bug Reports

- File a bug ticket at JIRA system: https://issues.apache.org/jira/browse/EAGLE
- Or send bug report to user mailing list: user@eagle.apache.org
Contributing Code Changes

When you contribute code, you affirm that the contribution is your original work and that you license the work to the project under the project’s open source license. Whether or not you state this explicitly, by submitting any copyrighted material via pull request, email, or other means you agree to license the material under the project's open source license and warrant that you have the legal authority to do so.

This section to help anyone who want to involve and contribute code to eagle project. Apache site have great documentation on how apache build the community and help the open source project. If one wants to know more detail on how apache works, he/she might want to check http://community.apache.org/.

JIRA

For who is new to eagle project, we have a label “new” to start with.

<table>
<thead>
<tr>
<th>T</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>P</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EAGLE-1070</td>
<td>Storm metrics ELK reporter posting metrics with same id</td>
<td>Hao Chen</td>
<td>Kesava Reddy</td>
<td></td>
<td>OPEN</td>
<td>Unresolved</td>
<td>Nov 29, 2017</td>
<td>Jan 02, 2018</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EAGLE-1066</td>
<td>All TimeZone values are not considered for UI</td>
<td>Unassigned</td>
<td>Jay Sen</td>
<td></td>
<td>OPEN</td>
<td>Unresolved</td>
<td>Oct 12, 2017</td>
<td>Jun 14, 2018</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EAGLE-251</td>
<td>Run eagle on flink/samza</td>
<td>Unassigned</td>
<td>Ralph Su</td>
<td></td>
<td>OPEN</td>
<td>Unresolved</td>
<td>Apr 13, 2016</td>
<td>Jan 04, 2018</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EAGLE-250</td>
<td>Metadata on mongodb</td>
<td>Ralph Su</td>
<td>Ralph Su</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
<td>Apr 13, 2016</td>
<td>Oct 12, 2016</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EAGLE-241</td>
<td>Beautify script output or logs</td>
<td>Hao Chen</td>
<td>Hao Chen</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
<td>Apr 12, 2016</td>
<td>Oct 08, 2016</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EAGLE-234</td>
<td>Clean maven build warning in eagle</td>
<td>Michael Wu</td>
<td>Hao Chen</td>
<td></td>
<td>OPEN</td>
<td>Unresolved</td>
<td>Apr 11, 2016</td>
<td>Jul 20, 2017</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EAGLE-196</td>
<td>eagle-topology.sh should have jar file path as parameter</td>
<td>Qingwen Zhao</td>
<td>Edward Zhang</td>
<td></td>
<td>RESOLVED</td>
<td>Fixed</td>
<td>Mar 10, 2016</td>
<td>Apr 13, 2016</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EAGLE-135</td>
<td>Run Eagle in Spark Streaming</td>
<td>Zhaokun Qin</td>
<td>Zhaokun Qin</td>
<td></td>
<td>CLOSED</td>
<td>Fixed</td>
<td>Jan 21, 2016</td>
<td>Aug 03, 2018</td>
<td></td>
</tr>
</tbody>
</table>

8 issues

- Search the user@eagle.apache.org and dev@eagle.apache.org mailing list archives for related discussions
- Search JIRA for existing issues: https://issues.apache.org/jira/browse/EAGLE
- Proposing a JIRA and pull request is appropriate only when a clear problem or change has been identified.

Pull Request

If you are ready to contribute code change. Here in this page we just focus on a consistent dev process to keep the project run smoothly.

1. Find the existing Eagle JIRA that the change pertains to. If confirmed the change is new, create a new JIRA ticket if required with required fields
   - Issue Type
   - Priority
   - Affects Version
2. Fork one’s own feature branch from eagle official github (https://github.com/apache/eagle/)
3. Fix bug / Develop feature in the feature branch. One might have multiple
4. After development is done, send a github pull request. Like https://github.com/apache/eagle/pull/19. NOTE: Please make sure the pull request title is in format of EAGLE-${ticket_number} ${ticket_title}, so that ASF bot could help sync the pull request comments to JIRA automatically.

Review Process

The review process is for Apache Eagle committer only.

The committer should follow below process to review the contribution and merge the code.
1. **Review:** Committers should review the pull request changes, leave comments and sigh off about whether accept the code change. (Reject/Recall/Pass)

2. **Merge:** merge the patch of the pull request and push the merged code to apache/eagle (following process is temporary solution for manual merging, the automatic tool is under development EAGLE-84)

   **Pre-requisite**
   Committers should have already cloned repo from apache github
   
   `git clone https://git-wip-us.apache.org/repos/asf/eagle.git`

   **git-apply**
   
   ```
   > git pull https://git-wip-us.apache.org/repos/asf/eagle.git master # get latest
   # anyway you like to get the patch and apply
   > # review the code; run the test;
   > git commit   # Commit with the commit MSG described below if everything looks good
   > git push https://git-wip-us.apache.org/repos/asf/eagle.git master # push to upstream
   ```

   **git-rebase when necessary**
   
   ```
   > git pull https://git-wip-us.apache.org/repos/asf/eagle.git master # get latest
   > git pull https://github.com/<contributor>/eagle.git EAGLE-ID
   > git rebase -i # squash and modify the final committed message
   > git commit
   > git push https://git-wip-us.apache.org/repos/asf/eagle.git master
   ```

   Or **git-reset**
   
   ```
   > git pull https://git-wip-us.apache.org/repos/asf/eagle.git master # get latest
   > git pull https://github.com/<contributor>/eagle.git EAGLE-ID
   > # review the code; run the test;
   > git reset origin/master
   > git add
   > git commit -a # Commit with the commit MSG described below if everything looks good
   > git push https://git-wip-us.apache.org/repos/asf/eagle.git master # push to upstream
   ```

   Committers should follow an convention format to add appropriate commit message. The format should be:

   ```
   EAGLE-{$jira_id} {$title}
   ${description}
   ${jira_url}
   
   Author: @${committer_github_id} <author_email>
   Reviewer: @${reviewer_github_id} <reviewer_email>
   
   Closes #${pull_request_id}
   ```

   - Author/Reviewer could be optional when the committer is the single reviewer.
   - For example: https://github.com/apache/eagle/commit/52b8e58b1af5327378245f52e61b4f4700e26c9
3. **Close**: Committer to close the JIRA ticket. (Pull request will be close automatically)

Here is a diagram to show the workflow above: