GitHub

Getting started with Git

Important: We are now fully migrated to a native git server solution. All procedures around using git is therefore slightly different, since there is no longer subversion synchronization process. To clone the new git repository, you can use

```
$ git clone https://github.com/apache/trafficserver.git
```

To work on a branch in git

```
$ git checkout -b 7.1.x origin/7.1.x
```

GitHub PR

1. Assignment of the PR should be for the person that created the PR
2. Set the labels to the appropriate ones
3. Set the milestone to the next major release (e.g. 10.0.0)
4. Set the project to the current release if you would like to have it backported (e.g. 9.0.x)
   a. Create another separate PR for each earlier release branch (e.g. 7.1.x or 8.0.x)
5. Reviews should be done by so
6. When merging a commit make sure there is ***only 1 commit*** (squash and merge if necessary).
7. Commit messages should follow the standard formatting described on the CommitPolicies page.
8. If the PR has documentation add a comment with: [approve ci docs]

Cherry-Pick (only for release managers)

When some commits in master branch are needed to be back-ported, cherry-pick and send Pull-Request.

1. Make sure the commits are already on master
2. Cherry-Pick with "-x"

   ```
   $ git cherry-pick -x <COMMIT>
   ```

3. Uncomment the conflics in commit message if there
4. Send a Pull-Request
5. Add "Backport" label

Example:

```
commit 90ad8484773095459f81242ec1ed61822da7142b
Author: Leif Hedstrom <zwoop@apache.org>
Date:   Fri Dec 9 08:08:31 2016 -0700

    Update the RAT exceptions

    (cherry picked from commit 7487ff750cd214ee956557273c8059b9b3580983)

    Conflicts:
            ci/rat-regex.txt
```

More details

More details for using Git and Subversion (particularly here at the ASF), see the following links:

- https://git-wip-us.apache.org/

Some older docs available at

- http://wiki.apache.org/general/GitAtApache
- http://git.apache.org/