

## art - Necessary Maintenance #7262

### Repair required to art repository on master and develop branches due to faulty tag and merge of v1\_12\_03

11/04/2014 12:08 PM - Christopher Green

<b>Status:</b>	Closed	<b>Start date:</b>	11/04/2014
<b>Priority:</b>	Immediate	<b>Due date:</b>	
<b>Assignee:</b>	Christopher Green	<b>% Done:</b>	100%
<b>Category:</b>	User Code	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.12.03	<b>Spent time:</b>	2.00 hours
<b>Scope:</b>	Internal	<b>SSI Package:</b>	
<b>Experiment:</b>	-		

#### Description

On Tuesday 2014/10/21 at 10:55 CDT, an erroneous tag and merge was made to the main repository's master and develop branches. The tag was **v1\_12\_03**.

The changes to be made to the repository will change the origin/master and origin/develop branches to preserve their states at tag time and HEAD while cleaning up the merge history to retain readability. The v1\_12\_03 tag will be moved to a new branch v1\_12-branch which will be constructed to allow the code representing the tag to remain identical while allowing the aforementioned cleanup.

If you have a local clone of the repository via any method (e.g. http or ssh), please for each clone:

1. Make a tarfile backup with (e.g.):

```
( cd .. ; tar jcf <tarfile> <local-repo-name> )
```

Do not skip this step unless you like losing work!

2. If you have not pulled or fetched origin/develop or origin/master from cdcvs.fnal.gov since before the faulty merge 2014/10/21, then please refrain from doing so until you have been given the all-clear. This is the only action required of you in this case.
3. If you have, or you think you may have, any local branches with the bad merge in the tree (including develop), you may check with the following command:

```
git branch -a | grep -v remotes | sed -e 's&*& &g' | while read branch; do git rev-list "${branch}" | grep -q 41d89bdb99254440fb52d00d79ce8ad6db2991cb && echo "${branch}"; done
```

4. For each branch whose name was printed as a result of the above command (i.e. has the offending merge):
  1. Commit any local changes you wish to preserve. This includes anything you currently have stashed, since the stash will also be no good after the repair has been made.
  2. Then, execute the following command to preserve your changes:

```
git checkout <branch>  
git format-patch --stdout <first-good-sha1>..<last-good-sha1> > <branch>-rescue.txt
```

Note the literal .. separating the two SHA1s. git log or gitk can help you decide which commits you need to keep.

5. Update the metadata for upstream branches:

```
git fetch origin
```

6. For each branch identified by step (3), when **and only when** you receive the all clear that the repair has been made:
  - If there is a remote tracking branch (e.g. master and develop):

```
git checkout <branch>  
git reset --hard origin/<branch>  
# If you have a saved patch:  
git am <branch>-rescue.txt
```

- If your branch has no remote tracking branch:

```
git branch -D <branch>  
git checkout -b <branch> origin/develop
```

```
# If you have a saved patch:  
git am <branch>-rescue.txt
```

#### 7. Update the tag:

```
git fetch --tags
```

As of right now, there are no published feature branches which include the bad merge commit, so no-one should have to deal with that eventuality.

## History

---

### #1 - 11/04/2014 02:26 PM - Christopher Green

- *Description updated*

Update description to include the extra checklist item to fetch the new definition of the tag.

### #2 - 11/10/2014 08:39 AM - Christopher Green

- *Status changed from Assigned to Resolved*

### #3 - 12/03/2014 09:57 AM - Christopher Green

- *Status changed from Resolved to Closed*

- *% Done changed from 0 to 100*