Release Process

This page describes the process of releasing a new version of Mynewt through the ASF.

Overview

A Mynewt release consists of three pieces:

1. Core
2. Newt, Newtvm
3. newtmgr
4. Blinky
5. NimBLE (on a separate release track, see Apache NimBLE release process)

Each of these pieces has its own git repository:

Piece	Git Repository
Blinky	https://github.com/apache/mynewt-blinky
Core	https://github.com/apache/mynewt-core
Newt, newtvm	https://github.com/apache/mynewt-newt
newtmgt	https://github.com/apache/mynewt-newtmgr

When you are ready to release a new version of Mynewt, use the below steps to guide you through the process.

Step 0: Create the branch (optional step) - major release rc1 is done on master branch

> Release branch is NOT created for major releases RC1.
>
> Release branch is created only for major releases eg 1.4.0, for minor releases eg. 1.4.1 use major release branch (and name it as major event if not present).

If this is RC1 of major release skip this step.

For RC2 of major release create branch of RC1 tag

```
git fetch origin
git checkout major-release-rc1_tag    (eg.    git checkout mynewt_1_10_0_rc1_tag)
```

For point release create branch of major release final tag

```
git fetch origin
git checkout major-release_tag    (eg.    git checkout mynewt_1_10_0_tag)
```

To create the branch, use the following git command:

```
git branch <branch_name>
```

Generally, you want to switch to the branch after you have created it. To switch to the branch, use the following git command:

```
git checkout <branch_name>
```

The shorthand way to do both of these (create and switch to the branch):

```
git checkout -b <branch_name>
```

The naming convention for the branch is based on the release number. The naming convention is the release number separated by underscores followed by _dev.

Point releases use major release naming.

Examples:

1. Release 1.0.0: git branch 1_0_0_dev
2. Release 2.1.2: git branch 2_1_0_dev

You now need to push the branch to the remote. You should be in that branch currently when doing this with the command listed here.

```
git push origin <branch_name>
```

Always a good idea to see if it all worked. To view both local and remote branches:

```
git branch -a
```

Step 1: Review github pull requests

> For minor release eg 1.4.1 PRs should be made against major release branch eg mynewt_1_4_0_dev and should contain only cherry-picks (or equivalent if cherry-pick is not feasible) from master branch.

There may be some changes intended for the release that never made it in. Review the outstanding pull requests in the three github mirrors and make sure none of them need to make it into the release. If any of them should be in the release, they need to be merged before you can proceed with the release process.

Pull request URLs

Piece	URL
Blinky	https://github.com/apache/mynewt-blinky/pulls
Core	https://github.com/apache/mynewt-core/pulls

Newt, newtvm	https://github.com/apache/mynewt-newt/pulls
newtmgr	https://github.com/apache/mynewt-newtmgr/pulls

Step 2: Ensure all tests pass

The Mynewt testing procedure is underdeveloped. Testing consists of two steps:

1. Run the core unit tests (see example output below). Run the unit tests under both OS X and Linux.
2. Manually execute the Apache NimBLE Test Plan.
3. Do Coverity Scan

```
$ newt test all
...lots of compiling and testing...
...about 2 minutes later ...
Archiving bootutil.a
Linking test_bootutil
Executing test: /Users/ccollins/tmp/myproj/bin/unittest/libs/bootutil/test_bootutil
Passed tests: [net/nimble/host fs/nffs libs/os hw/hal libs/mbedtls libs/util sys/config libs/bootutil]
All tests passed
```

Step 3: Test build newt and newtmgr on Linux, MacOS and Windows.

Step 4: Audit licenses of third-party code

Every third-party library included in a release that is not Apache-licensed has to be accounted for. For information about what licenses are Apache-compatible, see the following pages:

- Allowed: http://www.apache.org/legal/resolved.html#category-a
- Not allowed: http://www.apache.org/legal/resolved.html#category-x

Download Rat

To perform a license audit, you will need to use a tool called Apache Rat. Download Rat here: http://creadur.apache.org/rat/download_rat.cgi

Rat Usage

Rat checks all the files in the specified path for something resembling an Apache license. Files that do not contain an Apache licensed get flagged. A *.rat-excludes* file is used to indicate which files should be ignored by Rat. Once a non-Apache file has been accounted for, its name can be added to the excludes file.

To execute the Rat tool from the command-line:

1. Change to the base directory of the repository you wish to audit (e.g., *~/dev/core*).
2. Run the following command:

```
java -jar <rat-directory>/apache-rat-0.11.jar -E .rat-excludes -d .
```

Where *<rat-directory>* is the location where you unpacked the rat tool.

Example:

```
[ccollins@ccollins-mac:~/repos/core]$ java -jar ~/Downloads/apache-rat-0.11/apache-rat-0.11.jar -E .rat-
excludes -d .
*****************************************************
Summary
-------
Generated at: 2016-03-03T19:28:50-08:00
Notes: 4
Binaries: 2268
Archives: 0
Standards: 3926
[...lots of output snipped...]
```

Audit Procedure

1. Run the Rat tool from the base directory of the repository being audited.
2. Look at the source files flagged by Rat. For each file:
    a. If the file was created for Mynewt (i.e., not third-party), then it is missing the Apache license text. Copy the text from another Mynewt source file and paste it at the top of the problematic file. You can verify that the license text is correct by comparing it to the text at the bottom of this page: http://www.apache.org/licenses/LICENSE-2.0
    b. If the file is third-party and it uses a license that is **not** Apache-compatible (http://www.apache.org/legal/resolved.html#category-x), then the files must be removed from the release.
    c. If the file is third-party and it uses a license that **is** Apache-compatible:
        i. Add a pointer at the end of the repository's LICENSE file indicating that the library is included in the product and that it is licensed in an Apache-compatible way. The LICENSE file should be located in the root of the repository. A single pointer can account for an entire library if the library is isolated in the directory tree. If Mynewt sources files are intermixed with third-party files, each third-party file must be specified in the LICENSE file.
        ii. Add the filename to the repository's *.rat-excludes* file. This file should be in the repository's base directory. Preface the addition with a comment explaining which library the files are from and how they are licensed. This change prevents Rat from flagging the same files during subsequent executions.

Note

Unfortunately, Rat does not appear to support more than one directory in filename paths in the exclude file. I.e., the following entries are OK:

- crc16.*
- system_nrf52.h
- libs/baselibc

But these aren't:

- fs/nffs/src/crc16.c
- hw/mcu/nordic/nrf52xxx/include/mcu/system_nrf52.h

This could become problematic if a repository contains two files with the same name.

Step 5: Update the newt version numbers

Update the newt tool so that:

- "newt version" reports the new version number
- "newt new" downloads the correct version of blinky

The version strings are defined at the top of *newt/newtutil/newtutil.go* in the newt repository:

```
var NewtVersion Version = Version{0, 9, 0}
var NewtVersionStr string = "Apache Newt (incubating) version: 0.9.0"
var NewtBlinkyTag string = "mynewt_0_9_0_tag"
```

Official releases should have versions like "1.2.0", "1.3.0". Development versions should have versions like "1.2.0-dev", "1.3.0-dev" an so on. Please remove the "-dev" postfix when doing a new release!

Getting the *NewtBlinkyTag* string right is a bit tricky, since you have not created the new tag yet. Refer to step 6 to see how the tag will be named.

Make sure you commit and push the change!

Step 5a: Update the newtmgr version number

Update version string in newtmgr/newtmgr.go so that `newtmgr version` prints proper version.

```
VersionString: "1.5.0",
```

Step 5b: Update repository.yml on master branch

Note: no update to 0-latest and 1-latest yet !

```
@@ -35,9 +35,10 @@ repo.versions:
     "1.4.1": "mynewt_1_4_1_tag"
     "1.5.0": "mynewt_1_5_0_tag"
     "1.6.0": "mynewt_1_6_0_tag"
+    "1.7.0": "mynewt_1_7_0_tag"

     "0.8-latest": "0.8.0"
@@ -49,6 +50,7 @@ repo.versions:
     "1.4-latest": "1.4.1"
     "1.5-latest": "1.5.0"
     "1.6-latest": "1.6.0"
+    "1.7-latest": "1.7.0"

 repo.newt_compatibility:
     # Allow all versions for 0.0.0.  This is a workaround to prevent a warning
@@ -61,6 +63,9 @@ repo.newt_compatibility:
     # Core 1.4.0+ requires newt 1.4.0+ (feature: sync repo deps).
     # Core 1.5.0+ requires newt 1.5.0+ (feature: transient packages)
     # Core 1.6.0+ requires newt 1.6.0+ (feature: choice)
+    # Core 1.7.0+ requires newt 1.7.0+ (feature: range)
+    1.7.0:
+        1.7.0: good
     1.6.0:
         1.6.0: good
     1.5.0:
@@ -87,6 +92,7 @@ repo.deps:
             mynewt_1_4_1_tag: 1.0.0
             mynewt_1_5_0_tag: 1.0.0
             mynewt_1_6_0_tag: 1.1.0
+            mynewt_1_7_0_tag: 1.2.0

     mcuboot:
         type: github
@@ -94,3 +100,4 @@ repo.deps:
         repo: mcuboot
         vers:
             master: 0-dev
+            mynewt_1_7_0_tag: 1.3.1
```

Repeat this procedure for each of the following repositories:

- https://github.com/apache/mynewt-core
- ~~https://github.com/runtimeinc/mynewt_arduino_zero~~

Step 5c: Update project.yml on mynewt-blinky

In beetwen releases blinky template should point to mynewt-core development branch (version 0.0.0) while for releases it should point to version being released. This commit should be tagged (see point 7)

```
@@ -27,6 +27,6 @@ project.repositories:
 #
 repository.apache-mynewt-core:
     type: github
-    vers: 0.0.0
+    vers: 1.11.0
     user: apache
     repo: mynewt-core
```

Step 6: Update *README.md* and *RELEASE_NOTES.md* files

Each of the main repositories (core, newt, blinky) contains a *README.md* file in its root directory. Most of the repositories also contain a *RELEASE_NOTES.md* file in the same location. Make sure these files contain up to date information. In particular, ensure these files:

- Don't contain old dates or version numbers
- Update copyrigths dates if needed

- Mention TODO items which are complete in this release
- Mention upcoming releases which have already been released (or are in the process of being released).

Note: Technical documentation for the release gets generated according to the process described in the "Cut the new release documentation set" in the README here: https://github.com/apache/mynewt-site.

Step 6a: Update *LICENSE and NOTICE* files if needed

Make sure that LICENSE lists all non-APL components, and remove those no longer shipped.

Step 7: Tag the Repositories

Each git repository must be tagged with the name of the release. Tagging the repositories makes it easy to determine exactly what went into the release. The tag should be named as follows:

- mynewt_<*version-number*>_rc<#>_tag

The *rc<#>* is the release candidate number. Start with an rc of 1; increment this number each time you have to cancel and restart the release process. Replace periods in the version number with underscores. If this is a beta release, append *b<number>* to the version-number. For example, 0.8.0 beta 2 rc3 should have the following tag:

- *mynewt_0_8_0_b2_rc3_tag*

Tag the repository with the following git commands:

```
git tag -a <tag-name> -m <commit-message>
git push origin --tags
```

The first command creates a tag in your local repository. The second pushes the tag to the remote server.

newt starting from version 1.8 supports handling of rc tags so there is no need to create final tag at this step.

If something goes wrong and you need to delete the tag here are the commands to do this. Note that the first command deletes it from the remote, the second locally.

git push --delete origin <tag_name>

git tag --delete <tag_name>

Example:

```
[ccollins@ccollins-mac:~/repos/core]$ git tag -a mynewt_0_8_0_b2_rc3_tag -m 'Mynewt 0.8.0 B2, rc3'
[ccollins@ccollins-mac:~/repos/core]$ git push origin --tags
```

Repeat this procedure for each of the following repositories:

- https://github.com/apache/mynewt-blinky
- https://github.com/apache/mynewt-newt
- https://github.com/apache/mynewt-core
- https://github.com/apache/mynewt-newtmgr
- ~~https://github.com/runtimeinc/mynewt_arduino_zero~~
- ~~https://github.com/runtimeinc/mynewt_nordic~~
- ~~https://github.com/runtimeinc/mynewt_stm32f3~~

Step 8: Expose the new release in the repository.yml files

People will need a way to try out the new release while it is being voted on. You should update each repo's *repository.yml* file with a pointer to the corresponding tag. If needed update dependencies tags as well (similar to step 5c)

> A repo's *repository.yml* file is only contained in the master branch; that is the branch you must modify!

Since the release is not approved yet, the old tags should still be used by default. In other words, don't modify any pointers, just add new ones. Here is the git diff showing the changes that were made to the core *repository.yml* file for the 0.9.0 release:

```
 repo.name: apache-mynewt-core
 repo.versions:
     "0.0.0": "develop"
     "0.7.9": "mynewt_0_8_0_b2_tag"
     "0.8.0": "mynewt_0_8_0_tag"
+    "0.9.0": "mynewt_0_9_0_tag"
     "0-latest": "0.8.0"
     "0-dev": "0.0.0"
     "0.8-latest": "0.8.0"
+    "0.9-latest": "0.9.0"
```

Repeat this procedure for each of the repositories:

- https://github.com/apache/mynewt-core
- ~~https://github.com/runtimeinc/mynewt_arduino_zero~~
- ~~https://github.com/runtimeinc/mynewt_nordic~~
- ~~https://github.com/runtimeinc/mynewt_stm32f3~~

Step 9: Create the source release artifacts

Note: To complete this step you will need to publish a GPG code signing key. This something you only have to do once. If you have not done this, see: Co de Signing Keys.

Each source release consists of three artifacts:

1. Gzipped tarball.
2. A SHA512 hash of the gzipped tarball.
3. A detached signature generated by signing the gzipped tarball with your code signing key.

Source release artifacts must be named as follows (file extension excluded):

- apache-*<product>*-*<version>[-b#]*-incubating

The intermediate *-b#* is only included if this is a beta release.

For example, for 0.8.0 beta 2 rc3, the three source artifacts would have the following names:

- apache-mynewt-core-0.8.0-b2-incubating
- apache-mynewt-newt-0.8.0-b2-incubating
- apache-mynewt-blinky-0.8.0-b2-incubating

The following procedure can be used to produce the source artifacts:

1. Grab the *mynewt-archive-src.sh* script (https://github.com/runtimeinc/mynewt_tools/raw/master/mynewt-archive-src.sh) if you don't already have it. Put it somewhere in your PATH, or adjust the following steps as needed.
2. Change to the base directory of the git clone.
3. *mynewt-archive-src.sh* <*destination-dir/artifact-name*> <*tag-name*> (note: you will be prompted for your key password)

The <*tag-name*> is the name of the tag you created in step 3. The <*destination-dir/artifact-name*> field is best illustrated with an example:

```
[ccollins@ccollins-mac:~/repos/core]$ mynewt-archive-src.sh ~/releases/0.8.0-b2/rc3/apache-mynewt-core-0.8.0-b2-incubating mynewt_0_8_0_b2_rc3_tag
```

The above invocation will create the following three files:

```
# Source tarball.
~/releases/0.8.0-b2/rc3/apache-mynewt-core-0.8.0-b2-incubating.tgz

# Detached signature.
~/releases/0.8.0-b2/rc3/apache-mynewt-core-0.8.0-b2-incubating.tgz.asc

# SHA512.
~/releases/0.8.0-b2/rc3/apache-mynewt-core-0.8.0-b2-incubating.tgz.sha
```

Apply this process to each repository (core, newt, newtmgr and blinky). Do not include the rc-number in the artifact names.

Note: You do not need to make a fresh git clone for the above procedure. The *mynewt-archive-src.sh* script uses the *git archive* command, which packages the source files exactly as they were tagged in git.

Step 10: Create the binary release artifacts

Note: To complete this step you will need to publish a GPG code signing key. This something you only have to do once. If you have not done this, see: Code Signing Keys.

Binary releases contain the same artifacts as source releases:

1. Gzipped tarball.
2. A SHA512 hash of the gzipped tarball.
3. A detached signature generated by signing the gzipped tarball with your code signing key.

Binary release artifacts must be named as follows (file extension excluded):

- apache-*<product>*-bin-*<platform>*-*<version>[-b#]*-incubating

The intermediate *-b#* is only included if this is a beta release.

For example, for 0.8.0 beta 2 rc3, the two binary artifacts would have the following names:

- apache-mynewt-newt-bin-osx-0.8.0-b2-incubating
- apache-mynewt-newt-bin-linux-0.8.0-b2-incubating

The following procedure can be used to produce the binary artifacts:

1. Grab the *mynewt-archive-bin.sh* script (https://github.com/runtimeinc/mynewt_tools/raw/master/mynewt-archive-bin.sh) if you don't already have it. Put it somewhere in your PATH, or adjust the following steps as needed.
2. Build the binary from the appropriate source artifact for the appropriate platform (more details TBD).
3. *mynewt-archive-bin.sh <destination-dir/artifact-name> <file-1> [file-2] [file-3] [...]* (note: you will be prompted for your key password)

The file list should consist of the following:

- newt
- LICENSE
- NOTICE
- README.md
- RELEASE_NOTES.md

The *LICENSE*, *NOTI*CE, *README.md*, and *RELEASE_NOTES.md* files should be exactly those from the corresponding source release.

Example:

```
[ccollins@ccollins-mac:~/repos/newt]$ mynewt-archive-bin.sh ~/releases/0.8.0-b2/rc3/apache-mynewt-newt-bin-osx-
0.8.0-b2-incubating newt/newt LICENSE NOTICE README.md RELEASE_NOTES.md
```

The above invocation will create the following three files:

```
# Binary tarball.
~/releases/0.8.0-b2/rc3/apache-mynewt-newt-bin-osx-0.8.0-b2-incubating.tgz

# Detached signature.
~/releases/0.8.0-b2/rc3/apache-mynewt-newt-bin-osx-0.8.0-b2-incubating.tgz.asc

# SHA512.
~/releases/0.8.0-b2/rc3/apache-mynewt-newt-bin-osx-0.8.0-b2-incubating.tgz.sha
```

Apply this process for each platform (osx and linux). Do not include the rc-number in the artifact names.

Step 11: Verify the signatures

The *mynewt-archive.sh* script verifies the signatures after they are created, but you might want to double-check that everything is good. Run the following command for each detached signature (.asc) file:

```
    gpg2 --verify "<asc-file>" "<tgz-file>"
```

The output will indicate that the signature matches the tarball (or not!).

If all your artifacts are in the same directory, you can simplify the process with a script, e.g.,

```
for i in *.asc ; do echo ">>> $i <<<" && gpg2 --verify "$i" "${i%.asc}" ; done
```

Step 12: Upload the artifacts to the release candidate svn server

⚠ Steps 12 and 15 will require checking out mynewt on both `release` and `dev` SVN trees, as well as moving from one directory to another. One way to checkout both mynewt trees is given below:

```
$ svn co --depth immediates https://dist.apache.org/repos/dist/ apache-svn
$ cd apache-svn
$ svn up --set-depth infinity dev/mynewt
$ svn up --set-depth infinity release/mynewt
```

1. Check out the dev/mynewt directory from the svn server:

```
svn co https://dist.apache.org/repos/dist/dev/mynewt
```

2. Make sure the KEYS file is up to date

If you used a new key to sign the release, you need to append the text representation of your key to the *KEYS* file. The *KEYS* file is located in the base mynewt path of the svn server.

```
cat <your-key-file> >> KEYS
```

3. Create a new subdirectory for the release in the mynewt directory

The directory should be named as follows:

- *apache-mynewt-<version>[-b#]-incubating/rc<#>*

```
[ccollins@ccollins-mac:~/repos/dist.apache.org/repos/dist/dev/incubator/mynewt]$ mkdir -p apache-mynewt-0.8.0-b2-incubating/rc3
```

4. Add the build artifacts to the new subdirectory

```
[ccollins@ccollins-mac:~/repos/dist.apache.org/repos/dist/dev/incubator/mynewt]$ cp ~/releases/0.8.0-b2/rc3/* apache-mynewt-0.8.0-b2-incubating/rc3/
[ccollins@ccollins-mac:~/repos/dist.apache.org/repos/dist/dev/incubator/mynewt]$ svn add apache-mynewt-0.8.0-b2-incubating
```

5. Commit

```
[ccollins@ccollins-mac:~/repos/dist.apache.org/repos/dist/dev/incubator/mynewt]$ svn ci
```

Step 13: Send [VOTE] and [DISCUSS] emails to the dev list

Call for a vote for the new release by sending an email to dev@mynewt.apache.org. A separate "[DISCUSS]" thread should also be opened to allow for discussion of the pros and cons of the release. These threads are kept separate so that the vote thread is not polluted with discussion.

[VOTE]

- Template: mynewt-vote-dev.txt (shamelessly adapted from a Taverna vote email)
- Example: http://mail-archives.apache.org/mod_mbox/incubator-mynewt-dev/201602.mbox/%3C20160224012927.GC14480%40iori.nightmare.heaven%3E

[DISCUSS]

- Template: mynewt-discuss-dev.txt
- Example: http://mail-archives.apache.org/mod_mbox/incubator-mynewt-dev/201602.mbox/%3C20160224013028.GD14480%40iori.nightmare.heaven%3E

The vote must remain open for at least 72 hours. You can keep the vote open longer if you wish (e.g., if the necessary number of +1 votes have not been accumulated yet). The vote passes if the following two criteria are met:

- At least three binding +1 votes.
- The total vote tally is >= half the number of binding votes (i.e., majority consensus).

Before sending the email, it is a good idea to test every URL that it references. Broken links won't look good!

Step 14: Send a [RESULT][VOTE] email to the dev list

- Template: mynewt-result-dev.txt
- Example: http://mail-archives.apache.org/mod_mbox/incubator-mynewt-dev/201602.mbox/%3C20160227021125.GO14480%40iori.nightmare.heaven%3E

If the vote doesn't pass, the process ends here. You will need to restart the process after the problems are fixed. If the vote does pass, proceed to next step.

Step 15: Tag the Repositories with release tag

Each git repository must be tagged with the name of the release. Tagging the repositories makes it easy to determine exactly what went into the release. The tag should be named as follows:

- mynewt_<*version-number*>_tag

Tag should be similar to rcX tags but without *_rcX _* part. For example, 0.8.0 beta 2 should have the following tag:

- *mynewt_0_8_0_b2_tag*

Tag the repository with the following git commands:

```
git tag -a <tag-name> -m <commit-message>
git push origin --tags
```

The first command creates a tag in your local repository. The second pushes the tag to the remote server.

Example:

```
[ccollins@ccollins-mac:~/repos/core]$ git tag -a mynewt_0_8_0_b2_tag -m 'Mynewt 0.8.0 B2'
[ccollins@ccollins-mac:~/repos/core]$ git push origin --tags
```

Repeat this procedure for each of the following repositories:

- https://github.com/apache/mynewt-blinky
- https://github.com/apache/mynewt-newt
- https://github.com/apache/mynewt-core
- https://github.com/apache/mynewt-newtmgr
- ~~https://github.com/runtimeinc/mynewt_arduino_zero~~

Step 16: Move the release artifacts to the release directory on the svn server

The release directory is: https://dist.apache.org/repos/dist/release/mynewt

The release subdirectory (and the *KEYS* file, if you changed it) should be moved to the above directory. Be sure to use the `svn mv` command to move the files from dev to release; **don't** make local copies of the dev files and add them separately with `svn add`. If you moved the files properly with `svn mv`, their svn logs should be augmented with a new entry corresponding to the move operation. In other words, svn history should not be lost during the move.

Example:

```
[ccollins@ccollins-mac:~/repos/dist/release/incubator/mynewt]$ mkdir apache-mynewt-1.0.0-b1-incubating
[ccollins@ccollins-mac:~/repos/dist/release/incubator/mynewt]$ svn add apache-mynewt-1.0.0-b1-incubating/
A         apache-mynewt-1.0.0-b1-incubating
[ccollins@ccollins-mac:~/repos/dist/release/incubator/mynewt]$ cd apache-mynewt-1.0.0-b1-incubating
[ccollins@ccollins-mac:~/repos/dist/release/incubator/mynewt/apache-mynewt-apache-mynewt-1.0.0-b1-incubating]$
svn mv ~/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/* .
A         apache-mynewt-blinky-1.0.0-b1-incubating.tgz
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
blinky-1.0.0-b1-incubating.tgz
A         apache-mynewt-blinky-1.0.0-b1-incubating.tgz.asc
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
blinky-1.0.0-b1-incubating.tgz.asc
A         apache-mynewt-blinky-1.0.0-b1-incubating.tgz.sha
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
blinky-1.0.0-b1-incubating.tgz.sha
A         apache-mynewt-core-1.0.0-b1-incubating.tgz
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
core-1.0.0-b1-incubating.tgz
A         apache-mynewt-core-1.0.0-b1-incubating.tgz.asc
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
core-1.0.0-b1-incubating.tgz.asc
A         apache-mynewt-core-1.0.0-b1-incubating.tgz.sha
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
core-1.0.0-b1-incubating.tgz.sha
A         apache-mynewt-newt-1.0.0-b1-incubating.tgz
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
newt-1.0.0-b1-incubating.tgz
A         apache-mynewt-newt-1.0.0-b1-incubating.tgz.asc
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
newt-1.0.0-b1-incubating.tgz.asc
A         apache-mynewt-newt-1.0.0-b1-incubating.tgz.sha
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
newt-1.0.0-b1-incubating.tgz.sha
A         apache-mynewt-newt-bin-linux-1.0.0-b1-incubating.tgz
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
newt-bin-linux-1.0.0-b1-incubating.tgz
A         apache-mynewt-newt-bin-linux-1.0.0-b1-incubating.tgz.asc
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
newt-bin-linux-1.0.0-b1-incubating.tgz.asc
A         apache-mynewt-newt-bin-linux-1.0.0-b1-incubating.tgz.sha
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
newt-bin-linux-1.0.0-b1-incubating.tgz.sha
A         apache-mynewt-newt-bin-osx-1.0.0-b1-incubating.tgz
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
newt-bin-osx-1.0.0-b1-incubating.tgz
A         apache-mynewt-newt-bin-osx-1.0.0-b1-incubating.tgz.asc
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
newt-bin-osx-1.0.0-b1-incubating.tgz.asc
A         apache-mynewt-newt-bin-osx-1.0.0-b1-incubating.tgz.sha
D         /Users/ccollins/repos/dist/dev/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating/rc2/apache-mynewt-
newt-bin-osx-1.0.0-b1-incubating.tgz.sha
[ccollins@ccollins-mac:~/repos/dist/release/incubator/mynewt/apache-mynewt-1.0.0-b1-incubating]$ svn ci
Adding         .
Adding  (bin)  apache-mynewt-blinky-1.0.0-b1-incubating.tgz
Adding         apache-mynewt-blinky-1.0.0-b1-incubating.tgz.asc
Adding         apache-mynewt-blinky-1.0.0-b1-incubating.tgz.sha
Adding  (bin)  apache-mynewt-core-1.0.0-b1-incubating.tgz
Adding         apache-mynewt-core-1.0.0-b1-incubating.tgz.asc
Adding         apache-mynewt-core-1.0.0-b1-incubating.tgz.sha
Adding  (bin)  apache-mynewt-newt-1.0.0-b1-incubating.tgz
Adding         apache-mynewt-newt-1.0.0-b1-incubating.tgz.asc
Adding         apache-mynewt-newt-1.0.0-b1-incubating.tgz.sha
Adding  (bin)  apache-mynewt-newt-bin-linux-1.0.0-b1-incubating.tgz
Adding         apache-mynewt-newt-bin-linux-1.0.0-b1-incubating.tgz.asc
Adding         apache-mynewt-newt-bin-linux-1.0.0-b1-incubating.tgz.sha
Adding  (bin)  apache-mynewt-newt-bin-osx-1.0.0-b1-incubating.tgz
Adding         apache-mynewt-newt-bin-osx-1.0.0-b1-incubating.tgz.asc
Adding         apache-mynewt-newt-bin-osx-1.0.0-b1-incubating.tgz.sha
Committed revision 17368.
```

Step 17: Send an [ANNOUNCE] email to the dev and announce lists

Send a release announcement to the following two recipients using your apache.org address.

- dev@mynewt.apache.org
- announce@apache.org

Some resources for writing this email:

- Template: mynewt-announce.txt
- Example: https://mail-archives.apache.org/mod_mbox/www-announce/201603.mbox/%3C20160306161202.GE7806%40iori.nightmare.heaven%3E

Notes:

- The announce list is moderated; you won't see your email in the archives until it is approved by the moderator.
- Send this email from your apache.org email address (announce@apache.org destination bounces otherwise)

Step 17: Update the default tags in the repository.yml files

Now that the release is out, it should be the version that new users get by default. Update the appropriate labels in each repo's *repository.yml* file. The below git diff shows what was done for the 0.9.0 release:

```
 repo.name: apache-mynewt-core
 repo.versions:
     "0.0.0": "develop"
     "0.7.9": "mynewt_0_8_0_b2_tag"
     "0.8.0": "mynewt_0_8_0_tag"
     "0.9.0": "mynewt_0_9_0_tag"
-    "0-latest": "0.8.0"
+    "0-latest": "0.9.0"
     "0-dev": "0.0.0"
     "0.8-latest": "0.8.0"
     "0.9-latest": "0.9.0"
```

Repeat this procedure for each of the repositories:

- https://github.com/apache/mynewt-core
- ~~https://github.com/runtimeinc/mynewt_arduino_zero~~
- ~~https://github.com/runtimeinc/mynewt_nordic~~
- ~~https://github.com/runtimeinc/mynewt_stm32f3~~

Step 18: Update mynewt website with latest release info

Update https://github.com/apache/mynewt-site with release info. See https://github.com/apache/mynewt-site/pull/528 for example.

Update https://github.com/apache/mynewt-documentation with release info. See https://github.com/apache/mynewt-documentation/pull/105 for example.

Step 19: Update newt and newtmgr version strings to next release -dev

Make sure newt and newtmgr build from master show next release X-dev versions. See points 5) and 5a) for reference.

Step 19a: Update blinky master to point to mynewt-core master

```
@@ -27,6 +27,6 @@ project.repositories:
 #
 repository.apache-mynewt-core:
     type: github
-    vers: 1.11.0
+    vers: 0.0.0
     user: apache
     repo: mynewt-core
```

Step 20: Merge any fixes and updates done on release branches back to master

Make sure that master branch contains all fixes and updates (or equivalents) pushed to release branch.

Step 21: Delete old releases tarballs from dist/release on SVN

dist/release should only contain latest release (https://www.apache.org/legal/release-policy.html#when-to-archive)

If release is combined with NimBLE release delete both apache-mynewt-X.Y.Z and apache-nimble-X.Y.Z releases, if only Mynewt is released remove only apache-mynewt-X.Y.Z

```
[janc@ix mynewt]$ pwd
~/mynewt_release/svn/dist/release/mynewt
 [janc@ix mynewt]$ svn remove apache-mynewt-1.8.0 apache-nimble-1.3.0
D         apache-mynewt-1.8.0
D         apache-mynewt-1.8.0/apache-mynewt-blinky-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-blinky-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-blinky-1.8.0.tgz.sha512
D         apache-mynewt-1.8.0/apache-mynewt-core-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-core-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-core-1.8.0.tgz.sha512
D         apache-mynewt-1.8.0/apache-mynewt-newt-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-newt-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-newt-1.8.0.tgz.sha512
D         apache-mynewt-1.8.0/apache-mynewt-newt-bin-linux-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-newt-bin-linux-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-newt-bin-linux-1.8.0.tgz.sha512
D         apache-mynewt-1.8.0/apache-mynewt-newt-bin-osx-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-newt-bin-osx-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-newt-bin-osx-1.8.0.tgz.sha512
D         apache-mynewt-1.8.0/apache-mynewt-newt-bin-windows-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-newt-bin-windows-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-newt-bin-windows-1.8.0.tgz.sha512
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-1.8.0.tgz.sha512
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-bin-linux-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-bin-linux-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-bin-linux-1.8.0.tgz.sha512
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-bin-osx-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-bin-osx-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-bin-osx-1.8.0.tgz.sha512
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-bin-windows-1.8.0.tgz
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-bin-windows-1.8.0.tgz.asc
D         apache-mynewt-1.8.0/apache-mynewt-newtmgr-bin-windows-1.8.0.tgz.sha512
D         apache-nimble-1.3.0
D         apache-nimble-1.3.0/apache-mynewt-nimble-1.3.0.tgz
D         apache-nimble-1.3.0/apache-mynewt-nimble-1.3.0.tgz.asc
D         apache-nimble-1.3.0/apache-mynewt-nimble-1.3.0.tgz.sha512
[janc@ix mynewt]$ svn ci
D       apache-mynewt-1.8.0
D       apache-nimble-1.3.0
Committing transaction...
Committed revision 46953.
```