# Patch Check List

So, you want to apply a patch? Here are tips, traps, etc. for dealing with patches (in no particular order):

1. Get a fresh copy of trunk. Or at least make sure you are up to date and clean your build area. For complex patches, it is recommended you deal with a fresh checkout.
2. Look at the patch and see where it is applied. Ideally it is generated from the root, but not everyone does this, especially for contrib areas.
3. patch -p 0 -i <path to patch> Throw a --dry-run on there if you want to see what happens w/o screwing up your checkout.
4. Did the author write unit tests? Are the unit tests worthwhile?
5. How are the benchmark results? contrib/benchmarker may be used to test performance in before/after scenarios.
6. Are the licenses correct on newly added files? Has an ASF license been granted?
7. Update CHANGES.txt. Give proper credit to the authors.
8. Make sure you update JIRA by assigning the issue to you so that others know you are working on it.
9. If it is a complex change and you have added to the original author's patch, it is suggested that you create a new patch and attach that to JIRA so that it can be discussed.
10. How's the documentation, esp. the javadocs?
11. Before committing, make sure you add any new documents to SVN. Just b/c the patch added them doesn't mean you have.
12. Run all unit tests, verify all tests pass.
13. Generate javadocs, verify no javadoc errors/warnings were introduced by the patch.
14. Put in a meaningful commit message. Reference the JIRA issue when appropriate.
15. Remember to update the issue in JIRA when you have completed it.
16. From the top directory "ant rat-sources" to make sure all the files have license headers.