Release Certified Application

1. Review
   1. Was the *doxyfile* version updated? This may not apply to C++ libraries following the certified documentation convention where version is not supplied in the *doxyfile*, but in the User’s Guide.
   2. Were release notes updated? This includes the audit trail link.
   3. Was the web landing page (index.html) updated? This is typically not necessary for applications and shared series compatible library updates. Since previous version of libraries (not compatible) are still available for use; their documentation remains pertinent.
   4. Examine outstanding *Jira* issues. Issues addressed with this release should be resolved and mentioned in the release notes. Should other open issues be addressed with this release? Close issues as appropriate. Create new *Jira* project version, or new project if this is first release.
   5. User’s Guide optionally updated. If the release is a non shared series compatible library update, the document must at least have the version number updated. This is because we use *doxygen* to merge User’s Guide and reference information. The reference information will change in this type of update.
   6. Was a test suite provided?
   7. Were temporary files removed from directory tree?
   8. Perform “make –f mkall compact”.
   9. Create and examine difference list. Focus on seeing the source code embedded version number was changed and any debug lines were removed. Examine source code as deemed necessary.
   10. Perform “make –f mkall all”. Watch for build warnings or errors, doxygen warnings, or test suite failures.
2. Install
   1. Update certified software web site. Copy difference list, *doxygen* output, release notes, web landing page (index.html), and any pdf documents as appropriate. For non shared series compatible library updates, previous User’s Guides remain pertinent and will be moved to the “past versions” folder.
   2. Become user *sqam* on a *dev* fiefdom host.
   3. Copy built application or library to “/cs/certified/apps/…” folder. Execute *install.sh*.
   4. If this is a library, there will be a “./lib” folder and possibly a “./cfg” folder that needs to be updated; which will be found at the application’s root level directory. Use the *./lib/relink.sh* script to either create a new shared series or update the previous series to point to a new compatible release.
   5. If this is the first release of an application or the release has a new architecture build, use the /cs/certified/linkPro.sh script to make links in the per architecture folders to the production version of the application. Use the “bin” argument if the application should appear in user’s search path, otherwise use the “libexec” argument if not.
   6. Execute /cs/certified/admin/rsync\_certified to update all fiefdoms.
3. Wrap up
   1. Return to where the new application was built and execute “make –f mkall compact”. Create a “.tar.gz” of the application version and move it to the certified software repository (O:\accsft\certified\backup).
   2. Update the certified web page directory to account for the new release. I use a custom GUI for doing this (/usr/devuser/cjs/bin/certDB).