How to Release a High Level App

HLA Java Release procedures

0) For brand new designs/applications, ensure the following software development cycle is followed:

- Gather Requirements from users / colleagues
 - Write Reqs Document (Insert links to Reqs template)
 - Perform Reqs Review w colleagues / users
 - If GUI is required:
 - Perform GUI Mockup Design
 - Preform GUI Review w engineers / users
 - Write SW Conceptual / Detailed Design (insert links to Design template)
 - Ensure all significant functions have an API
 - If GUI, ensure Model/View/Controller design Pattern
 - Perform Design Review w colleagues
 - · Write Systems Integration Regs, such as host systems and network considerations, database support, matlab PVs, etc,
 - Implement/Code (Insert link to Coding Standards)
 - Follow relevant Design Patterns where applicable
 - Unit Test (test in development)
 - Conduct Code Reviews / demo on development
 - ° Write Help / User Manual / Developer's Guide
 - Write Test Plan
 - $^{\circ}~$ Commission
 - Perform Training

Software Release Procedures

Run eclipse as softegr@lcls-builder on your workspace. After completing the code reviews and implementing the changes, check in the code to CVS from Eclipse.

Update RELEASE_NOTES with latest version tag (create if first time)

Ensure the application has an ANT script that sets the right classpath and builds the application with all the dependencies.(example: See build.xml in any of the High Level Apps)

Ensure the application has a launching script (example: score.bash)

After committing all the changes to CVS, tag the release to the new version published in the RELEASE_NOTES using eclipse.

Check out newly created tagged version in "\$PHYSICS_TOP/release" area

softegr@lcls-builder release > cvs co -r <tagName> -d <tagName> <moduleName>

Example:

softegr@lcls-builder release > cvs co -r score-R0-2-30 -d score-R0-2-30 score

Go to the new folder that was created.

Example: cd score-R0-2-30

Run the ANT script.

Example: ant

Make sure there are no build errors and a new "jar" folder is created.

Verify the <app>.jar is created inside the jar folder.

For example, score-R0-2-30/jar contains score.jar.

Technical and Social Release procedures

Place your tagged release into production following these steps.

Schedule testing time by submitting a Software Mini-Test Plan.

The day you are scheduled to test a new release, make sure you follow instructions to announce your release ahead of time via controls-software-release email.

The day you are scheduled to test you can modify \$PHYSICS_TOP symlinks in order to place your tagged release into production:

Go to \$PHYSICS_TOP and insert or modify symlink to point to release/tagName

In -s release/tagName <Module Name>

example:

\$ In -s release/score-R0-2-30 score

cd to <Module Name> and run the tester launch script.

\$ cd score-R0-2-30

\$./score.bash

Verify that it works!