Jefferson Lab

THOMAS JEFFERSON NATIONAL ACCELERATOR FACILITY

12000 Jefferson Avenue
Newport News, VA 23606

HALL B PROCEDURE NO.:
B000000402 –P005 Rev - 0

TITLE: **Hall B CLAS12 - LabVIEW and PLC Software Storage and Control**

BY: T. Lemon

DATE: 12 /16 /2016

Intended Checker and Approvers:
CHK: N. Sandoval
1. APP: R. Fair
2. APP: 2nd Approver (if necessary)
3. APP: 3rd Approver (if necessary)

REV.	ECO#	DESCRIPTION	BY	CHK.	APP.	APP.	DATE

SUMMARY OF CHANGES FROM PREVIOUS REVISION:

Hall B Magnet GitHub Version Control

All LabVIEW, PLC, and EPICS programs for Hall B Torus and Solenoid will be maintained through GitHub repositories. The GitHub repositories for LabVIEW and PLC are private and can only be accessed by members of the admin group. To be added to the admin group for a repository, users need a GitHub account and should contact the admin group owner to be added. The table below lists repositories and their admin group.

GitHub Repository	System	Admin Group Name
clas12-epics	CLAS12 EPICS	N/A
clas12-plc	PLC for Torus and Solenoid	CLAS12-PLC-Admins
clas12-crio-tor-fast	Torus Fast-Daq	CLAS12-cRIO-Admins
clas12-crio-tor-lv	Torus LV Excitation Chassis	CLAS12-cRIO-Admins
clas12-crio-sol-fast	Solenoid Fast-Daq	CLAS12-cRIO-Admins
clas12-crio-sol-lv	Solenoid LV Excitation Chassis	CLAS12-cRIO-Admins

Advantages of GitHub over JLab M Drive are that GitHub acts as an off-site backup, GitHub allows for better version control, and the repository can easily be accessed. GitHub is maintained external to JLab, allowing repositories to be accessed from any computer at any time via the GitHub desktop app or github.com. GitHub also provides version control by requiring users to commit and document changes to programs.

Users should frequently commit changes to the respective GitHub repository. Programs can be reverted to an old version if new commit causes unforeseen issues. Major development changes should be done in a branch repository that will be merged with the master branch after changes have been tested.

LabVIEW:
Contact: Tyler Lemon

When changes are made to a VI, the modified VI should be committed to the corresponding GitHub repositories. As mentioned above, if major changes are being made, development should occur in a branch repository and be merged with the master repository after changes are tested.

A new release should be generated on GitHub if the program deployed to the cRIO is changed. The most recent release should be the code that is currently running on the cRIO. Version numbering for new releases will follow semantic versioning principles. Release titles should be in the format *vA.B.C*, where A is the major version number, indicating major development changes; B is minor version number, indicating minor, backwards-compatible changes; C is patch number, indicating bug fixes or small improvements. Determining which version number to iterate is left to the discretion of users.

All executables created by the project file should be included when making a new release. The table below details the executables that should be included in a release.

cRIO System	Number of Executables	File Extension of Executables
Torus Fast-Daq	2	*.lvbitx* *.rtexe*
Torus LV Excitation Chassis	1	*.rtexe*
Solenoid Fast-Daq	2	*.lvbitx* *.rtexe*
Solenoid LV Excitation Chassis	2	*.rtexe*

PLC:
Contact: Pablo Campero

The *clas12-plc* repository contains folders for each PLC system. The folders within the repository are detailed in the table below. When changes are made to a PLC program, the project file should be committed to the *clas12-plc* repository in the correct folder.

Users should commit both the *.ACD* file and a *.L5X* file version of a PLC program when committing to GitHub. The *.ACD* file is the main project binary file. The *.L5X* file is an XML representation of the binary file used to generate the PV database for EPICS. In contrast to *.ACD* files, *.L5X* files also give an indicator of any changes made to the program.

Directory Name	Description
CryoDBox	PLC code for the Cryogenic Distribution Box
Docs	Documentation for all PLC systems
Solenoid	PLC code for the Solenoid PLC system
Torus	PLC code for the Torus PLC system

EPICS:
Contact: Wesley Moore

The *clas12-epics* repository houses EPICS applications, drivers, and the operator interfaces for CLAS12. Development is done and changes are committed using "fork-and-pull" strategy. Users should: make a copy (fork) of the repository to their GitHub account and create a separate branch for development where changes will not affect the master branch; develop and make commits to branch in fork; submit a pull request detailing any changes made in comments. Pull requests will be processed and forked repository will be merged with the main branch if there are no conflicts.

Releases are made when EPICS screens are pushed live. A release contains all EPICS applications, drivers, and operator interfaces that will be used.