ALS-U

ADVANCED LIGHT SOURCE

EPICS IOC Development Guide

ALS-U Controls Technical Document

Jeong Han Lee

Document Number: **AL-1451-7629** Revision: **A**

Document Status: Released
Document Type: Note
Category Code: AL7000

TABLE OF CONTENTS

1 REVISION HISTORY

Rev.	CM number	Description of Change
A		Add the standard procedure for IOC

2 APPROVALS

The following individual(s) shall approve this document:

Approver	Project Role
William Wardon	Accelerator Electrical Systems Lead
Windchill Approved / Concurred By	

3 ABBREVIATIONS AND ACRONYMS

ALS	Advanced Light Source
ALS-U	Advanced Light Source Upgrade
LBNL	Lawrence Berkeley National Laboratory
N/A	Non Applicable
EPICS	The Experimental Physics and Industrial Control System
IOC	Input-output controller

4 INTRODUCTION

4.1 Scope

- The purpose of this document is to describe the engineering procedure and troubleshooting about how the EPICS IOC should be developed and be maintained in cooperation with the ALS-U EPICS Environment.
- This document attempts to be a simple guideline, not to be a mandatory procedure.

4.2 Target Audience

This document is targeted to ALS/ALS-U Controls System engineers and technical stakeholders. It is assumed that the target audience has a technical background in the EPICS development, a Unix/Linux environment, and a revision control system, specifically, `git`.

5 BUILD EPICS APPLICATION AND AN IOC

5.1 IOC Name Naming Convention

The first step is to define `IOCNAME`, its directory name, and repository name according to the IOC Name naming conventions [1]. The critical name is Device Name, which can be used in multiple names, such as the repository name, and its EPICS application name. Each engineer has a different preference. Thus, please consult other engineers if one does not sure how these names are defined and one wants to follow a common standard name. Table 5.1 shows the IOC Name Naming example. Here we have two TC-32 devices in difference locations (B46 and B6).

Description	Name	EPICS Variable
Location	TEST, ALSU	
Device Name	TCTEMP	
Common IOC Stats Name	`test-tctemp`	`$IOCNAME`
Full IOC Name (Dir Name)	`ioctest-tctemp`	`$IOC`
Git Repository Name	`tctemp`	
Application Name	`tctemp`	

Table 1 TC-32 IOC Name Naming Example

5.2 Requirements

The `EPICS` environment must be defined. Thus, one must check the `EPICS_BASE` variable and all other EPICS-related environment variables. For example,

```
export EPICS_BASE=/somewhere/epics_base
export EPICS_HOST_ARCH=darwin-aarch64
export PATH=${EPICS_BASE}/bin/${EPICS_HOST_ARCH}:${PATH}
export LD_LIBRARY_PATH=${EPICS_BASE}/lib/${EPICS_HOST_ARCH}:${LD_LIBRARY_PATH}
```

Several packages (screen, git, bash, and make) are essential.

5.3 Structure an EPICS IOC

The script, such as `generate_ioc_structure.bash`, was developed in cooperation with the customized EPICS template to reduce tedious jobs. With the tools repository, one can do the following steps together. It is highly recommended to use this repository to initiate one's IOC structure.

- the consistent `IOCNAME`, its application names and its directory structure through EPICS IOC application structure
- the initial `git` configuration, such as `git init`, `.gitignore`, and `.gitattributes`
- the ALS Gitlab Continuous Integration (CI) [2]
- ALS site specific IOC Deployment scripts and its configuration by using the site-specific EPICS templates

The simple procedure is

- Clone https://git.als.lbl.gov/alsu/tools
- Run `generate_ioc_structure.bash` outside the cloned `tools` folder.

Here is the real example,

```
$ git clone https://git.als.lbl.gov/alsu/tools.git
$ mkdir -p testing
$ cd testing
$ testing $ ../tools/generate_ioc_structure.bash -p tctemp -l alsu -c
../generate_ioc_structure.bash -p tctemp -l alsu -c
Using target architecture darwin-aarch64 (only one available)
hint: Using ’master’ as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:    git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of ’master’ are ’main’, ’trunk’ and
hint: ’development’. The just-created branch can be renamed via this command:
hint:
hint:    git branch -m <name>
Initialized empty Git repository in /Users/JeongLee/gitsrc/tools/testing/tctemp/.git/
$ testing $ tree --charset=ascii
.
‘-- [JeongLee  352]  tctemp
    |-- [JeongLee  900]  Makefile
    |-- [JeongLee   33]  README.md
    |-- [JeongLee  352]  configure
    |   |-- [JeongLee  878]  CONFIG
    |   |-- [JeongLee   61]  CONFIG_IOCSH
    |   |-- [JeongLee 1.6K]  CONFIG_SITE
    |   |-- [JeongLee  157]  Makefile
    |   |-- [JeongLee 2.1K]  RELEASE
    |   |-- [JeongLee  120]  RULES
    |   |-- [JeongLee   39]  RULES.ioc
    |   |-- [JeongLee   41]  RULES_DIRS
    |   ‘-- [JeongLee   40]  RULES_TOP
    |-- [JeongLee  128]  iocBoot
    |   |-- [JeongLee  121]  Makefile
    |   ‘-- [JeongLee  288]  iocalsu-tctemp
    |       |-- [JeongLee  124]  Makefile
    |       |-- [JeongLee   84]  attach
    |       |-- [JeongLee   65]  run
    |       |-- [JeongLee   68]  rund
    |       |-- [JeongLee  192]  screenrc
    |       |-- [JeongLee 1.8K]  st.cmd
    |       ‘-- [JeongLee   73]  st.screen
    ‘-- [JeongLee  192]  tctempApp
        |-- [JeongLee   96]  Db
        |   ‘-- [JeongLee 1.1K]  Makefile
        |-- [JeongLee  363]  Makefile
        |-- [JeongLee  128]  iocsh
        |   |-- [JeongLee  155]  Makefile
        |   ‘-- [JeongLee  172]  tctemp.iocsh
        ‘-- [JeongLee  128]  src
            |-- [JeongLee 2.6K]  Makefile
            ‘-- [JeongLee  402]  tctempMain.cpp
8 directories, 25 files.
$ testing $ ls -a tctemp/
.     .git     .gitignore  Makefile configure   tctempApp
..    .gitattributes .gitlab-ci.yml README.md   iocBoot
$ testing $ make -C tctemp
$ testing $ cd tctemp/iocBoot/iocalsu-tctemp/
$ iocalsu-tctemp (master)$ tree --charset=ascii
.
|-- [JeongLee  124]  Makefile
|-- [JeongLee   84]  attach
|-- [JeongLee  261]  envPaths
|-- [JeongLee   65]  run
|-- [JeongLee   68]  rund
|-- [JeongLee 1.7K]  screenlog.0
|-- [JeongLee  192]  screenrc
|-- [JeongLee 1.8K]  st.cmd
‘-- [JeongLee   73]  st.screen
0 directories, 9 files
$
$ iocalsu-tctemp (master)$ ./run
$ iocalsu-tctemp (master)$ ./attach
```

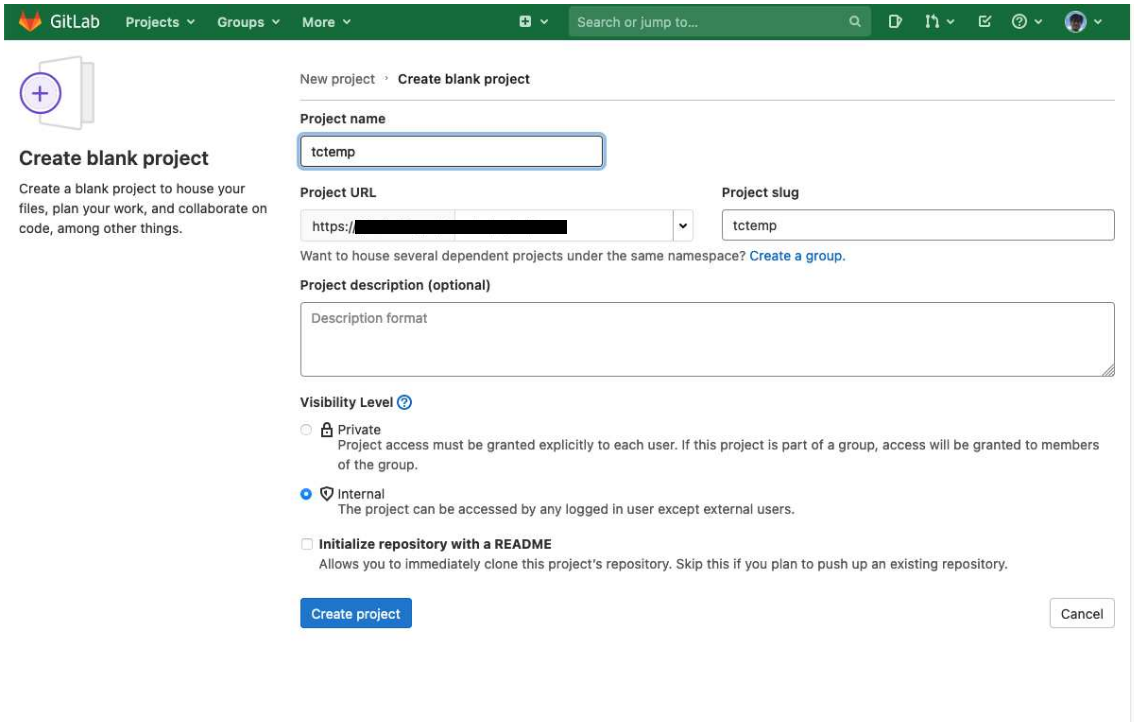
One can see many unusual files in `iocalsu-tctemp`, such as `attach`, `run`, `rund`, etc. These files are used to deploy the IOC within the ALS EPICS Environment. Reference [3] shows its deployment guide.

6 A REMOTE REPOSITORY

There are many ways in which we can create a repository, but here we limit our scenario to create a repository through the web interface.

- Login the gitlab server
- Move the proper IOCs directory
- Create blank project
- Initialize the git repository according to its configuration
- Customize codes to match IOC requirements

6.1 Create a remote repository



Create blank project

Create a blank project to house your files, plan your work, and collaborate on code, among other things.

Project name: tctemp

Project URL: https://[redacted]

Project slug: tctemp

Want to house several dependent projects under the same namespace? Create a group.

Project description (optional): Description format

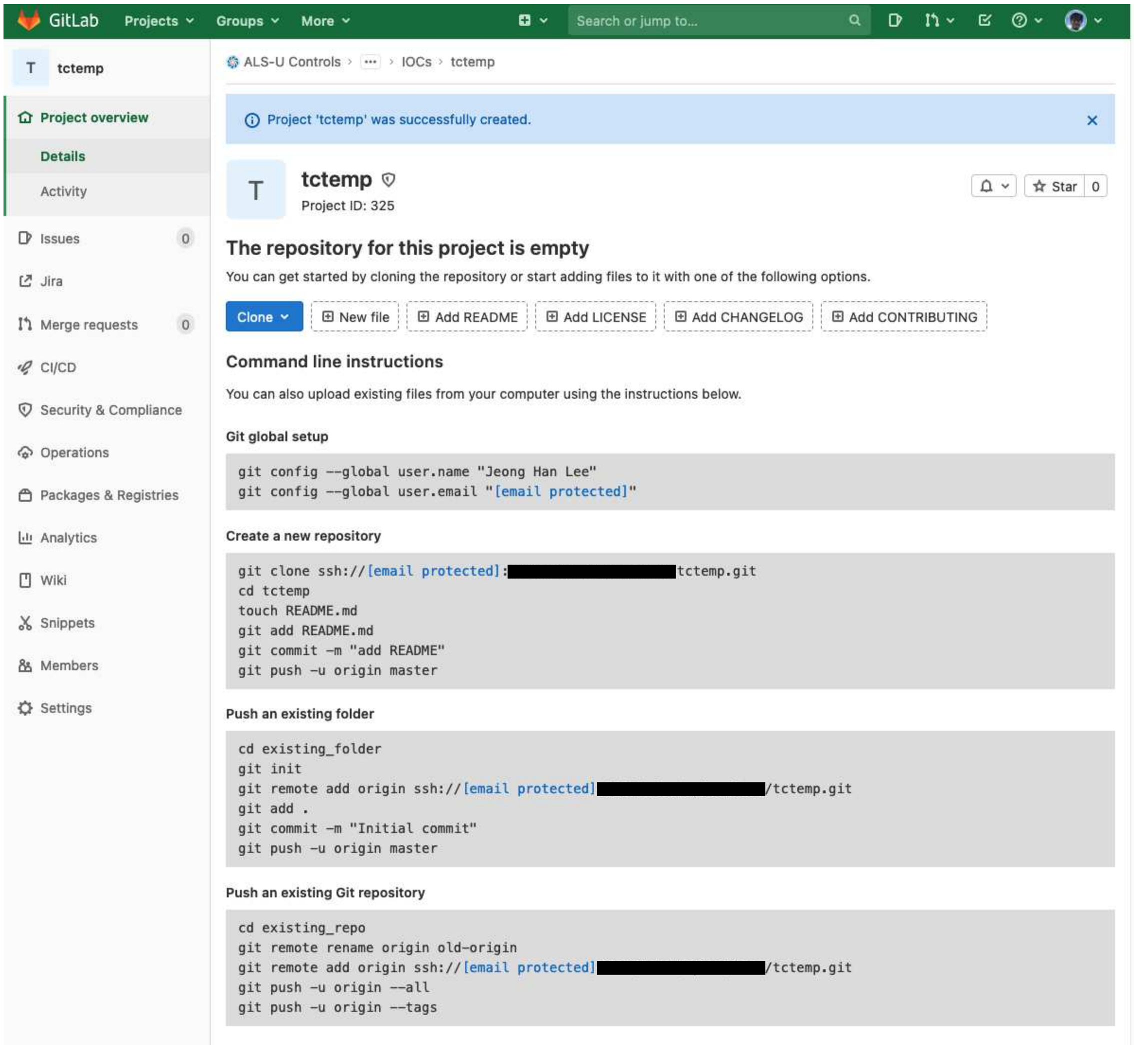
Visibility Level

○ Private
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.

● Internal
The project can be accessed by any logged in user except external users.

□ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Figure 1 Create a Project in the ALS GitLab repository.



Project 'tctemp' was successfully created.

tctemp

Project ID: 325

The repository for this project is empty

You can get started by cloning the repository or start adding files to it with one of the following options.

Command line instructions

You can also upload existing files from your computer using the instructions below.

Git global setup

```
git config --global user.name "Jeong Han Lee"
git config --global user.email "[email protected]"
```

Create a new repository

```
git clone ssh://[email protected]:[redacted]tctemp.git
cd tctemp
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master
```

Push an existing folder

```
cd existing_folder
git init
git remote add origin ssh://[email protected][redacted]/tctemp.git
git add .
git commit -m "Initial commit"
git push -u origin master
```

Push an existing Git repository

```
cd existing_repo
git remote rename origin old-origin
git remote add origin ssh://[email protected][redacted]/tctemp.git
git push -u origin --all
git push -u origin --tags
```

Figure 2 Project `git` configuration.

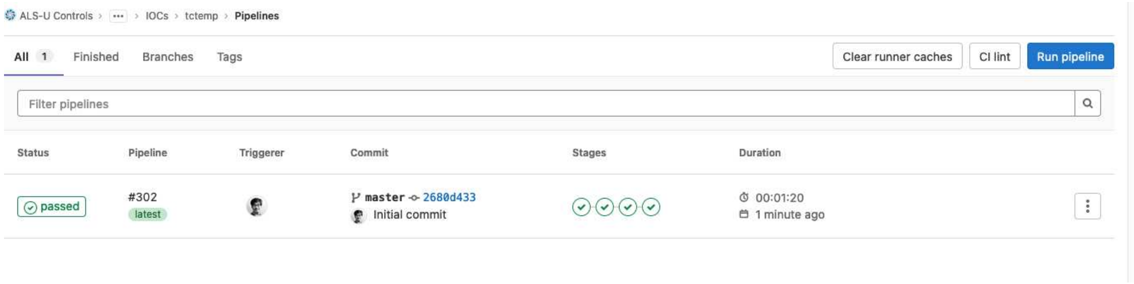
6.2 Push local source files to the remote repository

```
testing $ cd tctemp/
tctemp (master)$ git add .
tctemp (master)$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
```

```
   new file:   .gitattributes
   new file:   .gitignore
   new file:   .gitlab-ci.yml
   new file:   Makefile
   new file:   README.md
   new file:   configure/CONFIG
   new file:   configure/CONFIG_IOCSH
   new file:   configure/CONFIG_SITE
   new file:   configure/Makefile
   new file:   configure/RELEASE
   new file:   configure/RULES
   new file:   configure/RULES.ioc
   new file:   configure/RULES_DIRS
   new file:   configure/RULES_TOP
   new file:   iocBoot/Makefile
   new file:   iocBoot/iocalsu-tctemp/Makefile
   new file:   iocBoot/iocalsu-tctemp/attach
   new file:   iocBoot/iocalsu-tctemp/run
   new file:   iocBoot/iocalsu-tctemp/rund
   new file:   iocBoot/iocalsu-tctemp/screenrc
   new file:   iocBoot/iocalsu-tctemp/st.cmd
   new file:   iocBoot/iocalsu-tctemp/st.screen
   new file:   tctempApp/Db/Makefile
   new file:   tctempApp/Makefile
   new file:   tctempApp/iocsh/Makefile
   new file:   tctempApp/iocsh/tctemp.iocsh
   new file:   tctempApp/src/Makefile
   new file:   tctempApp/src/tctempMain.cpp
tctemp (master)$ git remote
tctemp (master)$ git remote add origin ssh://git@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx/tctemp.git
tctemp (master)$ git remote
origin
tctemp (master)$ git commit -m "Initial Commit"
tctemp (master)$ git push -u origin master
```

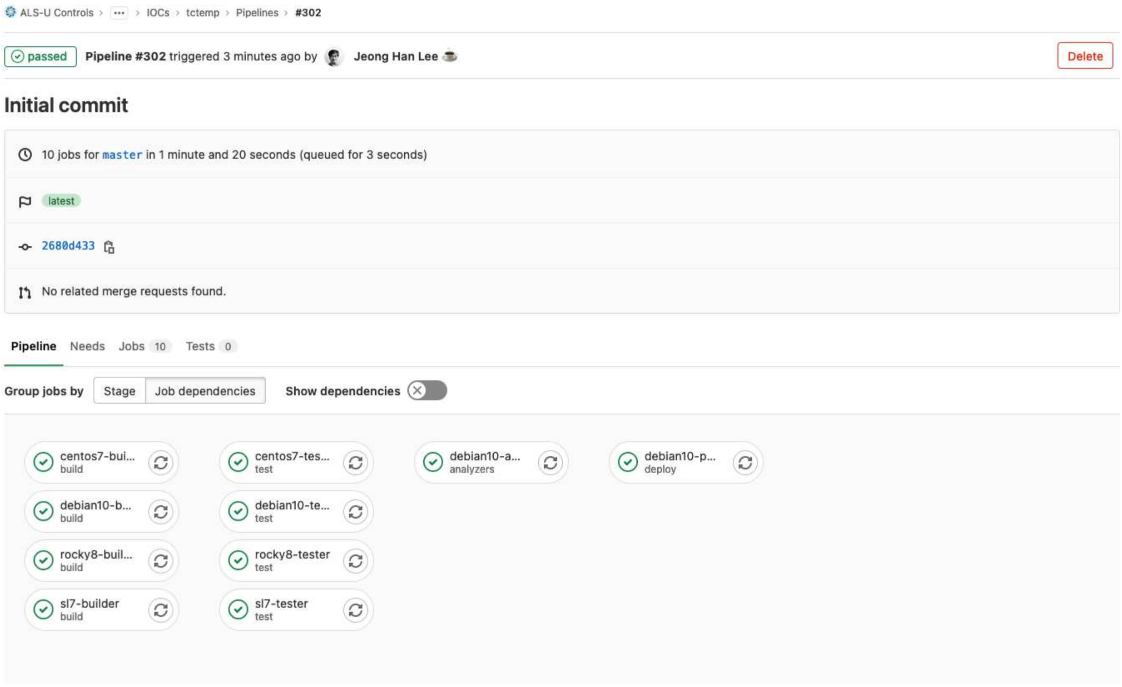
6.3 GitLab CI/CD

The `git push` activity automatically triggers the GitLab CI/CD Pipelines. One can check the default CI configuration through the generated `.gitlab-ci.yml` file. Figure 3 and 4 show the default CI/CD status when all files are located in a remote repository.



Status	Pipeline	Triggerer	Commit	Stages	Duration
passed	#302 latest		master -o- 2680d433 Initial commit	✓ ✓ ✓ ✓	00:01:20 1 minute ago

Figure 3 GitLab CI/CD Pipelines status after the first `git push`.



passed Pipeline #302 triggered 3 minutes ago by Jeong Han Lee

Initial commit

10 jobs for master in 1 minute and 20 seconds (queued for 3 seconds)

latest

2680d433

No related merge requests found.

Figure 4 GitLab Pipeline on the specific detailed ID status after the first `git push`.

6.4 Customization

One can develop an EPICS IOC and its application within pre-defined structure. Typically, one should do the following procedures.

- Edit `configure/CONFIG_SITE` if necessary
- Edit `configure/RELEASE` if necessary
- Add the proper database files into `xxxApp/Db`, and edit `Makefile` in `xxxApp/Db`
- Add the additional source files, sequencer files into `xxxApp/src` if necessary
- Edit `Makefile` into `xxxApp/src`

This template allows users to add necessary `db`, `dbd`, and its corresponding librariesi easily. Moreover, one can add its own local `iocsh` file into the EPICS Application. Please see `Makefile` in `xxxApp/iocsh`.

7 ADD ANOTHER IOC TO THE EXISTING EPICS APPLICATION

In case, one wants to add another IOC into the existing EPICS application, with the different `LOCATION name`, one can create the seperated `iocBoot` directory, and its assicioated files within that directory. After finishing local works, one can add them into `git` repository locally and remotely.

```
testing (master)$ ../generate_ioc_structure.bash -p tctemp -l test
$
testing (master)$ tree --charset=ascii
.
‘-- [JeongLee  448]  tctemp
    |-- [JeongLee  900]  Makefile
    |-- [JeongLee   33]  README.md
    |-- [JeongLee  352]  configure
    |   |-- [JeongLee  878]  CONFIG
    |   |-- [JeongLee   61]  CONFIG_IOCSH
    |   |-- [JeongLee 1.6K]  CONFIG_SITE
    |   |-- [JeongLee  157]  Makefile
    |   |-- [JeongLee 2.0K]  RELEASE
    |   |-- [JeongLee  120]  RULES
    |   |-- [JeongLee   39]  RULES.ioc
    |   |-- [JeongLee   41]  RULES_DIRS
    |   ‘-- [JeongLee   40]  RULES_TOP
    |-- [JeongLee  160]  iocBoot
    |   |-- [JeongLee  121]  Makefile
    |   |-- [JeongLee  288]  iocalus-tctemp
    |   |   |-- [JeongLee  124]  Makefile
    |   |   |-- [JeongLee   84]  attach
    |   |   |-- [JeongLee   65]  run
    |   |   |-- [JeongLee   68]  rund
    |   |   |-- [JeongLee  192]  screenrc
    |   |   |-- [JeongLee 1.0K]  st.cmd
    |   |   `-- [JeongLee   73]  st.screen
    |   `-- [JeongLee  288]  ioctest-tctemp
    |       |-- [JeongLee  124]  Makefile
    |       |-- [JeongLee   84]  attach
    |       |-- [JeongLee   65]  run
    |       |-- [JeongLee   68]  rund
    |       |-- [JeongLee  192]  screenrc
    |       |-- [JeongLee 1.0K]  st.cmd
    |       `-- [JeongLee   73]  st.screen
    `-- [JeongLee  192]  tctempApp
        |-- [JeongLee   96]  Db
        |   `-- [JeongLee  901]  Makefile
        |-- [JeongLee  363]  Makefile
        |-- [JeongLee  128]  iocsh
        |   |-- [JeongLee  102]  Makefile
        |   `-- [JeongLee  172]  tctemp.iocsh
        `-- [JeongLee  128]  src
            |-- [JeongLee 2.0K]  Makefile
            `-- [JeongLee  402]  tctempMain.cpp

testing (master)$ diff tctemp/iocBoot/iocalus-tctemp/st.cmd tctemp/iocBoot/ioctest-tctemp/st.cmd
16,17c16,17
< epicsEnvSet("IOCNAME", "alus-tctemp")
< epicsEnvSet("IOC", "iocalus-tctemp")
---
> epicsEnvSet("IOCNAME", "test-tctemp")
> epicsEnvSet("IOC", "ioctest-tctemp")

tctemp (master)$ git add iocBoot/ioctest-tctemp/
tctemp (master)$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
   new file:   iocBoot/ioctest-tctemp/Makefile
   new file:   iocBoot/ioctest-tctemp/attach
   new file:   iocBoot/ioctest-tctemp/run
   new file:   iocBoot/ioctest-tctemp/rund
   new file:   iocBoot/ioctest-tctemp/screenrc
   new file:   iocBoot/ioctest-tctemp/st.cmd
   new file:   iocBoot/ioctest-tctemp/st.screen

tctemp (master)$ git commit -m "add ioctest"
[master 669ad72] add ioctest
 7 files changed, 70 insertions(+)
 create mode 100644 iocBoot/ioctest-tctemp/Makefile
 create mode 100755 iocBoot/ioctest-tctemp/attach
 create mode 100755 iocBoot/ioctest-tctemp/run
 create mode 100755 iocBoot/ioctest-tctemp/rund
 create mode 100644 iocBoot/ioctest-tctemp/screenrc
 create mode 100755 iocBoot/ioctest-tctemp/st.cmd
 create mode 100755 iocBoot/ioctest-tctemp/st.screen

tctemp (master)$ git push
```

A MAKE BASE APPLICATION: MANUAL PROCEDURE

- Create a directory, e.g., `tctemp`, and change the current path to `tctemp`.
- Run `makeBaseApp.pl` to create the EPICS application.
- Run `makeBaseApp.pl` to add the IOC `test` into the created EPICS application.

```
$
$ echo ${EPICS_BASE}
/Users/JeongLee/epics/macOS/11.2.1/e881cb1/base

$
/Users/JeongLee/epics/macOS/11.2.1/e881cb1/base/bin/darwin-aarch64/makeBaseApp.pl

$ mkdir tctemp
$ cd tctemp

$ makeBaseApp.pl -t ioc tctemp
$ tree -L 1
.
+---[JeongLee  900]  Makefile
+---[JeongLee  320]  configure
+---[JeongLee  160]  tctempApp

$ makeBaseApp.pl -i -t ioc -p tctemp test-tctemp
$ tree -L 2
.
+--- [JeongLee  900]  Makefile
+--- [JeongLee  320]  configure
|     +-- [JeongLee  838]  CONFIG
|     +-- [JeongLee 1.6K]  CONFIG_SITE
|     +-- [JeongLee  157]  Makefile
|     +-- [JeongLee 1.6K]  RELEASE
|     +-- [JeongLee  120]  RULES
|     +-- [JeongLee   39]  RULES.ioc
|     +-- [JeongLee   41]  RULES_DIRS
|     +-- [JeongLee   40]  RULES_TOP
+-- [JeongLee  128]  iocBoot
|     +-- [JeongLee  121]  Makefile
|     +-- [JeongLee  128]  ioctest-tctemp
+-- [JeongLee  160]  tctempApp
      +-- [JeongLee   96]  Db
      +-- [JeongLee  304]  Makefile
      +-- [JeongLee  128]  src
```

BIBLIOGRAPHY

[1] Jeong Han Lee and Tyna Ford. *AL-1451-7452 : IOC Name Naming Convention at ALS*, June, 2021. ALS-U Document AL-1451-7452.

[2] Jeong Han Lee. *ALS GitLab CI Templates*, 2021 (accessed June 29, 2021). https://git.als.lbl.gov/accelerator-controls/environment/ci.

[3] Jeong Han Lee. *AL-1453-7006 : EPICS IOC Deployment Guide*, June, 2021. ALS-U Document AL-1453-7006.