./figs/alsulogo.eps

EPICS IOC Reference Guide

ALS-U Controls Technical Document

Sangil Lee

Document Number: **AL-xxxx-xxxx** Revision: **A**

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

TABLE OF CONTENTS

1 REVISION HISTORY

Rev.	**CM number**	**Description of Change**
A		Add reference note for IOC development

2 APPROVALS

The following individual(s) shall approve this document:

Approver	**Project Role**
William Wardon	Accelerator Electrical Systems Lead
Carlos Serrano Pareja	Controls and Instrumentation CAM
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 organize the references necessary for EPICS IOC development.

- 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 RECOMMEND NETWORK CONFIGURATION

The EPICS provides a framework to integrate a lot of distributed control components. The main point of the distributed control system is to config the efficient and flexible network.

5.1 Dividing EPICS Network and Local Network

Many equipment devices support for UDP/TCP/IP protocol to control them. Therefore, it is good to separate the network for controlling field I/O and the network for EPICS IOC. Figure 1 shows that the network configuration is divided into an EPICS network and a local network based on the IOC. EPICS IOC server through two NIC interfaces isolates each other. A network switch connected to the IOC server uses an L2 switch. This can prevent packet looping errors caused by the L3 switch even if the network is incorrectly configured. This configuration also minimizes IP address list management based on IOC.

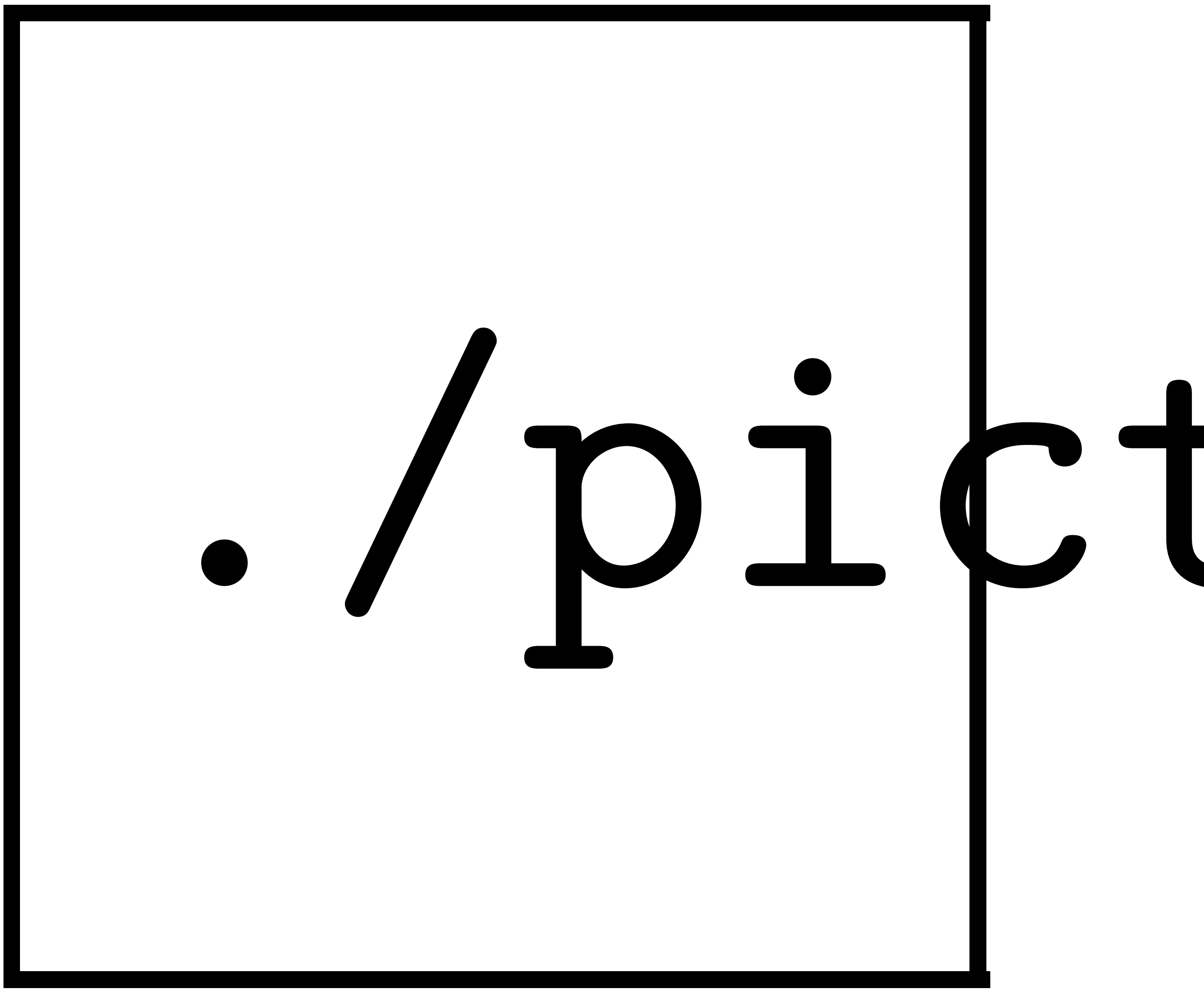

Figure 1 Dividing Network Configuration by IOC.

6 HIERARCHICAL LOGIC-PARTITIONING

For the configuration of the integrated control system of large and distributed facility, it is necessary to define the location of the control logic. Figure 2 shows three logic configuration with three layers. By setting the position of the appropriate logic partition according to the layer composition, the actual control software operation and configuration management are carried out.

Layer	Name	EPICS Variable
Location	TEST, ALSU	
Device Name	TCMD	
Common IOC Stats Name	`ctrlslab-tcmd`	`$IOCNAME`
Full IOC Name	`iocctrlslab-tcmd`	`$IOC`
Application Name	`tcmd`	

Table 1 TC-32 IOC Name Naming Example

6.1 Layer 1 - EPICS IOC Logic related field I/O

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 6 shows the IOC Name Naming example. Here we have two TC-32 devices in difference locations (B46 and B6).

6.1.1 Requirements

The `EPICS` environment must be defined. Thus, one must check the `EPICS_BASE` variable and all other EPICS-related environment variables. For example, in the ALS-U Standard EPICS environment, one can run the specific version of an EPICS enviornment via

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

6.2 First higher EPICS IOC logic

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.

./pictures/logic_partition_layer.eps

Figure 2 Logic Partitioning by Layer 3

- the consistent `IOCNAME`, its application name 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` to be prepared.
- Create a working folder where you can keep all repositories, which you want to generate. This folder should not be in the 'tools' folder.
- Go that working folder, and call `generate_ioc_structure.bash`.

6.3 Second higher EPICS IOC logic

6.4 Top level application logic

- Python
- Matlab / Octave
- Go / Rust
- Shell(Bash, csh, zsh)

6.5 OPI logic - Phoebus User Interface

Here is the real example,

One can see many unusual files in `iocctrlslab-tcmd`, 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.

7 ALARM

8 RECOMMENDED TIPS FOR OPTIMIZED IOC

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

8.1 Create a remote repository

./pictures/gitlab_create_project.eps

Figure 3 Create a Project in the ALS GitLab repository.

./pictures/gitlab_config_project.eps

Figure 4 Project `git` configuration.

8.2 Push local source files to the remote repository

8.3 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 5 and 6 show the default CI/CD status when all files are located in a remote repository.

./pictures/gitlab_ci_pipelines01.eps

Figure 5 GitLab CI/CD Pipelines status after the first `git push`.

./pictures/gitlab_ci_pipelines02.eps

Figure 6 GitLab Pipeline on the specific detailed ID status after the first `git push`.

8.4 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 libraries easily. Moreover, one can add its own local `iocsh` file into the EPICS Application. Please see `Makefile` in `xxxApp/iocsh`.

9 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 different `iocBoot` directory, and its associated files within that directory. One should run `generated_ioc_structure.bash` within a parent directory which holds a repository which one wants to add another IOC. For example, the following example shows that the location will be `cloning_folder` directory. After finishing local works, one can add them into `git` repository locally and remotely.

10 SOFTWARE REQUIREMENT DOCUMENT

11 PRACTICAL GUIDELINE

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.

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.