ALS-U EPICS Environment Quick Reference

EPICS-env 1.5.0 | EPICS Base 7.0.10 | PVXS / QSRV2 | Local Loopback Exercises

ALS-U Controls Training | Ref: jeonghanlee.github.io/epics-trainings

SINGLE-PAGE QUICK REFERENCE

Author: Jeong Han Lee, Dr.rer.nat

Core Session and Loopback Rules

1. Activate one matching OS tree per shell: source "$TREE/setEpicsEnv.bash" (TREE=<dist>/1.5.0/<os-id>/7.0.10).
2. Pin CA and PVA clients to local loopback (127.0.0.1) with AUTO_ADDR_LIST=NO in every client terminal.
3. Run one active IOC per PV prefix and TCP port (9399-9401); stop periodic TCP emulators before exit at iocsh.

1. Environment Activation and Loopback Discovery

Setting / Check	Shell Command	Role in Exercises
Identify OS ID	. /etc/os-release; echo "${ID}-${VERSION_ID}"	Match <os-id> directory (for example debian-13, rocky-8.10)
Activate tree	source "$TREE/setEpicsEnv.bash"	Set PATH, EPICS_BASE, and EPICS_MODULES in current shell
CA loopback	export EPICS_CA_AUTO_ADDR_LIST=NO EPICS_CA_ADDR_LIST=127.0.0.1	Send Channel Access UDP searches only to 127.0.0.1
PVA loopback	export EPICS_PVA_AUTO_ADDR_LIST=NO EPICS_PVA_ADDR_LIST=127.0.0.1	Send pvAccess UDP searches only to 127.0.0.1
Simulator ports	ss -ltn '(sport >= :9399 and sport <= :9401)'	Verify exercise TCP ports 9399-9401 have no listener

2. Module Loader (iocsh.bash) CLI Options and Startup Directives

Invocation / Directive	Long Form or Scope	Behavior
iocsh.bash st.cmd	(default interactive)	Resolve modules, generate /dev/fd/3 + /dev/fd/4, start softIocPVX
iocsh.bash -n st.cmd	--show	Print generated startup (dlload, dbd, macros) and exit without IOC
iocsh.bash -S st.cmd	--non-interactive	Run startup with the interactive iocsh prompt disabled
iocsh.bash -v st.cmd	--verbose	Echo startup commands as softIocPVX executes them
iocsh.bash -e "$TREE" st.cmd	--environment "$TREE"	Source "$TREE/setEpicsEnv.bash" inside loader process only
iocsh.bash -h	--help	Print loader command usage summary and exit
module <Name> [<Ver>]	Top-level st.cmd only	Load module + transitive deps; export <MACRO> from cfg/iocsh.conf
dbLoadGroup("<f>.json", "<M>")	Before iocInit()	Register QSRV2 structured PVA group PV (built into softIocPVX)

3. IOC Shell (iocsh) Runtime Inspection and Control Commands

IOC Shell Command	Provider	Inspection or Control Action
dbl / dbl("<recordType>")	EPICS Base	List loaded record names (all records or one record type)
dbpr("<pv>", <level>)	EPICS Base	Print record fields at detail level 0-4 (for example dbpr("PV", 2))
asynReport(<level>, "<port>")	asyn	Show port connection state and counters (omit port for all drivers)
asynSetTraceMask(<port>, -1, ERROR|FLOW|DRIVER)	asyn	Enable driver and flow diagnostic messages on an asyn port
pvxgl(<level>, "<pattern>")	PVXS / QSRV2	List QSRV2 group PVs and atomic members (for example pvxgl(1, "*"))
reGrep("<regex>")	retools	List loaded record names matching a regular expression
exit	EPICS Base	Stop the IOC cleanly and return to the host Bash prompt

4. Channel Access (CA) vs pvAccess (PVA) Client Commands

Operation	CA Client (Base)	PVA Client (PVXS)	Usage Notes
Read value	caget <pv>	pvxget <pv>	pvxget prints Normative Type / group fields + alarm + time
Confirmed put	caput -c <pv> <val>	pvxput -r "record[block=true]" <pv> <val>	Waits for async completion; plain pvxput writes non-blocking
Monitor	camonitor -t sI <pv>	pvxmonitor <pv>	-t sI (capital I) prints incremental seconds; Ctrl+C stops
Inspect type	caget -d CLASS_NAME <pv>	pvxinfo <pv>	CLASS_NAME prints record type; pvxinfo prints PVA layout
Check alarm	caget <pv>.STAT <pv>.SEVR	pvxget <pv>	Confirm NO_ALARM; COMM / INVALID marks disconnected device

DIAGNOSTIC REMINDERS: Place module directives only in top-level st.cmd (never inside iocshLoad fragments).

On runtime errors, /dev/fd/4 line numbers match st.cmd directly; /dev/fd/3 holds generated loader commands.

With commonIocsh autosave.iocsh, wait past SETTINGS_PERIOD (default 5 s) before stopping an IOC to persist settings.