OAK RIDGE National Laboratory

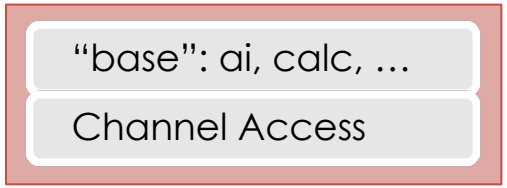
‘makeBaseApp’ IOC

Kay Kasemir

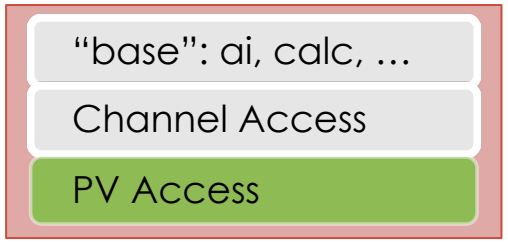
Feb. 2022

U.S. DEPARTMENT OF ENERGY

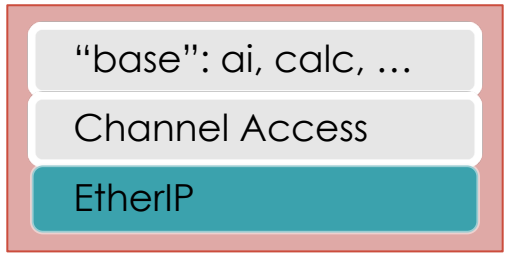
IOC binaries



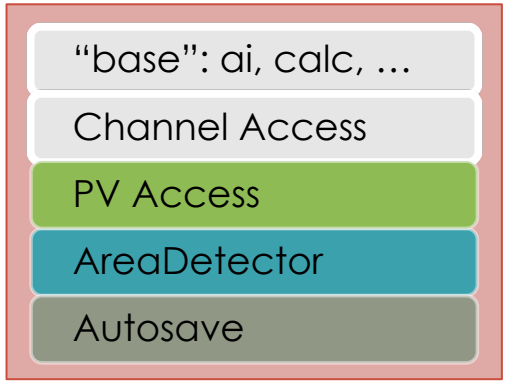
- softIoc
  - Executes *.db files with ai, calc, …
- softIocPVA
  - Adds PVAccess server

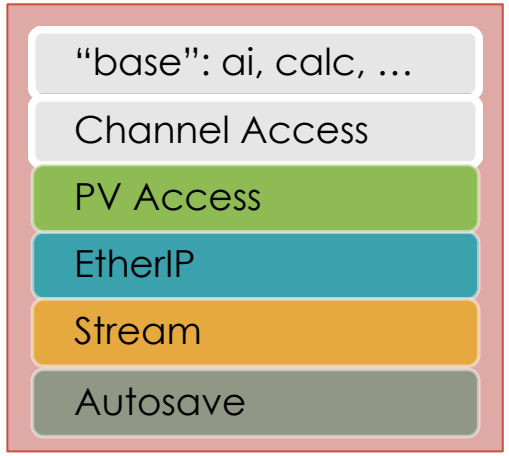


- eipIoc
  - Understands 'field(DTYP, "EtherIP")'



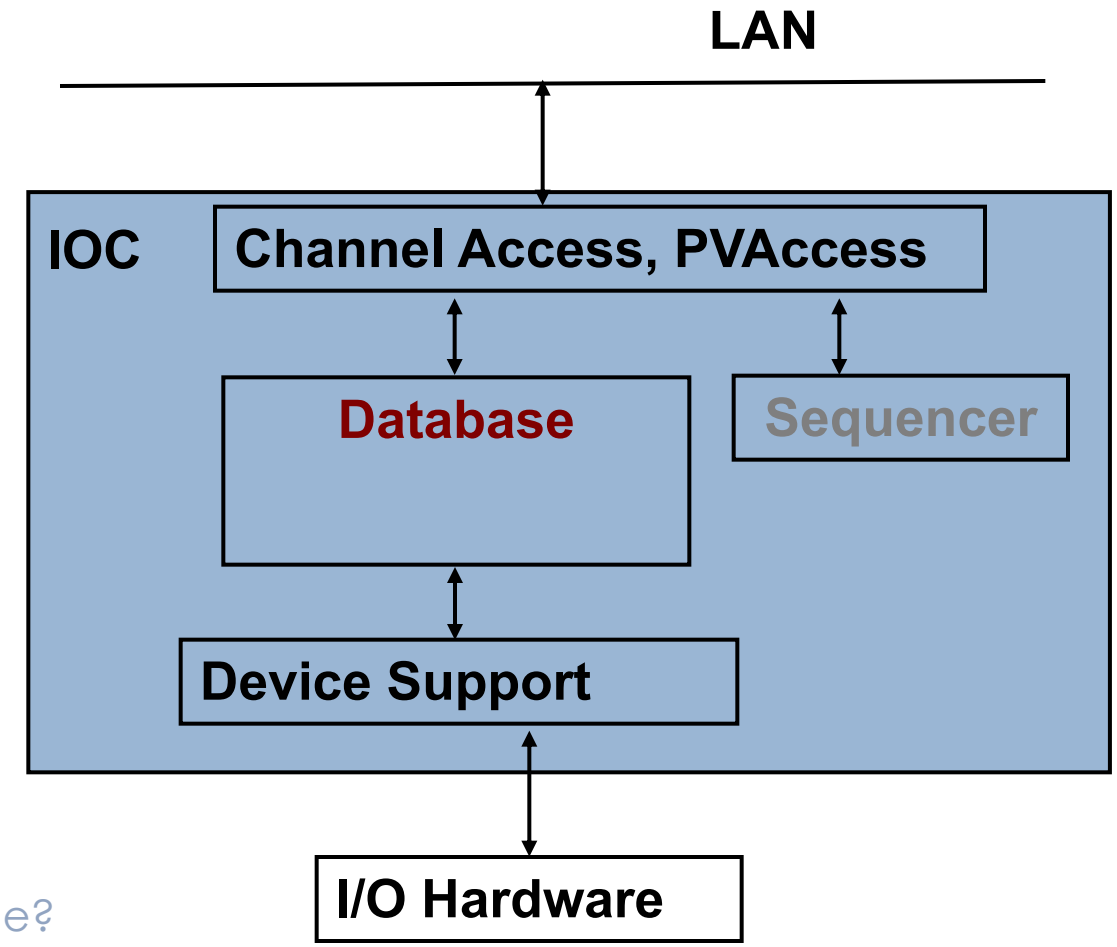
What if I want 'myCustomIoc' that includes PVA, EtherIP, Autosave, Stream Device, …?





EPICS IOC

- Database
  - Records
  - Remote access
  - Access security
- Sequencer
  - C code for state machine
- Device Support
  - Include existing device support?
    Easy enough
  - Have to write new device (driver) code?
    Varying degrees of difficulty

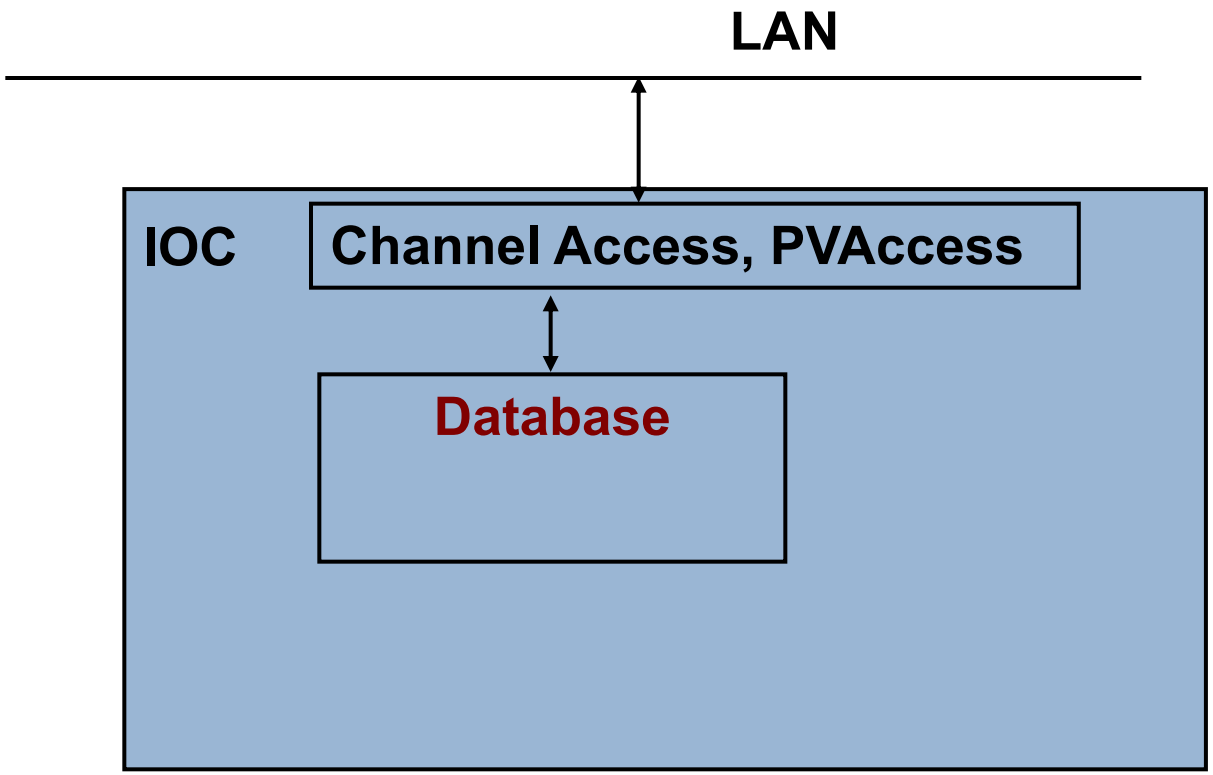


3

softIoc, softIocPVA

IOC with
Database engine,
Channel Access,
opt. PVA.

Comes with EPICS.



Need autosave, sequencer, device support?

→ Create your own IOC application binary!

‘Host’ vs. ‘Target’ IOCs

- Host-based, aka “soft” IOC
  - Runs on same type of host (Linux, Mac, Windows) on which it’s compiled
  - IOC is just another program on the host
  - May run many IOCs on the same host
  - Examples: `softIoc`, `softIocPVA`
- Target IOC
  - Cross-compiled from e.g. Linux to VxWorks
  - Runs on VxWorks, RTEMS
  - IOC is the primary, maybe only program running on the target

A lot of EPICS code can be used on both

- Records
- Device support for networked I/O

How many custom IOC binaries?

Each IOC may use its own IOC binary

- More work?
- Allows updating binary for IOC 1 while IOCs 2-99 keep running

Or define common needs

- Vacuum IOC: Autosave, Support for AllenBradley PLC
- LLRF IOC: Autosave, Support for LLRF hardware
- Camera IOC: Autosave, AreaDetector
- Sample environment IOC: Autosave, Motor Record, Stream Device

… and update/restart all the vacuum IOCs together when changes are needed

‘makeBaseApp.pl’

Creates skeleton for custom IOC

- Directory structure
- Makefiles
- Examples: *.db, *.st, driver/device/record *.c
- IOC startup file

Two extremes

- makeBaseApp.pl –t example
  - Get most everything; you delete what’s not needed
- makeBaseApp.pl –t ioc
  - Just dirs & Makefiles; you add what’s needed

EPICS Build Facility

Is outstanding

- make, perl
- Builds on Linux, Mac, Windows, for Linux, FreeBSD, OS X, Windows, vxWorks, RTEMS, x86, x86_64, ppc, arm, …
- AppDevGuide
- Functioned for decades across many changes of OSs, compilers, …

Is aggravating

- Why is it not a Visual C++, Kdeveloper, VSCode, … project? What about CMake, GNU automake, … ?
- What's the name of that option again?
- What's causing this error now?

‘demo’ based on ‘example’ template

```
# Go somewhere
mkdir -p /ics/mine
cd /ics/mine


# Create IOC application of type ‘example’,
# using ‘demo’ in the generated names
makeBaseApp.pl -t example demo


# Create IOC startup settings of type ‘example’,
# call it ‘demo’ because it’s for the app of that name
makeBaseApp.pl -t example -i demo
# When prompted, use the previously created ‘demo’
# application as the one that the IOC should load

# Compile everything
make

# Start IOC
cd iocBoot/iocdemo
chmod +x st.cmd
./st.cmd
```

Repeat after changes

Directory Layout: Key Files

```
# makeBaseApp.pl -t example demo
configure/RELEASE
configure/CONFIG_SITE
demoApp/Db/*.db
demoApp/Db/*.substitutions
demoApp/Db/Makefile
demoApp/src/Makefile

# makeBaseApp.pl -t example -i demo
iocBoot/iocdemo/Makefile
iocBoot/iocdemo/st.cmd
```

To study the skeleton, check files before the first ‘make’ or after a ‘make distclean’

configure/RELEASE

- Defines the path to EPICS base and other modules

```
EPICS_BASE=/ics/tools/base-7.0.6
SNCSEQ = /ics/tools/seq-2.2.9
AUTOSAVE = /ics/tools/autosave-R5-10-2
```

- Since about 3.15, includes ../RELEASE.local

```
basedir/RELEASE.local: Lists all the modules

basedir/top1/configure/RELEASE   - includes ../../RELEASE.local
basedir/top1/abcApp/             - uses EPICS base etc.
basedir/top1/iocBoot/            - IOC bootups

basedir/top2/configure/RELEASE   - includes ../../RELEASE.local
basedir/top2/xyzApp/             - uses EPICS base etc.
basedir/top2/iocBoot/            - IOC bootups
```

demoApp

- xyzApp/Db **Database files**
- xyzApp/src *Main.cpp,
  Sequences,
  custom device support,
  Makefile that lists required *.dbd and libs

HowTo: Add Database files

1. Create `xyzApp/Db/another.db`

   For simple database, can test via
   `softIoc –d another.db`

2. Add to `xyzApp/Db/Makefile`:

```
DB += another.db
```

3. `make`

   Now it's under `db/another.db`

4. Add to `iocBoot/iocwhatever/st.cmd`

```
dbLoadRecords "db/another.db", "macro=value"
```

5. (Re-)start the IOC

Directory Layout: Generated Files

```
**/O.Common
**/O.linux-x86_64
**/O.*
db/*
dbd/*
include/*
lib/*
bin/*
```

Beware of difference:

- xyzApp/Db/*
  - Database ‘Sources’. Edit these!
- db/*
  - ‘Installed’ databases, may have macros replaced. Will be overwritten by next ‘make’!

*.dbd: Database Descriptions

IOC record types, device support, … are extensible

- Implement new record type, new device support:
  Write C/C++ code for certain interfaces, compile.
- Then 'register' your addition with core IOC code:
  *.dbd file

Internals:
VxWorks RTOS, the original IOC target, had runtime loader and symbol table.
RTEMS, .. don't necessarily offer this.
EPICS build facility generates IOC startup source code from *.dbd file.

HowTo: Add Support Modules (Device, …)

Example: 'Autosave'

1. Define path in `configure/RELEASE` or better in `../RELEASE.local`

```
AUTOSAVE=/ics/tools/autosave-R5-10-2
```

2. Add binary and DBD info to `xyzApp/Db/Makefile`:

```
YourProduct_DBD += asSupport.dbd
YourProduct_LIBS += autosave
```

3. Use the support module in the IOC startup file:

```
cd ${AUTOSAVE}
dbLoadRecords "db/save_restoreStatus.db", "P=demo"
set_requestfile_path("/home/controls/var")
create_monitor_set(...)
```

Details on how to use a support module depend on the specific one, including names of provided *.dbd, binary, *.db, IOC commands

Summary

makeBaseApp.pl creates the IOC skeleton

Good practice:

- Use `makeBaseApp.pl -t example…` for copy/paste.
- Create empty operational setup,
  and only paste-in what you need.
- Do it in small steps.

Much more:
EPICS Application Developer's Guide