OAK RIDGE National Laboratory

IOC 'makeBaseApp'

Kay Kasemir

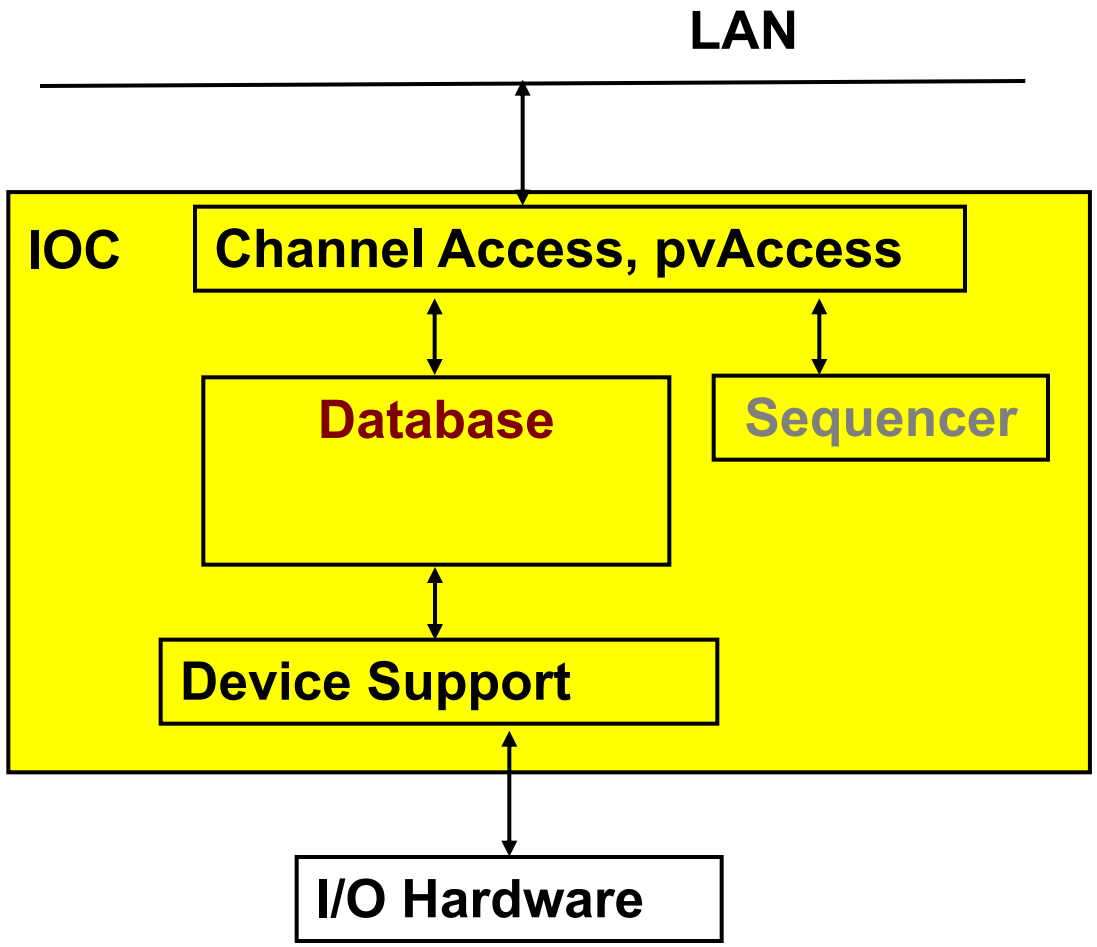
Jan. 2019

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

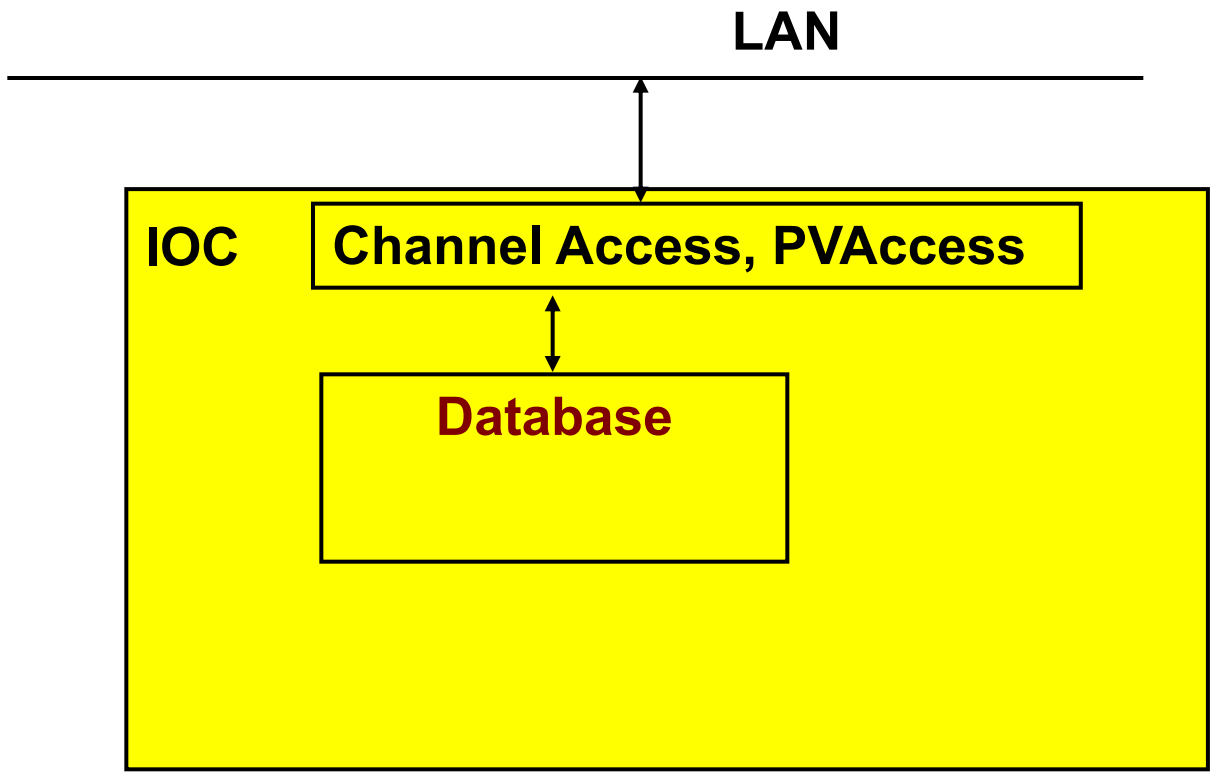
EPICS IOC

- Database
  - Known & well tested records
  - Remote access
  - Access security
  - ‘bumpless’ reboot
- Sequencer
  - Others might not understand your C code
- Device Support
  - Include existing device support? Easy enough
  - Have to write new device (driver) code? Varying degrees of difficulty



2

softIoc, softIocPVA



Pre-built IOC with Database engine, Channel Access, opt. PVA.

Run as many instances as needed.

Need autosave, sequencer, device support?
→ Create your own IOC application binary!

‘Host’ vs. ‘Target’ IOCs

- Host-based
  - 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, RTLinux
  - 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?

Accelerator

- Vacuum: Autosave, Support for AllenBradley PLC
- LLRF: Autosave, Support for LLRF hardware

Beamlines

- Cameras: Autosave, AreaDetector
- Various sample environments:
  Autosave, Motor Record, Stream Device

‘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 an Eclipse, Visual C++, KDeveloper … 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 ~/epics-train/mine
cd ~/epics-train/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
```

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

```
BASE=/home/training/epics-train/tools/base-7.0.1.1
SNCSEQ = /home/training/epics-train/tools/seq-2.2.6
AUTOSAVE = /home/training/epics-train/tools/autosave-R5-9
```

- 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.
- Somehow ‘register’ this 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 resp. ../../RELEASE.local`

```
AUTOSAVE=/home/training/epics-train/tools/autosave-R5-9
```

Path to the support directory is usually pulled into a macro, since you often include more than one support module:

```
TOOLS =/home/training/epics-train/tools
AUTOSAVE=$(TOOLS)/autosave-R5-9
```

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