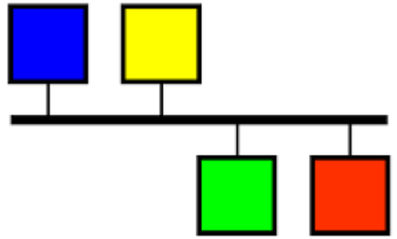
EPICS IOC

Notes and quick guides

EPICS IOC is a ChannelAccess server & client



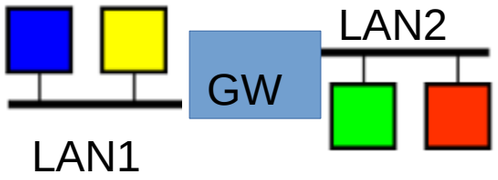
Run many of CAS/CAC → Integrate subsystems into your Control System
– LabIOC, EPICS IOC, java-based (CSS, PV-Archivers), python-based, NI-SV-based , etc.
– and also CA gateways

Be aware of LAN
– EPICS_CA_AUTO_ADDR_LIST=”” ; EPICS_CA_ADDR_LIST=””’;
– EPICS_CAS_IGNORE_ADDR_LIST=”” ; EPICS_CAS_INTF_ADDR_LIST=”” ; # starts with 3.15.x
– Other libs & SW should implement this somehow: env, config, INI-file,….
– be aware of localhost

We have mostly three LAN in each controller – only one should talk via CA !

Double-check broadcast strom settings on network switches (10% is fine, but …)
– your CA implementation might not be perfect (PV search, beacons are UDP bcasts in default)

CA-gateway subsystems
– cagateway or the new pva-based gateway
– Read-only GW, CAC-Archiver specific GW



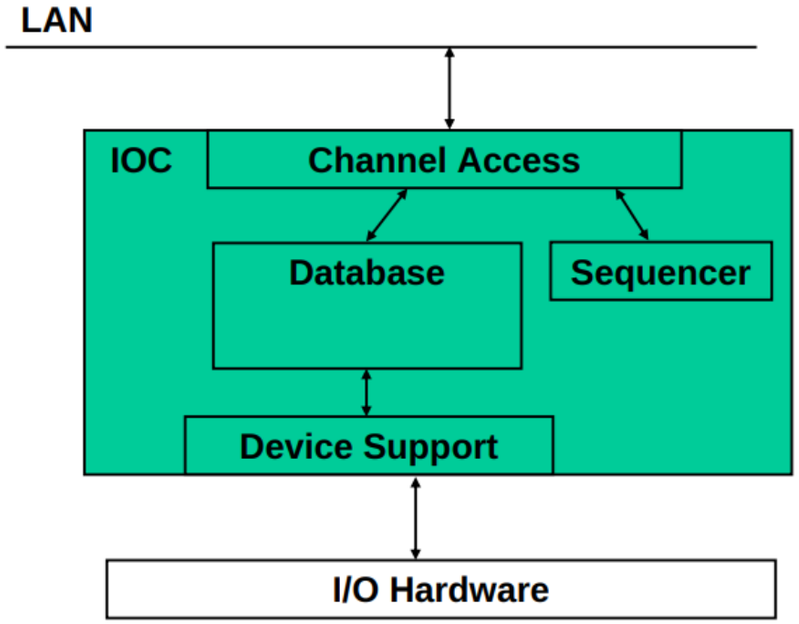
EPICS IOC – Why to like it ?

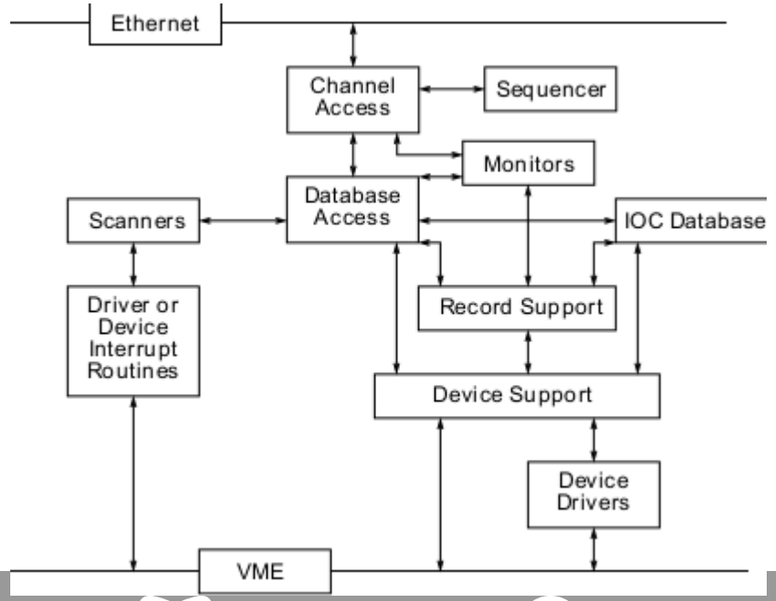
– because of records – that implement control system process

– because of auto channel access access to values & attributes

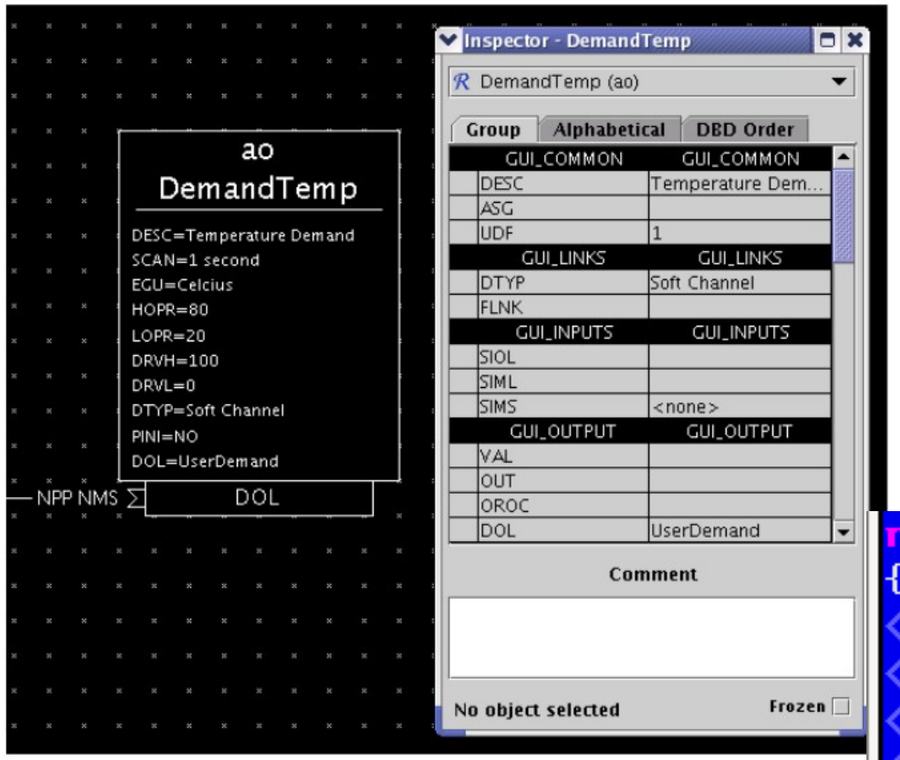
– mutlti-threaded, multi-platform SW
– from ARMv7 to server CPU beasts (base-3.15.8/configure/os/*)
– Linux, Windows, and others (base-3.15.8/configure/os/*)

IOC shell
– epicsThreadShow
– casr ; casr 255;
– dbpf DEMO:aiExample.TPRO 1





Graphical view of a record

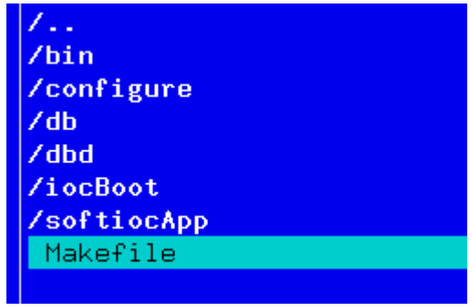



```
record(ai, "$(NAME):aiExample")
{
        field(DESC, "Analog input")
        field(INP, "$(NAME):calcExample.VAL  NPP NMS")
        field(EGUF, "10")
        field(EGU, "Counts")
        field(HOPR, "10")
        field(LOPR, "0")
        field(HIHI, "8")
        field(HIGH, "6")
        field(LOW, "4")
        field(LOLO, "2")
        field(HHSV, "MAJOR")
        field(HSV, "MINOR")
        field(LSV, "MINOR")
        field(LLSV, "MAJOR")
}
```

EPICS IOC

Folder structure – really ?
– the Makefile includes includes …
– very well managed “chain” but not for the start

```
makeBaseApp.pl -t ioc mytest_SoftIOC
makeBaseApp.pl -i -t ioc mytest_SoftIOC
```



bin
configure
db
dbd
iocBoot
softiocApp
Makefile

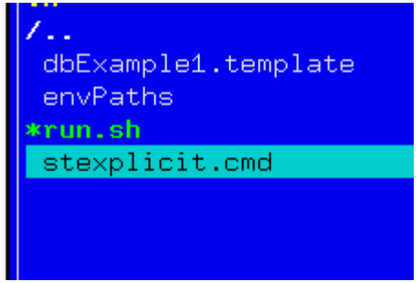
Advanced:

```
/epics/base-3.15.4/src/template/base/top/eli_sw_SoftIOCApp
/epics/base-3.15.4/src/template/base/top/eli_sw_SoftIOCBoot
```

Easy DEMO:

```
PATH=
LD_LIBRARY_PATH=
/usr/local/epics/R3.15/bin/linux-x86_64/softIoc stexplicit.cmd
```

dbExample1.template
envPaths
run.sh
stexplicit.cmd



Getting prebuilt stuff – mainly the base

* NSLS-II Debian packages https://epicsdeb.bnl.gov/debian/
and its source from epicsdeb project https://github.com/epicsdeb
– e.g. Dockerfile

* Conda package already contains the binaries
#conda install -c gsecars pyepics
Look for filename “softIoc”

* Shown yesterday: epics-containers project https://github.com/epics-containers
https://github.com/epics-containers/epics-base/archive/refs/tags/23.3.1.tar.gz
https://github.com/epics-containers/epics-base.git

* And some others
https://github.com/prjemian/epics-docker/blob/main/docs/docker_images.md

* Make the base on your own
– Dockerfile nowadays ?

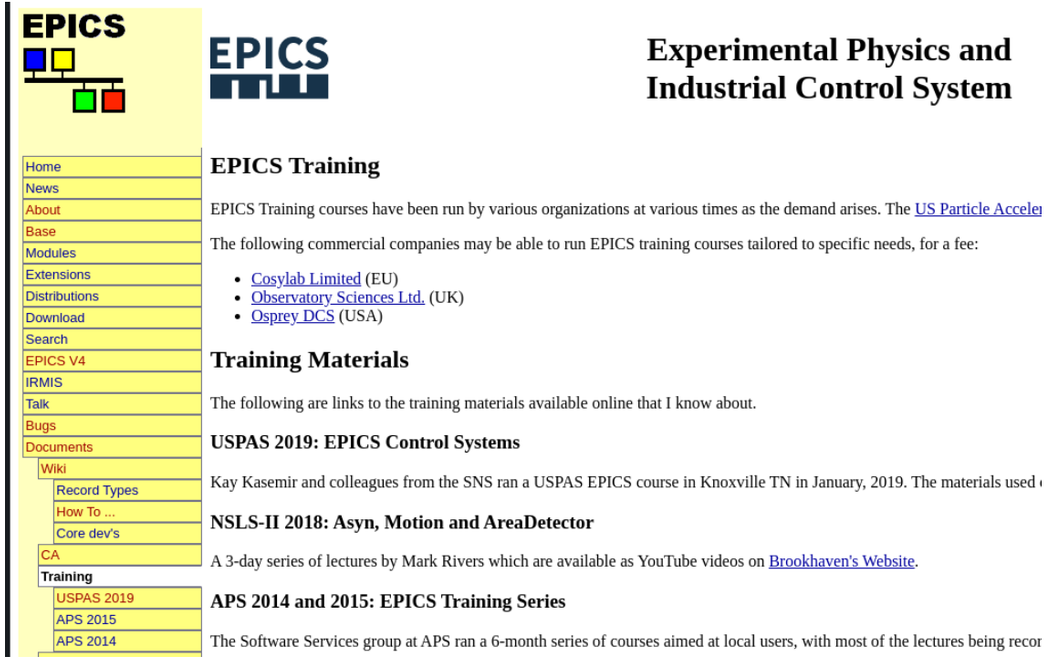
```
#!/bin/bash
source  ~/env-anaconda3.sh
conda activate py36
/home/anaconda3/envs/py36/opt/
epics/bin/softIoc stexplicit.cmd
```

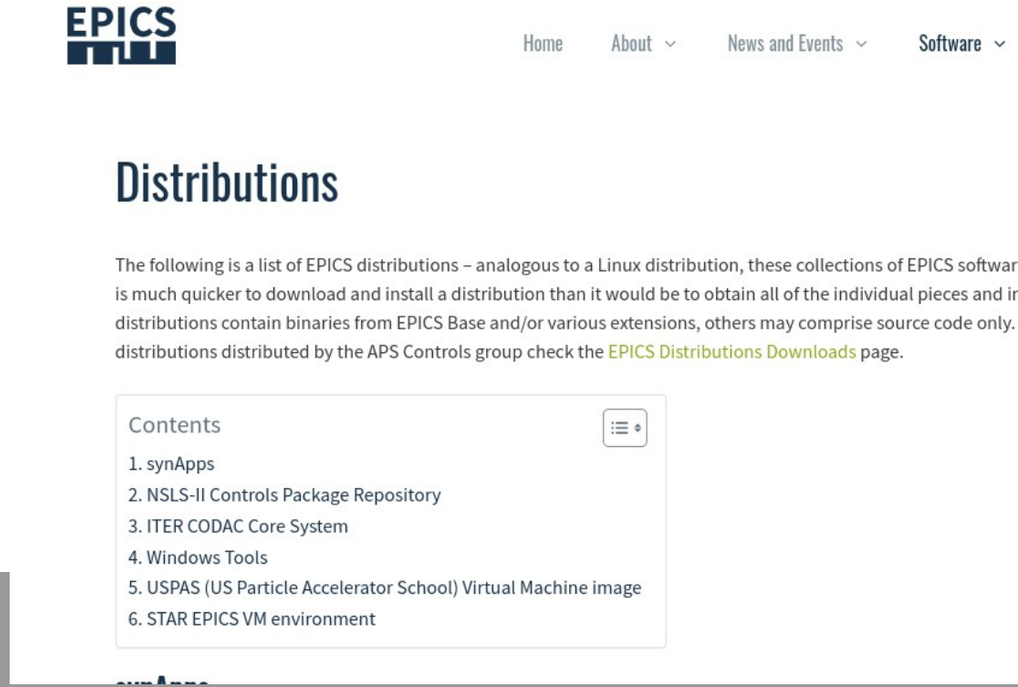
```
./
Dockerfile

base-3.15.8.orig-patch.diff
ca-gateway.orig-patch.diff
CONFIG_SITE.local
RELEASE.local

./files:
base-3.15.8.tar.gz
extensions.tar.gz
softioc.tar.gz
```

Windows CA tools prebuilt - also search somewhere else
https://epics.anl.gov/download/distributions/index.php
-> camonitor.exe via cmd.exe

Searching throught the https://epics.anl.gov/docs/training.php
– prebuit *.EXE with tutorials
– there is also a VM from USPAS (US Particle Accelerator School) at https://epics-controls.org/resources-and-support/distributions/

https://epics-controls.org/resources-and-support/distributions/#USPAS_US_Particle_Accelerator_School_Virtual_Machine_image





Thank you for your attention

tomas.mazanec@eli-beams.eu

./Dockerfile

```
FROM alpine:3 AS epics315

RUN apk update && apk add --no-cache \
    perl \
    libstdc++ \
    readline \
    libexecinfo \
    pcre \
    tzdata \
    bash \
    git \
    vim \
    curl \
    mc \
    tini \
    iputils \
\
    && adduser -u 7777 -h /home/eli_sw -s /bin/sh -D eli_sw \
\
    && ln -snf /usr/share/zoneinfo/UTC /etc/localtime

ENV EPICS_HOST_ARCH="linux-x86_64"

# # epics base
ADD --chown=7777:7777 files/base-3.15.8.tar.gz /home/eli_sw

# # CA-GATEWAY
ADD  --chown=7777:7777 files/extensions.tar.gz /home/eli_sw

## apk del --purge .build-deps  afterwards deletes also DEPENCIES , nice !
RUN apk update && apk add --no-cache --virtual .build-deps \
    gcc \
    g++ \
    build-base \
    linux-headers \
    readline-dev \
    libexecinfo-dev \
    pcre-dev \
    re2c \
\
    && ln -s /home/eli_sw/base-3.15.8/ /home/eli_sw/R3.15 \
    && cd /home/eli_sw/R3.15 \
    && make -j3 \
    && make clean \
    && chown -R eli_sw:eli_sw /home/eli_sw/base-3.15.8 \
    && chown eli_sw:eli_sw /home/eli_sw/R3.15 \
    && cd /home/eli_sw/extensions/ca-gateway \
    && rm -rf ./git \
    && make -j3 \
    && make clean \
\
    && cp /home/eli_sw/extensions/ca-gateway/bin/linux-x86_64/gateway
/home/eli_sw/extensions/ca-gateway/ \
    && chown -R eli_sw:eli_sw /home/eli_sw/extensions \
    && ln -s /home/eli_sw/extensions/rungw.sh /home/eli_sw/rungw.sh \
\
    && apk del --purge .build-deps

# # SoftIOC
ADD --chown=7777:7777 files/softioc.tar.gz /home/eli_sw

ENTRYPOINT ["/usr/bin/env"]
CMD ["sh"]

# docker-compose build
# docker build -t pokus/alpine3 .
```