Getting Started with EPICS

<div align="center">**COLLABORATORS**</div>

	TITLE : Getting Started with EPICS		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Michael Davidsaver	September 2015	

<div align="center">**REVISION HISTORY**</div>

NUMBER	DATE	DESCRIPTION	NAME
1	February 2009	Initial revision	MAD
2	February 2012	Revise for newer software versions. Updates links etc.	MAD
3	February 2013	Correct testing section. The EPICS_CAS_INTF_ADDR_LIST is ignored by RSRV.	MAD
4	September 2015	Update with Base 3.15.2. Remove ALH references.	MAD

Contents

List of Tables

1   Introduction

This article describes how to build and run a simple software IOC and the EDM display manager. The goal is to provide the reader a basic functional system with which to further explore the EPICS control system.

The intended audience is anyone with an interest in setting up a standalone EPICS system for development, testing, or just to try something new. Some familiarity with a Bourne shell and with building software in a Unix(like) environment is assumed. No knowledge of programming or the C/C++ languages is assumed as none is required to understand the examples presented here.

2   Environment

The EPICS build system carries some assumptions about the layout of its source code tree. We will be building the EPICS Base package along with several extension packages (EDM and ALM). For the remainder of this document it is assumed that the following directory structure is present under a base directory *$HOME/myepics*. This name can be replaced with an arbitrary directory. No privileged operations are used.

```
- myepics
  - base
  - extensions
    - configure
    - src
      - edm
    - etc
  - caqtdm
  - cs-studio
```

Several environment variables must also be set when building and using EPICS. They will be set when appropriate in the following sections.

EPICS_BASE $HOME/myepics/base

```
    By convention the location of the EPICS Base tree.
    Used to reference executables, headers, libraries, and other built files.
```

EPICS_EXTENSIONS $HOME/myepics/extensions

```
    Home of the various addons to the Base package including EDM.
    Also contains separate build infrastructure with potentially different  ↩
        settings.
```

EPICS_HOST_ARCH linux-x86, solaris-sparc-gnu, win32-x86-mingw, etc.

```
    The identifier for the build host.
    Unix users can run the 'EpicsHostArch' script found in '$EPICS_BASE/startup'
```

EDMOBJECTS $EPICS_EXTENSIONS/etc

```
    Directory containing edmObjects
```

EDMPVOBJECTS $EPICS_EXTENSIONS/etc

```
    Directory containing edmPvObjects
```

EDMFILES $EPICS_EXTENSIONS/src/edm/edmMain

```
    Directory containing misc EDM configuration.
```

EDMHELPFILES $EPICS_EXTENSIONS/src/edm/helpFiles

```
Location of EDM online help data.
```

EDMLIBS $EPICS_EXTENSIONS/lib/$EPICS_HOST_ARCH

```
Location of EDM plugin libraries.
```

The following are used to ensure that this example remain self-contained even if run on a machine attached to a network with Channel Access traffic. Their meanings are discussed below, and explained fully in the *Channel Access Reference Manual* [CAref].

EPICS_CA_ADDR_LIST 127.255.255.255

EPICS_CA_AUTO_ADDR_LIST NO

3   Getting the Source

The source for most EPICS software can be found at or through the EPICS Home Page. The specific versions listed below were used when writing this document.

Name	File	Download
Base	base-3.15.2.tar.gz	http://www.aps.anl.gov/epics/base/R3-15/2.php
Extensions	extensionsTop_201210904.tar.gz	http://www.aps.anl.gov/epics/extensions/configure/-index.php
EDM	edm-1-12-101.tgz	http://ics-web.sns.ornl.gov/edm/
caQtDM	V3.9.4.tar.gz	http://epics.web.psi.ch/software/caqtdm/
cs-studio	CSS-NSLSII-linux64-3.2.16a.zip	http://cs-studio.sourceforge.net/

Table 1: EPICS Components

Download all the files listed in components table into *$HOME/myepics*. First unpack Base and Extensions. Either rename or symlink *base-3.15.2* to *base*. Create the directory *extensions/etc* Now unpack EDM in the *extensions/src* directory.

4   Building

4.1   Base

Before starting the compile it is worth mentioning that there are several files which contain user options which effect the build process. *CONFIG_SITE* and *RELEASE* in *$EPICS_BASE/configure*. In particular *CONFIG_SITE* is where the cross-build targets would be specified.

Now simply switch to the base directory and issue the `make` command.

```
cd $EPICS_BASE
make
```

More information on compiling EPICS Base can be found in *base/documentation/README.html* or online at http://www.aps.anl.gov/-epics/base/R3-15/2-docs/README.html.

Before leaving this directory, determine the host target name, and set the `EPICS_HOST_ARCH` environment variable. Several scripts in the startup/ directory can help with this.

```
export EPICS_HOST_ARCH=`./startup/EpicsHostArch`
```

4.2 cs-studio

While the source for the cs-studio client tools is available, the build process is quite complex. It is recommended to use a pre-compiled version http://cs-studio.sourceforge.net/.

The latest stable NSLS2 version is suggested http://cs-studio.sourceforge.net/nsls2/nsls2.html Although any site's version will suffice for the examples given below.

Fetch the .zip and extract intos *$HOME/myepics*. This should create an executable *$HOME/myepics/css-nsls2/css-nsls2*, which is the entry point.

When run it will ask for the location of a workspace, the default will suffice, or *$HOME/myepics/workspace*.

Files should be placed under the workspace directory to be visible. When started for the first time a sub-directory *CSS* is created. The .opi file(s) mentioned in this guide should be placed here, then either refresh the workspace (F5) or restart cs-studio.

4.3 caQtDM

This package requires Qt, Qwt, and python to be available on the host system.

Once unpacked, set the appropriate environment variables, or edit the file caQtDM_Env.

Set 'EPICS_BASE` to the full path `$HOME/myepics/base'.

It may be necessary to change other variables (eg. *PYTHONVERSION*) depending on the host.

Build it

```
./caQtDM_BuildAll
```

The result can be started with the *startDM_Local* and *qtdesigner* wrapper scripts which respectively run and edit .ui files.

4.4 Extensions

The `extensions` build files come pre-configured to expect that EPICS Base is installed along side them. This can be changed by changing the `EPICS_BASE` line in `extensions/configure/RELEASE`.

You may also need to edit `extensions/configure/os/CONFIG_SITE.${EPICS_HOST_ARCH}.${EPICS_HOST_ARCH}` and change `X11_LIB` and `MOTIF_LIB` to the directory where these libraries are found (eg. */usr/lib* or */usr/lib/x86_64-linux-gnu*)

4.5 EDM

Building the EDM display manager can be complicated. It has many dependencies in addition to Motif including libgif, libpng, and many X11 libraries (X11, Xm, Xau, etc.). These are assumed to be present and the build system does not check that they are. If a compile or link fails look for messages indicating missing headers or libraries and install the appropriate package. More information of building EDM can be found in the EDM User Manual [EDMman].

On Debian Linux (and derivative) these are

```
apt-get install libgif-dev libpng-dev libz-dev libmotif-dev\
 libx11-dev libxp-dev libxmu-dev libxtst-dev x11proto-print-dev
```

Now build EDM.

```
cd ${EPICS_EXTENSIONS}/src/edm
make
```

The GIF plugin needs to link against the ungif library which may not be available, and has been unnecessary for some time.

```
sed -i -e 's| ungif||g' giflib/Makefile*
make
```

4.5.1 Generate Plugin Config Files

EDM depends on several configuration files. The two files which list the locations of various plugins must be generated.

The following is a simpleminded way to do this which takes advantage of the fact that EDM can recognize if a library is a valid plugin and which type it is. It will only add valid plugins to the list files.

```
cd ${EPICS_EXTENSIONS}/lib/${EPICS_HOST_ARCH}
export EDMOBJECTS=$PWD
export EDMPVOBJECTS=$PWD
export EDM=${EPICS_EXTENSIONS}/bin/${EPICS_HOST_ARCH}/edm
for pv in Epics Calc Log Loc;do $EDM -addpv $PWD/lib$pv.so;done
for ff in lib*.so;do ${EDM} -add $PWD/$ff; done
sed -i -e "s|$PWD|"'$(EDMLIBS)|' edmObjects edmPvObjects
rm edmObjects~ edmPvObjects~
unset EDMOBJECTS EDMPVOBJECTS EDM
install -d ${EPICS_EXTENSIONS}/etc
mv edmObjects edmPvObjects ${EPICS_EXTENSIONS}/etc/
```

The additional step of replacing the absolute paths with *$(EDMLIBS)* is taken to allow the built tree to be relocated without requiring modifications to these files. It may be omitted, and the `EDMOBJECTS` and `EDMPVOBJECTS` environment variables left unset.

4.5.2 Additional Environment

Several more environment variables are required to run EDM.

```
export EDMOBJECTS=$EPICS_EXTENSIONS/etc
export EDMPVOBJECTS=$EPICS_EXTENSIONS/etc
export EDMFILES=$EPICS_EXTENSIONS/src/edm/edmMain
export EDMHELPFILES=$EPICS_EXTENSIONS/src/edm/helpFiles
export EDMLIBS=$EPICS_EXTENSIONS/lib/$EPICS_HOST_ARCH
```

EDM can now be executed, but it can't show anything interesting, yet.

```
export PATH="${EPICS_EXTENSIONS}/bin/${EPICS_HOST_ARCH}:${PATH}"
edm
```

5 Testing

The remainder of this document will be devoted to creating a trivial IOC which will then be used to demonstrate the functionality of EDM.

It is assumed that this example is being executed on a disconnected system, or at least on a system which sees no EPICS traffic and hosts no IOCs. In the interest of safety the following will prevent clients on the local system from connecting to IOCs on other systems. However, remote clients may connect to local IOCs. Complete isolation requires a firewall set to block TCP and UDP traffic on ports 5064 and 5065.

```
export EPICS_CA_AUTO_ADDR_LIST=NO
export EPICS_CA_ADDR_LIST=127.255.255.255
```

So in summary, the environment constructed for the following examples is:

```
export EPICS_BASE="${HOME}/myepics/base"
export EPICS_EXTENSIONS="${HOME}/myepics/extensions"
export EPICS_HOST_ARCH=`${EPICS_BASE}/startup/EpicsHostArch`
export PATH="${EPICS_BASE}/bin/${EPICS_HOST_ARCH}:${PATH}"
export PATH="${EPICS_EXTENSIONS}/bin/${EPICS_HOST_ARCH}:${PATH}"
```

```
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${EPICS_BASE}/lib/${EPICS_HOST_ARCH}"
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${EPICS_EXTENSIONS}/lib/${EPICS_HOST_ARCH}"
export EDMOBJECTS=$EPICS_EXTENSIONS/etc
export EDMPVOBJECTS=$EPICS_EXTENSIONS/etc
export EDMFILES=$EPICS_EXTENSIONS/src/edm/edmMain
export EDMHELPFILES=$EPICS_EXTENSIONS/src/edm/helpFiles
export EDMLIBS=$EPICS_EXTENSIONS/lib/$EPICS_HOST_ARCH
export EPICS_CA_AUTO_ADDR_LIST=NO
export EPICS_CA_ADDR_LIST=127.255.255.255
```

6   SoftIoc

Now we will construct a IOC which will run on the host machine. This will be done without compiling any additional code through the use of the softIoc executable.

The IOC will have only tree Process Variables (PVs). One will be automatically fill with the sum of the other two. That is if PV A is 1 and PV B is set to 2 then PV SUM will be set to 3.

Create the two short files: sum.db and sum-start.cmd. Then start the IOC in the directory containing them.

```
mkdir $HOME/myepics/myioc
cd $HOME/myepics/myioc
```

6.1   sum.db

https://github.com/mdavidsaver/epics-doc/raw/master/starting/sum.db

```
record(ao,"$(INST):a"){
  field(DTYP,"Soft Channel")
  field(VAL,0)
  field(UDF,1)
  field(FLNK,"$(INST):sum")
}
record(ao,"$(INST):b"){
  field(DTYP,"Soft Channel")
  field(VAL,0)
  field(UDF,1)
  field(FLNK,"$(INST):sum")
}
record(calc,"$(INST):sum"){
  field(INPA,"$(INST):a")
  field(INPB,"$(INST):b")
  field(CALC,"A + B")
}
```

This file will be used to create several process variables in the database of the softIoc instance. They consists of two Soft Analog Output records and a Calculated record. The *$(INST)* will be replaced with an actual prefix string when loaded at runtime.

sum.cmd

```
https://github.com/mdavidsaver/epics-doc/raw/master/starting/sum.cmd[https://github.com/ ←
    mdavidsaver/epics-doc/raw/master/starting/sum.cmd]


--------------------------------
dbLoadRecords("sum.db","INST=calc")
iocInit
--------------------------------
```

```
Here we have a start script containing commands to be
given to the IOC shell.
It simply sets several environment variables, loads the
records previously defined, with the given prefix,
and starts running.

The same db file could be loaded again with a different
prefix.

Running
```

Now start the IOC.

```
softIoc sum.cmd
```

When running it provides three PVs: calc:a, calc:b, calc:sum.

7   PV Access from a Shell

EPICS Base provides several utilities for reading and modifying Process Variables from command shell. Here *caget* and *caput* are demonstrated by accessing our softIoc (running in another shell).

```
$ caget calc:a calc:b calc:sum
calc:a                          0
calc:b                          0
calc:sum                        0
$ caput calc:a 2
Old : calc:a                            0
New : calc:a                            2
$ caget calc:a calc:b calc:sum
calc:a                          2
calc:b                          0
calc:sum                        2
$ caput calc:b 3
Old : calc:b                            0
New : calc:b                            3
$ caget calc:a calc:b calc:sum
calc:a                          2
calc:b                          3
calc:sum                        5
```

8   PV Access with EDM

Run EDM. First place two *Text Control* s and one *Text Monitor* widgets. In each one set the PV to one of the three PVs from our IOC. Also set *Null Condition* to *Disabled* and check *Editable*, *Keypad*, *Alarm Sensitive*, and *Alarm Border*.

The file sum.edl is provided as a starting point.

https://github.com/mdavidsaver/epics-doc/raw/master/starting/sum.edl

Execute the display and try setting the three fields. Try setting A and B so that the sum exceeds 15. Also try using slider controls for A and B.

To open in edit mode.

```
edm -m 'INST=calc' sum.edl
```

To go directly to run mode add *-x*. Switching between edit and run modes can also be done from the context menu (middle mouse button).

```
edm -x -m 'INST=calc' sum.edl
```

Check EDM's *Help* menu for more information. EDM has a decent on-line help/reference.

8.1 Alarms

We will revisit the record specification file *sum.db* and add several fields. Some of these additional fields are used by EDM to determine how records should be displayed. Others are used by the IOC for alarming.

Same the new file *sum-alarm.db*.

Stop and restart the softIoc, either by editing *sum.cmd*, or by giving the macro definition and .db files as arguments (two ways to achieve the same result).

```
softIoc -m 'INST=calc', -d sum-alarm.db
```

Now set *calc:a* and *calc:b* so that the sum exceeds 15. The border of the sum widget should change color (the alarm sensitive/border properties).

8.2 sum-alarm.db

https://github.com/mdavidsaver/epics-doc/raw/master/starting/sum-alarm.db

```
record(ao,"$(INST):a"){
  field(DESC,"Input A")
  field(DTYP,"Soft Channel")
  field(PREC,1)
  field(DRVH,10)
  field(HOPR,10)
  field(VAL,0)
  field(LOPR,-10)
  field(DRVL,-10)
  field(UDF,1)
  field(FLNK,"$(INST):sum")
}
record(ao,"$(INST):b"){
  field(DESC,"Input B")
  field(DTYP,"Soft Channel")
  field(PREC,1)
  field(DRVH,10)
  field(HOPR,10)
  field(VAL,0)
  field(LOPR,-10)
  field(DRVL,-10)
  field(UDF,1)
  field(FLNK,"$(INST):sum")
}
record(calc,"$(INST):sum"){
  field(DESC,"Sum")
  field(PREC,1)
  field(INPA,"$(INST):a")
  field(INPB,"$(INST):b")
  field(CALC,"A + B")
  field(HOPR,20)
  field(HIHI,18)
  field(HIGH,15)
  field(LOW,-15)
  field(LOLO,-18)
  field(LOPR,-20)
```

```
  field(HHSV,"MAJOR")
  field(HSV,"MINOR")
  field(LSV,"MINOR")
  field(LLSV,"MAJOR")
}
```

9   PV Access with caQtDM

Where EDM uses .edl files, caQtDM uses .ui files. These are created using QT Designer using plugins provided by caQtDM. These will not appear unless the appropriate environment variables are set (see the *qtdesigner* wrapper script).

Run

```
./startDM_Local -macro 'INST=calc' sum.ui
```

https://github.com/mdavidsaver/epics-doc/raw/master/starting/sum.ui

10   PV Access with cs-studio

```
./cs-studio --launcher.openFile /CSS/sum.opi
```

https://github.com/mdavidsaver/epics-doc/raw/master/starting/sum.opi

11   Have Fun

For more information on record types see the *Record Reference Manual* in the EPICS wiki http://www.aps.anl.gov/epics/wiki/index.php/RRM_3-14. Information on writing more functionality for IOCs can be found in the Developer's guide [AppDev].

12   References

12.1   References

[1] [CAref] Jeffrey O. Hill & Ralph Lange. Channel Access Reference Manual. http://www.aps.anl.gov/epics/base/R3-15/2-docs/CAref.html

[2] [AppDev] Marty Kraimer et al. 'EPICS Application Developer's Guide '. http://www.aps.anl.gov/epics/base/R3-15/2-docs/AppDevGuide/

[3] [EpicsRTEMS] W. Eric Norum. Getting started with EPICS on RTEMS. http://www.aps.anl.gov/epics/base/RTEMS/tutorial/

[4] [EDMman] John Sinclair. Extensible Display Manager. http://ics-web.sns.ornl.gov/edm/edmUserManual/index.html