

Epics Tutorial 3: Creating IOC's

Carlos Galdino

Last update: 2 March 2016

- IOC stands for Input Output Controller. We need IOC's to build our applications, in the next tutorial we are going to build a simple calculator using and IOC.
- In this tutorial we are going to talk about the basics about creating an IOC.
- The IOC will be called "**myCalculator**".
- This tutorial aims to make you understand a little better how the IOC works, but we're not going to run a full example. This document is more like a support for next tutorial.
- **The IOC *myCalculator* will be used in the next tutorials.**
- Feel free to send comments, critics, and suggestions.

Making things easier

Before we get started, we are going to define a few variables. You do not need to define these variables, but if you do, you are able to copy and paste future commands directly from this tutorial to the terminal.

Long story short, this is just to make things easier for us. In the next tutorials, for example, most of the time I need the epics base path I will use *EPICS_BASE*, and not *"/opt/epics/base"*. So maybe, when you are reading de next tutorials, you might need to came back here to recall these definitions.

Do not forget to have your *LD_LIBRARY_PATH* variable (only linux) and the *PATH* variable set correctly (check installation tutorials again if needed).

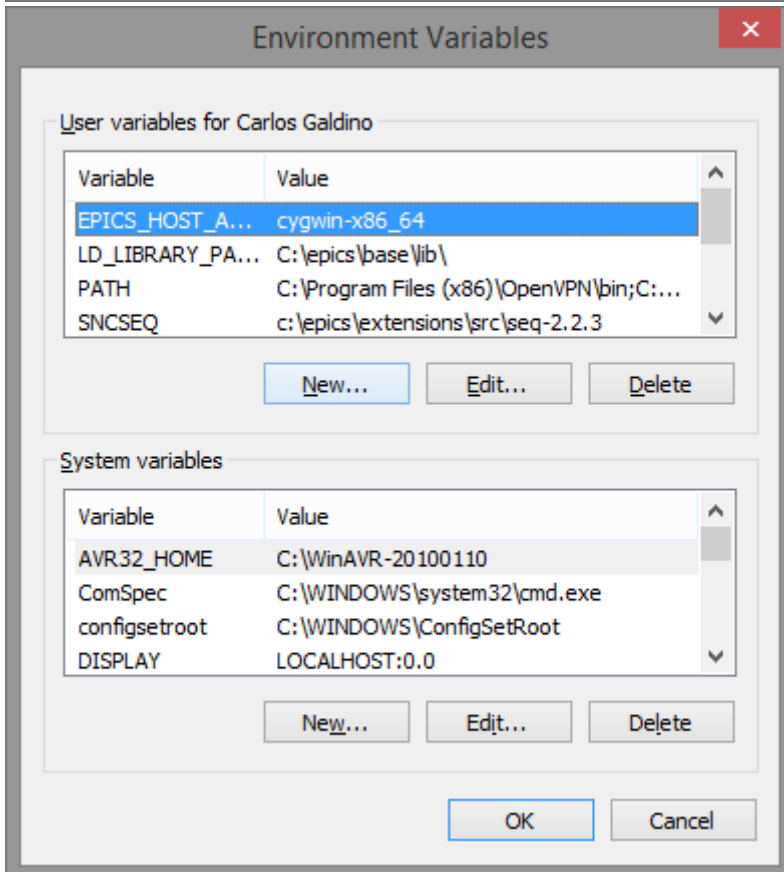
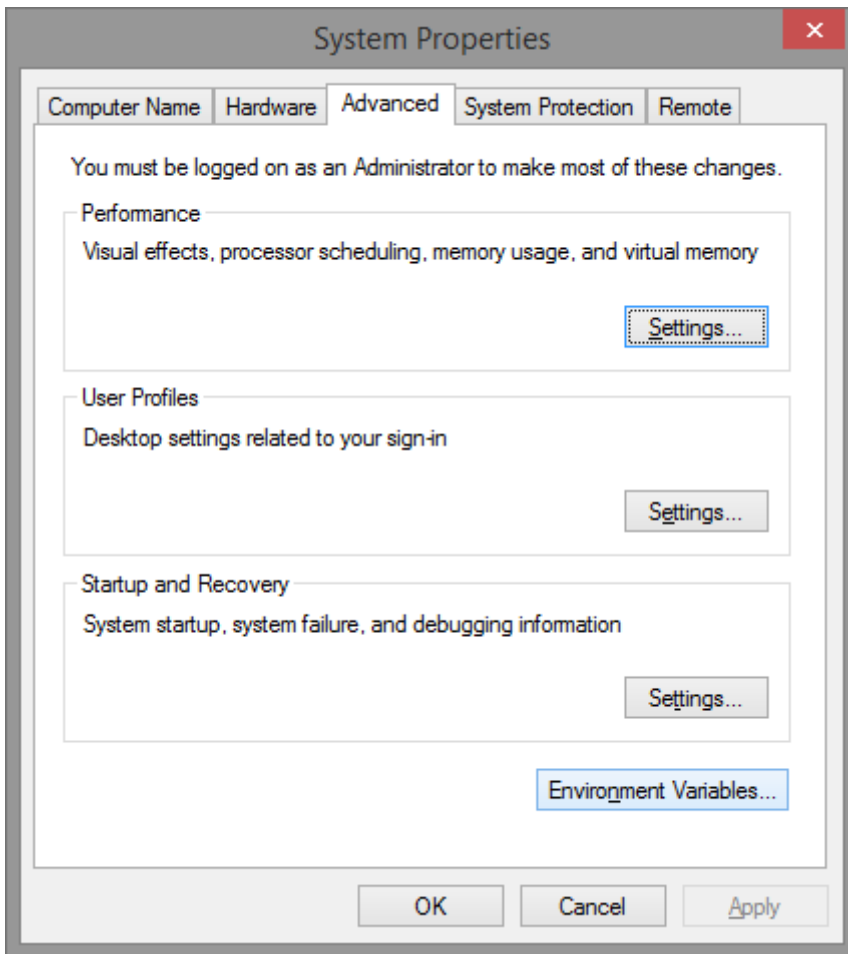
Linux

Copy the following commands in the terminal (or the Cygwin Terminal in Windows).

- **Path to base**
`export EPICS_BASE = "/opt/epics/base"`
- **OS type** (here we are using a bash that automatically assigns the right OS type):
`export EPICS_HOST_ARCH = ' ${EPICS_BASE}/startup/EpicsHostArch '`
- **Name of our IOC**
`export WORKINGNAME=myCalculator`

Windows

Go to System Properties > Advanced > Environment Variables (If you do not know how to do it, jut google it).

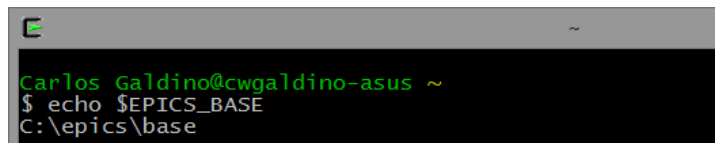


Click on **New** (the User variables for Carlos Galdino) and add the following variables:

- **Path to base**
`EPICS_BASE = C:\epics\base`
- **OS type** (check epics manual to see available options):
(32bit)
`export EPICS_HOST_ARCH = cygwin-x86`
(64bit)
`export EPICS_HOST_ARCH = cygwin-x86_64`
- **Name of our IOC**
`WORKINGNAME=myCalculator`

You have to Restart Cygwin terminal for the changes to apply.

Use the command `echo $EPICS_BASE` to check if you set the variables correctly.



```
Carlos Galdino@cwgaldino-asus ~  
$ echo $EPICS_BASE  
C:\epics\base
```

Setting IOC Folder

Go to Epics folder and make a new folder for the IOC.

```
cd $EPICS_BASE
```

```
cd ..
```

```
mkdir $WORKINGNAME
```

```
cd $WORKINGNAME
```

Sometimes the IOC folder will be blocked and you will not be able to modify files. Use `chown` and `chmod` to unlock the folder (remember to use in the folder above myCalculator). Use it anytime you need.

```
chown carlogaldino $WORKINGNAME
```

```
chmod -R 777 $WORKINGNAME
```

Run the make base App.

```
$EPICS_BASE/bin/$EPICS_HOST_ARCH/makeBaseApp.pl -t ioc $WORKINGNAME
```

```
$EPICS_BASE/bin/$EPICS_HOST_ARCH/makeBaseApp.pl -t ioc -i $WORKINGNAME
```

If it asks something to you, just press enter. Take a time to check the IOC folders and files.

IOC

The IOC structure is:

iocName

configure

CONFIG
CONFIG_SITE
Makefile
RELEASE
RULES
RULES.ioc
RURLES_DIRS
RULES_TOP

iocNameApp

Dd

Makefile

src

iocMain.cpp
Makefile

Makefile

iocBoot

iocName

Makefile
st.cmd

Makefile

- **Bold** names are folders.
- **Blue** names are files that we usually modify before building the IOC.
- **Red** names are files that we usually modify after building the IOC.

Let's take a moment to talk a little bit about the files and folder in a IOC.

- The TOP folder is the most "top" folder of an application.
- The TOP Epics folder is ie. /opt/epics
- The TOP folder of an IOC is ie. /opt/epics/myCalculator
- The IOC has a config folder from which the installation looks for the settings related to the installation process.
- The IOC is also composed of a folder *Boot, which initializes the IOC.
- The App folder is the application itself. Most "core" changes are made in this folder. Note you can have several "applications" in a single IOC.
- The Makefile becomes necessary so you can install the IOC. The Makefile enters the aforementioned folders and executes makefiles that are inside these folders. The installation is not over until all makefiles have been executed.

We are now going to give you a brief description of **each file**. Do not bother reading it all. For this example (myCalculator IOC) you are not going to edit anything, then you can just skip to the next session.

The TOP/makefile

- The Makefile of TOP folder is the first file to “play” (you must run make in the TOP folder of the IOC).
- The makefile of TOP folder more or less like this:

```
DIRS := $(DIRS) $(filter-out $(DIRS), configure)
```

```
DIRS := $(DIRS) $(filter-out $(DIRS), $(wildcard *App))
```

```
DIRS := $(DIRS) $(filter-out $(DIRS), $(wildcard iocBoot))
```

By the structure above, we note the make command runs first in the configure folder, then the folder *App and finally the *Boot folder.

TOP/configure/CONFIG

Do not edit this file (unless you know what you’re doing).

TOP/configure/RULES

Do not edit this file (unless you know what you’re doing).

TOP/configure/RULES.ioc

Do not edit this file (unless you know what you’re doing).

TOP/configure/RULES_DIRS

Do not edit this file (unless you know what you’re doing).

TOP/configure/RULES_TOP

Do not edit this file (unless you know what you’re doing).

TOP/configure/CONFIG_SITE

- Used to set the architecture of your installation.
- You can set your IOC to be built in different Architectures using the “CROSS_COMPILER_TARGET_ARCHS”. Don’t worry about this right now, just leave it blank.

```
CROSS_COMPILER_TARGET_ARCHS =
```

- Read instructions in the file for more information

TOP/configure/RELEASE

- If your IOC is simple and you are not using extensions you **do not** need to edit this file.
- Check this file where there is a lot of extra modules:

#FILENAME: RELEASE

#USAGE: Specify directory paths to synApps support modules

#Version: \$Revision: 16643 \$

#Modified By: \$Author: mooney \$

#Last Modified: \$Date: 2013-06-25 14:06:40 -0500 (Tue, 25 Jun 2013) \$

#HeadURL: \$URL:

https://subversion.xray.aps.anl.gov/synApps/configure/tags/synApps_5_7/RELEASE \$

#NOTES

- To remove modules from the build, delete or comment out the

module name.

SUPPORT=/opt/epics/extensions/src/synApps_5_7/support

-include \$(TOP)/configure/SUPPORT.\$(EPICS_HOST_ARCH)

EPICS_BASE=/opt/epics/base

-include \$(TOP)/configure/EPICS_BASE

-include \$(TOP)/configure/EPICS_BASE.\$(EPICS_HOST_ARCH)

#/usr/local/iocapps/cvsroot/modules/bus/allenBradley

ALLEN_BRADLEY=\$(SUPPORT)/allenBradley-2-2

AREA_DETECTOR=\$(SUPPORT)/areaDetector-1-9-1

#<https://svn.aps.anl.gov/epics/asyn/>

ASYN=\$(SUPPORT)/asyn-4-21

AUTOSAVE=\$(SUPPORT)/autosave-5-1

BUSY=\$(SUPPORT)/busy-1-6

CALC=\$(SUPPORT)/calc-3-2

CAMAC=\$(SUPPORT)/camac-2-7
DAC128V=\$(SUPPORT)/dac128V-2-7
DELAYGEN=\$(SUPPORT)/delaygen-1-1-0
#http://www.slac.stanford.edu/grp/cd/soft/epics/site/devlocStats/
DEVIOCSTATS=\$(SUPPORT)/devlocStats-3-1-7
DXP=\$(SUPPORT)/dxdp-3-2
IP=\$(SUPPORT)/ip-2-14
IP330=\$(SUPPORT)/ip330-2-7
#https://svn.aps.anl.gov/epics/ipac/
#IPAC=\$(SUPPORT)/ipac-2-11
(V2-12? have outstanding commits since V2-11)
IPAC=\$(SUPPORT)/ipac-2-12
IPUNIDIG=\$(SUPPORT)/ipUnidig-2-9
LOVE=\$(SUPPORT)/love-3-2-5
MCA=\$(SUPPORT)/mca-7-4
MEASCOMP=\$(SUPPORT)/measComp-1-0
MODBUS=\$(SUPPORT)/modbus-2-4
MOTOR=\$(SUPPORT)/motor-6-8
OPTICS=\$(SUPPORT)/optics-2-9-1
QUADEM=\$(SUPPORT)/quadEM-4-0
#http://www-csr.bessy.de/control/SoftDist/sequencer
SNCSEQ=\$(SUPPORT)/seq-2-1-13
SOFTGLUE=\$(SUPPORT)/softGlue-2-3
SSCAN=\$(SUPPORT)/sscan-2-9
STD=\$(SUPPORT)/std-3-2
#http://epics.web.psi.ch/software/streamdevice/
STREAM=\$(SUPPORT)/stream-2-6
VAC=\$(SUPPORT)/vac-1-4-1
VME=\$(SUPPORT)/vme-2-8-1
XXX=\$(SUPPORT)/xxx-5-7-1

TOP/*App

- This folder is the application itself. Most "core" change are made in this folder.
- You can have several "applications" in a single IOC.
- In App folder there are two folder, Db and src (and the Makefile, of course).

TOP/*App/Makefile

- The *App makefile is more or less like this:

```
DIRS := $(DIRS) $(filter-out $(DIRS), $(wildcard *src*))
```

```
DIRS := $(DIRS) $(filter-out $(DIRS), $(wildcard *Src*))
```

```
DIRS := $(DIRS) $(filter-out $(DIRS), $(wildcard *db*))
```

```
DIRS := $(DIRS) $(filter-out $(DIRS), $(wildcard *Db*))
```

By the structure above, we note the make command runs first in the src folder and then the db folder.

TOP/*App/src

- The Src folder is very important. This folder serves to build the definitions of databases along with the libraries. In other words, this folder installs the .dbd files of your IOC.
- In this folder we have two files makefile and a file ".cpp". In the makefile you should put all dbd's you want to install. Usually the annoying part is knowing what you want to install.
- It's common to make a new folder inside TOP/*App called "Op" and another one inside Op called "adl" to put .adl files (these are GUI files from medm, we'll talk about it in another tutorial).

TOP/*App/src/myCalculatorMain.cpp

Do not modify this folder, unless you need to.

TOP/*App/src/Makefile

- Here you will find a short description of some of the commands in the Makefile file.
- If you are not using any other external module you probably **don't need to modify** this file, but if you're using synApps, or other things you will need.

TOP=../.. Relative path until you reach the TOP folder, it's usually right


```

include $(TOP)/configure/CONFIG
#-----

# ADD MACRO DEFINITIONS AFTER THIS LINE

If you want, you can add definitions and stuff

#=====

#=====

# Build the IOC application

Build IOC
PROD_IOC = myCalculator
# myCalculator.dbd will be created and installed
DBD += myCalculator.db

# myCalculator.dbd will be made up from these files:
myCalculator_DBD += base.dbd           This line add the dbd's from epics base (all IOC's
need these).

# Include dbd files from all support applications:
#myCalculator_DBD += xxx.dbd           Here you add extra .dbd files that you
might need, normally when you will build bigger applications you will have a lot of extra dbd
files.

# Add all the support libraries needed by this IOC
#myCalcalatos_LIBS += xxx             Here you add extra libraries files that you might
need, normally when you will build bigger applications you will have a lot of extra libraries.

# myCalcalatos_registerRecordDeviceDriver.cpp derives from myCalcalatos.dbd
myCalcalatos_SRCS +=
myCalcalatos_registerRecordDeviceDriver.cpp   Registers the information read from
the database definition files. Don't need to be modified

```

```

# Build the main IOC entry point on workstation OSs.

myCalcalatos_SRCS_DEFAULT += myCalcalatosMain.cpp

myCalcalatos_SRCS_vxWorks += -nil-

# Add support from base/src/vxWorks if needed

# myCalcalatos_OBJS_vxWorks += $(EPICS_BASE_BIN)/vxComLibrary

# Finally link to the EPICS Base libraries

myCalcalatos_LIBS += $(EPICS_BASE_IOC_LIBS)

#=====

include $(TOP)/configure/RULES

#-----

# ADD RULES AFTER THIS LINE

```

Note that:

- Record instances (PV's) files have the extension ".db"
- Database definition files have the extension ".dbd".
- Note that the order in which .dbd files are added to a list variable doesn't matter, but the order of libraries does.

TOP/*App/Db

- Inside this Db folder there is makefile which will generate the databases. This folder is very important and should be treated with care.
- In this makefile you will add all the db's that you will use.
- You can do it two ways:
 - In the makefile, place the full path so the build can find the .db's.

```
DB += /opt/epics/extensions/src/synApps/support/motor-6-8/db/motor.db
```

```
DB += /opt/epics/extensions/src/synApps/support/motor-6-8/db/basic_motor.db
```

- You can put all the db's files the the Top/*App/Db folder and declare this files without any path.

DB += basic_asyn_motor.db

DB += IMS_extra.db

- You can create your own .db's. We will do it later.
- You do have to worry about .db's but not now, you can edit .db's later (after building IOC).
- This is what the makefile looks like:

TOP=../..

include \$(TOP)/configure/CONFIG

#-----

ADD MACRO DEFINITIONS AFTER THIS LINE

#-----

Optimization of db files using dbst (DEFAULT: NO)

#DB_OPT = YES

#-----

Create and install (or just install) into <top>/db

databases, templates, substitutions like this

#DB += xxx.db << Add .db's here

#-----

If <anyname>.db template is not named <anyname>*.template add

<anyname>_template = <templatename>

include \$(TOP)/configure/RULES

#-----

ADD RULES AFTER THIS LINE

TOP/iocBoot

- This Folder contains the makefile and another folder for the startup of the IOC.
- You Do not need to worry much about this folder because you can edit it later (after building IOC).

TOP/iocBoot/Makefile

- The Makefile just "run's" the installation into the Startup folder of ioc.

TOP/iocBoot/myCalculator

- The Startup folder of the IOC have two files, a makefile and st.cmd.

TOP/iocBoot/myCalculator/Makefile

- Create the "env Paths". env Paths is very important because it gives the st.cmd file the paths of things it needs.

TOP/iocBoot/myCalculator/st.cmd

- This file tells what will happen when the IOC runs.
-

Building the IOC

After you have checked all the files and stuff just go to ioc's TOP folder and run "make"

```
cd $EPICS_BASE/$WORKINGNAME
```

```
make
```

If you find some error along the way use the command `make clean uninstall` to undo the build.

```
make clean uninstall
```

Try figure out what is missing, correct it and run the command make again.

Built IOC

The built IOC will have two or three additional folders in the ioc's TOP directory:

- **TOP/Bin** This one has the executable file of the IOC.
- **Top/dbd** Has the .dbd file we set up in the Top/*App/src/Makefile
- **TOP/db** If you have set any .db in the Top/*App/Db/Makefile, you will have this TOP/db folder. If you haven't set, then you have to create this folder and place all your .db's in there.

CONGRATULATIONS

Now you have a IOC ready to go. Check the next tutorial to learn how to use it.