
amControls Documentation

Release 0.1

Argonne National Laboratory

Apr 22, 2022

CONTENTS

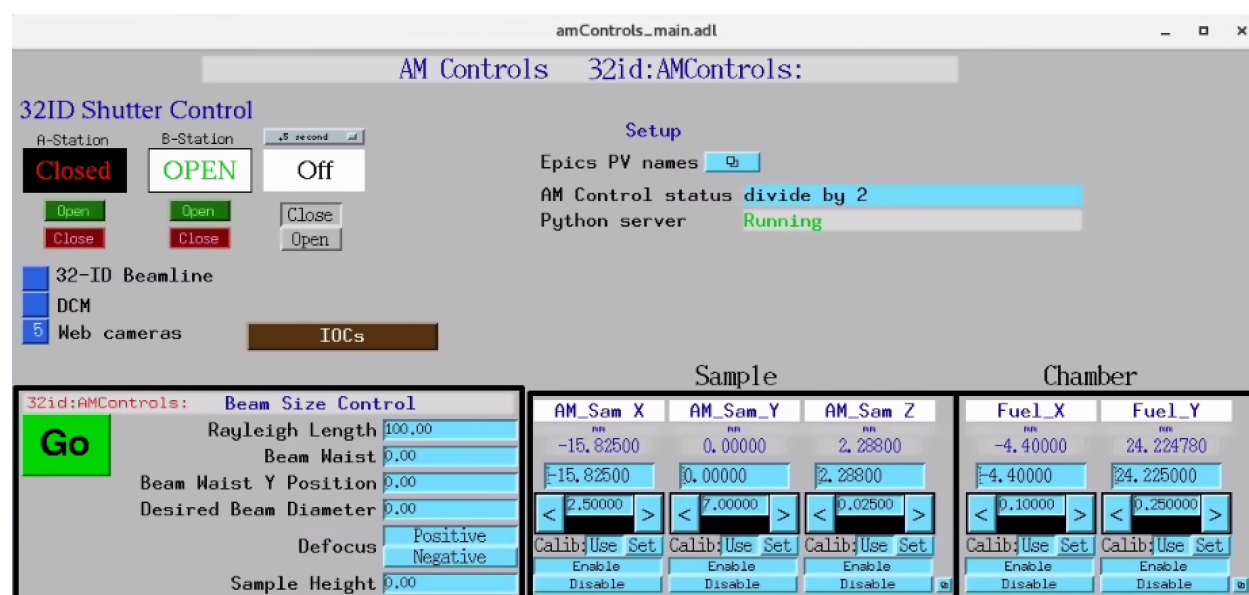
1	Content	1
----------	----------------	----------

CONTENT

1.1 About

This is a fully functional epics IOC that provide an example on how to create&serve epics PVs, how to connect to PV served by other epics IOCs (both referring to them as PV prefix or full PV name customizable at run time) and how to support a callback via python.

amControls is an EPICS IOC supporting the Additive Manufacturing at beamline 32-ID of the Advanced Photon Source.



1.2 Instrument

1.2.1 Laser

More infor

At beamline 32-ID we use ...

1.3 Usage

1.3.1 Start EPICS IOC

```
fast@merlot $ cd ~/epics/synApps/support/amcontrols/iocBoot/iocAMControls
fast@merlot $ ./start_IOC
```

at the end of the start up process you will get the IOC console:

```
Starting iocInit
#####
## EPICS R7.0.6.2-DEV
## Rev. R7.0.6.1-75-g91941af992f6c32ef4f4
#####
reboot_restore: entry for file 'auto_settings.sav'
reboot_restore: Found filename 'auto_settings.sav' in restoreFileList.
*** restoring from './autosave/auto_settings.sav' at initHookState 6 (before record/
↳device init) ***
reboot_restore: done with file 'auto_settings.sav'

reboot_restore: entry for file 'auto_settings.sav'
reboot_restore: Found filename 'auto_settings.sav' in restoreFileList.
*** restoring from './autosave/auto_settings.sav' at initHookState 7 (after record/
↳device init) ***
reboot_restore: done with file 'auto_settings.sav'

cas WARNING: Configured TCP port was unavailable.
cas WARNING: Using dynamically assigned TCP port 44367,
cas WARNING: but now two or more servers share the same UDP port.
cas WARNING: Depending on your IP kernel this server may not be
cas WARNING: reachable with UDP unicast (a host's IP in EPICS_CA_ADDR_LIST)
iocRun: All initialization complete
create_monitor_set("auto_settings.req", 30, "P=32id;R=AMControls:")
save_restore:readReqFile: unable to open file amControls.req. Exiting.
epics> auto_settings.sav: 8 of 8 PV's connected
epics>
```

If you do any modification to the **amControls_settings.req** or **amControls.template** files in:

```
~/epics/synApps/support/amcontrols/amcontrolsApp/Db
```

you need to rebuild the epics IOC:

```
epics> exit
fast@merlot $ cd ../../
fast@merlot $ make -sj
fast@merlot $ cd iocBoot/iocAMControls
```

and restart the IOC:

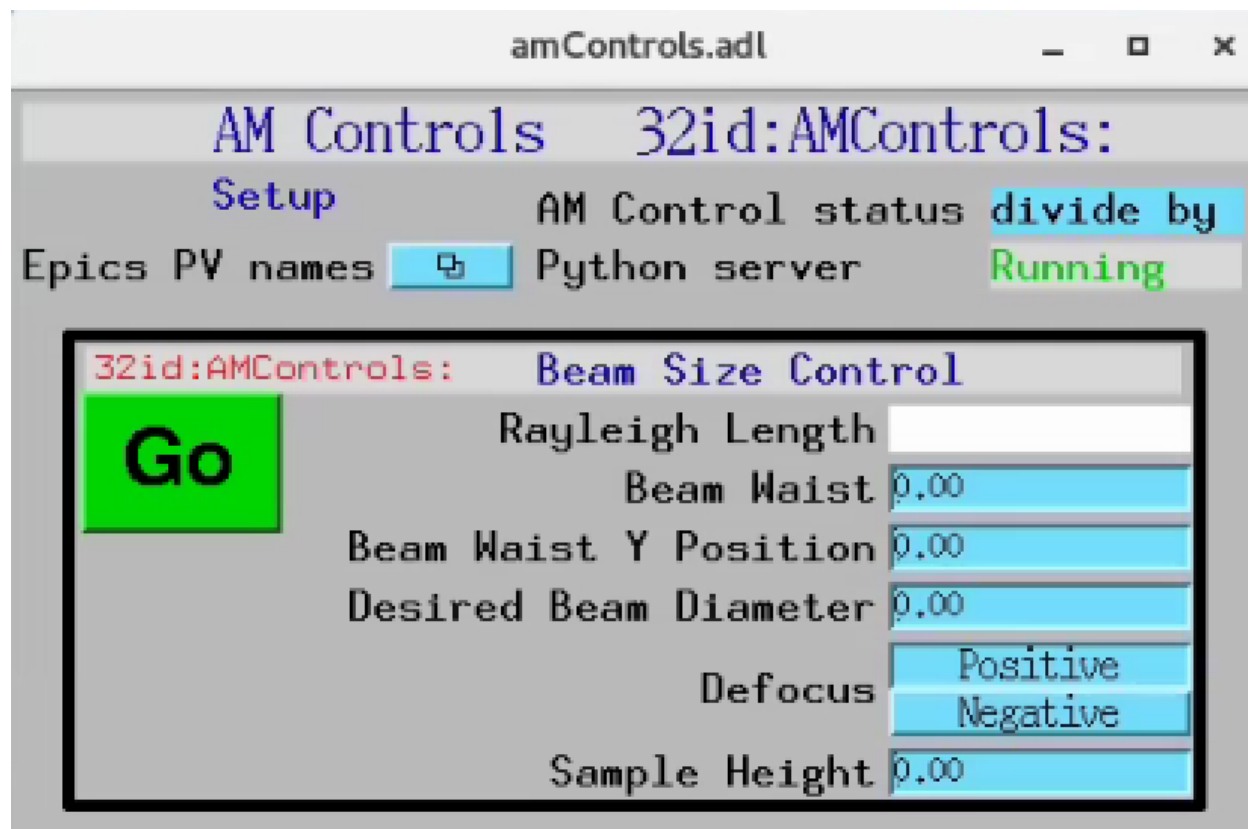
```
fast@merlot $ ./start_IOC
```

You can accomplish the same with a single line command:

```
fast@merlot $ cd ../../ ; make -sj ; cd iocBoot/iocAMControls ; ./start_IOC
```

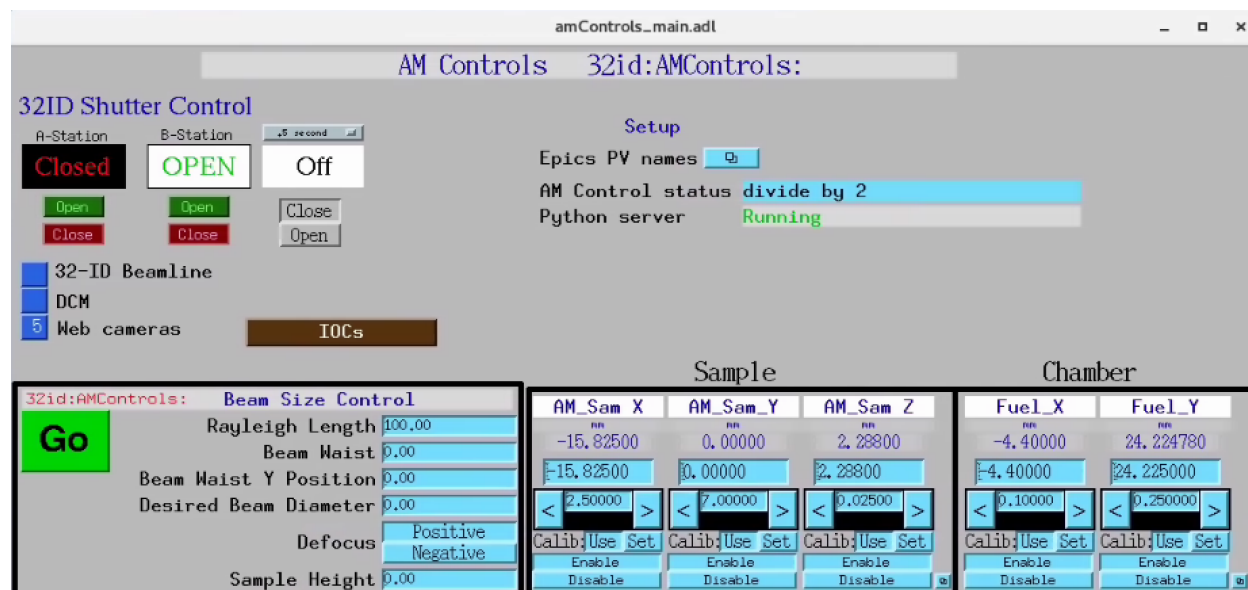
1.3.2 Start MEDM screen

```
fast@merlot $ cd ~/epics/synApps/support/amcontrols/iocBoot/iocAMControls
fast@merlot $ ./start_medm
```



or:

```
fast@merlot $ ./start_medm_user
```



1.3.3 Start python server

```
$ bash
(base) $ conda activate amcontrols
(amcontrols) fast@merlot $ cd ~/epics/synApps/support/amcontrols/iocBoot/iocAMControls
(amcontrols) fast@merlot $ python -i start_amcontrols.py
configPVS:
CameraPVPrefix : 2bmbSP2:
ExamplePVName : 32id:m1
DefocusSelect : Positive
RayleighLength : 50.00
BeamWaist : 0
BeamWaistYPosition : 0
DesiredBeamDiameter : 0
SampleHeight : 0

controlPVS:
Example : None
Go : Done
AMStatus :
Watchdog : -38

pv_prefixes:
Camera : 2bmbSP2:
>>>
```

If you do any modification to the python source code files in:

```
~/epics/synApps/support/amcontrols/amcontrols/
```

you need to rebuild the python server code:


```
>>> exit()
(amcontrols) fast@merlot $ cd ../../
(amcontrols) fast@merlot $ python setup.py install
(amcontrols) fast@merlot $ cd iocBoot/iocAMControls/
(amcontrols) fast@merlot $ python -i start_amcontrols.py
```

You can accomplish the same with a single line command:

```
(amcontrols) fast@merlot $ cd ../../; python setup.py install; cd iocBoot/iocAMControls/;
→ python -i start_amcontrols.py
```

1.4 Demo

amControls allows for ...

1.5 Install directions

1.5.1 Build EPICS base

Warning: Make sure the disk partition hosting ~/epics is not larger than 2 TB. See [tech talk](#) and [Diamond Data Storage](#) document.

```
$ mkdir ~/epics
$ cd epics
```

- Download EPICS base latest release, i.e. 7.0.3.1., from <https://github.com/epics-base/epics-base>:

```
$ git clone https://github.com/epics-base/epics-base.git
$ cd epics-base
$ make -sj
```

1.5.2 Build a minimal synApps

To build a minimal synApp:

```
$ cd ~/epics
```

- Download in ~/epics [assemble_synApps.sh](#)
- **Edit the assemble_synApps.sh script as follows:**
 1. Set FULL_CLONE=True
 2. Set EPICS_BASE to point to the location of EPICS base. This could be on APSshare (the default), or a local version you built.

For amcontrols you need

1. ASYN=R4-37

2. AUTOSAVE=R5-10
3. BUSY=R1-7-2
4. XXX=R6-1

You can comment out all of the other modules (ALLENBRADLEY, ALIVE, etc.)

- Run:

```
$ assemble_synApps.sh
```

- This will create a synApps/support directory:

```
$ cd synApps/support/
```

- Edit asyn-RX-YY/configure/RELEASE to comment out the lines starting with:

```
IPAC=$(SUPPORT)/  
SNCSEQ=$(SUPPORT)/
```

Warning: If building for RedHat8 uncomment **TIRPC=YES** in asyn-RX-YY/configure/CONFIG_SITE

- Clone the amcontrols module into synApps/support:

```
$ git clone https://github.com/tomography/amcontrols.git
```

- Edit configure/RELEASE add this line to the end:

```
AMCONTROLS=$(SUPPORT)/amcontrols
```

- Edit Makefile add this line to the end of the MODULE_LIST:

```
MODULE_LIST += AMCONTROLS
```

- Run the following commands:

```
$ make release  
$ make -sj
```

1.5.3 Testing the installation

- Edit /epics/synApps/support/amcontrols/configure to set EPICS_BASE to point to the location of EPICS base, i.e.:

```
EPICS_BASE=/APSShare/epics/base-3.15.6
```

- Start the epics ioc and associated medm screen with:

```
$ cd ~/epics/synApps/support/amcontrols/iocBoot/iocAMControls  
$ start_IOC  
$ start_medm
```

- A database file and corresponding autosave request file that contain the PVs required by the amcontrols.py base class.
- OPI screens for medm
- An example IOC application that can be used to run the above databases. The databases are loaded in the IOC with the example substitutions file, amControls.substitutions.

Defocus select

Record name	Record type	Description
\$(P)\$(R)\$(B)DefocusSelect	\$(P)\$(R)\$(B)DefocusSelect	Direction selector, Positive or Negative

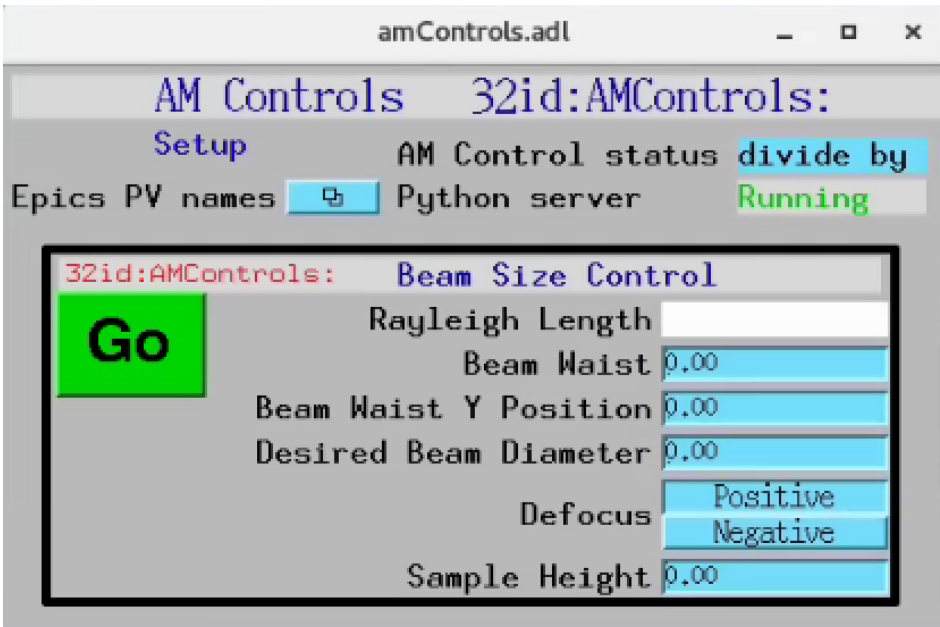
AM status via Channel Access

Record name	Record type	Description
\$(P)\$(R)\$(B)Go	\$(P)\$(R)\$(B)Go	Setting this record to 1 starts a control action.
\$(P)\$(R)\$(B)AMStatus	\$(P)\$(R)\$(B)AMStatus	This record will be updated with the scan status while scanning.
\$(P)\$(R)\$(B)ServerRunning	\$(P)\$(R)\$(B)ServerRunning	This record will be Running if the Python server is running and Stopped if not. It is controlled by a watchdog timer, and will change from Running to Stopped within 5 seconds if the Python server exits.

medm files

amControls.adl

The following is the MEDM screen amControls.adl during a scan. The status information is updating.



amControlsEPICS_PVs.adl

The following is the MEDM screen `amControlsEPICS_PVs.adl`.

If these PVs are changed amControls must be restarted.

