
COSMIC XPCS Documentation

Ronald J Pandolfi

Apr 29, 2021

Contents:

1	FastCCD Startup Process	1
2	XPCS system locations	3
3	Services on Tsuru	5
4	Troubleshooting Network Issues	7
5	Things to do	9
6	Tsuru and BL701-IOC Services and Configuration Guide	11
7	iptables	15
8	Indices and tables	17

FastCCD Startup Process

1.1 Cautions

Before running the FastCCD, check the state of the power supplies and cooling. The FCRIC and timing configuration can be modified live, however bias values should not be modified without turning off camera bias and clocks; otherwise damage to the sensors may result.

1.2 Setup the FastCCD

The FastCCD can be initialized and checked from fastcameraATCA1.dhcp.lbl.gov when coming online.

- Connect to fastcameraATCA1.dhcp.lbl.gov via SSH and start the VNC server using the following commands

```
vncserver -list
    shows last vnc process; the process shown may actually have ended, but you
    → can restart it with below commands

vncserver -kill :11
    kills the vnc server on window 11

vncserver :11 -geometry 2400x1400
    starts a vnc server on window 11 with a screen size of 2400x1400
```

- Start the VNC client and connect to fastcameraATCA1.dhcp.lbl.gov
- Run the auto-start script
~/CameraControl/python/fccd_auto_start.py Sets up the FCCD. When running this script the Camera Interface Node led panel's bottom-right light will cycle in order red>yellow>green as it configures the device.
- Check the fiber optics status
~/CameraControl/python/get_FOPS_Status.py This script prints the monitor values of the fiber optics hardware. It should show approx. 4.49 V and 2.71 A.

- Launch the FCCD gui
~/CameraControl/fccd_gui/CINController.py Launches the main
- Click [Connect]
- Under [Expose], select Exposure Mode -> Continuous

When all modules are masked in the fcris, the standard test pattern is sent. The fiber optic test pattern can also be sent; this is toggled with: - setFOtestpatter_on.py - setFOtestpattern_off.py

1.3 Starting Tsuru

Tsuru is fully configured to start automatically. The FCCD GUI must be closed for Tsuru to receive data. The FCCD GUI also interrupts the EPICS IOC. After closing the FCCD GUI, the EPICS IOC must be restarted.

1.4 Restarting the EPICS IOC

The EPICS IOC is configured to run as a *systemd service*. To restart it, use:

```
sudo systemctl restart epics.service
```

For more info on systemctl, see [Arch's systemd](#).

1.5 Notes

- Configure Tsuru networking
 - 2: ens4f0 = 10G -> 10.0.6.42 (Not sure if 5 or 6)
 - 4: enp179sf0 = Base -> 192.168.1.42

XPCS system locations

The COSMIC-XPCS beamline has controls resources spread across multiple systems.

Table 1: Controls Resources

Hostname	Static IP
tsuru.dhcp.lbl.gov	131.243.73.224
fastcameraatca1	131.243.73.253

CHAPTER 3

Services on Tsuru

Three services exist on Tsuru to automate startup of resources: a Caproto IOC service, an EPICS IOC service, and a Jupyter Hub service. Their unit files are at `/etc/systemd/system/`.

The log of these services can be read by `journalctl -u ...`. The services can be configured with `systemctl`.

3.1 Caproto IOC

```
[Unit]
Description=Caproto IOC Service
Requires=network.target
After=network.target

[Service]
ExecStart=/home/rp/PycharmProjects/alsdac/venv/bin/python /home/rp/PycharmProjects/
↪alsdac/alsdac/caproto/__init__.py

[Install]
WantedBy=multi-user.target
```

3.2 EPICS IOC

```
[Unit]
Description=EPICS IOC Service
Requires=network.target
After=network.target

[Service]
WorkingDirectory=/usr/local/epics/R7.0.1.1/support/areadetector/3-2/ADFastCCD/iocs/
↪FastCCDIoc/iocBoot/iocFastCCD
```

(continues on next page)

(continued from previous page)

```
ExecStart=/usr/local/epics/R7.0.1.1/support/areadetector/3-2/ADFastCCD/iocs/  
↪FastCCDIOC/bin/linux-x86_64/FastCCDApp /usr/local/epics/R7.0.1.1/support/  
↪areadetector/3-2/ADFastCCD/iocs/FastCCDIOC/iocBoot/iocFastCCD/st.cmd  
StandardInput=tty  
StandardOutput=journal  
TTYPath=/dev/tty50  
KillMode=process  
Restart=always  
  
[Install]  
WantedBy=multi-user.target
```

3.3 Jupyter Hub

```
[Unit]  
Description=Jupyterhub  
After=syslog.target network.target  
  
[Service]  
User=root  
ExecStart=/usr/bin/jupyterhub -f /etc/jupyter/jupyterhub_config.py  
  
[Install]  
WantedBy=multi-user.target
```

The above unit file is at `/lib/systemd/system/`.

3.4 Note

For more information, see [Arch's systemd](#).

<https://serverfault.com/questions/821575/systemd-run-a-python-script-at-startup-virtualenv>

Troubleshooting Network Issues

4.1 Step 1: Check Network Addresses (on tsuru)

After ssh'ing to tsuru, run:

```
ip address
```

Interfaces 2, 4, and 5 should be UP.

- 2 - fabric switch
- 4 - local network
- 5 - lab network

If DOWN, run Ron's setup scripts to configure the ip addresses for the network interfaces:

```
bash ~rp/setup.sh  
ip address
```

4.2 Step 2: Initializing Camera

We want to make sure the camera is initialize when testing the network connections.

First, log in to tsuru (vnc is helpful here to run Xi-CAM).

If the Controls GUI does not show in Xi-CAM Acquire after selecting `fastccd` device, check that you have `$PASSWD_MONGO` and `$USER_MONGO` set.

If initializing the camera (via the `Initialize` button) fails, check the service logs:

```
journalctl -f -u fastccd_support_ioc
```

This may show something like `socket.timeout=timed out`.

Re-check the network interface addresses (Step 1).

Retry the initialization of the camera in Xi-CAM.

4.3 Step 3: Check connection between tsuru and CIN

We will then want to test the connection between tsuru and the camera interface node.

CIN IP: `10.0.5.207`

On tsuru, run:

```
sudo ping -f 10.0.5.207
```

After a few seconds, CTRL-C and check packet loss (20% is usual).

Next, we will want to look at the connection from the fastccdATCA1 server to the fabric switch.

4.4 Step 4: Check connection between fastccdATCA1 and CIN

ssh to `fastccdatca1.dchp.lbl.gov` and repeat the ping flood from step 3.

This may also show packet loss of around 20%.

4.5 Step 5: Check fastccdATCA1 and tsuru network configuration

is this relevant??? – recording for note purposes, modify later

On `fastcameraatca1` and `tsuru`, check the `net.core.rmem_default` and `net.core.rmem_max` values:

```
systemctl net.core.rmem_default
systemctl net.core.rmem_max
```

Max should match default. If not, modify `net.core.rmem_default` to match the `net.core.rmem_max` in the following files:

- `/etc/sysctl.conf`
- `/etc/rc.d/rc.local`

Then, reboot:

```
sudo shutdown -r 0
```

After rebooting, reconnect and repeat steps 3 and 4.

CHAPTER 5

Things to do

Tsuru and BL701-IOC Services and Configuration Guide

This guide will show:

- how to start up IOCs on tsuru and bl701-ioc1
- which unit files need to be active
- how the network is configured on tsuru
- which network service should be active
- some useful commands for troubleshooting

6.1 Starting up IOCs

IOCs:

- **tsuru**
 - epics
 - caproto
 - fastccd_support_ioc
- **bl7011-ioc1**
 - delay_generator_ioc
 - lakeshore_ioc

6.1.1 tsuru

1. Check IOC Units Are Enabled

ssh to tsuru, then run:

```
systemctl list-unit-files --state=enabled
```

to see which services are enabled. The `caproto`, `epics`, and `fastccd_support_ioc` units should be enabled.

If they are disabled, they can be enabled as follows:

```
sudo systemctl enable epics caproto fast_support_ioc
```

2. Start/Restart the IOCs

```
sudo systemctl restart epics
sudo systemctl restart caproto
sudo systemctl restart fastccd_support_ioc
```

3. Check the Services

```
systemctl status epics caproto fastccd_support_ioc
```

4. Check the Network

```
ip address
```

This will show the ip addresses for the interfaces on tsuru.

- 2: enp25s0f0, fabric connection
- 4: enp179s0f0, local network
- 5: enp179s0f1, lab network

Make sure these interfaces have an **UP** state and have **inet** addresses set to valid IPv4 type addresses.

```
netctl list
```

This will show all the configured profiles. Active profiles will have a **‘*’** next to them:

```
* enp179s0f0
* enp179s0f1
* enp25s0f0
```

These three profiles (interfaces) should all be active.

6.1.2 bl7011-ioc1

1. Check IOC Units Are Enabled

ssh to `bl7011-ioc1`, then run:

```
systemctl list-unit-files --state=enabled
```


to see which services are enabled. The `delay_generator_ioc` and `lakeshore_ioc` units should be enabled.

Note: If they are disabled, they can be enabled as follows:

```
sudo systemctl enable delay_generator_ioc lakeshore_ioc
```

2. Start/Restart the IOCs

```
sudo systemctl restart delay_generator_ioc
sudo systemctl restart lakeshore_ioc
```

3. Check the Services

```
systemctl status epics caproto fastccd_support_ioc
```

6.2 Resources

6.2.1 tsuru

Unit File	Description
<code>caproto.service</code>	caproto IOC
<code>dhcpcd.service</code>	DHCP client
<code>epics.service</code>	EPICS IOC
<code>fastccd_support_ioc.service</code>	FastCCD IOC
<code>iptables.service</code>	firewall/routing
<code>mongodb.service</code>	Mongo database
<code>netctl.service</code>	net interface config

The only networking service running should be `:code:'netctl'` (with `:code:'dhcpcd'`)! If other networking services like `:code:'NetworkManager'` or `:code:'systemd-networkd'` are running, connection errors may occur!

6.2.2 bl7011-ioc1

Unit File	Description
<code>delay_generator_ioc.service</code>	Delay Generator IOC
<code>lakeshore_ioc.servie</code>	Temperature Sensor IOC

There is no custom configured networking service here. By default, something like the `:code:'NetworkManager'` services will be enabled.

6.2.3 systemctl

`systemctl list-unit-files --state=enabled` list all enabled installed units

`ls /etc/systemd/system` show system-level unit configuration files

ls /usr/lib/systemd/system show user-level unit configuration files

journalctl -u UNIT look at journal (logs) for the unit (service)

systemctl start UNIT start (activate) the unit(s)

systemctl stop UNIT stop (deactivate) the unit(s)

systemctl restart UNIT stop then start the unit(s)

6.2.4 netctl

netctl list list all profiles, active ones marked with a '*'

netctl status PROFILE check the status of the PROFILE

/etc/netctl directory where profiles (configurations) are located

netctl start PROFILE start the PROFILE (will make it active)

ip address show ip addresses on this machine

7.1 mongo

On tsuru, the iptables service is configured to accept incoming connections to mongo for only a few whitelisted internal IPs. All other attempts to connect to mongo will be rejected.

To see the current firewall rules, run:

```
less /etc/iptables/iptables.rules
```

Danger: When modifying rules, be sure to activate root's cronjob for auto-flushing and resetting rules; otherwise, lockout may occur!

You can toggle on/off the safety root cronjob by running:

```
sudo crontab -e
```

then removing/adding a '#' to the beginning of the line that contains the `/root/fwstop.sh` command.

You can list root cronjobs by running:

```
sudo crontab -l
```

This will show you if it is enabled or not (if it is commented out or not).

Warning: When you are finished modifying rules, it may be useful to save the current iptables rules to a backup file. Then, you will need to save the rules to `/etc/iptables/iptables.rules` by running `iptables-save -f /etc/iptables/iptables.rules`.

You will also need to disable the root cronjob for flushing the rules!

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`