

## **Diskless Boot of LinuxRT or Centos7 - Quick Start-up**

## **i. Definitions**

- BOLD: refers to files that need to be modified or generated by IOC developer

- *Italics: symbolic link*

## References (documents need to be updated)

- [Block Diagram](#)
- [FAQ](#)
- [Setup EtherCat](#)

## How-To setup the CPU and IOC boot directories

The instructions differ between production and development due to permission restrictions in production.

**Production:** lcls, facet, spear, acctest

1. Send email to Ken Brobeck requesting:
  - a. Which facility the CPU will boot (e.g. facet, lcls, spear, acctest). This indicates the path to the startup.cmd script and the DHCP server.
  - b. MAC Address of the CPU
  - c. The CPU EPICS target architecture (e.g. linuxRT-x86, linuxRT-i686, linuxRT-x86\_64, rhel7-x86\_64, etc)
  - d. create controls infrastructure for a new linuxRT CPU and / or IOC to a specific facility production server.
    - i. includes \$EPICS\_IOCS/<cpu,ioc> directories
    - ii. includes \$IOC\_DATA/<ioc>/<autosave, autosave-req, restore, iocInfo, archive, yaml> directories
  - e. add CPU boot information to the DHCP configuration file, and reboot the DHCP server. The DHCP Configuration file is at \$TFTPBOOT/dhcpd.conf, Ken, Arjun or Charlie will need the following details to update DHCP.
    - CPU Node name
    - CPU MAC Address
    - LinuxRT or Centos7

### Example Development DHCP snipped

```
host cpu-b024-pm01 {
    hardware ethernet 74:FE:48:16:3D:D5;
    fixed-address 134.79.217.162;
    option host-name "cpu-b024-pm01"
    if ( substring( option vendor-class-identifier, 0, 5 ) = "udhcp" ) {
        filename "/afs/slac.stanford.edu/g/lcls/epics/iocCommon/cpu-b024-pm01/startup.cmd";
        option root-path "afsnfs:/afs/slac.stanford.edu:/afs/slac.stanford.edu";
    }
}
```

1. Proceed to **Development Step #4**

## Development:

1. . Create the \$IOC\_DATA directory for your cpu and iocs as follows:
  - a. cd \$IOC\_DATA
  - b. mkdir iocInfo archive restore autosave autosave-req yaml
2. Create the \$EPICS\_IOCS cpu and ioc boot directories
  - i. cd \$EPICS\_IOCS
  - ii. mkdir cpu-<area>-<subsystem-prefix><2-digit#>
  - iii. mkdir sioc-<area>-<subsystem-prefix><2-digit#>
  - iv. Modify your application to add the cpu boot scripts. If your IOC runs on a Shared Platform cpu then modify the SharedPlatform IOC Application :
    1. add cpuBoot/<faciilty>/<cpu node>/st.cmd
    2. add any dhcp.config or ethercat xml scripts
    3. tag and push shared the SharedPlatform tag to production using cram.
3. Setup DHCP Server
  - a. Add cpu boot information to the DHCP configuration. Contact Ken to update the DHCP file and provide the following information:
    - CPU Nodename
    - CPU MAC Address
    - LinuxRT or Centos7

### Example LCLS DHCP snippet:

```
host cpu-lii10-pm01 {
    hardware ethernet 74:FE:48:55:1D:A9;
    fixed-address 172.27.72.103;
    option host-name "cpu-lii10-pm01";
    if ( substring( option vendor-class-identifier, 0, 5 ) = "udhcp" ) {
        filename "/usr/local/facet/epics/iocCommon/cpu-lii10-pm01/startup.cmd";
        option root-path "mccfs2:/export/mccfs/usr/local:/usr/local";
    }
}
```

4. Create the <cpu>.ipxe file at:

```
for LinuxRT: $TFTPBOOT/linuxRT/boot/ipxe/<cpu>.ipxe
for Centos7: $TFTPBOOT/boot/ipxe/<cpu>.ipxe

for 64-bit cpu: cp example.ipxe <cpu>.ipxe
for 32-bit cpu: cp example-32bit.ipxe <cpu>.ipxe
```

**Note: the buildroot version in the IPXE file must match the IOC Application buildroot version defined in configure/CONFIG\_SITE**

5. Setup CPU Boot directory files:

**Original Method: (steps 3b and 3c)**

- a. `cd $EPICS_IOC/cpu-<area>-<subsystem-prefix><2-digit#>`
  - b. Setup relative symbolic links to the common startup script and EPICS version:
    - `ln -s ../common/st.cmd.<type>cpu startup.cmd` where type is soft or linuxRT
    - `ln -s ../common/bin/<epics base version> epicsSpecificRelease`
  - c. Generate the file `kernel-module.cmd`. This file load kernels modules and drivers: `kernel-modules.cmd` for examples click [here](#).
- New Method: (steps 3d and 3e) - must be base release R7.0.3.1-1.0 or greater**
- d. Automatic:
    - `createLinuxRTCpu --help`
  - b. Manual:
    - i. `cd $EPICS_IOC/cpu-<area>-<subsystem-prefix><2-digit#>`
    - ii. Setup relative symbolic like to the TOP of the IOC Application first time only, thereafter use `cram` to upgrade link.
      - `ln -s ../iocTop/<app> iocSpecificRelease`
      - for linuxRT: `cp ../template/startup.cmd.linuxRT.cpu.<facility> startup.cmd`
      - for centos7: `cp ../template/startup.cmd.soft.cpu.<facility> startup.cmd`
        - replace `<cpu>` with your cpu name
        - replace `<full path>` with the full path to the boot directory in `$EPICS_IOC`

6.. Setup ioc directory files

- a. for linuxRT running iocsh: `cp ../template/startup.cmd.linuxRT startup.cmd`
- b. for linuxRT running cexp: `cp ../template/startup.cmd.linuxRT.cexp startup.cm`
- c. for centos7: `cp ../template/startup.cmd.soft.ioc startup.cmd`
- d. `cd sioc-<area>-<subsystem-prefix><2-digit#>`
- e. copy the ioc startup.cmd script from `$EPICS_IOC/template`
- f. edit `startup.cmd`
  - i. replace `<ioc>` with your ioc name and for cexp replace `<facility>`
  - ii. change path of the `EPICS_IOC_APP` if your `st.cmd` file is in a non-standard location.
- g. `cp ../template/screenrc screenrc`
- h. edit `screenrc` and replace `<ioc>` with your ioc name
- i. Setup a relative symbolic link to the TOP of the IOC Application release.
  - i. `ln -s ../iocTop/<app>/<version> iocSpecificRelease`

6. Modify `$EPICS_IOC/screeniocs` – Please note that this is a symbolic link to a CVS:

- a. `cd`
- b. `cd <work>`
  - cvs co `epics/iocCommon/All/<facility>` , where facility is dev,prod,facet,acctest
  - i. add line for cpu  
format: `cpu-<area>-<subsystem-prefix><2-digit #> <ts> <ts port #> <screen host> [ # comment]`
  - ii. add line for each ioc  
format: `sioc-<area>-<subsystem prefix><2-digit #> cpu-<area>-<subsystem-prefix><2-digit #> <account> [ # comment]`
  - iii. The "comment" field in **screeniocs** is optional and must be precede but "#".
  - iv. `cd epics/iocCommon/All/<facility>/screeniocs`
  - v. edit **screeniocs**
- c. verify that the file in use has not been modified without being committed.
  - i. `cd $EPICS_IOC/All/<facility>` , where facility is Dev,Prod,facet,acctest
  - ii. `cvs diff -r HEAD screeniocs`
  - iii. if there are issues, then send email to whoever made the change or to controls-software that you are about to update this file and uncommitted edits will be lost, so please commit your changes to CVS. If you don't get a response, you can either add the changes to CVS in addition to yours or just add yours.
- d. check out **screeniocs** from CVS into your work area
- e. commit your changes of **screeniocs** to CVS as follows:
  - i. `cvs commit -m "added cpu and ioc blah" screeniocs`
- f. update the `screeniocs` file in use
  - i. `cd $EPICS_IOC/All/<facility>`
  - ii. `rm -f screeniocs`
  - iii. `cvs update screeniocs`

**IOC Applications Specific RT constraints:**

1. Create script `<app>/iocBoot/<facility>/<ioc>/rtPrioritySetup.cmd` to set real-time priorities for threads
2. Load `rtPrioritySetup.cmd` from `<app>/iocBoot/<facility>/<ioc>/st.cmd`

[How-to Release a LinuxRT IOC Application:](#)

[How-to boot a linuxRT IOC from a Linux Host:](#)

1. Log onto the host machine for example [softgr@lcls-srv01](#) type the following:  
siocRestart <ioc>

#### How-to boot an IOC from the linuxRT host

The following scripts are located under the directory \$IOC/common and should be in PATH on the CPU.

1. siocStart.sh <ioc> - starts a specific ioc. However, if the ioc is up nothing will be done, if no ioc is specified all iocs on cpu not already up will be started.
2. siocStart.sh - starts all iocs not already running
3. iocConsole.sh <ioc> -d (starts the specified ioc, but does not attache to the screen session)
4. iocConsole.sh <ioc> -R (graceful reset)
5. iocConsole.sh <ioc> -r (fast restart)
6. iocConsole.sh --help

#### How to connect to CPU from a Linux host:

Use either iocConsole or ssh to CPU as follows:

1. iocConsole <cpu>
2. Using ssh
  - from lcls or dev: ssh laci@<cpu>
  - from facet: ssh flaci@<cpu>

#### How to check if IOC is running on the linuxRT CPU:

- from laci or flaci type: screen -ls

#### [How-to use screen program:](#)

- ctrl a d : detach from the screen session
- ctrl a [ : scroll up through the start-up messages.
- View all vioc's on host cpu:
  - logged in as laci or flaci for FACET iocs
  - screen -ls
- Attach to vioc: screen -r <ioc>
- Attaching to a linuxRT screen session multiple times with "laci"concurrently: (Note: from FACET use flaci)  
screen -x <ioc>
- Note: The "screen" program supports multi-display mode. When you attach an existing screen session, you can use "-x". It allows an user to attach to "a not detached screen session." The -r option allows a user to attach to a "detached screen session".

#### **Additional References:**

- [Additional Information on Cross-compiling EPICS Applications:](#)
- [LinuxRT Startup Script:](#)
- [How the initial boot-up process works:](#)