

iocConsole user, maintenance and troubleshooting guide

Table of Contents

- [Overview and operation](#)
- [Configuration](#)
- [Troubleshooting](#)
- [Maintenance and monitoring](#)

Overview and operation

- iocConsole is an EPICS Extension that creates a persistent console session into an IOC, EIOC or SIOC which can be accessed by multiple users simultaneously from their workstations. Each session can also optionally log to a file: `$IOC_DATA/iocname/screenlog.0`.
- iocConsole connects to RTEMS or other "hard" IOCs by telnet-ing to the IOC's terminal server port.
- iocConsole starts up a soft IOC process, and establishes a console session for the IOC process.
- iocConsole uses the Linux screen program (<http://www.slac.stanford.edu/grp/cd/soft/epics/extensions/iocConsole/screen.1.html>) to create and manage the console sessions and logging.

iocConsole EPICS Extension doc: <http://www.slac.stanford.edu/grp/cd/soft/epics/extensions/iocConsole/index.html>

contacts:

- Judy Rock
- Jingchen Zhou

technical details:

an iocConsole session has several moving parts:

- perl scripts that handle the session (start, stop, cleanup, etc.)
- Linux screen process that runs the console
- pipe file (socket) that associates the ioc name with the screen process id

basic commands

Once your IOC is configured in the system (see Configuration below) you can start an iocConsole session like this:

iocConsole *iocname*

e.g.

iocConsole sioc-sys0-al00

note: *doing iocConsole to a soft IOC that is down will start it!*

- (snip from the *iocConsole EPICS Extension doc* referenced above):
In the iocConsole session, the most important screen commands are ("C-" below indicates the ctrl key)
 - C-a d : Detach the screen session from this terminal (C-a C-d also works).
 - C-a ? : Show all screen commands.
 - C-a * : Show a list of all users attached to this screen session.
 - C-a [: Enter copy/scrollback mode (C-a C-[or C-a <esc> also works). In copy/scrollback mode you can use 'vi' or 'emacs' commands to move around the scrollback history. Some useful scrollback mode commands are:
 - Arrow keys move line by line or column by column.
 - h, j, k, l move line by line or column by column (vi-style).
 - + - move one line up/down.
 - C-u C-d move up/down one half a screen.
 - C-b C-f move up/down a full screen.
 - g moves to the beginning of the scrollback history.
 - % preceded by a number jumps to that percentage of the scrollback.
 - / or ? followed by some characters followed by a return searches forward or backwards for those characters.
 - <esc> exits copy/scrollback mode.
 - **important:** do not type "exit", or do ctrl-C from within a soft iocConsole session: this will terminate the soft IOC process! Instead use C-a d to detach from the screen session.
- iocConsole *iocname* -cleanup
as mentioned above, iocConsole has a couple of components that need to be in sync in order to work properly. iocConsole *iocname* -cleanup can help (see troubleshooting section below)
 - **please note:** doing iocConsole -cleanup will kill any concurrent iocConsole sessions - i.e. another colleague interacting with the IOC.
 - **please note: doing iocConsole -cleanup for a running soft IOC will kill the IOC process - please use with care!**

Configuration

screeniocs

To use iocConsole, your IOC must be configured in the system screeniocs file (**contact Judy Rock to add your IOC to screeniocs in production**)
When iocConsole starts up a session, it connects to the IOC using info found in the screeniocs file. screeniocs is CVS-ed.

- screeniocs location :
\$IOC/All/Prod/screeniocs or facet_screeniocs
- screeniocs contents:
 - RTEMS IOC:
 - ioc name
 - terminal server
 - port
 - screen host (where the shared screen session runs)
 - Soft IOC
 - ioc name
 - executable
 - screen host (where the shared screen session and the soft IOC process run)

screen logging

logging turned on is the default (not need for action on your part to make it happen)

- default screen logging parameters are found in the \$IOC/Prod/All/screenrc file
 - to override defaults, copy screenrc into your own ioc's \$IOC/*iocname* directory and edit it
 - for example, to turn off screen logging, **in your own local copy** comment out the "deflog on" line: "#deflog on"

Troubleshooting

iocConsole's error messages and conditions are not always 100% clear!

Feel free to call Judy Rock (or Jingchen Zhou) for assistance.

Here are a few situations you may encounter at some point:

- **Your IOC hasn't been added to screeniocs yet:**
looks like this:
[softegr@lcls-builder script]\$ iocConsole xyz
Error: iocConsole is terminating...ioc was not included in screeniocs file
- **Unsuccessful attempt to start an iocConsole session, with *no socket found* error message from iocConsole:**
looks like this:
screen writes out a message with something like "blah blah blah **no socket found** blah blah blah" and returns to the Linux prompt.
<when a real example arises, I will cut/paste it in here>
 - Possible causes: **soft iocs - for a soft ioc, this error condition often indicates that the soft IOC cannot start due to some IOC software or configuration problem, or has hung in some way.**
 - the IOC process has crashed.
 - rare: the IOC process has hung, but is still "running"
 - the IOC is incorrectly entered in screeniocs (e.g. wrong directories or executable)
 - Things to try **soft iocs**:
 - check screeniocs to make sure the data for your IOC are correct
 - have a look at the screenlog file: \$IOC_DATA/*iocname*/screenlog.0. There are often error messages from the IOC indicating why it's having problems. Problems previously observed in screenlog.0 include:
 - \$IOC/*iocname* bin directory or executable name not found (problem with make)
 - segmentation fault - IOC executable cannot run
 - for a hung but running soft ioc: as <adminuser> on <softIOCmachine>: ps -ef | grep <iocname>
 - If a process exists, but you can't iocConsole to it, this indicates a hang in the process. You'll need to kill the process from the <adminuser>@softIOCmachine command line. Then try iocConsole again.
 - Possible causes: **hard iocs**
 - the IOC or terminal server are powered down, or not connected to the network
 - there is a hardware problem with the terminal server or IOC
 - the IOC is incorrectly entered in screeniocs (e.g. wrong port)
 - the terminal server port needs resetting.
 - prior iocConsole session needs cleaning up
 - Things to try: **hard iocs**
 - check screeniocs to make sure the data for your IOC are correct
 - check to make sure the IOC and terminal server are up, and connected properly to the network (ping, etc.)
 - try resetting the terminal server port for the IOC:
iocTSmgr *iocname*
 - try iocConsole -cleanup
(you must be logged in as iocegr or fiocegr to run iocTSmgr)
 - iocConsole *iocname* -cleanup
- **Unsuccessful attempt to start an iocConsole session, with *dead screen* error message from iocConsole:**
looks like this:
...
iocScreen: /home/screen/bin/screen -r -S laci/ioc-li17-ky00
There is a screen on:
10660.ioc-li17-ky00 (Dead ???)
Remove dead screens with 'screen -wipe'.
There is no screen to be attached matching ioc-li17-ky00.
...
 - Cause:
 - a previously running SCREEN session was killed without deleting its socket pipe file.

- To fix:
 - `iocConsole <iocname> -cleanup`
 - *please note: this will kill a soft IOC that is running.* (unlikely that the ioc can run under these circumstances though)
- **Hard IOC console session starts and connects to the terminal server, but there is no EPICS prompt or response from the IOC**
 looks like this:
 Trying 172.27.68.102...
 Connected to ts-li20-nw02.slac.stanford.edu (172.27.68.102).
 Escape character is '^]'.
 < then no response to a carriage return or other IOC command >
 - Possible causes:
 - the IOC is powered down, or not connected properly to the terminal server
 - the IOC is having hardware or software problems and needs restarting, or other application-specific action
 - the IOC is incorrectly entered in screeniocs (e.g. wrong port)
 - the terminal server port needs resetting.
 - Things to try:
 - check screeniocs to make sure the data for your IOC are correct
 - check to make sure the IOC and terminal server are up, and connected properly to the network (ping, etc.)
 - try resetting the terminal server port for the IOC
`iocTSMgr iocname`
 (you must be logged in as iocegr or fiocegr to run `iocTSMgr`)
 - Reboot or power cycle your IOC
- **IOC console session stops logging** (also see the maintenance section below - logging is monitored daily by a cron job)
 looks like this: `screenlog.0` is missing OR `screenlog.0` is not growing as expected
 There are a couple of possible causes:
 - your IOC has its own copy of `screenrc` with "deflog on" commented out
 - check to see if there's a `$IOC/iocname/screenrc` file
 - if so, make sure the "deflog on" is not commented out ("#deflog on")
 - the `iocConsole` session has terminated or is in an inconsistent state
 - **for HARD IOC only** do `iocConsole -cleanup`:
 - First do
`iocConsole iocname -cleanup`
 - then
`iocConsole iocname`
 - the screen process logging function has hung up somehow (this happens most commonly for soft IOCs, but can probably occur for all IOCs)
 - if `screenlog.0` exists, make a backup copy
`cd $IOC_DATA/iocname`
`cp screenlog.0 screenlog.0SAVE`
 - `iocConsole iocname`
 - within the `iocConsole` session, toggle logging off and then on again:
`ctrl-a H`
`ctrl-a H`
 then exit `iocConsole` with `ctrl-a d`

Maintenance and monitoring

`iocConsole` hosts and sessions

- **LCLS**
 - All LCLS hard and soft IOC processes and screen sessions run on `lcls-daemon1` EXCEPT:
 - The klystron support soft IOC processes and screen sessions run on `lcls-daemon0` (due to high loads)
 - The process and screen session host is configured by the entry in the `screeniocs` file "Host where screen is run" column (the last column)
- **FACET**
 - All FACET hard and soft IOC processes and screen sessions run on `facet-daemon1`.

`screenlogd.bash`

For each accelerator facility (LCLS and FACET) a nightly cron job running the `screenlogd.bash` script handles `screenlog.0` file cleanup and maintenance of screen logging `iocConsole` sessions:

- manages `screenlog` backups
 - `screenlog.0` files larger than 1MB are cycled into a series of backup copies, with timestamp extensions. 5 backup copies are kept, with the oldest deleted first. If you would like to keep more than 5 copies, please copy them to another directory.
- manages `iocConsole` sessions to maintain screenlogging
 - "hard" IOCs
 - if a hard IOC's `iocConsole` screen session is found to be absent, `screenlogd.bash` attempts to restart an `iocConsole` session. If the restart fails, `screenlogd.bash` tries to reset the IOC's terminal server port using `iocTSMgr`, and then retries establishing an `iocConsole` screen session.
 - "soft" IOCs
 - `screenlogd.bash` does NOT restart `iocConsole` screen sessions for soft IOCs, since this will restart the IOC in production.
 - `screenlogd.bash` reports soft IOCs that are missing `screenlog.0` files (by e-mail to Judy Rock and Jingchen Zhou)
 - **to re-establish logging for a soft IOC missing `screenlog.0`:**
 - `iocConsole` to the soft IOC
 - in the IOC's console session:
`ctrl-a H` to stop screen logging

ctrl-a H again to restart screen logging
ctrl-a d to exit from the iocConsole session

iocConsole mass restart scripts

If a server that supports iocConsole (lcls-daemon1, lcls-daemon0, facet-daemon1) goes down for some reason, iocConsole sessions are often left in an inconsistent state.

When the servers come back up again:

- **soft IOCs** are automatically restarted upon system startup, which includes an automatic cleanup and restart of their iocConsole sessions.
 - **hard IOCs** require manual intervention to cleanup and restart their iocConsole sessions (for screenlogging). Here are the scripts that run the cleanup and iocConsole/screenlogging restart en masse, that should be run after system restart. They start iocConsole console sessions and will execute iocTsmgr to reset terminal server ports that need it.
 - **LCLS**
/usr/local/lcls/tools/script/restartALLIOCLogging.bash
 - **FACET**
/usr/local/facet/tools/script/restartALLIOCLogging_facet.bash
-