How to operate CODA2.0, and Hall B online FAQ
S.P. Barrow, TJNAF, FSU
September 25, 1997
A copy of this postscript file is kept at /usr/local/clas/doc/coda_2.ps

Contents

1 Introduction

A few days before the December 1996 commissioning run of Hall B, The Hall B online group released a document to guide first time users of CODA2.0 through a tutorial, using real hardware. That tutorial hardware has been rearranged, and some of it is now down in Hall B. The remaining hardware still works, but the tutorial is out of date. This document will provide a more general outline of the steps needed to start a runcontrol session. It's worth pointing out that there is a great deal of documentation about the Hall B DAQ system on web pages. A good place to check is http://clon00.cebaf.gov/clas/. For some topics the web pages go into much greater detail about the inner workings of the online system than is presented here. At some point the FAQ section of this document may also be added to a web page.

First, walk/ride to the Hall B counting house. Find an unoccupied clon workstation, choose one of CLON01, CLON02, CLON03, or CLON04. Login as clasrun.

2 Message logging (NEW SECTION, IMPORTANT STUFF)

After the June run ended, the online group finished the work on the message logging for the various components of the online system. Therefore it's no longer essential to have a telnet connection for every ROC that participates in the configuration. If you are reading this section for the first time, I would recommend that you do the run the way you are accustomed to, namely with a telnet window for each ROC. Try the new message logging GUI concurrent with the telnet windows, and it should be quite clear to you how things have changed. Of course this is the only section of this document that was written after the advent of message logging, so the other sections may still refer to a telnet window. As long as you are aware of this there should be no confusion.

To start up the preliminary message logging GUI, first create a window on clon00. Set the DISPLAY variable to your local workstation. Now perform the following actions:

- Type cmlog $< RET >$
- Under the "File" menu, select "Connect"
- Under the "Options" menu, select "Update", then click the "OK" button

As soon as any component of the online system executes a CODA transition messages should start appearing in this window.

3 Runcontrol

The data acquisition software used by Hall B is CODA2.0, which consists of five states and seven transistion, as shown in Figure 1. Only when all components of the online system are "active" will data acquisition occur.

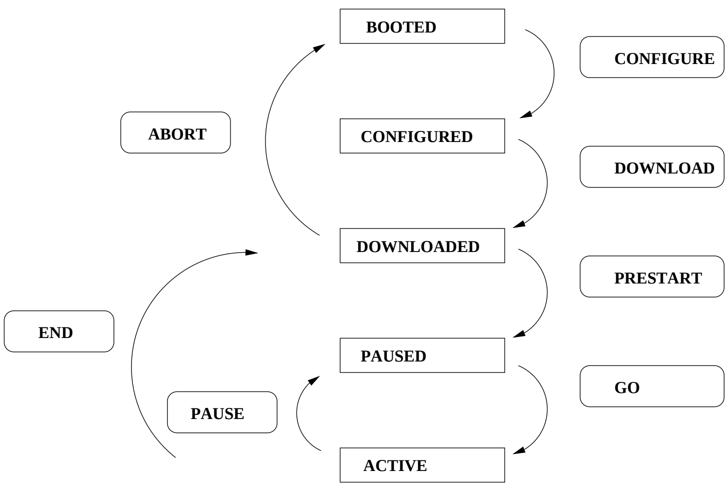

Figure 1 States and transitions of CODA2.0.

Most readers of this document already have some experience with runcontrol, and know that runcontrol uses sessions and configurations. For Hall B the session is "clasprod", the configuration will be chosen from roughly a dozen candidates. To view the list of available configurations start up “codaedit” on any UNIX window. An interface will pop-up. Find the “File” menu from the list of menus along the top of the GUI. Perform the following actions:

- Select “open database”
- click on “clasrun”
- click on “ok”
- Select the menu “Configuration”
- click on “open”
- select the configuration “TAG_TEST” from the list
- (EDITOR’S NOTE: The configuration “TAG_TEST” no longer exists, choose another.)

These actions have selected the configuration TAG_TEST. Currently there are quite a few configurations available. Many of the configurations are removed before each production run. There should now be some icons in the main display area of the GUI. Study these icons carefully. Note that each icon has an IP address associated with it, and that the IP address is indicated above the icon. This is a schematic of this particular configuration. There are three synchronous ROCs, one trigger supervisor, an event builder and an event recorder. For this upcoming run we may still use telnet connections to each ROC, so for this configuration “TAG_TEST”, we would need a telnet connection to each ROC shown in this figure. These ROCs are single user environments; if someone already has a telnet connection to the ROC it will be impossible to create a new connection. To determine if there are any telnet connections type “find_telnet”. This will search all the clons and summarize the telnet connections to crates in the hall. If you want to stop one of these telnet connections type “kill_telnet crocXX”, where XX is the IP alias. The script “kill_telnet” works for all ROCs, even for names such as clastrig2 and scaltag.

To create a telnet connection from any unix window type

- telnet IP address, for example croc08, clastrig2, etc.
- exit
- $< RET >$

This should start up the codaroc shell (clasprod::crocXX), and exit from the vxworks shell (->). Watch the messages as they scroll by. Without knowing anything they can still give you a good feel for the status of the ROC. The ROCS are no longer password protected. Many of the crates are connected to terminal servers. The website http://clon00.cebaf.gov/clas/ explains how to use the terminal servers.

Sometimes it’s useful to start up an event builder and event recorder by hand, but it’s not necessary. If they don’t exist runcontrol will start them automatically. Since this particular configuration uses EB1 and ER1, they can be started on two separate windows (on clon00) by typing

- /usr/local/clas/bin/coda_eb -s clasprod -o “EB1 CDEB” -i

and

- /usr/local/clas/bin/coda_er -s clasprod -o “ER1 ER” -i

From a different unix window, type “rlogin clon00”. Once the login is finished type

- setenv DISPLAY clonXX:0.0, where XX=01,02,03, or 04
- runcontrol -o

The runcontrol GUI will appear on your workstation. Customize to suit your taste. Click on "connect", then in the new widget select "clasrun" for "database" and "clasprod" for "session". Leave the database directory empty. Click "spawn". You should now be seeing the main runcontrol GUI. Click on the "load" button. Now you need to choose exactly which run you wish to start. All the commands you have entered in the runcontrol GUI up to this point are generic to all runcontrol sessions, with possible changes in the session name and database name the only caveats.

From this point forward the actions you take apply to a particular configuration. Click on "configure". Choose "TAG_TEST" from the pop-up widget. If there is a "configure failed" message, try again. Under the "options" menu, select "How to monitor components". Deselect the option "End a run automatically when errors are found". Now click

- download
- prestart
- go

in that order. If everything worked then you have just started a run.

4 ROCs, rebooting of

This section will explain how to reboot a ROC if the need should arise. There are now two ways to reboot the fastbus crates. As of this writing all the crates in the hall contain the cards necessary to perform a remote reset using a LAT terminal server.

Select one of the telnet windows. From the coda_roc shell (the "clasprod::crocXX" prompt) type

- $< CTRL - C >$
- reboot

to reboot the ROC. From a vxworks prompt ("->") simply type

- reboot

so type whichever is appropriate for your situation. If you see no prompt type <ret> in the telnet window until you see a prompt. If that doesn't work type $< CTRL - C >$. Once the ROC is rebooted the connection will close. Wait a few seconds (~ 20) while the crate reboots. Since you have already made a telnet connection from the window, the telnet command can be resurrected with the $\uparrow$ key. If instead of a telnet connection a terminal server is used, it's possible to watch the power PC as it reloads its kernal and reboots.

To create a telnet connection from any unix window without using the $\uparrow$ key, type

- telnet IP address, for example croc08, clastrig2, etc.
- exit
- $< RET >$

This should start up the codaroc shell (clasprod::crocXX), and exit from the vxworks shell (->).

4.1 Remote reset of ROCs

It is now possible to use a GraphOn terminal in the counting house to remotely reboot the fastbus crates in the hall. Eventually all the crates will have this capability. This is done through a handful of LAT connections that daisy chain to several ROCs each using an RS485 cable. Each ROC contains a microprocessor card that responds only to commands directly addressed to it. One of the commands each microprocesser understands is “RES”, which reboots the crate, and is equivalent to performing a hard reset of the Power PC. Use the GraphOn terminal which is currently resting on the bookcase near the glass door that opens into the SMP room.

The crates that are remotely resetable are divided into five terminal servers; the first four have sessions names:

Session	ROC Name	LAT name
Session 1	SC3, SC12, DC1, DC2	LAT4P7
Session 2	CCTAG1, TOF1, SCALE1	LAT1P7
“	SC8, EC1, EC2	“
Session 3	DC3, DC4, DC9, DC10, CLASTAG	LAT5P7
Session 4	DC5, DC6, DC7, DC8	LAT10P7
-	LAEC	LAT6P7

The first step to rebooting a ROC is to find to which session you are currently connected. Type

- Show Session

If you are connected to a session skip down a few steps to the connected area. If you want to resume a session which is already connected, type

- RESUME 1 $< RETURN >$

where the 1 can be replaced by whichever session you want (1, 2, 3, or 4). If you are having trouble getting through on an existing connection, you can close it with the command “DISCONNECT SESSION 3”, where in this example session 3 is closed. you should then try to reopen the session.

The last possibility is that you are not connected to any session, and you will need to reconnect to a session. At the local “>” prompt type

TYPE	For
CONN LAT4P7	Session 1
CONN LAT1P7	Session 2
CONN LAT5P7	Session 3
CONN LAT10P7	Session 4

At this point you are prompted for a password which is HALLBC. Once you are connected you must type the following (all in capital letters):

- !NAME:

where “NAME” is the name listed above for the ROC that you want to reboot. There is no need to press return after the colon. A new prompt will appear on your screen: NAME>. You are now connected to the ROC. At this prompt you have two options. One is to type STA. This will give you the status of the ROC. The other option is to type RES which will reboot your crate.

Once you have finished using the GraphOn terminal, press the “break” key, F5. This will return the terminal prompt back to the “>” prompt, and makes things easier for the next user. Thanks to Si Mcaleer for help with this section.

5 Ending a run, starting the next

Click on the "end" button on the runcontrol GUI. Now, on the runcontrol GUI, click on the following buttons:

- prestart
- (configure the trigger supervisor with the ts_config GUI at this point)
- go

to start another run. If you are having the problem that the event files are frequently empty, follow the steps that are discussed under the item "The event file is empty" in the FAQ section.

6 Trigger supervisor interface (NEW SECTION)

We now have a GUI which is used to configure the trigger supervisor directly, and is shown in the accompanying figure. It is no longer necessary to edit and recompile the trigger supervisor readout list every time you wish to change to a new configuration or enable a new trigger bit. To start up this GUI type "ts_config" (of course you'll need to set your display variable before typing this). This GUI mainly consists of columns of checkbuttons. The columns of checkbuttons are sandwiched between two labels. The yellow foreground label to the left indicates the IP alias for that crate. The blue background label to the right indicates the BOS data that originates from each crate. Note that between these two labels there are two checkbuttons per fastbus crate (the crates that only have one button are the scaler VME crate). The left button with the white background allows you to choose whether or not you want to save the raw BOS bank from each crate for the upcoming run. By default all these buttons are selected. The buttons on the right are used to inform the trigger supervisor that this crate is participating in the upcoming run. For the configuration you wish to run, select (i.e. make the green dot appear) all the checkbuttons that correspond to synchronous ROCs for that configuration. Then select which crates will include the RAW banks. Finally select only the trigger bits you wish to use in the upcoming run. If you are prescaling the triggers set those values also.

Once all that is done perform the usual CODA transitions needed to begin the next run. After the "prestart" transition but before the "go" transition, click the "Set trigger bits", "Set prescale factors", "Set acknowledgement bits" buttons of the ts_config GUI. This will configure the trigger supervisor exactly as you have specified. To repeat:

- Click the "Set trigger bits", "Set prescale factors", "Set acknowledgement bits" buttons **AFTER THE "PRESTART" BUT BEFORE THE "GO" TRANSITIONS.**

Failure to follow this sequence will result in a seemingly unpredictable turn of events. The "Set raw banks" button can be pressed at any time, this information is not cleared by the prestart transition, and will remain in effect until the relevant crates are rebooted.

If for some reason you need to reboot the trigger supervisor (clastrig2), the next time you use the ts_config GUI you will get an error message. This is because it's complaining about the fact the coda_roc shell which used to run on clastrig2 had disappeared. Get rid of the error message and "ts_config" should keep working, but if it doesn't, then "exit" the current ts_config and start up a new one. Likewise, the first time you press "Set raw banks" after rebooting fastbus crates you may see an error message pop-up. Again, don't worry about it.

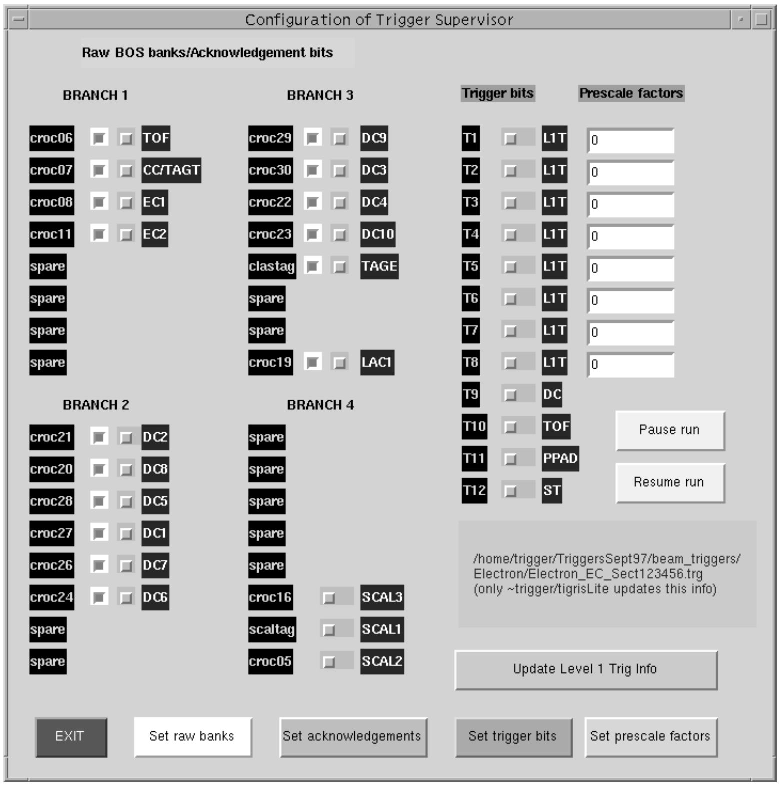

Figure 2 Trigger supervisor GUI.

7 Pausing a run (NEW SECTION)

The "ts_config" GUI discussed in the previous section now has two new buttons. One of these buttons, "Pause run" switches the trigger supervisor out of the "go" state, the other "Resume run" switches it back to "go" state. These buttons can be used to pause a run. For example, suppose the beam is unstable or just stopped, but you have every reason to believe it will be back momentarily. In this case press the "Pause run" button. Events will no longer accumulated, but the fastbus crates will remain in "active" state. Under the current arrangement almost all scalers will stop counting while a run is paused. Only some beamline scalers will continue to count. Once the beam is back and stable press the "Resume run" button.

8 RAID partitions swapping (NEW SECTION)

There is now a script to swap the RAID disks and write the current disk's data to tape silo. The following is taken directly from an email message by Mike Johnson. /usr/local/system/dumpdisk does the following:

- Kills all processes using /raid or /access1
- Removes everything in /access1 and remounts the empty partition as /raid. This happens very quickly so that the ER can start up again.
- Takes the full partition and dumps all non-zero files to tape. (Actually, to another RAID acting as a buffer before the tape.)

- Remounts the full partition as /access1 for online analysis.
- exits

Note that if any errors occur, a systems guru (Mike Johnson, Dave Rackley..) will have to be called before the next swap can happen. Also note that any “bogus” files in /raid/stage_in will still be copied over to tape silo. Clean any garbage out of /raid/stage_in before running dumpdisk, or just don’t put any there in the first place.

Here is the command to type to execute the swap:

- sudo /usr/local/system/dumpdisk

This won’t work without the ”sudo”. Only the clasrun account can use dumpdisk, and you will be prompted for the clasrun password. This can only be run from clon00.

9 Terminal servers

As with many topics in this document, there is a lengthy discussion of terminal servers at the website. This section will only review how to use the terminal server from the VAX account. For every serial connection you wish to establish you will need a telnet connection to latsrv.

- telnet to latsrv, login as clon, enter password
- type “set host/lat lat2pX”, where $1 \leq X \leq 8$
- enter the password, same as clon password from above

This will establish a serial line to lat2pX, which is a useful way to watch the crate reboot. The list of serial connections is presented in the following table. These connections sometimes change, so be prepared for discrepancies.

host	IP alias	Name
lat2p1	croc10	pretrigger 1
lat2p2	croc06	TOF
lat2p3	croc05	scaler
lat2p4	clastrig2	TS
lat2p5	croc09	CAMAC
lat2p6	croc11	EC2
lat2p7	croc07	CC
lat2p8	?	?
lat4p3	croc27	DC1
lat4p6	croc21	DC2
lat5p1	croc30	DC3
lat5p2	croc22	DC4
lat10p1	croc28	DC5
lat10p2	croc24	DC6
lat10p3	croc26	DC7
lat10p4	croc20	DC8
lat5p3	croc29	DC9
lat5p5	croc23	DC10

If you want to terminate the serial connection type “$< control > \backslash$”. If the serial connection does nothing but respond to the “$< RET >$” key, then there exists a telnet or rlogin connection to that crate which is negating the serial connection. Close the telnet connection and the serial connection should work.

10 Scaler display

If you are following the steps of this tutorial, you are using the configuration TAG_TEST, which includes scaler events. This section outlines the steps needed to view the scaler information with the scaler GUI. It is not necessary to do anything in order to insert the events into the data stream; that has already been done. There is a process that connects to the DD_RING and sends the scaler events to a GUI. Check to see if it is already running by typing "find_all dd2scaler". If it's already running it's not necessary to start another one. If you need to start this process, the following environment variable must be set on a clon00 window:

- setenv DD_NAME clasprod

where clasprod is the name of the DD_RING currently being used. Use this same clon00 window to start the process that plucks the scaler events from the DD_RING:

- dd2scaler -a clasprod >/dev/null

Finally, start up the scaler GUIs. The scaler GUI doesn't use the DD_NAME variable, so this program can be run from any window.

- cd /home/clasrun/scalers
- start_scaler

The vertical axes of these scaler displays can be either linear or log scale. The "Print" button will create a postscript file of the current scaler display, but does not send this file to the printer automatically. The .ps can be sent to the printer in the Hall B counting house by typing "lp -d clonhp XXXXX.ps" from the /home/clasrun/scalers directory, where XXXXX.ps is the name of the postscript file. For this TAG_TEST configuration, only the TAGGER GUIs have the potential of displaying any scaler data. The other GUIs might as well be closed.

If no scaler events are plotted in the scaler display, check to be sure sca_server is running. You can do this by typing "find_all sca_server". If it's not running, start it by typing "sca_server -a clasprod >/dev/null &".

Recently John Hardie prepared a much more complete summary of the scaler system. A copy of that document is kept at /usr/local/clas/doc/sca_disp.ps.

11 Online histograms

Before you get started with the monitor program, keep in mind the "ldir" command doesn't work for some unknown reason. Therefore, from inside PAW it's impossible to view the directory structure. However, if you change directory to the hardware system of interest to you and type "h/lis", you will see the list of histograms for that system. You might also have problems with the PAW HIGZ window. If you do, try the step discussed below.

First, check if a process needed to use the the online monitoring is running. On any UNIX window type "find_all monitor". This will search the clon cluster to see if the program "monitor" is already running. If it is not running, type the following:

- setenv DD_NAME clasprod
- monitor -a clasprod

Now, on a clon00 window, type

- paw
- < *ret* >
- global_sect MON
- cd TAG (for example)

The last two commands will be typed in at a "paw>" prompt. The last command will change to the directory that contains the tagger histograms. If a run is in progress and tagger data is being acquired some of these histograms should contain events. If the HIGZ screen fails to appear, instead of hitting < *ret* > in the step discussed previously, type "1.clon0X:0.0", where "X" specifies exactly which clon satellite you wish to use (i.e. X=1,2,3 or 4).

Here is a list of the directories immediately under MON:

CC	Cherenkov
DC0	Drift chamber
EC1	Large angle cal.
DC	More DC ?
EC	Calorimeter
SC	TOF
ST	Start counter
TAG	Tagger

12 CED display

It's possible to have CED connect directly to the dd_ring to view events within a few seconds of their occurrence. Login to clon00. Set the DISPLAY variable to your local workstation display. Set the environment variable DD_NAME to clasprod. Type ced to start a display process. Under the "Events" menu at the top of the ced display, select "Events from DD ring". Again, under the "Events" menu select the option "Event every 5 seconds". If there is a run in progress events should appear in the display every five seconds. This section is under construction.

13 Useful phone numbers

Counting House	5244-5247
Forward carriage level 2	5371
Space frame level 2	5170
"	5171
Space frame level 3	5173
Cellular	7107

14 Loading a level 1 trigger

The programs used to load level 1 triggers down to clastrig2 are called Tigris and TigrisLite. From the clasrun account you can access either program from any directory by typing the script names, either "Tigris" or "TigrisLite". TigrisLite is Tigris without the GUI, and it requires the complete path and filename of the desired level 1 trigger as a command line argument. Here's an example of the required syntax:

- TigrisLite /home/trigger/TriggersSept97/beam_triggers/Charged_Particle/Charged_ECTOF_Sect123456.trg

Using TigrisLite has the added advantage that the information in the trigger supervisor GUI "ts_config" will be current.

To download a trigger from Tigris, under the "Action" menu on the far left of the GUI, select "Open Configuration" to select the level 1 trigger you wish to download. Then under the "Download" menu, select "download" (the only option under this menu), then click the "okay" button. This will download the trigger to the trigger supervisor (clastrig2).

15 Solutions to frequent problems

- **"When I'm starting runcontrol and click "spawn", nothing happens, or I get a message "Impossible things happened""**

There is probably already an rcServer running which is using the session name you have chosen. Type "find_all rcServer" to determine if this is indeed the case. If there is already a server running, type "cleanup -a". Now try to start runcontrol again.

- **"When I try to start runcontrol I get a message "Warning: using local msql server may need CODADB environment variable""**

Ignore this message, and all other messages that relate to CODADB.

- **"When I try to telnet to a fastbus crate, I get the message "Sorry, shell is locked.""**

Someone else already has a telnet connection to that crate. From a UNIX window type find_telnet. If the telnet connection originates from the clasrun account, type "kill_telnet crocXX" to terminate the existing telnet connection. If you get a similar message, "telnet: Unable to connect to remote host: Connection refused", when you try to login to a ROC, it means the crate is currently booting up. Try again in a few seconds.

- **"When I telnet to the fastbus crate, I can't start up the coda_roc shell"**

The previous user of the crate may have left the shell in a funny state. Reboot the crate and try again. Another possibility is that the coda_roc shell is having trouble starting up due to problems external to the crate. Wait a few minutes to see if it eventually gets started. If you have waited several minutes and the shell still hasn't appeared then it could be a problem with the runcontrol server rcServer or else the MSQL deamon msqld. Terminate the current rcServer and startup a new one. This will require restarting runcontrol from scratch. A generic operator such as yourself doesn't have high enough privileges to restart the MSQL deamon.

- **"When I select a configuration I get a message "configuration failed""**

The last time this configuration was used it ended badly. CODA2.0 uses an MSQL database. According to the database the configuration you have selected is still in use, even though it obviously is not. Now that you have requested this configuration, the MSQL database status has been changed for this configuration. Try selecting the configuration again. It will work the second time.

- **"I've started coda_eb and coda_er shells, but the runcontrol GUI says it can't connect to them"**

Make sure both shells are running on clon00.

- **"I have selected a configuration, but when I click "download", eventually the download fails after a long time delay of several minutes "**

Does it appear that some of the ROCs are successfully downloading code while others are dormant? If you click the "RESET" button of the runcontrol GUI every time you restart or reboot a component of the online system this problem is much less frequent. Remember to do that next time you change something. Read the following question of this FAQ.

This situation occasionally happens. This seems much more likely to occur if the same server has been used for several different configurations. For some reason either the rcServer sometimes fails to send the download message to the ROCs, or else the ROCs sometimes fail to receive the message. I suspect the rcServer is getting confused about the status of the online components.

If you are using telnet windows to all the ROCs, watch the messages as they scroll by on the windows. Try downloading again, it should work the second time. If it still doesn't work you'll need to try downloading the configuration by hand. As an example I'll show how to manually download the configuration TAG_TEST to the crate CROC07. At the "clasprod:croc07" prompt, type the following: "croc07 download TAG_TEST $< ret >$". This is identical to the message the rcServer tried to send, and there is no difference between typing it by hand at the telnet connection prompt and clicking the "download" button of the runcontrol GUI. Try applying this procedure to each ROC in your configuration.

This problem also might occur if you don't have a telnet connection to all the participating ROCs. Use codaedit to make sure you have started a coda_roc shell on all the needed ROCs. Do this by comparing the graphical representation of the configuration as indicated by codaedit with the list of crates to which you currently have telnet connections. Don't forget the trigger supervisor (clastrig2) crate. If you are doing a calibration run, such as a pedestal run, you also need to start a coda_roc shell on a VME crate.

- **"What does the "RESET" button on the runcontrol GUI do? "**

This reconfigures the run control server, rcserver, to its startup values. If you have just rebooted a fastbus crate or have restarted the event builder or the event recorder, clicking the reset button is a good idea, since it ensures the rcserver will not have trouble communicating with the new CODA shells you have just started.

- **"How can I tell if the triggers are being prescaled?"**

Go to the telnet window for clastrig2. From the % prompt, type "exec prescale_info". This will tell you what the current prescale factors are. A prescale of zero means that trigger is not being prescaled.

- **"How can I change the prescale factor on the triggers?"**

Use the ts_config GUI described earlier.

- **"How can I tell if scaler events are being inserted into the data stream?"**

Go to the telnet window for clastrig2. From the % prompt, type "exec i". Check the list of jobs running on this single board computer. If a process called "FORCE_SYNC" is running then scaler events will be inserted into the data stream once a run is started.

- **"How can I tell which trigger bit is producing the triggers?"**

There are several ways to check this. Since many people know how to use ced that method will be explained here. This will only explain how to get this information from a data file, but it will also work from the DD_RING. Type "ced -f xxxx.a00" to open the current data file. Under the "events" menu, open the "event control" window. Step through several events and look at the information in the HEAD bank. The line " Event Type Physics Type: 8" tells you which trigger bit fired. Currently the physics type corresponds exactly to the latched trigger bit. In this example trigger bit 8 fired. Scaler events have event type zero.

- **"The event file is empty"**

End the run. Exit the coda_eb and coda_er shells. Do a "dd_cleanup" before restarting them. Click the "RESET" button of the runcontrol GUI. Check to be sure you have actually specified that the data be written to a file. If you have successfully specified that the data be written to a file you should see the name of the data file in the runcontrol GUI. Try again.

- **"I've started a run, but the event builder hasn't received any events"**

If you have a very large token interval and a low trigger rate, it could be several minutes before the first token of data gets sent to the EB. Check your token interval, compare that with the number of triggers taken by the ROCs. If the ROCs have taken less than one tokens' worth of events, wait.

- **"I've started a run but I'm not getting any triggers"**

It is somewhat difficult to answer this for the general case, but there are basically only two types of triggers: level 1 triggers or triggers that connect directly to the trigger supervisor. If you are using a level 1 trigger, be sure you have downloaded the correct trigger using Tigris or Tigris_Lite. Try downloading it again. Have you checked the signals going into the level 1 trigger? If you are connecting the trigger directly into the trigger supervisor and are convinced triggers are being sent to that trigger bit, it's possible that particular trigger bit has been disabled by the software. Try using the ts_config GUI to set the trigger bits of the trigger superviosr.

- **"I've started a run, the fastbus crates received one trigger, and now everything is idle"**

This is a communication breakdown between the trigger supervisor and the ROCs. For every event, the trigger supervisor waits for an acknowledgement from each ROC participating in the run. Unfortunately, the trigger supervisor is currently programmed for a different configuration that uses a different combination of crates. Therefore it is waiting for acknowledgements that will never arrive. The trigger supervisor will not accept any more triggers until it gets all the acknowledgements it expects. Ask a CODA expert to check if the acknowledgements set in the trigger supervisor are correct for the current configuration, or use the ts_config GUI to modify them yourself.

- **"I've started a run, everything seems to be working, but when I look at the event information for clastrig2 on the runcontrol GUI it claims clastrig2 hasn't had any events"**

This is normal. Everything is working as it should. The crate that contains the trigger supervisor and the level 1 trigger cards (clastrig2) is not a synchronous ROC. Its job is to determine if an event is interesting enough to bother reading out, and if it is, to notify all the synchronous ROCs that an interesting event worthy of readout has just occurred. This crate doesn't keep track of how many events have been accumulated, so its entry in the runcontrol GUI will be zero. More generally, whenever a configuration includes an asynchronous ROC, don't be surprised if its event information is very different from the other crates.

- **"I've started a run, everything was working fine, but suddenly the event builder printed the message "EVENT TYPE MISMATCH", and now data acquisition is halted"**

Normally this is a very rare problem, but it could be quite frequent if their is a hardware failure down in the hall. In order to distinguish events, such as scaler events from physics events, each event fragment is assigned an event type. Sometimes the power PCs incorrectly determine the event type, and sometimes the event type information is lost as it travels from the trigger supervisor to the fastbus crates through a twisted pair ribbon cable.

Since the event builder is the first process to see all the event fragments from all the crates, it is the first process to notice that one of the crates is reporting an event type which is different from the event types reported by the other crates. If this happens end the run, exit the coda_eb and coda_er shells, cleanup the dd_ring and start a new run. If you have tried several runs and this problem persists, then it's probably a hardware problem that will require access into the hall to fix.

- **"I've started a run, everything was working fine, but suddenly the event builder started printing messages about "waiting for volatile memory""**

In principle this is not a problem and the DAQ system should continue indefinitely. What this means is that several very large events have recently been put on the DD ring, so many in such a short period of time that the DD ring has run out of volatile buffers. The fact this has occured suggests there is a noise problem with the hall electronics, or else the luminosity is too large, or the minitorus might be off, or something else along those lines is wrong.

- **"One of my fastbus crates keeps printing the message "datascan = 0x0" or some other hexadecimal number"**

That crate is not receiving any gates. Until this problem is fixed there will not be any data from that crate, but the other crates will operate normally. This is probably a hardware problem that will require access to the hall to fix.

- **"One of the fastbus crates prints the message "fpbr error pa=7 res = ... maxbytes = 800 returnBytes = 804", or something similar, quite often"**

There may be a problem with the LeCroy card in slot "pa=7". The fastbus blockread that transfers the data from the ADC/TDC in slot "pa=7" to the power PC failed with an error message, and probably all data from the slot was lost for that event. The rest of the cards in this crate are working normally. This is probably not a big enough problem to justify stopping a run, although for some events the information from this crate for slot "pa" may be missing.

- **"When I start a run I get a pop-up message "starting a run failed" in front of the runcontrol GUI, but the ROCs are getting triggers and accumulating events"**

This is not a serious problem. Try clicking the "go" button of the runcontrol GUI one more time. It may work the second time. The most likely cause of this problem is a combination of two unrelated developments. First, there is probably a very high trigger rate, and second, one of the fastbus crates is probably printing an error message for every event, such as the error messages discussed in the previous two questions. These error messages are so frequent they are disrupting the communication between that fastbus crate and the runcontrol server. Either fix the problem that is causing the error message in the fastbus crate or else reduce the trigger rate using the prescale factor for the appropriate trigger bit, and this problem should cease.

- **"My current run seems to have decided to suddenly end, without any actions on my part"**

Find the runcontrol GUI. Under the "options" menu, select "How to monitor components". Deselect the option "End a run automatically when errors are found". Probably what just happened was that ethernet was loaded down with a lot of traffic. Sometimes that can be misconstrued as an error by the monitoring code. I recommend that this option always be turned off. Start another run.

Another possibility is that either the event limit or data limit has been set for the configuration you are using. Find the runcontrol GUI. Near the upper right corner of the display there are several display boxes. Check the entries for "Event Limit" and "Data Limit". Make sure they are both set to zero, which disables these limits. If they are not set to zero you can change them yourself by changing the values in these entry widgets.

- **"When I tried to end a run, it failed"**

The first things to check are the coda_er and coda_eb shells. Find those two UNIX windows. Is there an obvious problem with either shell, for example a core dump or segmentation fault? Does one of the shells no longer display a prompt when you type "$< return >$" in that window? Is one of the windows scrolling an infinite amount of information? If you answer "yes" to any of these questions, use your favorite UNIX tricks to kill both coda shells. Once they are both gone, type "dd_cleanup" at either window, wait until that script is finished, and then start these two coda shells up again. The simplest way to do this is by using the ↑ key to resurrect previous commands.

If both the coda_er and coda_eb shells appear normal to you, check the ROCs. First, type "$< return >$" at each telnet window to be sure the coda_roc shells haven't crashed. If you find a ROC that is frozen, reboot that crate. Don't forget to check clastrig2.

If you want to run again with the same configuration, repeat the "configure" and "download" transitions of runcontrol. From that point on repeat the steps you normally do between runs (see the "Ending a run, starting the next" section of this document).

- **"I have analyzed a recent calibration run and would like to load a new set of pedestals into my ADCs"**

The safest, easiest way to load these new pedestals is to reboot your crate. As part of the booting procedure we load the most recent pedestal file for all crates that contain ADCs. Of course it's your job to make sure you have put the new pedestal file in the same directory as the old pedestal file, or change the symbolic link if you use one. A reboot takes about 20-30 seconds to complete. See the section "ROCs, rebooting of" earlier in this document.

- **"The fastbus crates stopped accumulating events after some number of events between 384 and 1024"**

It has been so long since this problem has occurred it's probably no longer necessary to include this. Use the runcontrol GUI to determine how many events the event builder has received. Compare that number with the token interval. The current token interval value can be found under the "options" menu of the runcontrol GUI ("token interval value"). If the number of events accumulated by the event builder is equal to the token interval, then there is a problem with the token passing scheme. Notify a CODA expert.

- **"A fastbus crate seems competely hung up. How can I reboot it from down in the hall?"**

The Power PC is positioned on the right side of the fastbus crate, as seen from the front of the crate. It's inside the Struck Fastbus interface, SFI, which is a grey device that occupies three slots of the fastbus crate. Find the PPC inside the SFI. Now, near the top of the PPC, there is a toggle switch with two options: "HRST" and "SRST". Push the button towards the "HRST". All the LED displays on the SFI should strobe. Wait about twenty seconds. While you are waiting examine the SFI. There are several rows of LEDs near the top of the SFI, in the middle of the second row there is a LED called "SFF". When that light comes on the crate has successfully rebooted and is ready to take data. If after several minutes the "SFF" light doesn't come on, reboot the PPC again.

- **"How can I recycle the power on a fastbus crate?"**

The only reason to try this is if you have tried several hard resets (HRST) of the fastbus crate and it's not booting up. For the crates in the forward carriage, simply turn off the power supply to the front of the crate with the ON/OFF switch. If it's a crate that uses a power sequencer (i.e. in the space frame) turn off the power to the crate, set the sequencer to "off", wait a few seconds, turn on the power to the crate, wait a few seconds, set the sequencer to "on".

Printed from clon00

Sep 17 1997 09:47 **../crate_info.dat** Page 1

	IP address	Boot script	Name	TS acknowledge	bit pattern	Trans. ID
clastrig2	129.57.167.140	vx_go_clastrig2	TS	-	-	-
croc05	129.57.167.33	vx_go_croc05	Scaler-1	4,7	0x80000000	-
croc06	129.57.167.34	vx_go_sfi_croc06	TOF	1,0	0x00000001	13
croc07	129.57.167.35	vx_go_sfi_croc07	CC	1,1	0x00000002	12
croc08	129.57.167.36	vx_go_sfi_croc08	EC1	1,2	0x00000004	14
croc09	129.57.167.37	vx_go_croc09	CAMAC	-	-	-
croc10	129.57.167.38	vx_go_croc10	Pre-trigger(1)	-	-	-
croc11	129.57.167.39	vx_go_sfi_croc11	EC2	1,3	0x00000008	15
croc12	129.57.167.40	vx_go_croc12	Pre-trigger(2)	-	-	-
croc16	129.57.167.44	$VXKERN_CROC16	Scaler-3	4,5	0x20000000	-
croc17	129.57.167.45		LAC-VME	-		
croc18	129.57.167.46	-	-reserved-	-	-	-
croc19	129.57.167.47	vx_go_sfi_croc19	Ital. Cal.	3,7	0x00800000	16
croc20	129.57.167.48	vx_go_sfi_croc20	DC8(2,5,4S,3)	2,1	0x00000200	8
croc21	129.57.167.49	vx_go_sfi_croc21	DC2(3,5,4S,1)	2,0	0x00000100	2
croc22	129.57.167.50	vx_go_sfi_croc22	DC4(3,6,1S,1)	3,2	0x00040000	4
croc23	129.57.167.51	vx_go_sfi_croc23	DC10(2,6,1S,3)	3,3	0x00080000	10
croc24	129.57.167.52	vx_go_sfi_croc24	DC6(1,4-6,2)	2,5	0x00002000	6
croc25	129.57.167.53	vx_gpib_pulser	GPIB Pulser	-	-	-
croc26	129.57.167.54	vx_go_sfi_croc26	DC7(2,3,4A,3)	2,4	0x00001000	7
croc27	129.57.167.55	vx_go_sfi_croc27	DC1(3,3,4A,1)	2,3	0x00000800	1
croc28	129.57.167.56	vx_go_sfi_croc28	DC5(1,1-3,2)	2,2	0x00000400	5
croc29	129.57.167.57	vx_go_sfi_croc29	DC9(2,2,1A,3)	3,0	0x00010000	9
croc30	129.57.167.58	vx_go_sfi_croc30	DC3(3,2,1A,1)	3,1	0x00020000	3
clastag	129.57.167.137	vx_go_sfi_clastag	Tagger	3,4	0x00100000	11
scaltag	129.57.167.141	vx_go_scaltag_162	Tagger-scaler	4,6	0x40000000	-

Drift Chamber = (region,sector(A,S),levels)

Physical layout of branches: 1: F.C 2: S.F.-north 3: S.F.-south 4:F.C.