Control System Data Management from an Operational Point of View [1]

B. Dunham, Continous Electron Beam Accelerator Facility,
12000 Jefferson Ave., Newport News, VA 23606

Abstract

Day-to-day operation of the Continuous Electron Beam Accelerator Facility (CEBAF) requires the management of over 25,000 signals on the EPICS control system. Operators need to carry out tasks such as saving the present state of the machine, restoring old machine states and rebooting IOCs. Several tools for performing data management tasks are described in this paper. The saving and restoring of signals from a user friendly GUI using the Tcl/Tk programming language is presented, along with a method for recovering from IOC reboots, both intentional and unintentional.

Introduction

The management of the EPICS control system signals for the accelerator at CEBAF is an important issue from the standpoint of machine reliability and reproducibility. Machine uptime for the nuclear physics program is of critical importance and downtime due to poor data management techniques cannot be tolerated. Thus, one must be able to take a snapshot of the state of the machine at a particular time in such a way that the signals are easily accessible in the future for restoring or studying, with the knowledge that all signals necessary for these operations have been saved.

There are two main data management tasks of particular interest to the operations group at CEBAF: saving and restoring operational parameters, and living through IOC reboots (EPICS runs on computers known as Input-Output Controllers, or IOCs). The first section of this paper describes a save and restore tool that has been developed using the Tcl/Tk programming language[1]. It provides a simple graphical user interface to allow operators to save, restore, view and compare control system signals.

The second section of this paper describes a method for recovering from IOC reboots, both intentional and unintentional. Operationally, the main requirement when performing a reboot is to return the machine to its previous state with little or no effort. The methods described here provide a framework for meeting this goal.

I. SAVINGS AND RESTORING OPERATIONAL SIGNALS

Requirements

The save and restore program is required to manage all of the signals necessary for daily operation of the accelerator. These signals include magnet setpoints, parameters for the RF system and for diagnostic systems. The signals must be accessible by area and by type within an area, where typical areas are the injector, linacs, arcs and experimental halls, and possible types are quadrupoles, dipoles, sextupoles, correctors and RF cavities.

The program must perform several tasks. The first task is to save all of the operational signals for the entire machine (called an *ALL save*). These signals, defined by various groups (i.e. magnets, rf or diagnostics) are what is required to run the accelerator; for example, the fields in magnets and cavity gradients and phases. One should also be able to save a portion of the machine for troubleshooting or a quick backup during machine setup (called a *Partial save*).

The next task is to restore signals, either from an *ALL save,* a *Partial save,* or a special file. The user can select which data set to restore, as well as which area and device type within that data set. The special files are to be user generated files for downloading signals for particular machine settings, such as different optical lattices.

The final important task is the ability to display the saved settings and to compare them with the present settings on the machine. The compared signals should be displayed side by side, and the signals differing by more than a given tolerance should be highlighted for ease of visual comparison.

Methods

The BackUp and Restore Tool (BURT) written by N. Karonis[2] of Argonne National Laboratory is the program used for saving and restoring EPICS database signals. The present version of BURT has been tested extensively at CEBAF and has proved to be very reliable. To save signals, the basic command is

---

1. Supported by U.S. DOE Contract DE-AC05-84-ER40150

`burtrb -f` *`reqfile`* `-o` *`snapfile`*

while

`burtwb -f` *`snapfile`*

will restore signals to the control system. The term *reqfile* refers to a file containing a list of signals that one requests to save or backup, and *snapfile* refers to the file generated by BURT that holds the saved signals and their values. Error messages are generated if a signal cannot be connected to for some reason (wrong name, IOC down, network traffic, etc.).

As BURT saves all of the signal values into one *snapfile*, it is not possible to index the values in a way so that they can be retrieved by area and device type as is required. To implement this desired feature, individual *reqfile*s were generated for each area and device type, and each file has a descriptive name. For example, the rf signals for the north linac are stored in a file *NLrf.req* and their values are stored in a file *NLrf.snap.* Presently, 68 such files are needed to cover the entire accelerator.

Whenever a save is performed, all of the *snapfile*s are put in a directory tagged by the date and time of the save. This allows a large number of files to be managed by keeping track only of the date and time of the save and the areas of the machine that were saved.

File System Organization

The file system resides on the operations cluster at CEBAF (see figure 1). The *burt* subdirectory contains the directories that hold the *reqfile*s and the *snapfile*s (from *ALL saves, partial saves* and special download files). The *tcl_apps/burt* subdirectory contains all of the source code as well as scripts for automatically generating all of the *reqfile*s. These scripts (for rf, bpms and magnets) are located in separate subdirectories.

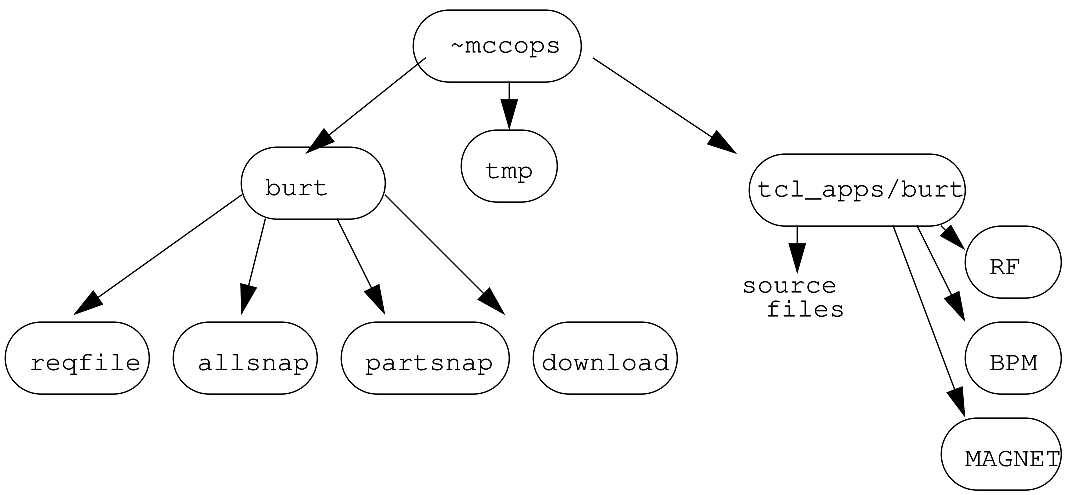

Figure 1. The file system hierarchy for saving and restoring signals

User Interface

A graphical user interface using the Tcl/Tk windowing language controls the save and restore program (see [3] for complete details). Tcl/Tk is extensively used at CEBAF as it provides a way to quickly develop complex programs in a windowing environment. The main menu allows the user to execute various save and restore commands, select the areas of the machine to be saved or restored, select options, or display a help screen. Figure 2 shows the main menu and the commands under the *Execute* menu button, each of which are described below.

The first task is to save a set of signals and it is initiated by selecting *Perform A Save of the Selected Areas*. Before saving the signals, a comment must be entered in the display that appears and an area(s) must be selected from the *Area-Selection* menu (see figure 3). The save can then begin, with the name of each *reqfile* showing at the top of the display as it is processed. Each corresponding *snapfile* is sent to a directory whose name contains the date and time that the save was started (a subdirectory of *allsnap* or *partsnap),* for example *08-20-95_12:05:27.* At the conclusion of the save, the vital information (user, date, time, comment, and selected area) is appended onto the file *master.save*, which is used as a pointer to the location of the saved signals. If there were any errors or signals not connected to during the save, they will be displayed as well as appended to a log file.

The *ALL* selection will obviously select all of the signals to be saved. This area selection is different in that the signal values will be directed to files in the *allsnap* directory (see the section *File System Organization*), whereas selecting an individual area(s) will direct the signal values to the *partsnap* directory. Since saving all of the signals can take several minutes, this feature was implemented to allow a quick save of a small portion of the machine.

Figure 2. The *Execute* Menu Item

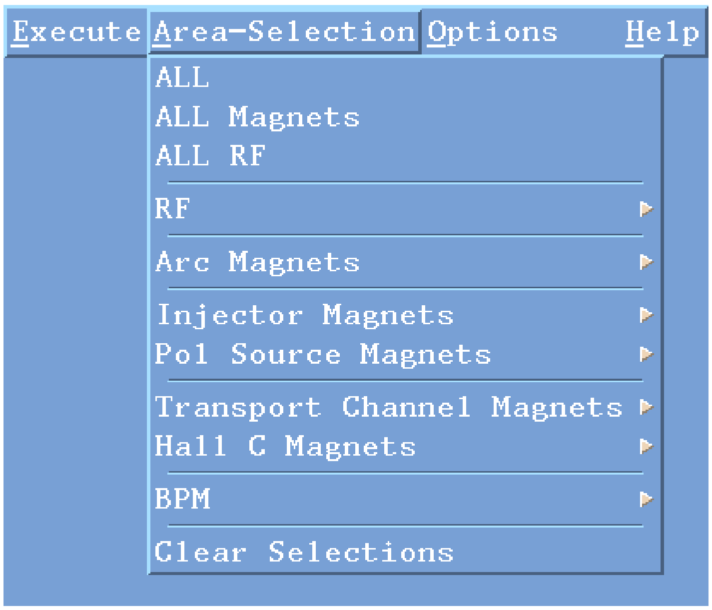

Figure 3. The Area Selection Menu Item

The next major task is restoring signals. Selecting *Perform a Restore of Selected Areas* under the *Execute* button on the main menu will generate the window shown in figure 4. Again, an area and device type must be selected before attempting to restore. The desired data set is chosen by clicking on the line showing the date and time or on the corresponding line showing the comments. Choosing *Restore* will initiate the restore, but a window will first pop up asking for verification before actually loading the files. The *snapfile* names will be displayed on the top of the window as they are being restored. When the restore has concluded, any signals that were not restored will be displayed and appended to a log file.

One of the most useful (and most used, over 100 times on some days) features is the ability to take a set of data, look at all or part of it, and compare it to what is presently on the machine. After selecting a data set, clicking on the *Filter/View/Compare* button will bring up the window shown in figure 5. All of the files corresponding to the selected areas are then concatenated together and displayed in the listbox.

A number of options are available on the *Filter/View/Compare* window. It is often desirable to be able to filter the signal list to examine only a portion of them. For example, to look at only the rf phase setpoints, typing *PSET* in the filter box and hitting either return or the *Filter* button will select only signals that have the symbols *PSET* in them. The *Print* button will dump the signals names and their values to the printer (the printer destination can be set under *Options* on the main menu bar).

Pressing the *View Current Values* button retrieves the present settings for the signals from the control system and displays them in a listbox next to the old settings. If the actual settings differ from the saved settings by more than a given tolerance, the value is highlighted. The *Print* button will dump the differences to the printer.

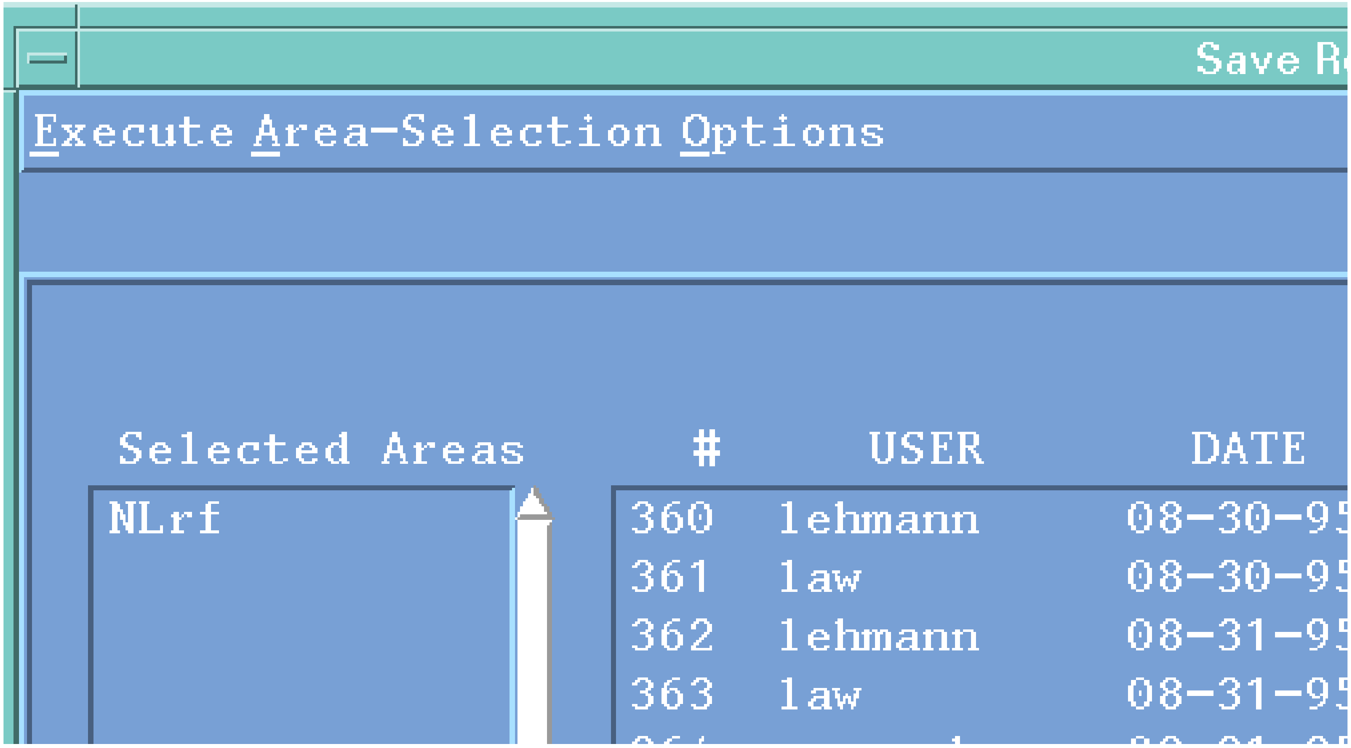

Figure 4. The window for restoring signals

II. IOC REBOOTS

Introduction

At CEBAF, the EPICS control system presently oversees 37 IOCs which handle all of the real-time database processing for the machine. Each IOC controls specific devices at various locations around the accelerator, some handling mostly RF systems, while others handle magnets or a wide range of diagnostics. They typically run continuously, with occasional planned reboots for software updates and less occasionally, unplanned crashes.

For both planned and unplanned reboots, it is necessary to save all of the signals on the IOC as close as possible to the time before the reboot occurs, and then to restore the signal after the completion of the reboot. The signals to be saved are determined by the various software and hardware groups, and include all of the parameters needed to restore the machine to its pre-reboot state. If the reboot is planned, one can save the signals immediately beforehand, but if the IOC crashes, this is not possible. A crash can disrupt accelerator operations if any of the signals on that IOC have been changed since the last time a save was performed. To minimize such problems, a system has been established to save the signals on each IOC once every hour, thus the state of the machine no more than an hour old can be recovered. The methods for managing the data for both planned and unplanned reboots are similar and are described below.

Methods

A number of shell scripts oversee the saving and restoration of signals on the IOCs for reboots (see [4] for complete details). The script *ioc-save.csh <iocname> <type>* (the brackets indicate the arguments to the script) contains a list of the subsystems running on each IOC and it directs each of these subsystems to execute its own script which determines the appropriate *reqfiles* to save. The argument *<type>* can be either the word *normal* or *crash*. *Normal* indicates that a save for a planned reboot should be initiated and directs the *snapfile*s to a directory named *normal*. For planned reboots, the database signals are saved immediately before and restored immediately after a reboot, thus reducing the possibility that any signals will change just before the reboot. If the argument is *crash*, the *snapfile*s are sent to a directory with a two digit number (*00-23*) indicating the hour during which the save is performed. In both cases, the set of signals to save are identical, only the destination for their values is different. Any errors are re-directed to a log file for later inspection.

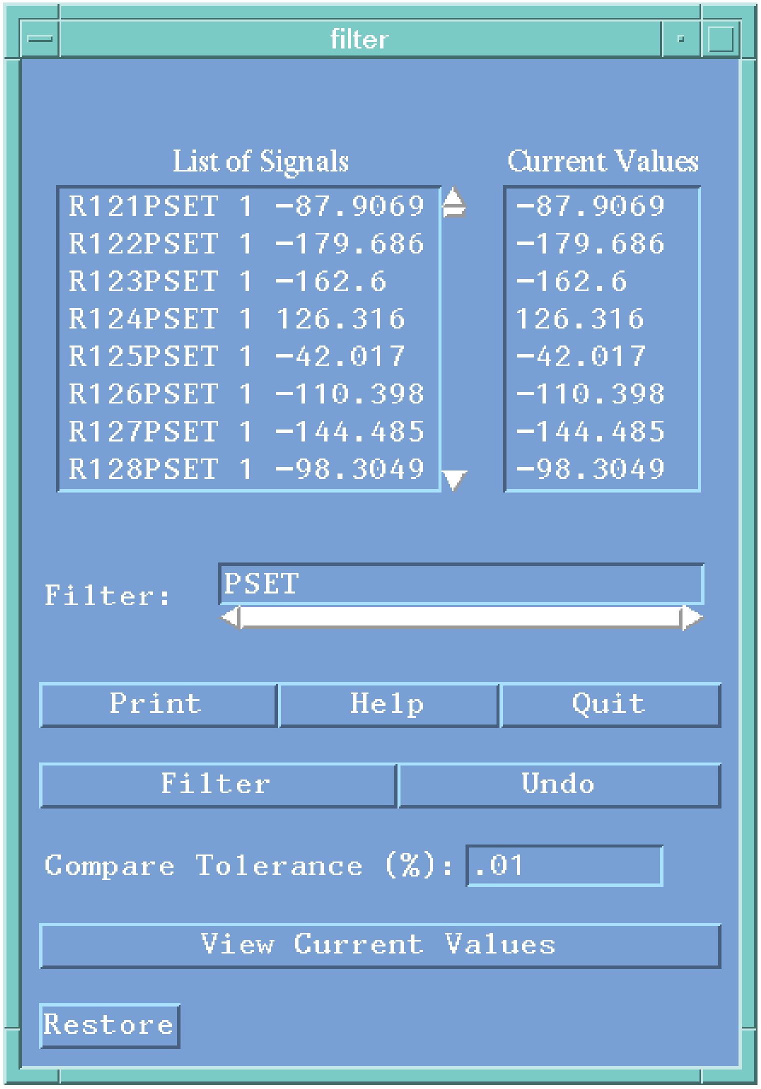

Figure 5. Comparing saved signals to the present machine state

The script *ioc-restore.csh <iocname> <location>* manages the restoration of signals. The *location* can either be *normal*, again referring to a planned reboot, or a number indicating the ‘hour’ from which to restore, *00, 01, . . . , 23* for crash recovery. This script also directs each subsystem to execute its own script to determine the *snapfiles* to restore for a given IOC. Error messages are echoed to the screen for the operators to see.

The Unix clock daemon *cron* controls the periodic execution of scripts on each IOC. At present, 37 IOCs are used at CEBAF, thus to backup each IOC once an hour requires a save approximately every 1.5 minutes. This distribution, rather than saving all at once, keeps the additional network load down so that it is not noticeable to the operators.

The entire system is designed so that adding additional subsystems and signals is simple. First, a subdirectory for the new subsystem must be added, along with the necessary signals to save and scripts for saving and restoring those signals in the appropriate order. Then the subsystem name must be added to the list for each IOC that it runs on in the scripts *ioc-save.csh* and *ioc-restore.csh.* More frequent backups may also be implemented by changing the crontab file and adding additional subdirectories for each new time.

The procedure for a reboot differs slightly for planned and unplanned reboots. For planned reboots, one must hit a button to start the save, perform the reboot and then hit another button to restore the signals. For an unplanned reboot, the process is more complicated. On a particular IOC, the restore scripts will show the the time, the total file size, the expected total file size and the log file size for each of the hourly saves. A zero log file size indicates that the save was completed successfully. The script instructs the operator to restore the signals from the last time which had a zero log file size before the crash using *ioc-restore.csh <iocname> crash <00-23>*.

File System Organization

The file system resides on the operations cluster under the *REBOOT* directory (see figure 6). There is a subdirectory for each accelerator system: beam position monitors, rf, magnets, fast shutdown system (fsd), beam loss monitors, viewers, vacuum and various high-level applications; a subdirectory for shell scripts to control execution; and a subdirectory *snapfiles* that contains the saved signals for each IOC.

For each accelerator system, there is a list of signals that need to be saved in a subdirectory *req*, and a shell script for saving and restoring those signals. The *snapfiles* directory contains all of the saved signals for the IOCs. There is a subdirectory for each IOC and under each IOC there are 25 subdirectories: one for each hour (00 through 23) and one for normal reboots. The *scripts* directory contains a number of shell scripts for controlling the saving and restoring of signals and the *cron* directory contains the file used for starting up the Unix cron clock daemon.

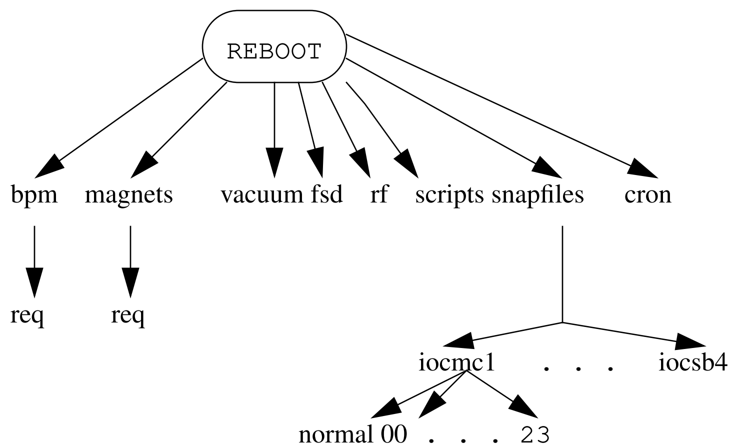

Figure 6. The file system hierarchy for IOC reboots

III. SUMMARY

Managing all of the signals necessary for accelerator operations at CEBAF is an important job in terms of machine reliability and restorability. The save and restore tool has provided a way to save, restore, view and compare the control system signals necessary for operations, and the reboot methods have provided a framework for a smooth transition through an IOC reboot. The tools described here have performed well in over a year of intensive use and have significantly improved the operation of the CEBAF accelerator.

IV. REFERENCES

[1] John K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1992.

[2] N. Karonis, *The Backup and Restore Tool: Theory and Use*, EPICS User Manual, Ch. 17, 1992, Los Alamos National Laboratory

[3] B. Dunham, *Recovery from IOC Crashes on the EPICS Control System at CEBAF*, CEBAF technote CEBAF-TN-95-048

[4] B. Dunham, *A Tool for Saving and Restoring Control System Signals*, CEBAF technote CEBAF-TN-95-057