Index Panel

SLC Control Program

SLC INDEX

SLAC's Software Engineering Newsletter

May 24, 1995 **All that Fits is News to Print** Vol. 9, No. 5

Control System Software

April 3, 1995

Author: *Spence Clark* **Subsystem:** *None* **User Impact:** *None*

Panel Changes: *None* **Documents:** *No* **Help File:** *No*

This article was originally published in May, 1990. It is amended and reprinted occasionally to remind all of our procedures:

We are pleased to find so many individuals outside the Controls Software group interested in writing software for use in the SLC Control System. This collaboration can lead to a better control system and help to alleviate the backlog of software requests. It can also lead to greater understanding of the systems from the user point of view. This insight will aid in the specification of new software based on everyone's enhanced perception of the complexities and subtleties of the software.

We need to put some procedures in place for implementing user-generated software in order to enhance overall effectiveness and to maintain system integrity. It is essential that the documentation and implementation are up to standards so that the Controls Software group can maintain the software in the future. We will be happy to work with you to explain how things work and to help you get your code running properly. We can achieve this if everyone uses the same set of code development procedures used in the Software group.

1. All software procedures need a specification of what the code should do and how it should interact with the user and the machine. This functionality should be supported by a detailed design which lays out how the new software will be implemented (e.g. new modules to be written, changes to existing modules, panels). Please feel free to come talk with us about this documentation, since the extent and detail required will scale with the complexity of the task.

2. We will review the requirements and the design and offer constructive suggestions on better implementation strategies, and guidance on where to find and how to use supporting software that already exists. We can also save you a great deal of time learning how to edit, compile, link, and test.

3. Read the Programmer's Guide, which describes the software development environment, including standard procedures and tools. Also read the Basic User's Guide which will give you

an introduction to the control system and the SCP environment and explain subsystems, support subroutines and utilities. This is the core reference manual for the SLC control system and how to work with it. You can get more details or assistance from your favorite software engineer.

4. Study the Fortran and C software standards to see how the actual code should be structured and written. This is critical for the maintenance and modification of the code in the future. These documents are in flux, so ask for a current version.

5. As the last phase of testing, a "walk through" should be arranged with one of the software supervisors to verify functionality and look for possible improvements to the user interface. When the software is ready for release, contact Chris Traller to arrange for Code Review. One of the programmers will go over the code to check for adherence to standards and look for potential problems. The librarian for the appropriate subsection of the SCP can then put the new code into CMS and release it into production.

Again, we are pleased to see user interest in writing code, and will do whatever we can to help make it a useful and successful experience. Feel free to approach your favorite software engineer, or talk to Nancy Spencer, Ron Chestnut or myself. Good luck.

SCP Dialog Box Upgrade

April 17, 1995

Author: *Daniel Van Olst* **Subsystem:** *SCP* **User Impact:** *Small*
Panel Changes: *None* **Documents:** *No* **Help File:** *None*

The dialog boxes that the SCP uses to prompt the user have been upgraded. Most of the improvements that are visible to the user can only be seen from X (non-terminal) SCPs.

The improvements that will be noticeable to the average user include:

- Dialog boxes are now moveable, so you can see what is underneath the dialog box or move it out of the way temporarily. (However, dialog boxes will continue to pop up over the touch panel window, to prevent confusion when two SCPs are running on the same workstation).

- Dialog boxes on SCPs with touch screens now have larger buttons and larger scroll bars. This makes dialog box buttons and scroll bars easier to select with fingers.

- Some dialogs boxes now appear with a highlighted default already present in the "text-entry" area of the dialog. Users can simply type over this default, edit it with the arrow keys, or just hit OK (or $< cr >$) to accept the supplied default value. (For an example, try the EXIT button from the main index panel).

- A new dialog box has been added. This dialog box allows the user to select one or more items from a list. Users can simply double-click on an item, select an item and hit OK (or $< cr >$), or type the first few letters of the desired item. One of the listed items must be selected; you cannot specify something that is not in the list. (For an example, try the HISTRY DISPLY TYPE button on any Klystron Panel).

More examples of the last two features listed above will appear as software is written (or modified) to take advantage of them.

For more information on writing SCP software to use the new dialog boxes, please see either BUG (Basic User's Guide) or the header to the file ref_c_inc:dlog.h.

Improved Limits for energy gain Actuators

May 24,1995

Author: *P. Grossberg* **Subsystem:** *Fast Feedback* **User Impact:** *Small*
Panel Changes: *None* **Documents:** *No* **Help File:** *Yes*

Normally, limits and tolerances for fast feedback actuator values are set by operator data entry, with software checks against actual hardware limits. However, for energy gain (EGAIN) actuators such as those on the FB31ENGY and EP01SCAV loops, the achievable limits are dependent upon the hardware limits of phase shifters and AMPLs as well as the energy state of the machine. This complex calculation is performed by the micro software, and was previously not available from the user interface. As a result, displays were misleading and users were often unsure about how much energy headroom was available. New software changes enable the correct, achievable energy limits to be available to the VAX software to improve the usefulness of the SCP displays.

Each time the FB31ENGY or EP01SCAV loop is cold-started or warm-started, the micro calculates the actual limits for the EGAIN actuators and sends the values to the VAX. The VAX calculates the tolerances within the limits and overwrites existing values of limits and tolerances in the database and in local memory. Thus, previously entered values for these items will no longer exist and the calculated values will appear on the displays and on the appropriate buttons on the fast feedback actuator panel.

FAST FEEDBACK PANEL REARRANGEMENT

May 09, 1995

Author: *P. Grossberg* **Subsystem:** *Fast Feedback* **User Impact:** *Small*
Panel Changes: *Few* **Documents:** *No* **Help File:** *Yes*

The fast feedback Diagnostic and Cascade/Calibration panels have been renamed and rearranged such that common user functions are now on one panel and more obscure functions are on the other.

Common functions are on newly named CALB/DIAG panel, reachable from the Main Fast Feedback panel, the Actuator panel and the Plot panels. This panel includes the following buttons:

- Create BPM Bufring
- Init Loop DB
- Get New Loop
- Load Matrix

- KISTST Help
- All Calibration functions
- Put RMAT model to DB
- Set CHISQ cut
- Set RMS cut
- Marginal BPM OK?
- Single Unit Loop
- Matrix Index
- FBCK Masks
- FBCK Rates

Less common functions are on newly named MORE DIAG/CASC panel, which is reachable from the CALB/DIAG panel, the State panel, the Gold Orbit panel and the two Plot panels. This panel includes the following buttons:

- All Cascade functions
- GAIN configuration file manipulation
- Select Feedback
- Select Element

Button-macro files have been checked and no problems are anticipated as a result of this rearrangement.

LI00 PHASE KNOBBING

May 17, 1995

Author: *P. Grossberg* **Subsystem:** *Fast Feedback* **User Impact:** *None*
Panel Changes: *One* **Documents:** *No* **Help File:** *Yes*

Capability has been added to the LTR energy feedback to control the electron and positron energy independently. This is expected to be commissioned for the next SLC run. In order to compensate for phase changes we will add control over the LI00 CID master phase and SBST phase (PHAS,LI00,96 and 64). A new capability has been added to allow manual control (knobbing) of these phases while the feedback is on. Two knob control panel buttons are provided so that it is possible to knob LI00 PHAS 96 by itself, and also to knob LI00 PHAS 96 and LI00 PHAS 64 together – by the same amount, in the same direction, at the same time.

This knobbing facility is similar to other knobbing facilities. However, delta knob values, rather than absolute knob values, are used to alter the phase actuator values. That is, when a phase is knobbed, the actuator value will change by the difference between the new knob position and the previous knob position.

The phase knob control buttons are on the NLTRSLTR PARAMETERS panel, reachable from the main FAST FEEDBACK panel by pressing MAGNET PANELS and then LTR PARAMS PANEL.

Upgrading EDITDBS

May 22, 1995

Author: *Lawrence Searcy* **Subsystem:** *None* **User Impact:** *None*
Panel Changes: *None* **Documents:** *No* **Help File:** *No*

EDITDBS uses a routine called Make_Miniedit to make miniedit files for the user when the user has changed a database reference file. There is a make_miniedit file specific to the editors EVE (Editdbs_Eve.Tpu) and LSE(Editdbs_Lse.Tpu). Both EDITDBS and the LSE Make_Miniedit have been modified.

In the case of the LSE Make_Miniedit, the file has been completely rewritten using Ken Underwood's new algorithm. This should stop all of the annoying special features(bugs) contained in the EVE version. I have also implemented some syntax checking and safety nets(error handling). It will check for a missing ";" as well as a double ";" at the end of secondaries. It checks for a missing ">" at the end of primaries and it will alert you to new secondaries and primaries. It now also continuously updates the miniedit window whenever it copies something from the modified file. On top of all that, it is much quicker than the EVE Make_Miniedit version because it now compares the modified file and original file primary by primary instead of line by line.

EDITDBS has been modified to allow LSE users with the definition EVE:=@Slccom:LSE to use the new LSE Make_Miniedit program. It has also been modified to check whether your current working directory is the reference directory REF_DBSFILE: and to limit you to only looking at a file. This will prevent users from trying to reserve the file from CMS while in the reference directory.

The EVE Make_Miniedit will be updated later.