Index Panel

SLC Control Program

INDEX

SLAC's Software Engineering Newsletter

Feb. 15, 1990 All that Fits is News to Print Vol. 4, No. 7

The SLC Database Under CMS

February 21, 1990

Author: *Ken Underwood* **Subsystem:** *Database* **User Impact:** *Moderate*
Panel Changes: *None* **Documents:** *Yes* **Help File:** *Yes*

On February 21, the SLC database source files will be placed under the control of the Code Management System (CMS). This will provide much-needed protection for these critical files. Following this upgrade, authorized users will access the SLC database source files from their own account and not the DBS account.

The current database procedures will remain in effect until February 21. About 8 a.m. the DBS account will be disabled. All files that have been modified since the last DBGEN on January 24 will be placed into the new CMS library. If you log onto the DBS account on February 21, you will see a message warning you that the conversion is in progress and then you will be logged off. Should you require emergency access to the database during this time, please call me at x2682. Let me know if you wish to be notified when the database files become available.

The conversion should take less than four hours and then the SLC database will be available under CMS. All access to the database source files will then be from your own account using the new procedures outlined below. These new procedures will be sufficient for most of the required work. Should you wish to deviate from these procedures, please contact me.

To make life easier under CMS, two new procedures will be added: FIND_DBSFILE and an EDITDBS procedure. FIND_DBSFILE will determine the filename of the database source file containing a specified device. EDITDBS will help you reserve a file from CMS, edit it, replace it and do a DBEDIT. Both of these procedures will be available on February 21. In the future, a more sophisticated DBEDIT procedure will be written to simplify the process of editting the SLC database and make it more foolproof.

Not everyone has the necessary privilege to access the SLC database files from their own account. If you see an error message that says you have insufficient privilege, please notify your supervisor who may request Software Engineering (x3667) to authorize your account.

LOOKING AT THE DBS FILES

A copy of the current database source files will reside in a READ ONLY directory with the logical name of REF_DBSFILE. You may look at or print these files, but modifications can be made only through the EDITDBS procedure. The CMS library for these files will be CMS_DBSFILE. If you merely want to look at the contents of the file QUADLI10.DBS, you may

```
$ TYPE REF_DBSFILE:QUADLI10.DBS
```

or look at the file with your favorite editor. You will NOT be able to save a modified file back into the reference directory.

You may also obtain a copy of any SLC database source file (or any older version) by using the SLCCMS FETCH command. Assuming you wish to look at the QUADLI10.DBS file, the following commands will leave a copy of the file in your default directory.

```
$ SLCCMS SET LIBRARY CMS_DBSFILE

$ SLCCMS FETCH QUADLI10.DBS “an informative comment”
```

FIND_DBSFILE

To look at a file, you will need to know the filename. In most cases the filename is simply the concatenation of the primary and micro names (i.e. QUADLI10.DBS), but there are some cases where the filename is not easily derivable. FIND_DBSFILE will locate the filename(s) for you. To make this search as quick as possible FIND_DBSFILE will search files in a specific order based upon the information you provide. In the worst case a complete search of all files may be necessary to find the requested device.

$ FIND_DBSFILE QUAD will list all files that have QUADs in them.

$ FIND_DBSFILE QUAD LI10 will list only those files with QUADs in LI10.

$ FIND_DBSFILE QUAD LI10 301 will list only that file that has this device.

A lengthy search would be necessary to determine that a non-existant device was not in any file.

EDITDBS

Since the database files will no longer be directly accessible, you will need to use SLCCMS to reserve a file before making changes. The new procedure EDITDBS written by Lawrence Searcy will make this fairly painless. EDITDBS will let you specify a filename (if you know it) or a device (e.g. QUAD,LI10 or QUAD,LI10,301). From your own account, type any of the following

```
$ EDITDBS filename

$ EDITDBS primary micro

$ EDITDBS primary,micro,unit
```

The parameters may be separated by commas or spaces. If a device is specified, EDITDBS will call FIND_DBSFILE to get the filename containing that device. Once a valid DBS filename is found, EDITDBS will do the following:

1. CMS RESERVE the file and prompt for your comment. SLCCMS will leave a copy of the database source file in your default directory.

2. EVE the file. If you plan to do a DBEDIT you MUST make a MiniEdit file as well as modifying the source file. The MiniEdit filename MUST be “Your_Username$Original_Filename”. This file MUST contain only the lines you wish to DBEDIT, not the entire source file.

3. TESTDBGEN the full DBS file. You may skip this step for very trivial changes, but it is strongly recommended to check for syntax errors.

4. CMS REPLACE the file. SLCCMS REPLACE will delete ALL versions of the specified filename from your directory.

5. DBEDIT "Your_Username$Original_Filename" on either SLC or MCC.

Prompts will guide you and remind you of the steps needed. You will be asked whether you are ready to REPLACE the file and whether you want to do a DBEDIT. If you do not complete the changes in a single session, you may leave the file RESERVED and in your directory. The next time you call EDITDBS for that file, it will find that you already have the file and not attempt to RESERVE it. It will ask if you want to EDIT the file or go directly to TESTGEN and REPLACE. If you want to DBEDIT an existing MiniEdit file, you may use the same procedure. The MiniEdit file MUST have a name in the standard edit format "Your_Username$Original_Filename". EDITDBS will recognize that the file is a MiniEdit file and ask if you want to DBEDIT.

When you reserve a file from CMS, you will be warned if the file is already reserved by someone else and asked if you want to continue. The file may still be reserved, but ALL parties have an obligation to see that the all of the changes are merged correctly back into the resultant file.

For quick online help, type $EDITDBS.

MAKING A MINIEDIT FILE

Assume you wish to change the DVI secondary for device QUAD:LI10,301. In addition to modifying the original source file QUADLI10.DBS, you must create a MiniEdit file, for example KKU$QUADLI10.DBS.

While you are editing the original file, the easiest way to make a MiniEdit file is to create a second buffer (e.g. KKU$QUADLI10.DBS). Then copy the lines you have changed to the new buffer. You will also need the device specification and the closing >. The new file would contain only

```
<:QUAD:LI10,301;

    :DVI : = 0.0, 100.0;

>
```

For help with problems or answers to questions, please contact me at x2682, or send mail to KKU on the SLC VAX.

Support for Long Messages from Micros

February 15, 1990

Author: *Nan Phinney* **Subsystem:** *All* **User Impact:** *Small*
Panel Changes: *None* **Documents:** *No* **Help File:** *None*

The message service utilities for communicating with the micros have been enhanced to allow transfer of longer data blocks. The previous limit was 8 Kbytes, unless a special utility was used which only supported transfers involving a single micro. Now even multi-micro transfers may have buffers up to 32 Kbytes.

This improvement is largely transparent to the user. The only applications immediately affected are the Final Focus deflection scans and the BPM buffered data acquisition. Previously, deflection scans were broken into two subscans if there were more than 30 points. This is no longer necessary, so the scans will be faster. For BPM buffered data, the number of pulses that could be read for multiple BPMs was limited by the 8K buffer size. Now up to 250 points may be taken even with four BPMs selected. Further improvements to the user interface for buffered data are planned in the future.

New MPS/PPS/Access Index

February 15, 1990

Author: *Daniel Van Olst* **Subsystem:** *MPS, PPS* **User Impact:** *Small*
Panel Changes: *Many* **Documents:** *No* **Help File:** *No*

A new index has been created for MPS, PPS, and access panels. This index can be selected from the main Index panel. Most buttons lead to familiar panels, such as the PPS panels. Some buttons go to panels which are new, such as LINAC MPS or BSY access.

Most of the existing access procedures and panels are due for review, and will be updated and reorganized in upcoming weeks. In addition, alarm panels are planned for various machine regions, and these panels will appear on the MPS/PPS/Access index.

New System Brownout Panel

February 15, 1990

Author: *Daniel Van Olst* **Subsystem:** *All* **User Impact:** *Small*
Panel Changes: *Few* **Documents:** *No* **Help File:** *Yes*

With the new access index, the old System On/Off panel (from the main index) has been revised, and is now the System Brownout panel.

This panel will now be used for brownouts, and possibly other situations where we wish to shut down the entire machine. The old brownout buttons are still present, and area-off buttons are provided as a backup to the Primary Brownout and Backup Brownout buttons.