Index Panel

SLC Control Program

SLAC's Software Engineering Newsletter

March 15, 1995 **All that Fits is News to Print** Vol. 9, No. 4

April Software Upgrades

March 13, 1995

Author: *Ron Chestnut* **Subsystem:** *SCP* **User Impact:** *Small*
Panel Changes: *Few* **Documents:** *Yes* **Help File:** *Yes*

Starting April 1, there will be a series of major software releases, which have, for the most part, been ready for several months, but have not been released in order to minimize impact on the program. These are scheduled as follows:

1. Take snapshot of current system. April 1 (Saturday) and 2, checkout on Monday early. TEL to organize.

2. Check all shareables for enough spare transfer vectors. Add where necessary. Add new shareable CAUTIL (Channel Access utilities) between UTILSHR and IOBSIC, parallel to NAUTIL. April 3 (Monday) and 4. VANOLST and GREG

3. Release 64K Database support, coupled with DBINSTALL. April 5 (Wednesday) and 6 for SLC checkout. April 7 (Friday) DBINSTALL and move to MCC. TEL and KKU - See following article.

4. Interlude: April 11 (Tuesday) to 13 MCC Preventive Maintenance and Network checkout. FAIRFIELD and ESM

5. Release new X-windows DIALOG support. April 13 (Thursday) or 14. VANOLST

6. BITID project - may become system recompile too. April 17 (Monday) to ? TEL et al. - See following article

Other scheduled items:

1. April 20 - DBINSTALL

2. April 19 to 21 - Sitewide ETHERNET work

3. April 26 to 27 - Possible upgrade to VMS 6.1

Other unscheduled items:

1. SCP event handling (X-windows) reorganization. RGJ
2. Inclusion and Exclusion masks; BPMP, BPMS HSTA expansion. At least BPMUTIL, BPMSHR, TIMSHR. TEG,LJH,MIKE
3. Increase number of micros per DGRP. At least DBSSHR. TEL

64K Database Support

March 15, 1995

Author: *Lahey,Underwood,Gromme* **Subsystem:** *Database* **User Impact:** *Small*
Panel Changes: *None* **Documents:** *Yes* **Help File:** *None*

The 64K database project expands the maximum allowable size of the database, allowing us to define more devices as needed.

Users and application code should see no changes, but we would like you to watch for unusual conditions in the control system that might be caused by this change. All interfaces between application code and database routines are UNCHANGED.

The changes are entirely hidden inside the database internals and low level database code on the VAX and Micro. We will release a new database and some new VAX and micro code.

BITID project

March 15, 1995

Author: *T. Lahey, et. al.* **Subsystem:** *All Software* **User Impact:** *Small*
Panel Changes: *None* **Documents:** *Yes* **Help File:** *None*

We are expanding the maximum number of SLCNET and other BITID micros that the control system supports. Any application code which calls MICROALL, MICR_EXPAND_RANGE, or GETMICROLIST must be able to accept more micros in the return arguments. Calls to BITMAP_TO_MICRONAME and MICRONAME_TO_BITMAP must support a larger number of defined micros.

Please review all of your private code and prepare to recompile and relink it after the release of the BITID code in April.

Another related project will expand the number of micros in a display group (DGRP). This project will be released separately. Watch for an article in a later index panel. While reviewing your private code for sensitivity to the total number of defined micros, please also look for code which allocates storage for the micros in one DGRP.

RIP – Matlab version 3

March 15, 1995

Author: *Keith Jobe* **Subsystem:** *None* **User Impact:** *Medium*
Panel Changes: *None* **Documents:** *Yes* **Help File:** *Yes*

Matlab version 4 is here, and version 3.5 is almost dead. On April 1, 1995 (no fooling) the license for Matlab version 3.5 will expire, and there is no possibility of extending the license.

We should be using version 4 by now. Transistioning to the newer version was describen in an article in the May 16, 1994 Index Panel. For those of us (myself included) who have misplaced it, here is a copy compliments of the computer:

Text of May, 1994 Index Panel article

The new version of Matlab is now available from the SLC VAX cluster mainframe computers. This version is a "major" release of Matlab, and has a significant number of new features and changes in the graphical interface. Unlike prior releases of Matlab, this release has broken a large fraction of the previously debugged scripts and functions developed by the SLC physicists and operations support personnel.

While the scripts and functions are being debugged, the old version (version 3.5h) will remain as the system "default" version for a while. To define the new version, enter:

```
mcc> MATLAB__4
```

and to run it, enter:

```
mcc> MATLAB
```

Both versions of Matlab are licensed to allow an unlimited number of users to run on the two major mainframes (MCC and SLC). The limited use licenses which allow a few users to run on the cluster workstations have not been upgraded to version 4 yet. Separate licenses also exist for operation on the SUN/UNIX workstations without any of the SLC extensions or database access routines. Please contact Bob Boeninger (BOE@SLAC.STANFORD.EDU) if you want more information regarding the SUN Matlab.

This is a new version of Matlab, the various standard SLC ".M" files, and additions will take their authors a while to debug. Please be patient and try to contact the author of the code when problems are uncovered.

Manuals for the new version of Matlab are available from Software Engineering upstairs in MCC. Those users with the Macintosh version of Matlab will probably not need a new manual since the manual sets are almost identical.

Note: Consistent with the Matlab help files and documentation, all Matlab commands in this article (such as `GET(0)`) are shown upper case. Note, however, that the commands must be entered using lower case characters (as `get(0)`) to work.

The major changes and incompatibilities are:

- Matlab graphics are now "object oriented." Specifically, this means that any changes in the attributes of a graph must now be done after the graphical object is created. For example, the command: `PLOT(SINC(-10:.03:10));` is followed by `AXIS([0 PI -INF INF])`. This will zoom the plot into the X-values of 0-π. Note the use of `INF` to extend the graph to the limiting datum. This sequencing also applies to the `SUBPLOT` command, which has a new format for it's arguments.

- The `DRAWNOW` command forces the graphics buffer to appear immediately. This is only required for command files where you are witnessing the analysis of the data. Typically, the rendering of a plot is deferred to the next user input prompt or `PAUSE` command, thus allowing any changes in the plot's rendering to proceed the actual drawing of the image.

- The continuation character is now definitely "**...**". Version 3.5 and earlier would accept either "**...**" or "**..**". Many of our files will be broken by this simple error.

- Matlab automatically senses for the presence of a X-Terminal type display, and uses it by default. If you *do not* want the X version of Matlab, enter `mcc> SET DISPLAY/DELETE` prior to evoking Matlab. Note that without a display or graphical interface defined, any graphics generated will not be seen or displayed, yet the PRINT family of commands will generate the expected pictures.

- Dumb terminal support: The Matlab command `TERMINAL` will present the user with a list of terminal types, and the command `TERMINAL VERSA` roughly replaces the old `VERSA` and `RETRO` commands. Graphs will appear, but do not expect any of the snazzy graphical user interface features to work.

- The `GLOBAL` command is changed. A variable becomes global when it is so declared, but is only visible to functions and scripts which explicitly declare the variable global. By Matlab convention, global variables should be UPPER-CASE names.

- Matlab now stores data in a variety of new ways, allowing a saving of disk space when saving large matrices of integer data. These files may not be readable by the old version of Matlab. The routines which save data to disk from the SCP have not been changed to use this feature, and will remain compatible with version 3 of Matlab for the present.

Printing: The usual SLC-type modifications have been applied to the Matlab print support – ignore the documentation on PRINT in the Matlab Reference Guide. These changes are required by the wealth of printer types available - Matlab presumes that the printer type and print command is fairly stable. The usual (Matlab 3.5) commands `PRINT`, `PRINTB`, `PRINTOUT`, `PRINTER_SET`, `PRINTCLEAR`, and `PRINT_WHERE` are supported.

Matlab aficionados might want direct access to the native Matlab PRINT command to generate EPS files, and will find it renamed as `PRINTM`.

The list of printers is smaller since Matlab insists on using Postscript level 1 or 2, and will not generate the Tektronix 4014 language for printing (a sad good-by to VM based Imagen printers). Your default printer may be defined in your login file by setting the symbol `MATLAB_PRINTER` to the printer name as seen from inside Matlab using the `PRINTER_SET` command. (Matlab 3.5 used a logical name to store this information.) For example, you might want to add the following to your own `LOGIN.COM` file:

```
$ MATLAB_PRINTER :== MCC$PRINT
```

User customization may be done in several ways.

1. Commands in the current directory (`[]`), followed by commands in your Matlab sub-directory (`[username.MATLAB]`) are used prior to the commands in any of the Matlab Toolboxes or SLC support packages. The special Matlab directory may be created with the `mcc> CREATE/DIR [username.MATLAB]` command from DCL.

2. You may have a local "startup" file called `my_startup.m`. If you have several such files, the version in your current directory will be used before the version in your `[.matlab]` sub-directory. Do not use the filename "startup.m" as described in the documentation, since this will circumvent some SLAC-specific startup code.

3. You may add a directory to the Matlab search path using the Matlab command `PATH(PATH,disk:'[directory]')`. The command `PATH` used by itself returns the current search path.

4. You may place your code in a production directory, allowing more direct and wider access to the routines than the above methods. Please see a Matlab expert prior to adding any new functions to the production directory.

Some of the snazzy new features are:

- Try `HELP`. The help is grouped by functions. Try `HELP GRAPHICS` and `HELP PLOT` for example.

- `WHICH <command>` returns the location of the command file you specified. See also `CD`, `PWD`, and `UIGETFILE`.

- Direct file read of binary files of user data (see `FOPEN`, `FREAD`).

- Colormapped plots and images (try `[X,Y]=MESHGRID(-3:.125:3); Z=PEAKS(X,Y); MESHC(X,Y,Z)`.

- Try `DEMO`. Try this *only* if you have some time to kill.

- After you plot something, explore the graph's objects with the commands `GET(GCF)` and `GET(GCA)`. To change something, try `SET(GCA,'VISIBLE','OFF')`. GCF means get current figure handle, GCA for axis.

- Matlab now supports M-file debugging. Try `HELP DBSTOP` for information.

- The command `AXIS` is different from `AXES`. The latter allows multiple scales on the same plot.

- Matlab supports a version of a movie, where multiple frames are displayed in a defined order. The `DEMO` displays this. This feature is known to break on X-displays with insufficient memory.

- `WHITEBG` toggles the background color to white. It does considerably more with various arguments.

Popping Up the Knobs Panel

March 14, 1995

Author: *Ralph Johnson* **Subsystem:** *Software Knobs* **User Impact:** *Some*
Panel Changes: *None* **Documents:** *No* **Help File:** *Yes*

The mouse button processing within the SCP has been changed so that you must push the center and right buttons simultaneously to pop up the software knobs panel. The rest of the operation of the panel remains the same.

For those using a two button mouse, holding down the control key while clicking the right mouse button is the now the same as clicking the center and right mouse buttons of a three button mouse.

Wire Scanner BPM Jitter Correction

March 9, 1995

Author: *Kenneth Underwood* **Subsystem:** *Wire Scanners* **User Impact:** *Small*
Panel Changes: *None* **Documents:** *None* **Help File:** *None*

Wire scanner X and Y BPM jitter correction can now be independently disabled. Until this change, BPM jitter correction could only be disabled for the entire wire scanner. BPM jitter correction for U/V wires is disabled only if both X and Y are disabled. All BPM jitter correction can still be disabled by toggling the [WIRE POSN CORRCT] button to OFF on the [SCAN OPTION PANEL] touch panel.

To modify the X or Y BPM jitter correction disables for a wire scanner you must DBEDIT the WIRE hardware descriptor secondary, HDSC. Add or remove the symbols %WIRENOXJ and %WIRENOYJ in the HDSC secondary for the desired WIRE unit while preserving all other symbols. After the DBEDIT, you must push the [REINIT DB VALUES] on the [SCAN OPTION PANEL] to update the SCP.

For example: one of the following lines would be used to disable X or Y BPM jitter correction for WIRE:FF11,1670. The %WIREQEMT and %WIREWEMT symbols describe other WIRE hardware features and must be preserved.

:HDSC: = %WIREQEMT + %WIREWEMT; ! X and Y BPM jitter correction enabled

:HDSC: = %WIREQEMT + %WIREWEMT + %WIRENOXJ; ! disable X only

:HDSC: = %WIREQEMT + %WIREWEMT + %WIRENOYJ; ! disable Y only

:HDSC: = %WIREQEMT + %WIREWEMT + %WIRENOXJ + %WIRENOYJ; ! both disabled