

# Little Console

## *Introductory guide*

Tue, Apr 22, 2003

The front end nodes that support the Fermilab Linac and many other projects at Fermilab include a suite of page applications that provide a user interface referred to as the "little console." This note is meant to introduce the basic operation of the little console interface.

There are a number of little consoles in the Linac gallery that can be used to operate page applications. But even for those nodes that lack little console hardware, page applications can be accessed indirectly via several platforms that support what is termed "Page G" access. (The name comes from the fact that a page application that supports this same interface is normally installed as page G on a little console.) Other platforms that support a Page G interface, and thus emulate a little console, include the Macintosh, via the Parameter Page written by Bob Peters, and the PC, via software written by Bob Florian.

The little console monochrome display consists of 16 lines of 32 characters each. Characters are limited to capital letters, digits, and several punctuation symbols. Each character can optionally be displayed in inverse video. The cursor location is indicated by the character at that position toggling between regular and inverse video at about 4 Hz.

The operation of the little console interface is not so dissimilar with that of the Acnet console pages; in fact, the little console interface predates Acnet. There is a pair of index pages that list the available page applications. From the index page, one can select a page application and "call it up." Interaction with the current page display is done via "keyboard interrupt," nowadays better referred to as a "mouse click," positioning of the cursor, typing onto the screen display via the keyboard, and a small set of console buttons and associated button lights. The hardware consoles also support a knob, but the Page G emulators do not.

The number of page application displays is limited to 31 in addition to the two index pages. They are referred to as Page 1 through Page V, employing a base 32 number system that uses 0-9, A-V. Page 0 is the index page itself. Calling up the index page from the index page simply toggles between the two index pages. This is most easily done by clicking with the cursor in the Home position, which is in the upper left corner.

To invoke another page, one can click on the indicated line of the index page, or one can type the single character that denotes that desired page in the Home position, then click with the cursor immediately following that character. If you know that you need to call up page P, for example, hit the Home key, type P, and click.

Each display page is associated with a chosen page application program whose name in the local nonresident memory file system is PAGExxxx, where xxxx here stands for a 4-character name. One can see these program names for those pages listed on the index page by pressing the Hex pushbutton. (In addition to the program names, one also gets the version date of the current system code displayed.) To change the program name attached to one of these pages, merely type the appropriate file name and click with the cursor immediately following the 8-character name. (A click anywhere else on the line will only call up the page, if any, that is already associated with that page.)

Each display page also has an associated page title, which is what one sees displayed on the index page and also on the top display line when the page is active. In order to establish a

page title, first call up the page. Then type the desired 16-character title beginning in the third character position on the top line. (This leaves one position at the Home position where the page number is shown, plus one space to separate the title from that character.) It is usually better to use only 15 characters in order to allow for a space to separate the title from the time-of-day shown on the top line. Having typed the desired title, *return the cursor to the Home position and click*. This returns to the index page, where one should be able to see the new title listed for that page. Calling the page up again results in the new title being displayed on the top line. This title change is permanently recorded in nonvolatile memory.

The bottom line of the display is usually reserved for use by the system, which supports two possibilities. A simple 8-byte memory display, called the Small Memory Dump, is activated by typing a local memory address at the start of the bottom line and clicking. The 8 bytes of memory are displayed as 4 words and are updated at 15 Hz, if necessary; *i.e.*, whenever their values change. This memory display can be removed by placing the cursor at the start of the bottom line and clicking.

The other available use of the bottom line of the display is to display alarm messages that are generated by the local system. They can only be displayed one at a time, of course, and one is given at least 2 seconds to view each one, before it is replaced by the next one, if any. This option is enabled via setting control Bits 00A2, 00A4. Activating the Small memory Dump option inhibits the alarm message display.

A suite of page application programs is available for use with these front end systems. Most are not suitable for users, but many are used by those of us who update and maintain these systems. The number has grown over the years as new page applications have been written; they are seldom unwritten. The main page application for users is the Parameter page; in fact, the design of the little console switches and lights was based upon the features needed by this application. The five mode switches (pushbuttons) are labeled A/D, D/A, Nom, Tol, Set. The three units switches are labeled Eng, Volts, Hex. The knob is used to adjust the analog channel indicated by the cursor position. The raise/lower buttons also can be used to adjust an analog channel. Any page has some associated nonvolatile memory, so that parameters can be retained across invocations of that page, or even across power cycles. This permits a user to establish a selected set of channels that are to be displayed on a page that is attached to the Parameter page program. Another page that is also attached to the same program can be set up to have a different set of channels associated with it. More details of the operation of this program can be found elsewhere.

Page applications can be written in Pascal or in C for the 68040-based systems, but PowerPC systems only support C programs. Two front ends serve as library nodes; all 68K page applications are kept in node0508, and all PowerPC page applications are in node0619. Each has a nonvolatile memory file system sufficient in size to house the full panoply of program files, including page application, local application, and data files.

A page application has from the start (1980) been given the ability to transparently access data from any set of front ends using the Classic data request protocols. The underlying system code automatically arranges to collect the requested data from any nodes included in the request. (A node number is part of the description of each analog channel in a request, for example.) The page application code itself does not have to deal with any of the resulting complexity of networking, etc.

If a page application is collecting data from other nodes, the system imposes a time out of this activity. After about an hour of running the page without any console interaction, the page is automatically switched back to the index page, canceling the active requests for such data. The idea is that if the page is collecting data from other nodes, and one leaves for home at the end of the day, it won't continue to run all night, performing unneeded network activity.

Here is a short list of page applications in common use:

<i>Name</i>	<i>Page</i>	<i>Function</i>
PARM	—	Parameter page, for up to 14 analog channels.
EVTS	—	Clock events, updated at 15 Hz.
EDAD	A	Analog descriptor, displays analog channel local database info
EDBD	B	Binary descriptor, displays binary bit local database info
DNLD	D	Download page, allows copying program/data files between nodes
LAPP	E	Local application parameters, configure LA instance
NETF	F	Network frames, captures recent datagram network activity
CRTI	G	Page G, allows access to page applications in another node.
ECHO	I	Ping, Echo, NTP, DNS Internet protocols
MDMP	M	Memory Dump, displays up to 64 bytes at 15 Hz, copy memory, etc.
SURV	—	Survey page, displays some generic info about any node

Besides those listed above, additional programs for system configuration and monitoring are:

<i>Name</i>	<i>Page</i>	<i>Function</i>
TASK	—	Capture task activity and output listing
PATS	—	Pattern search scans system tables in a set of nodes for selected patterns
PMEM	—	Print Memory, lists out contents of a range of memory for set of nodes
SNAP	—	Snapshot protocol (FTPMAN) test client
TFTP	—	TFTP log, especially for listing node0562 boot activity.
SRMC	—	SRM memory copy, copies from any SRM to another.
NVOL	C	Examines nonvolatile memory file system in 68K nodes for anomalies.
MBLK	—	Displays allocated blocks of dynamic memory, updated at 15 Hz.
ACRQ	L	Acnet request protocol (RETDAT) test client
REQR	R	Classic request protocol test client
SLOG	S	Setting log, shows recent settings logged by local node
SWFT	—	Swift digitizer log, shows recent Swift digitizer operations
T553	—	Test page for MIL-1553 protocol
EVTQ	U	Clock event queue, shows log of selected recent clock event activity
VERS	V	Program versions, compares version dates against library node, also lists local application table contents.

Of course, a page application can be attached to any page. Those shown here are merely the assignments in fairly common use. Page 0 is always the index page. It is supported by the system code directly; there is no associated page application program for it.

If a node is reset while a page application is active, the system will automatically re-invoke that page when it comes up following its TFTP boot, or for PowerPC nodes, following the execution of its boot script.