Chapter 6 --

DIALOGUE: TILES, KEYS, BUTTONS AND MENUS

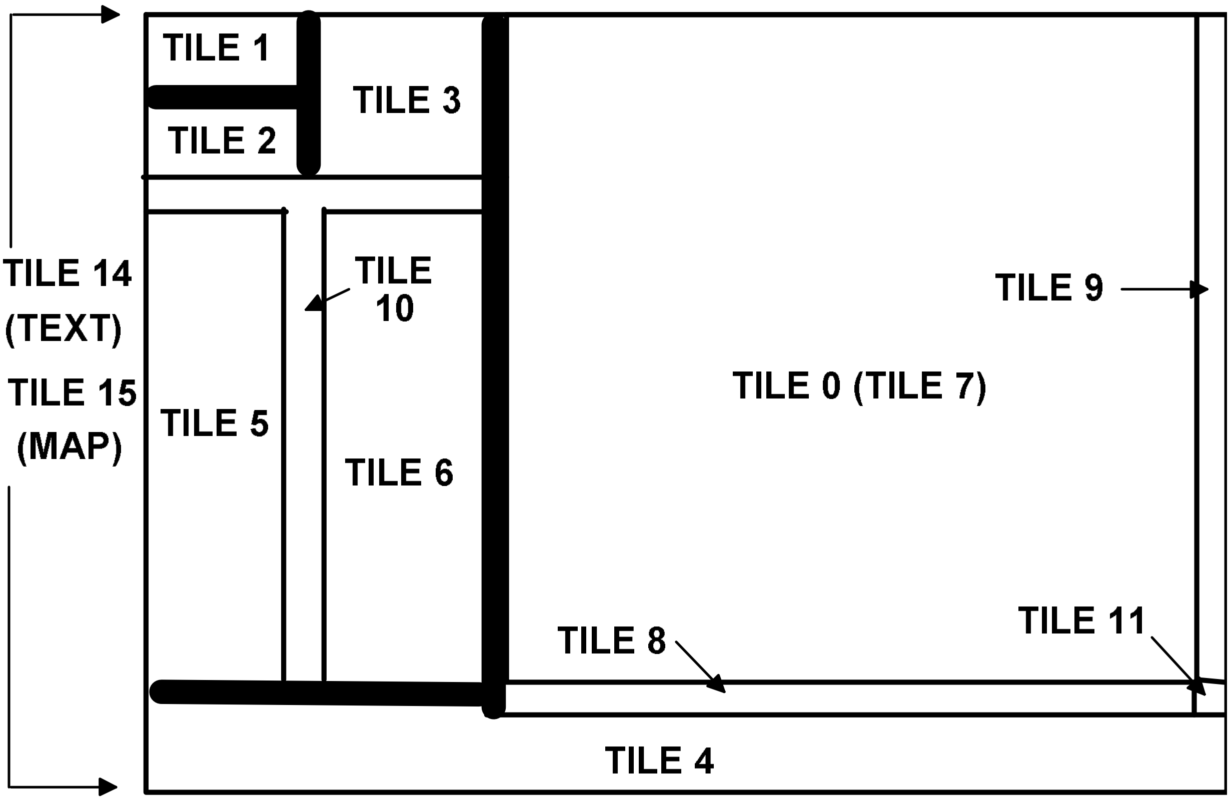
In the LTPlus processing environment, the human interface is greatly enhanced by the availability of four primary dialogue strategies. The TILES, or screen partitions, contain menus, function key correspondence information and menu displays. In addition to the menus displayed on the screen TILES, which are graphically implemented with either the mouse or the tablet cursor, LTPlus also provides the digitizer stickup menu, which can be activated by the digitizer cursor only. This menu type is created by the user.

Another form of communication available to the user is the concept of cursor key definitions, which are initiated by tablet cursor button presses. Both mouse and digitizer cursor keys may be pre-programmed by the user to invoke commands and/or macros. Another type of command implementation is the function key assignment facility; however, this process cannot be modified by the user (i.e., the function key assignments have been hard-coded). The function key assignments appear on the screen and are activated at any time by depressing the desired key, or combination of keys.

And, of course, a primary form of dialogue occurs when the user types on the keyboard instructions which appear in the Command Line TILE. Detailed information about all of these dialogue possibilities is included in this chapter, so read on.....

TILE OPERATIONS

All LTPlus operations are displayed on the screen; both the command choices and the command results will be reflected on one or more of the pre-defined screen blocks known as TILES. The LTPlus screen is comprised of 13 unique TILES, each with a specialized role, described below:



TILE 0 List display area (same as TILE 7).
Various types of lists, including map and file choices, directory catalogs, set selection criteria, etc., will be displayed in this TILE which is equivalent to the TILE 7 map display area. A text listing will turn off the graphics map display temporarily.

TILE 1 Status information area
This TILE contains information about the current, active map, the active graphic input device and the edit/display mode.

TILE 2 Cursor coordinate display
The type of units and the current, active cursor's X,Y location are displayed in this TILE.

TILE 3 Permanent function display
This TILE contains certain intrinsic LTPlus functions which are used frequently; their appearance in this TILE precludes the necessity of including them on more than one function menu.

TILE 4 Command line area

The command line area TILE is keyboard sensitive; all user commands may be entered from the keyboard in this area. All prompts and error messages will also be displayed in TILE 4. Activating TILE 14 or 15 will not overwrite this TILE space. The command line area is always considered active.

TILE 5 Menu display

This TILE contains a list of the menus available from the currently active menuset (as defined in the **header** file).

TILE 6 Current menu function display

The currently active menu, selected from the list displayed in TILE 5, is shown in TILE 6.

TILE 7 Map display area

This TILE contains the raster and/or vector map graphic representation, unless it has been temporarily displaced with a tex display for TILE.

TILE 8 Horizontal movement bar for map display

A click and drag operation on the bar in TILE 8 will cause a proportional horizontal movement (or displacement) of the currently active map display in TILE 7.

TILE 9 Vertical movement bar for map display

A click and drag operation on the bar in TILE 9 will cause a proportional vertical movement (or displacement) of the currently active map display in TILE 7.

TILE 10 Function key assignments

This TILE contains the unique numeric identifiers for the function keys which correspond to the adjacent functions listed in TILE 6. Note that the permanent functions in TILE 3 also have function key assignments.

TILE 11 Collapse/expand map display (graphics)

The small square called TILE 11, when activated, causes the map data in TILE 7 to be collapsed and replaced with the text display of TILE 0. The next activation of TILE 11 will cause the graphic contents of TILE 7 to be be re-displayed, effectively replacing TILE 0 text.

TILE 12 Digitizer stickup menu (not shown)

This TILE number is assigned to a user-designated digitizer stickup menu (optional).

TILE 13 Digitizer stickup menu (not shown)

This TILE number is assigned to a user-designated digitizer stickup menu (optional).

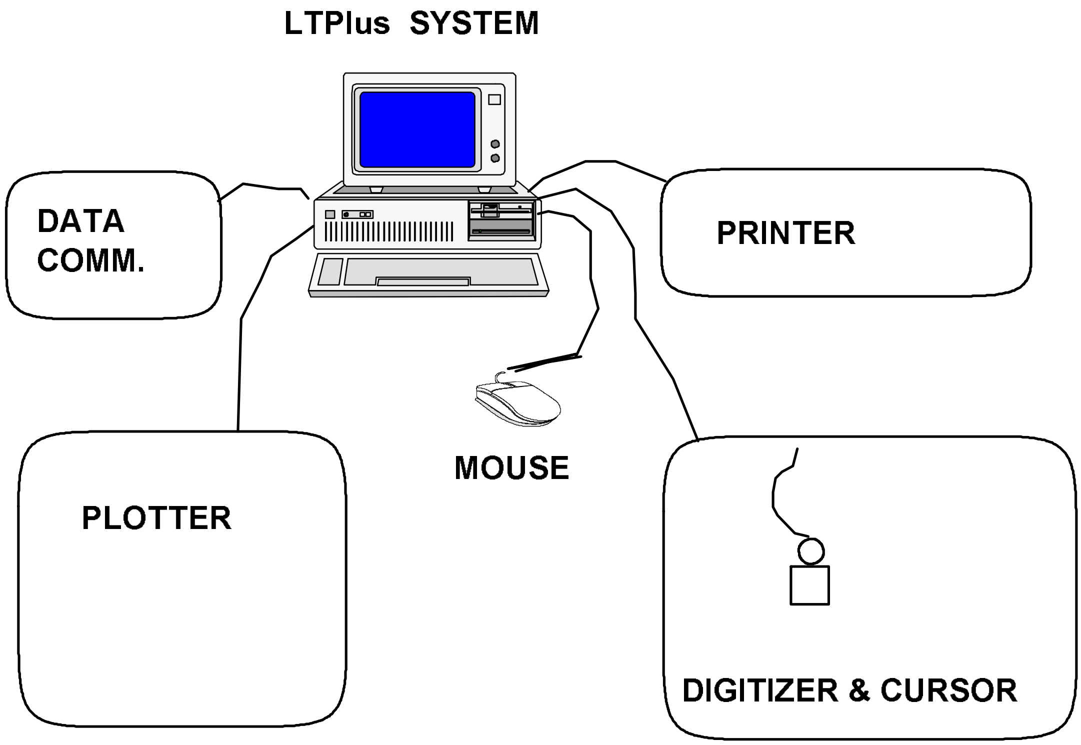
TILE 14 Whole screen as text (except for TILE 4)

TILE 14 is the entire screen (except for the command line area) opened for text display. It is used by the system whenever an extensive amount of textual information must be displayed.

TILE 15 Whole screen as graphics (except for TILE 4)

TILE 15 is the entire screen (except for the command line area) opened for graphic display. It is used by the system whenever an extensive amount of graphic data must be displayed.

As described in the TILE definitions above, some TILES contain user-selectable choices, functions or are, themselves, function initiators. To communicate interactively with LTPlus via a TILE, the operator may use the standard keyboard (to enter commands), the programmable function keys (defined in TILE 10), a mouse or digitizer cursor to highlight and select TILE options, and/or a digitizer cursor with a user-generated stickup menu (described in detail in *Chapter 6*). The typical user will likely incorporate a combination of these methods during the map processing activity.



INTERACTION WITH LTPlus

A typical, albeit simplified, system configuration scheme might look something like the above diagram.

By what process do you communicate with LTPlus? Here are some suggestions:

How To Select & Execute Commands, Menus or Macros

Use the digitizer cursor in relative mode, or the mouse, to move the screen crosshair into the menuset or function tiles to highlight your function/command/macro choice and then press the appropriate cursor or mouse button to select the option. If the command/menu/macro requires one or more parameters, LTPlus will prompt you for each. Your response(s) will be entered from the most appropriate device. For example, if you selected the **crit_astrg** command with the cursor or mouse, LTPlus will ask for the function argument string to be matched and will expect your response from the keyboard.

In contrast, if you select the **register** command with the cursor or mouse, with the intention of registering the map attached to the digitizer, LTPlus will respond by asking that you enter the coordinate type code from the keyboard. Your response of 4 (digitizer1, absolute coordinates) will then compel LTPlus to look for the required XY coordinate positions from the digitizer device, via a cursor button press while the cursor crosshair is centered over the desired map corner. It is entirely permissible to enter the XY coordinate values on the keyboard as parameters with the **register** command; i.e.,

register 4 sw=14.2,3.9

However, it is not a very appropriate or efficient method for **register**ing the manuscript since it is unlikely that you would know the XY values...that, after all, is why you own a digitizer!

Keyboard Shortcuts
A command can be executed from the keyboard by entering the least number of characters which comprise a unique name. For example, if you enter the letter **u**, and there are no menus or macros called **u**, LTPlus will execute the command **undo**, which is the only command which can be matched partially. If, however, there was either a menu or a macro called **u**, then it would be executed. It sounds somewhat confusing, but the rules which govern command, menu and macro execution are as follows:

Check for an exact match in the command file.
If one is found, then execute it and discontinue the search.
If no exact match is found, then search the menu file for an exact match.
If one is found, then execute it and discontinue the search.

If no exact match is found, then search the macro file for an exact match.
If one is found, then execute it and discontinue the search.
If no exact match is found, then search the command file for a partial match. This means that your entry must exactly match the start of one or more command(s).
If exactly one is found, then execute it and discontinue the search.
If more than one is found, then display them and allow the user to select one or none.
If one is selected, execute and discontinue the search.

If no exact partial match is found, then search the macro file for a partial match. This means that your entry must exactly match the start of one or more macro name(s).
If one is found, then execute it.
If more than one is found, then display them and allow the user to select one or none.
If one is selected, execute and discontinue the search.
If none is selected then discontinue the search.

WARNING: All user, keyboard entries are converted to lowercase; all menu and macro names must be lowercase or they will never be found by LTPlus and, thus, cannot be used.

Some examples:

Desiring to execute the **vertex_thin** command, you enter **ver**, thinking yourself clever for selecting this shortcut. However, you are immediately reprimanded by LTPlus for entering a non-unique substring; i.e., there are actually three intrinsic LTPlus functions which begin with those same three characters, and the program will not execute anything until you select the command that you want from that list.

Now, if you had already created a macro called **ver** which contained the single **vertex_thin** command that you wished to execute, then LTPlus would find the exact macro match first, then execute it, and never find the three partial matches in the command file.

If you enter the character string **verti**, LTPlus will find nothing to execute (assuming that you had no menu or macro matching or beginning with those characters). The program will not look for the command which comes closest to matching; it will only select commands, menus and macros which either match exactly or include the character string as the start of a valid command, menu or macro name.

MENUSETS

The LTPlus package is delivered with a standard set of TILE menusets. These may be modified, copied, removed or just plain ignored by the user at any time. A menuset is a collection of menus which is displayed in TILE 5. A menu is a collection of function, menu or macro names and is displayed in TILE 6. To display a specific menu, use the mouse to highlight and select it from the menuset shown in TILE 5, enter the name via the keyboard, or press the corresponding **CTL-Function** Key.

A menuset is nothing more than an ascii text file, formatted in a particular manner, which contains menus and their associated commands and/or macro names. The following sample menuset file diagram may help the user who wishes to create and/or modify a menuset file using the **edit** command, rather than the one-action-at-a-time **funct_** commands described later in his chapter.

File Contents Description of Contents

```
next_step            ---
view map             |
view block           |__ Permanent commands in TILE 3
view full            |
view zoom            |
save                 ---

setup                --- 1st menuset; appears in TILE 5
{  create            --- The { character starts
   acquire           |   the list of commands for
   status            |   this menuset.
                     |
   poly_map          |
   header            |
                     |--- Functions displayed in TILE
   regis_geo         |   6 when menuset setup is
   regis_dig         |   selected.  Note that blank
   reg_check         |   lines appear as blank fields
   default_head      |   in the TILE 6 block.
   config            ---
}                    --- The } ends the first menuset.

{                    --- This creates a blank line in
}                    ---  the TILE 6 block.

enter_lnwork         ---  2nd menuset; appears in TILE 5.
{                    ---  The { starts a list of functions.
   import            ---
   rotate            |
   reflect_h         |
   reflect_v         |
   negative_img      |
   regis_raster      |
   regis_geo         |
   reg_check         |
                     |
   ginput -a 1       |
   draw line         |-- Menu functions, TILE 6.
                     |
   draw point        |
                     |
   erase fatln       |
                     |
   undo              |
   recall            |
   clear_mode        |
  ginput mouse       ---
}                    --- The } ends the 2nd menuset.
```

TILE MENUS

There are five commands which relate to the development and use of TILE menus: **funct_add** and **funct_del** operate on the currently displayed menu in TILE 6 by allowing the user to add and delete functions. The menu in TILE 6 may be renamed with the **menu_define** command. The entire menuset may be saved to a name specified by the user. The active menuset for a given map is defined by a **menuset** = (name) entry in the **header** file. The user may select another menuset to activate during an LTPlus session by implementing the **menuset_get** command. Some commonly desired menu operations and how to accomplish them are:

Delete a menu entry (function, menu or macro)	Select a menu from TILE 5 Enter **funct_del** (function name)
Add a menu entry (function, menu or macro)	Select a menu from TILE 5 Enter **funct_add** (function name). Note that the function or macro so referenced must be valid and that the new function always appears at the bottom of the menu list.
Delete a menu	Select a menu from TILE 5 Delete all entries using **funct_del**. Save the menuset using the **menuset_save** function.
Delete a menuset	Use the **purge** command, specifying menuset as the file type. Remember that existing maps may still reference the **purge**d menuset name, causing a file management system error.
Copy a menuset	Use UNIX **cp** command. There is no LTPlus function for this action.
Create a new menuset	Copy an existing menuset to a different menuset name. Add/delete entries to the copied menuset menus and use **menu_define** to rename.
Use a menuset	Use the **header** command to set the desired menuset name for the active map.
Change a menuset	Use the **menuset_ge**t command to load a new menuset.

The five commands are more clearly detailed in the *Command Language Reference* section of this manual.

TILE MENU OPERATION GUIDELINES

Some menu operation guidelines:

> Only function, macro or other menu names have any meaning on a menu.
> The macro, menu or function name must be spelled correctly (sorry, JD).
> Functions, menus and macros may appear on more than one menu.
> Selecting a menu name from a menu is the same as selecting it from the menuset display in TILE 5--it causes it to be displayed in TILE 6.
> A maximum of 19 entries will be visible for any given menu (TILE 6).
> A maximum of 19 menus will be visible on the menuset display (TILE 5).
> Removing a function from a menu does not remove the function from the system! It simply cannot be activated from that particular menu.
> There is no limit to the number of menus contained in a menuset. Let some common sense regulate creativity, however.

DIGITIZER STICKUP MENU

Next menu please!! A second, common menu type is the digitizer tablet stickup menu. This tool is especially efficient for editing functions. LTPlus has five commands for the development, modification and use of stickup menus. The standard sequence of these commands is:

digmenu_init n row col

This initializes a new digitizer menu creation process. The user must supply the digitizer device number (n = 1 or 2), the number of rows (**row**) and the number of columns (**col**). A digitizer stickup menu is a rectangular grid of squares, each uniquely identified by a row and column location and each capable of containing the properly spelled name of a macro, menu or function.

digmenu_make row col name

A series of these commands can be used to assign menu, macro or function names to the menu boxes.

digmenu_save name

Once the menu, macro and function names have been assigned to box locations, the digitizer menu can be stored by supplying a unique menu name. The file will be stored as **/u/ltp/digmenus/(name)**.

digmenu_plot name p

This command will produce a hardcopy plot of the named menu on the plotter device specified (p = 1 or 2).

digmenu_get name n

Activates the named digitizer tablet menu and assigns it to the digitizer device specified (n = 1 or 2). The menu must be registered (see *Chapter 10*) prior to use.

Usage of the five **digmenu_** commands is detailed in the *Command Language Reference* section of this manual.

DIGITIZER STICKUP MENU GUIDELINES

Some digitizer stickup menu operation guidelines:

> The plotter and digitizer device assignments (1 and 2) are set by the **config** command and recorded in the **config** file.

> A digitizer menu file may be permanently removed from the system using the purge function.

> To modify an existing digitizer stickup menu, use the **digmenu_get** command to activate the desired menu, followed by the **digmenu_make** command(s) to assign menus, macros and functions to the menu boxes (replacing any existing ones), and then **digmenu_save** to store the modified menu.

How To Use a Stickup Menu

A stickup menu is, in reality, a data file which is physically represented by a paper or film plot of a row and column configuration containing command, menu and/or macro names. It is a very powerful feature because it allows the experienced LTPlus user to select and execute operations from the digitizer via the stickup menu entity.

Any menu, command or macro may be included on a stickup menu. There are five user commands available for designing, building, storing, retrieving and plotting stickup menus (see **digmenu_** commands). You may develop as many stickup menus as you want, storing each under a different file name and loading them for use as needed.

One of the primary activities to which this feature is applicable is for manuscript data input and editing. All of the input commands described in *Chapter 9* could be placed on a stickup menu and activated by aligning the cursor crosshair in the desired menu box and pressing a button. If you are really after efficiency and ease, then review all of the **ckeys_** commands in this chapter. Those functions allow you to map LTPlus commands, menus and/or macros to actual numbered cursor buttons which, when used in conjunction with the stickup menu feature, offers more power than most users can wield at any given time!

Here's a scenario where the user communication techniques have been optimized (note that the entry type is defined in parentheses):

ltpgo	(kbd)
digmenu_get mydigm 1	(kbd)
ckeys_get mykeys	(kbd)
ginput -a dig1	(kbd)
coord_type 3	(kbd)
register 7	(kbd)

196. Digitize the SW CORNER. (system prompt)
Center the cursor crosshair over the SW corner of the stickup menu and press the cursor button mapped as a start or continue/increment/decrement command

197. Digitize the NW CORNER. (system prompt)
Center the cursor crosshair over the NW corner of the stickup menu and press cursor button, described above

198. Digitize the NE CORNER. (system prompt)
You will then center the cursor crosshair over the NE corner of the stickup menu and press appropriate cursor button

199. Digitize the SE CORNER. (system prompt)
You will then center the cursor crosshair over the SE corner of the stickup menu and press appropriate cursor button

create new_map	(kbd)
header menu_set=mymenus	(kbd)
header scale=24000	(kbd)
header layer_name=ma	(kbd)
header stplzn=3601	(kbd)
header map_type=polygon	(kbd)
header map_ht=7.5	(kbd)
header map_wd=7.5	(kbd)
register 1 SW=121.5,45.125	(kbd)

(now slide on over to the digitizer)

register 6	(dig_menu)

196. Digitize the SW CORNER. (system prompt)
Center the cursor crosshair over the SW corner tick on your new_map manuscript and depress appropriate cursor button.

197. Digitize the NW CORNER. (system prompt)
Center the cursor crosshair over the NW corner tick on your new_map manuscript and depress cursor button 1.

198. Digitize the NE CORNER. (system prompt)
Center the cursor crosshair over the NE corner tick on your new_map manuscript and depress appropriate cursor button.

199. Digitize the SE CORNER. (system prompt)
Center the cursor crosshair over the SE corner tick on your new_map manuscript and depress appropriate cursor button.

(The system now displays a registration error value, in State Plane feet units invoked by the **coord_type 3** command, above. If a tolerance value has been set in the **config/tolerance** file and the error is within that tolerance, or if no tolerance value was set in the **config/tolerance** file, you may continue by selecting and executing another menu, command or macro. If the registration error exceeds the pre-established tolerance value, then no subsequent processing of this map will be permitted; however, you will be allowed to execute all registration-related commands in an effort to meet tolerance.)

register -d	(dig_menu)
frame_map d g	(dig_menu)
erase rectangle	(dig_menu)

Move the cursor to a position below and to the left of the map label data which lies within the interior of the map area (perhaps placed there in case the manuscript was scanned) and press cursor button 3, which has been programmed (with **ckeys_** commands) to execute the start command with the transmitted XY coordinate pair.

Now move the cursor to a position above and to the right of the offensive label, such that no other valid linework is enclosed in the imaginary rectangle that these two, diagonal points are defining. Press cursor button 1, which has been pre-programmed (and stored in the mykeys file that you loaded earlier in this session) to execute a continue with the transmitted XY coordinate pair.

All data (on pixels) contained within the defined rectangle will be erased (pixels off).

draw line	(ckey 6)

Using the pre-programmed cursor button 3 (start), digitize the line starting point. Continue to record significant points along the line using pre-programmed cursor button 1 (continue). To begin another line, again use the start button 3.

clear_mode	(ckey 9)
save	(dig_menu)
clean_up	(dig_menu)

clean_up is a user-defined macro which this scenario assumes was previously created and included on the **dig_menu mydigm** and constructed previously using the **macro_** commands (see *Chapter 16, The Macro Environment*). Its purpose is to execute several editing commands against the current, active map to delete spurs, fill holes, and generally prepare the manuscript linework for assembly. But, just to make sure.........

log clear	(dig_menu)
log spur	(dig_menu)

redisp_map (dig_menu)
goto log 1 (dig_menu)

Yep....there's a single-pixel spur there.....

wink (ckey 5)

Using the digitizer cursor, position the screen crosshair over the offending pixel and press the appropriate cursor button (start) which will activate the **wink** mode and kill (turn off) that nasty pixel.

goto log + (ckey 8)

....and fix whatever error you find there....

Repeat this goto-fix process until you have corrected all of the spur errors in the log file.

log clear (dig_menu)
log point (dig_menu)
goto log 1 (dig_menu)

...hmmm....nothing there....guess there were no isolated points after the editing macro was done.

assemble (screen_menu)

Prepare the plotter for a paper check plot by loading the appropriate size media and fine-tip pen(s).

plot 1 24000 1 250 (kbd)

Compare the check plot with the original manuscript, annotating any necessary modifications on the paper plot. Repeat portions of the previously described editing process to correct the map data, **clean_up** the updated file, re-**assemble** and, if necessary, re-**plot**. Repeat this entire procedure until you are satisfied that the map object data is accurate and complete.

Usage descriptions for these functions may be found in the *Command Language Reference* section of this manual.

CURSOR BUTTON DEFINITION

Isn't it just AWESOME how much flexibility and power LTPlus gives the user??!! One more example of this is the set of commands which allow the user to map specific commands to cursor buttons, described as follows:

ckeys_make b name

A properly spelled menu, macro or function name is assigned to a specific cursor button b, where permissible values are:

A,B,C,D,L,M,R,0,1,2,3,4,5,6,7,8,9,*,#

ckeys_save name

The currently active set of cursor button key assignments is stored under the name specified (which is located under **/u/ltp/cursor_dfns**).

ckeys_get name

This command will activate the named cursor key assignment set.

ckeys_disp [-p] name

The named cursor button assignment set will be displayed to the screen or printed to the default primary printer (see **config**), if the **-p** option is set.

The detailed usage descriptions for the **ckeys_** commands are found in the *Command Language Reference* section of this manual.

Menus are powerful tools which should be used with some degree of common sense in evidence. Remember....menu madness can be contagious!