Chapter 2 -- BASIC CONCEPTS

This section of the **User's Guide** was designed to provide the user with a generalized overview of the LTPlus process. Each major phase of the map processing activity is described conceptually. An overview of the LTPlus system may be found in *Chapters 1-3*. All of the system capabilities are thoroughly described in *Chapters 3-16*. The hands-on tutorials found in *Chapters 17* and *18* offer a detailed journey through the LTPlus processes defined in this chapter. *Chapter 19* contains all of the LTPlus intrinsic functions, sequenced alphabetically.

The basic LTPlus processing steps include:

> Input (digitize, scan, import)
> Edit (thin and clean the linework)
> Assembly (convert line sequences to objects)
> Attribute (attribute assignment)
> Output (report, plot, export)

The primary processing milestones discussed in this chapter are:

> Getting Started
> System Management Operations
> Help and Information Operations
> Map Information Operations
> Dialog: Tiles, Keys, Buttons and Menus
> File Management Operations
> Display Operations
> Input Modes
> Map Manipulation Operations
> Raster Manipulation Operations
> Attribute Operations
> Set and Criteria Selection Operations
> Plotting and Reporting Operations
> Import and Export Operations
> The Macro Environment

GETTING STARTED
At this point, it is assumed that the LTPlus package is available on your system. If this is not the case, please return to Chapter 1 for a full discussion of the software installation procedure. Once properly installed, LTPlus is initiated by entering:

ltpgo

The **ltpgo** function may be executed with one or more macro name(s) as argument(s); each will be executed, in the sequence entered, when LTPlus starts running (see *Chapter 16*). Another powerful feature available with the **ltpgo** startup command is that which initiates LTPlus in a batch mode, all operations willl occur without operator involvement. The **-m** option causes LTPlus to be initiated and instructs it to use the memory allocations (**malloc**) value found in the **config** file; this option should be tried if LTPlus catastrophically fails without it. The **-b** option automatically uses the **malloc** value in **config**.

SYSTEM MANAGEMENT
System management is the art and science of preventing other people from crashing your computer system. It also has a well-deserved reputation for being difficult. This document is not intended to teach UNIX. It assumes that you will find that training elsewhere. When you feel moderately comfortable with the basics of UNIX, review

Chapter 3 to learn how to customize your LTPlus environment and keep people from trashing, or otherwise degrading, the files generated by and for LTPlus. The **tolerance** value is also described in this section of the *User's Guide*. The intrinsic LTPlus functions which are classified as system management commands include:

config **execute** **exit** **pause** **shell** **wait**

HELP/INFO OPERATIONS
The human interface ranked very highly during the designing of LTPlus. There are six user commands available for displaying some level of help or operational information during an LTPlus session. Every function, menu and macro has a corresponding description which may be displayed interactively in either a long or short format. Both this *User's Guide* and the LTPlus *Technical Reference* may be printed on command. User prompts are displayed automatically. A prompt is a descriptive request from LTPlus which requires a keyboard response from the user. Prompts may be suppressed with the **prompt on/off** function. The following intrinsic LTPlus functions are discussed in greater detail in *Chapter 4*:

barf **burp** **help** **man_print** **next_step** **prompt**

MAP INFORMATION OPERATIONS
There are six commands which provide viewing and editing capability for the internal files associated with each map, with any *data dictionary* file, and with the optional *diary* file. The *diary* file is yet another example of the design team's desire to encourage communication--its sole purpose is to provide a space for each user to record useful information and comments for each map. Although optional, this file offers an excellent opportunity for formally tracking the procedures invoked for a particular map. The intrinsic LTPlus functions, discussed more thoroughly in *Chapter 5*, are:

default_header **diary_print** **dict_print** **header** **status** **raw_print**

DIALOGUE: TILES, KEYS, BUTTONS AND MENUS
The LTPlus system accepts and understands commands from any of four different sources, including TILE/mouse command selection, keyboard command entry, function key press and stickup menu/cursor input (a TILE is a portion of the LTPlus display screen). The user is encouraged to use whichever tool, or combination of tools that she/he finds most comfortable. *Chapter 6* contains valuable insight about menus, menusets, the tile structure, mouse operations, cursor button operations, function keys, keyboard command entry and the digitizer tablet stickup menu. The intrinsic LTPlus functions described in that chapter include:

ckeys_disp	**ckeys_get**	**ckeys_make**	**ckeys_save**
digmenu_get	**digmenu_init**	**digmenu_make**	**digmenu_plot**
digmenu_save	**stickup**	**funct_add**	**funct_del**
menu_define	**menuset_get**	**menuset_save**	

FILE MANAGEMENT OPERATIONS
This category of functions includes creating, acquiring, reading, writing, updating, editing, archiving, retrieving and removing files. LTPlus was developed with the ordinary user in mind; every effort was made to insure that the primary actions could be applied to designated files (or file types) without the need for a user to be a UNIX expert. Each of these file management operations is available to the user for use with individual files or with groups of files specified by a directory or by a file type. Thus, it is not necessary for a user to know that every map file in LTPlus is, in actuality, a group of files stored under the designated map name directory. The user need only know the map name and the desired operation--if it is legal, LTPlus will generate the required UNIX operating system commands to accomplish the request. The intrinsic LTPlus functions in this category, described in great detail in Chapter 7, include:

acquire **backup** **catalog** **create** **edit** **purge** **rename** **retrieve** **save**

DISPLAY AND LEVEL OPERATIONS
IN this category of functions are those which affect the display of a map in the LTPlus application. These commands alter the view screen contents, the coordinate display, the user's view location, automatically generate the names of adjacent maps in preparation for edgematching, and provide a side-lighted vertical view of a digital elevation model (dem) file. The powerful **level** commands are also discussed in this section of the *User's Guide*. LTPlus allows the user to import up to 63 levels of vector data which may be viewed and/or merged with the active map's linework. In addition, up to 35 levels of the current map may also be stored and viewed; some of these levels are pre-set while others are user-definable as sets. The commands described in *Chapter 8* include:

coord_type	**dem_view**	**edge_maps**	**ginput**	**graphics**	**goto**
level_clear	**level_dup**	**level_icopy**	**level_snap**	**redisp_map**	**step**
view					

INPUT MODES
Within this category of commands are four sub-categories: *condition-setting modes*, *condition-implementing modes, mode deactivation*, and a *reverse-action mode* (which is really not a mode at all). The primary characteristic of a mode command is that it does not perform any action, but, rather, sets an environment under which subsequent functions may be executed. Once implemented, a mode remains in effect until it is either cleared or replaced with another mode command. The purpose of the mode commands is to provide the user with environments in which linework may be effectively digitized and edited. The intrinsic LTPlus mode commands, described in more detail in *Chapter 9*, include:

atrbt_contur	**atrbt_dig**	**atrbt_line**	**bridge_gap**	**connect**	**decrement**	**draw**
erase	**erase_seg**	**increment**	**query**	**recall**	**separate**	
vertex_move	**wink**	**start**	**continue**	**clear_mode**	**undo**	

MAP MANIPULATION OPERATIONS
Map manipulation commands may be defined as those which interact with data elements which are associated with a specific map. These commands may also be distinguished from their raster manipultion counterparts because they operate primarily upon vector, or object, elements, and not at the building block, or pixel, level. This category includes commands for registering a map, constructing map topology, fixing a degraded map file, logging object type errors, snapping and thinning vertex points, creating and using map lists, and overlaying the vectors of two maps. Described in more detail in *Chapter 10*, these intrinsic LTPlus functions include:

assemble	**b_edit**	**fix_map**	**log**	**mass_update**	**mlist_create**
mlist_exec	**overlay**	**register**	**reg_copy**	**vertex_snap**	**vertex_thin**

RASTER MANIPULATION OPERATIONS
Every map has a raster, or pixel level, representation upon creation in the LTPlus environment (...you ought to be in pixels...). Nearly all of the editing operations performed upon a map are implemented against the map's raster. Once the raster appears to be correct, the raster data is transformed into vector, or object-oriented, information. If the raster is once again modified, it will no longer correspond to any previous vector representation until it is reassembled (*topologically structured*).

Users must be aware of the relationship between the raw raster data and the vectorized representation which is used to generate reports, export files and plot files. The raster manipulation commands are used to modify a map's raster in preparation for topological structuring, often in conjunction with the input mode commands described earlier. There are 24 intrinsic LTPlus functions, detailed in *Chapter 11*, which include:

brstr_create	**brstr_delete**	**brstr_switch**	**corridors**	**del_objects**	**del_points**	**del_spurs**
disprstr_ers	**dsprstr_get**	**dsprstr_mrg**	**dsprstr_put**	**erase_margin**	**fill_holes**	**frame_map**
grow_spurs	**margin**	**mask**	**merge_fill**	**negative_img**	**open_solids**	**reduce_res**
reflect	**rotate**	**thin_lines**				

ATTRIBUTE OPERATIONS
An attribute is a characteristic (or group thereof) associated with a given object. The attribute is generally portrayed by a string of one or more alphanumeric characters which represent the characteristic(s) of a point, line and/or polygon object. The equivalencies are often denoted in a *dictionary*, which may be stored as a file in a specific map's directory. The intrinsic attribute functions in LTPlus are designed to work on the attribute strings which are assigned to objects using one of the **atrbt_** mode commands described in the *Input Modes* section of the *User's Guide*. The attribute commands described in *Chapter 12* include:

atrbt_concat	**atrbt_edit**	**atrbt_print**	**atrbt_rec**
atrbt_serial	**atrbt_set**	**atrbt_thin**	

SET AND CRITERIA SELECTION OPERATIONS
A set is a group of objects whose members exhibit the characteristics defined by the user as set selection criteria. LTPlus provides 22 commands for the development of selection criteria and the manipulation of sets. Complex set operations may be defined, stored and applied to individual maps, or to groups of maps, as named in a map list. Set operations are extremely powerful; they provide the ability to generate selective plots and reports, and to extract specific information for exporting to other applications. The instrinsic LTPlus set and criteria selection functions, described in *Chapter 13*, include:

crit_adj	**crit_arang**	**crit_area**	**crit_astrg**	**crit_clear**	**crit_exec**
crit_length	**crit_newatr**	**crit_psides**	**crit_record**	**crit_report**	**crit_serial**
crit_slen	**crit_subel**	**crit_superel**	**crit_type**	**set_clear**	**set_comp**
set_exclude	**set_intrsec**	**set_report**	**set_union**		

PLOTTING AND REPORTING OPERATIONS
Once processed, data may be output in a number of ways. The user may generate printed or screen reports, color pen plots and a variety of ascii file formats for export to external applications. Reports and plots may also be output to an ascii file for editing and/or transmission to another computer or network. As previously mentioned, sets of objects may be defined and extracted for any of the aforementioned output activities. The six intrinsic plotting and reporting LTPlus functions, described in *Chapter 14*, include:

area_plot	**report**	**p_create**	**pers_plot**	**plot**	**plot_ascii**

IMPORT AND EXPORT OPERATIONS
LTPlus allows the user to generate an ascii export file in a predesignated format for input to any of a number of other, non-resident applications including MOSS, ARC/INFO, DWRIS and AutoCad. The following formats are available for exporting data from LTPlus:

cell	Generic grid-based raster data (USFS)
cff	Cartographic Feature File, vector quad-based layer data (USFS, GSC)
dem	Digital Elevation Model (USGS)
dig	A vector format for ERDAS
dlg3	Digital line graph, level 3, optional vector (USGS)
dxf	AutoDesk/AutoCad CAD vector format
moss	MOSS import/export vector format
rdef2	Raster Data Exchange Format (USFS, DWRIS/LIDES/LTPlus)
rstr	Generalized raster format
digit	Vector & attribute data for GRASS GIS (SCS)
scsgef	SCS Geographics Exchange Format, vector (SCS)

The LTPlus data file (internal format) is unaffected by the **export** function. A map may be exported any number of times, in any number of formats, because no modification to the internal data takes place.

LTPlus offers the following formats for **import** of data files generated by LTPlus itself and by other applications. It is also important to note that any file format described as vector must be imported into a user-designated level (see *Chapter 8*). In contrast, a file format defined as **raster** will be loaded directly onto the current map's raster, destroying the existing raster. The **import** formats include:

optr9	Raster format output by Optronics scanners (USFS, GSC)
ras	Raster format output by the Houston Instruments ScanCad system
rle1	tandard Run Length Encoded raster format (VIDAR Scanner, IDEAL Scanner)
asn	Strip run length encoded raster format (USFS, GRAMS output)
digit.ascii	Vector format used by GRASS (SCS)
digit.att	Attribute format used by GRASS (SCS)
map	Another LTPlus map (internal LTPlus format)

The import and export operations intrinsic LTPlus functions described in *Chapter 15*, include:

import **export**

THE MACRO ENVIRONMENT

A macro is a set of one or more LTPlus commands (or other macros) which can be stored and applied to individual or to groups of maps defined in a map list, at any time, and as many times as desired. It is most useful for applying a sequence of steps to a group of maps, or for applying the sequenced functions in a batch (non-interactive) mode. Macros are created by the user, described and stored for later use and may then be activated by anyone with access to the macro directory.

The LTPlus software package will likely contain some standard macros for generalized map processing. Any available macro is treated like an individual function--it can be executed by entering the macro name as a command, or placed on a menu for mouse selection, or assigned to a digitizer stickup menu for selection with a cursor. Ain't life wonderful?! The macro environment is discussed in greater detail in *Chapter 16*. The intrinsic LTPlus functions are:

macro_begin **macro_hist** **macro_end**

A FINAL HELPFUL HINT

Many of the LTPlus *Command Language* functions require some of the same parameters. This is especially true for the various codes, including, but not limited to: coordinate type, object type, level type, raw data file type, device type, action type, shape type, registration type, attribute type, label type, data file type, length units, area units, etc. Rather than list a common code equivalency with each function for which it is required, the table is usually listed once and then referenced from the other function descriptions.

We thought this was a good plan until this author had her fill of trying to find the original table references on a frequent basis. Disgust is a worthwhile motivator. Consequently, all of the code equivalencies which are used by more than one Command Language function are found in a set of code tables in *Appendix B*.

The original draft *User's Guide* included the function *help* files as part of the various chapter discussions. However, it soon became evident that locating that useful information for a specific function was difficult because it required remembering the appropriate chapter number. Thus, in this final version of the manual, the chapter discussions are limited to describing the functions in terms of the overall process invoked. The more complete usage reference for each function is now contained in the LTPlus *Command Language Reference*, an alphabetically sequenced listing found in *Chapter 19*.

Other valuable information will be added to this *User's Guide* as *Appendix Sections*, found at the back of the manual. In addition, an *Index* has been compiled to assist the user in locating manual references by subject or function. Note that all pages in the *User's Guide* are numbered as whole numbers with reference to the chapter (i.e., 1-2, 1-5, 10-2). Pages modified in future User Guide revisions will either be complete replacements or supplied as inserts, with appropriate page numbering for defining their locations (i.e., 1-2a, 1-5d, 10-20f). Happy trails.................