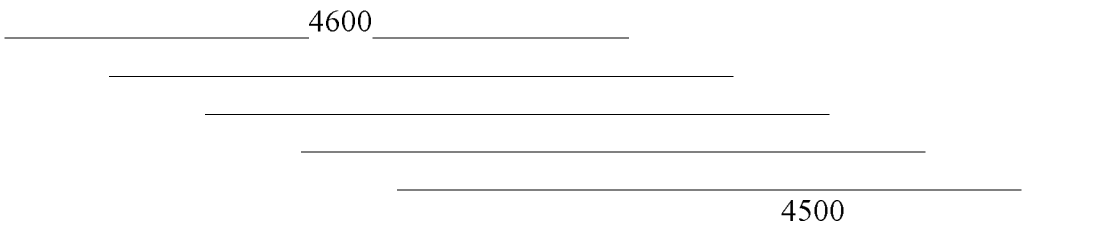
Chapter 18 -- A CONTOUR LAYER TUTORIAL

The LTPlus package was developed with two major applications in mind. As described in the previous chapter, LTPlus is an effective tool for digitizing, modifying and formatting resource data for entry into a GIS. The other primary purpose for which the system was designed is to replace the Line Trace software at the Geometronics Service Center (GSC). The major use of Line Trace is for the digitizing of scanned contour layers which will be processed and output in Digital Elevation Model (DEM) format. This data is then available for a wide range of applications, including orthophoto production and as a valuable layer in a GIS. This chapter provides a detailed description of LTPlus in this role.



PRE-WORK

The quad-based contour layer must include the four quad boundary corner ticks for proper geodetic registration in LTPlus. The layer will be scanned and a data file containing the raster representation of the map data will be transmitted to the LTPlus system in a pre-designated format (**optr9**).

The system manager will pre-set a data path where the raw scan files will be stored, awaiting LTPlus processing. For the purpose of this tutorial, we will assume that the import path has been established and that LTPlus has been advised of its existence via the **config** file **item=string** argument, **import_path=/xx/xx/xx/xx** (See *Chapter 3, System Management*, **config**). Once set in the **config** file, the path will appear as the default during the **import** operations. In addition, the system manager will have pre-set the raster height and width variables (**rasterh**, **rasterw**) in the **config** file to match the scanner output.

CONTOUR LAYER PROCESSING

The power of LTPlus is most evident when designing a sequence of processing steps; there is simply no single, logical route. What we are suggesting in this tutorial is only one possible method and should be modified and/or replaced to best meet your needs.

Located under the default **import** data path, the [fictitious] map used for this example is called **Browbeaten**. It is a standard 7.5-minute, 1:24,000 scale quad with a **SE** corner latitude of **44-07-30** and longitude of **122-30-00**. The contour interval (**CI**) is **40** feet. Here is a processing sequence which will produce an ascii DEM data file:

/u[sr]/ltp/bin/ltpgo This starts LTPlus going

First, the information specific to this map must be recorded in the map's **header** file:

header contour_interval=40
header cursor_file=contur datum=27
header layer_name=topography
header map_ht=7.5
header map_wd=7.5
header menu_set=cmenu
header quad_unit=Browbeaten
header scale=24000

header search_rad=5
header source_yr=83
header state=OR
header stplzn=3601

With this command, you have established that the quad called **Browbeaten**, which is a 7.5 x 7.5-minute, 1:24,000 scale map compiled in 1983 and based on the **1927 datum**, has a **contour interval** of **40** feet, is in the state of Oregon and has a State Plane Coordinate **zone code** of **3601**. In addition, the layer has been designated as topography and any raster file search will be confined to a **radius** of **5** pixels. And, finally, you have designated that two, previously created files are to be accessed for processing this map: a cursor definition **set** called **contur** and a **menuset** file called **cmenu** (See *Chapter 6, Menu Operations*).

Next, you must open a new set of files in which to process the raw scan data:

create browbeaten
Creates a set of files under the default directory called **.../data_path/maps/browbeaten**.

To move the raw scan data into the map file system:

import optr browbeaten 3 1
This operation is known as importing and may be done without prior creation of the map file; however, the data will be lost after the import action is exited unless a map file has been created.

Up until this point, your communication with LTPlus could have been accomplished by entering the displayed commands via the keyboard, or by selecting the primary command from a tile (screen or stickup) menu and then responding to the resulting LTPlus prompts for the additional parameter(s) required for each function. The next step is best accomplished by selecting the register function from your tile (screen) menu, with the cursor or mouse, and then providing the required information in response to the system prompts. This is the logical scenario:

Select **register** from the screen menu

The system will query:

50. Enter COORD type code.

Your response for this operation should be:

0, identifying pixels as the units of choice.

Next, the system will prompt:

167. Enter the CORNER IDENTIFIER.

Your response should reflect the corner which you are intending to next locate. All four must be entered and the order is not significant.

SW

Then, using the mouse or the cursor, whichever is the current graphics input device (**GIN** - See *Chapter 8, Display Operations*, **ginput**),

Position the screen cursor over the corner tick in the SW corner of the raw scan image and depress the data entry button (1 on digitizer cursor or any one on the mouse).

Repeat the **register** command procedure for all four corners.

Next, the map must be registered to its true ground location:

register 1 SE=44.125,122.5

This single command, best entered directly from the keyboard, supplies the geodetic registration for this map. Because you entered the **map_ht** and **map_wd** values into the **header** file earlier, LTPlus is capable of calculating all necessary ground-based corner data from a single-corner **lat,long** entry. Any corner will do, as long as it is correctly identified.

With the the scan image and the geodetic registration processes complete, LTPlus can provide a preliminary evaluation of your map registration activity by predicting how well the raw scan image shape matches the true geodetic map shape. You will be advised of any significant deviation and, thus, alerted to any possible data entry error. If this condition occurs, check all of the previously entered information (geodetics, corner identification and location on the scan image, etc.) and try it again.

To prepare for map editing, it is now necessary to securely attach the contour layer base to the digitizing tablet and properly register its location. To reset the graphics input device from the mouse to the cursor on the primary digitizer (again, see *Chapter 3, System Management*, **config**):

ginput dig1

As with the scan image registration procedure, it is most efficient to select the command from a screen menu, now using the **dig1** cursor, and allow LTPlus to request the required parameters.

Select the **register** command from the screen menu.

The system will query:

50. Enter COORD type code.

Your response for this operation should be:

4, identifying **dig1** relative coordinates as the unit of choice.

Next, the system will prompt:

167. Enter the CORNER IDENTIFIER.

Your response should reflect the corner which you are intending to next locate. All four must be entered and the order is not significant. However, users generally develop a pattern, often a clockwise format, beginning with:

SW

Then,

Position the digitizer cursor over the corner tick in the **SW** corner of the quad layer and depress the data entry button.

Repeat the register command procedure for all four corners.

Typically, a published contour layer requires editing to correct or eliminate the following conditions:

convergent lines, indicative of a steep slope or cliff
contour number labels
segments which fall short of the neatline
any specialized hatchuring or line dashing

Most of these conditions can be processed without operator involvement by developing and using a *macro* which applies the necessary commands to any number of maps, non-interactively (See *Chapter 16, The Macro Environment*). For the sake of expediency, this tutorial describes and utilizes a *macro* called **gsc_mac1**; the command sequence contained in **gsc_mac1** is shown.

However, prior to executing that *macro*, some manual, interactive editing may be necessary to correct the aforementioned convergent contour line condition. To prepare for this process, it is important to directly interface the digitizer cursor movement motion on the map with the screen cursor movement on the map image on the screen. To do so,

ginput -a dig1
the **-a** option indicates that any horizontal positions transmitted from the digitizer will be in *absolute* X,Y coordinates (i.e., relevant to a user-specified coordinate system, rather than *relative*, which supplies unscaled screen cursor motion which merely reflects the current cursor position relative to its previous position.

You may also wish to designate the coordinate display units by entering,

coord_type 3 units = STPln feet

And one more suggestion--use a digitizer *stickup menu* for command entry while using the digitizer in *absolute* mode. LTPlus offers a variety of ways to execute commands, including keyboard, mouse-screen menu, digitizer cursor-screen menu, cursor key definitions, and digitizer cursor-stickup menu. While each and every one is a valid means of command execution, some lend themselves more readily to particular processes than do others. Review *Chapter 6, Menu Operations*, for a more complete understanding of the data entry options.

In this tutorial, we will assume that you are going to use a *stickup menu* called **gsc_smenu1** and a set of cursor key definitions stored in a file called **gsc_ckeys1**. To activate both, enter (via the keyboard):

digmenu_get gsc_smenu1 1

and

ckeys_get gsc_ckeys1

Attach the physical *stickup menu* to the digitizer and **register** its location in the following manner:

Enter the **register** command from the keyboard

The system will query:

50. Enter COORD type code.

Your response for this operation should be:

7, identifying **dig1** *stickup menu* coordinates as the units of choice.

Next, the system will prompt:

167. Enter the CORNER IDENTIFIER.

Your response should reflect the corner which you are intending to next locate. All four must be entered and the order is not significant. However, users generally develop a pattern, often a clockwise format, beginning with:

SW

Then,

Position the digitizer cursor over the corner tick in the **SW** corner of the *stickup menu* and depress the data entry button.

Repeat the **register** command procedure for all four corners.

You may now execute commands from the *stickup menu* by positioning the digitizer cursor over the desired command menu box and depressing any button. The commands on this menu may be simple (without parameters), which will cause LTPlus to prompt for any missing data, or complex (with all required parameters). A *stickup menu* box can contain any command or *macro* (sequence of commands). The cursor key assignments may also be any command, simple or complex, or *macro* (sequence of simple and/or complex commands).

Visually review the map manuscript, looking for instances of convergent contour lines. For every occurrence, follow these steps:

Activate the desired viewing mode from the **stickup menu** (See *Chapter 8, Display Operations,* **view**).

Move the digitizer cursor (tracked in *absolute* mode by the screen cursor on the scan image) to the occurrence of convergent lines. The actual raster representation will likely be a mass of pixels from which no obvious individual contours may be accurately distinguished.

Use **separate** or **erase** modes in conjunction with the **start** and **continue** commands to clean up the mess (See *Chapter 9, Input Mode*).

Use the **connect** mode in conjunction with **start** and **continue** commands to replace the previous block of pixels with a single line of pixels connecting two segments which have the same elevational attribute.

Repeat these steps for all instances of convergent lines and then,

save for insurance!

All of the remaining, aforementioned editing conditions can probably be corrected by executing the *macro* called **gsc_mac1** which contains the following sequence of commands (*Chapter* references in parens):

frame_map d g	*(Chapter 11)*
thin_lines 0	*(Chapter 11)*
grow_spurs 2	*(Chapter 11)*
del_points	*(Chapter 11)*
del_spurs 30	*(Chapter 11)*
erase_margin	*(Chapter 11)*
frame_map e g	*(Chapter 11)*
assemble	*(Chapter 10)*
crit_clear	*(Chapter 13)*
set_clear 0	*(Chapter 13)*
crit_type 14	*(Chapter 13)*
crit_length 0 0 30	*(Chapter 13)*

crit_exec 0 sm int poly	*(Chapter 13)*
del_objects 0	*(Chapter 11)*
log clear	*(Chapter 10)*
log spur -1	*(Chapter 10)*

To execute this *macro* against the Browbeaten map, you must first create a maplist which includes the quad name (See *Chapter 10, Map Manipulation*):

mlist_create mlist1, browbeaten

Note that if you wish to execute this *macro* against several maps, you may do so by entering several map names into the **maplist** file.

You may now execute the **gsc_mac1** *macro* interactively by entering:

mlist_exec mlist1 gsc_mac1

When the processing is complete, you will be ready to view and correct any extraneous spurs which, if found, were recorded into the **log** file. Use the **goto** command (*Chapter 8, Display Operations*) to step though the errors, if any, and the **wink** and **erase** modes in conjunction with the **start** and **continue** commands (on your *stickup menu* or cursor key definitions) to remove or otherwise eliminate spurs from the map. If any changes are made, you will need to re-**assemble** the map.

ATTRIBUTE ASSIGNMENT

Every contour line segment is now an object in the LTPlus environment, and, as such, may have one or more associated attributes assigned and recorded. You must decide whether or not to allow more than one attribute (elevation) per line segment. The advantage of permitting multiple attributes is that the validity of the attribute assignment process may be tested by repeating the entire process and then comparing the first and second attribute strings, and identifying and correcting any conflicts which occur. If multiple attribute assignments are not allowed, then any subsequent attribute assignment for a specific object will replace the current one.

To start the attribute assignment process (allowing multiple assignments in this example) with a contour elevation of 5000 feet, enter:

atrbt_contur 3 5000

This activates the *contour attribute assignment mode* with an initial elevation of 5000 feet.

Next, determine the starting and ending points of the elevation profile line which you are intending to define.

Position the cursor to the outside of the 5000-foot contour line, with regard to the end point of the profile line, and press the cursor button to which the **start** function has been assigned.

Then, position the cursor over the desired end point of the profile line and press the cursor button which activates the appropriate slope-definition command. For example, if the movement from the start to the end of the profile defines a constantly ascending slope, select the button which executes the **increment** command. If the slope is constantly descending, then press the button to which the **decrement** command has been assigned. *Do not* define a profile line which crosses (contains) any mixture of ascending and descending contours. The profile must be either a consistent ascent or a consistent descent.
Every contour line which is crossed by the profile line can now be given an elevation value as per these rules--the first line crossed will be given the elevation supplied by an **atrbt_contur** mode command or from a calculated elevation value based on prior sequences of **increment** and **decrement** commands. At any time, you may re-initiate the elevation by executing an **atrbt_contur** command (with the appropriate parameters, of course).

Subsequent elevation attributes will be calculated based upon the profile starting point, the direction of slope of all subsequent contour lines crossed by the profile line (increment or decrement), and the *contour interval* as specified in the **header** file.

Theoretically, attributes for an entire contour map could be assigned from a single **atrbt_contur** statement followed by a **start** command followed by an appropriate sequence of **increment** and **decrement** commands.

ATTRIBUTE ASSIGNMENT ERRORS

There are some commands which can assist you with looking for any erroneous attribute assignments. The efficiency of these commands is dependent upon your initial choice about whether or not to allow multiple attributes. So. . . . .

If you permitted multiple attributes, you may wish to use one or more of the following methods/commands:

atrbt_thin -2 0	This will remove any redundant attributes for every object in the file (See *Chapter 12, Attribute Operations*).
atrbt_edit -1	Visually scan the ascii attribute file looking for instances where multiple attributes for a single contour line object do not match.
crit_clear **crit_type 13** **set_clear 0** **crit_exec 0** **plot 1 24000 1 250** or **report 1 2 2 2 6 0**	This sequence of commands will find all instances of contours with attribute assignments which are either greater or less than one contour interval from adjacent contour line attributes. The results may be plotted or reported (listed).
crit_clear **set_clear 0** **crit_slen 0 0** **crit_exec 0** **plot 1 24000 1 250** or **report 1 2 2 2 6 0**	This sequence of commands will find all instances of contour lines which have no attributes currently assigned. The results may be plotted or listed.

If you did not permit multiple attribute assignments, then only the last two methods will produce meaningful results.

REVIEWING THE RESULTS

It is left to the LTPlus operator and project manager to determine the level of quality checking which should occur for a given quad or quad set. In some cases, a report or screen plot may be sufficient to insure that the map has been adequately digitized and processed. In other cases, a hardcopy plot may be needed to compare with the original manuscript to determine if the data capture and processing were accomplished successfully and accurately.

EXPORTING THE DATA

Let's assume at this point that you have now completed processing of the **Browbeaten** map data and that you are ready to generate an ascii DEM-format file ready for transmission to another application. The command to accomplish this miracle is:

export dem 2 7 7 2 1 2 1

This, of course, contains some special parameters which have been preselected for this example. This command will generate an ascii file, located under the default **export_path** (defined in the **config** file) in a DEM format. That particular format requires seven additional pieces of information, in the order shown above:

> **grid coordinate type (2 = UTM)**
> **cell width (we said 7 meters, units based on previous parameter)**
> **cell height (again, we said 7 meters, ditto the rest)**
> **clipping coordinate type (how about UTM meters!?)**
> **map attribute units (feet)**
> **output dem units (meters, which is normal)**
> **smoothing filter passes (1 seems reasonable)**

That's about it for this tutorial...we sure hope that you had more fun following it than we did in attempting to describe it! Good luck!