Chapter 16 -- THE MACRO ENVIRONMENT

macro. Somewhere along the line, this innocuous little term has grown to represent a very powerful concept. In the lingo of LTPlus, a *macro* is defined as a collection, or sequence, of commands, stored under a unique name, which may then be referenced and executed in the same manner as an intrinsic LTPlus function. The inherent value of a *macro* is that it may be activated against one or more maps (map list), and it may be used repeatedly.

INTERACTIVE VS BATCH OPERATIONS

While LTPlus is a system which supports an active, online interaction with the user, many of its operations can be effectively implemented in a batch, or background, environment. The availability of *macros* and map lists allows the user to process large quantities of information during non-peak hours, or, under UNIX, in *separate windows, or partitions (*not implemented in Release 2.0).

Editing, assembly, reporting, backup and retrieval processes all operate very efficiently in batch. Clearly, those activities which normally demand user interaction, such as plotting and digitizing, are not recommended as batch activities.

BUILDING AND USING MACROS

The trick to building a useful **macro** lies in understanding the LTPlus process well enough to isolate those parts which are commonly applied to every map and which operate well without user interaction. The actual sequence of commands can then be entered and stored, and the macro name can be added to the menuset for easy access.

There are three commands which facilitate the development of macros--**macro_begin**, which opens the macro design environment, **macro_hist**, which allows the user to extract command sequences from the current history buffer, and **macro_end**, which closes the macro design environment and returns the user to an active command state.

INITIATING MACRO ELEMENT COLLECTION

The **macro_begin** commands invokes the *macro* collection mode. All subsequent commands are stored in the specified file name and no further action is taken on the current map until a **macro_end** is encountered.

ENDING A MACRO COLLECTION PROCESS

Ahhh...the easiest section of this manual to write. The **macro_end** command terminates the **macro_begin** mode.

CREATING A MACRO FROM A HISTORY FILE

Every map has an associated **history** file which contains the last 512 user-entered commands and/or *macros*. This is a running **history** of what has happened with the map. This file is available for generating a tested *macro* by an automated cut & paste operation invoked with the **macro_hist** command. After entering the **macro_hist** command, with no arguments, from the keyboard (there must be an active map), the user will be prompted to enter a new *macro* file name (into which the extracted **history** file commands will be copied). LTPlus then shells out to a listing of the contents of the current map's **history** file; each command line is numbered. The system prompts for a beginning and ending line number from which to develop the *macro*. The user may review the **history** file by entering a + or - and [**RETURN**]. When the beginning and ending line numbers have been identified, the user is asked to provide a description of the *macro* which is used to describe the *macro* when it is accessed with the **help** command. The **history** file lines so identified are then copied to the new *macro* file.

MACRO USAGE

Once a macro has been constructed, it may be initiated by entering its unique name, or by selecting it from a menu. To install a macro on a menu, review the discussion of Menu Operations (**funct_add, menu_define, menuset_save**) in *Chapter 6*. A macro may contain commands as well as other macro names (this is known as nesting macros). LTPlus can be instructed to run a specific macro upon startup by invoking the **ltpgo** command with the macro name as an argument.