

Laser-Scan Ltd.

LITES2

Cartographic Editing System

User's Guide

Issue 3.0 - 06-Mar-1986

Copyright (C) 2019 Laser-Scan Ltd
Science Park, Milton Road, Cambridge, England CB4 4FY tel: +44 (0) 1223 420414

Document "LITES2 User"		Category "USER"
Document Issue 1.0	R J Hulme	18-Feb-1985
Document Issue 2.0	R J Hulme	07-Mar-1985
Document Issue 3.0	R J Hulme	06-Mar-1986

CONTENTS

CHAPTER 1	INTRODUCTION	
1.1	Intended Audience	1-1
1.2	Use of this Manual	1-1
1.3	Program Description	1-1
1.4	Environment	1-2
1.5	Input Data	1-2
1.6	Feature Selection	1-2
1.7	Command States	1-2
1.8	Command Language	1-3
CHAPTER 2	STARTING A SESSION	
2.1	Running the program	2-1
2.2	Initialisation	2-1
2.3	Set Up	2-2
CHAPTER 3	SOME BASIC COMMANDS	
3.1	Command Information	3-1
3.2	Cursor Movement	3-1
3.3	Windowing and Redrawing the Map	3-2
CHAPTER 4	SIMPLE FEATURE OPERATIONS	
4.1	Identifying a Feature	4-1
4.2	Abandoning an Operation	4-1
4.3	Examining a Feature	4-1
4.4	Deleting a Feature	4-2
4.5	Recovering a Feature	4-2
4.6	Searching for Selected Features	4-2
CHAPTER 5	EDITING LINES	
5.1	Making Changes to a Feature	5-1
5.2	Moving a Feature	5-2
5.3	Copying a Feature	5-2
5.4	Joining Features	5-3
5.5	Tying Features	5-3
5.6	Closed Loops	5-4
5.7	Removing a Data Point	5-4
5.8	Inserting a Data Point	5-4
5.9	Editing a Data Point	5-5
5.10	Deleting Part of a Feature	5-5
5.11	Changing, or Copying and Changing Part of a Feature	5-6
5.12	Moving, or Copying and Moving Part of a Feature .	5-7
5.13	Extending a Feature	5-7

5.14	Replacing Part of a Feature	5-8
CHAPTER 6 EDITING TEXTS AND SYMBOLS		
6.1	Making Changes to a Text or Symbol	6-1
6.2	Moving a Text or Symbol	6-1
6.3	Copying a Text or Symbol	6-2
6.4	Changing the Size of a Text or Scaled Symbol . . .	6-2
6.5	Rotating a Text, Oriented Symbol or Scaled Symbol	6-2
6.6	Editing a Text String	6-3
6.7	Changing the Size and Orientation of a Scaled Symbol	6-3
CHAPTER 7 CREATING NEW FEATURES		
7.1	Creating a Line Feature	7-1
7.2	Closed Loops	7-1
7.3	Creating a Text Feature	7-2
7.4	Creating a Symbol Feature	7-2
CHAPTER 8 FINISHING A SESSION		
8.1	Normal Completion	8-1
8.2	Ignoring all Editing	8-1

CHAPTER 1

INTRODUCTION

1.1 *Intended Audience*

This manual is intended as the introductory guide to the use of the Laser-Scan Cartographic Editor LITES2. It describes the basic operations required to run the program, initialise it, read in and draw map data, make some edits, and finally preserve the edited data in a new file. It is assumed that users are already familiar with general use of the equipment and the DEC (Digital Equipment Corporation) VAX/VMS operating system, and with the principles of digital cartography.

For a more detailed description of particular commands, and for a description of many commands which are not used in a simple editing session, you should consult the "LITES2 Reference Manual". Note that the reference manual assumes that you are already familiar with the contents of this document.

1.2 *Use of this Manual*

Chapter 1 is devoted to background information, and introduces users to the concepts involved, as well as providing a broad description of some of the facilities available. It is advisable to read through this chapter before starting a LITES2 editing session.

The remaining chapters take you through a simple editing session, and the easiest way to learn about LITES2 is to follow these instructions at a LITES2 workstation. Your site will probably have an example data file, which may be used as the input map for this purpose.

In this manual the commands you have to give are printed in bold block capitals, and are separated from the rest of the text for easy reference. Sequences of commands separated by slashes (/) mean that any number of those commands can be used, in any order.

1.3 *Program Description*

LITES2, the Laser-scan (second generation) InTeractive Editing System, is an interactive graphical editing program which has been designed to be particularly suitable for work with cartographic type data. It can, however, be used on other types of feature-orientated data, and there are facilities for reading, drawing,

amending, creating, and deleting features. The program is a reimplementa- tion of the previous LSL (Laser-Scan) cartographic editor LITES1 (also variously known as LITES, MADES, IGES or SOLADI)

1.4 *Environment*

LITES2 will run on a DEC VAX series computer running under the VMS operating system, and can be used without graphic interaction on any DEC-compatible terminal. For graphic interaction an LSL-supported GKS (Graphical Kernel System) workstation configuration is required. A typical example might include an alphanumeric terminal, a graphics screen and a digitising table.

LITES2 needs to read various files of information in order to operate, and these, together with its data files are located by means of standard 'logical names'. These have to be set up by the system manager, and the operator should not have to be concerned with them. They are usually set up in a central standard login command file, together with definitions of the command abbreviations required to run the program.

1.5 *Input Data*

LITES2 is used to display and edit map data which is contained in IFF (Internal Feature Format) files. (IFF is the standard Laser-Scan format for map data.) Each IFF file contains a single map, and data within a map may be grouped into layers (sometimes known as overlays). The basic elements of map data are called features, and the editor deals with a total of twelve different categories or graphical types of features. For the purposes of feature identification in LITES2 these are divided into four groups, namely lines, circles, texts and symbols. Further details of feature representation can be found in the "Introductory Guide to Feature Representation on Laser-Scan Display and Plotting Devices", and in the "FRTLIB Reference Manual".

1.6 *Feature Selection*

Two fundamental concepts of the editor are those of the found-feature and the feature-in-hand. The found-feature is a feature selected by the operator to be the subject of editor operations. When an editing command is given, any found-feature becomes the feature-in-hand in order that a second feature can be found if needed for the editing operation. The current found-feature, and feature-in-hand, if any, are highlighted in refresh mode on the display screen.

1.7 *Command States*

LITES2 always operates in one or other of a small number of command states. Within each state, a given range of commands is valid and certain commands or operator actions cause the program to move between states. Some commands are valid in all states, while others are limited to only a few states. The states are listed below -

- * INITIAL state is entered at the start of a session, and it is in this state that things like options, menus, attribute files and data files are specified.
- * READY state is the principal operating level of the editor, and the majority of commands are available in this state
- * LINE state is entered when the found-feature is linear
- * CIRCLE state is entered when the found-feature is circular
- * TEXT state is entered when the found-feature is a text
- * SYMBOL state is entered when the found-feature is a symbol
- * EDIT state is entered when complex editing operations are performed on linear or circular features
- * MODIFY state is entered when complex editing operations are performed on text or symbol features
- * ON state is entered by commands which result in the cursor being constrained on a linear or circular feature
- * CONSTRUCT state is entered when a new linear or circular feature or part feature is being digitised
- * RECOVER state is entered when the found-feature is a deleted feature
- * WINDOW state is entered during definition of an area of map to be enlarged onto the graphics screen.
- * MACRO state is entered while defining macro commands.
- * AC state is entered to allow editing of ancillary coding

1.8 *Command Language*

A primitive command in LITES2 consists of at least a primary command, followed in some cases by one of several secondary commands. The primary command, with its secondary command if present, may be followed by compulsory or optional arguments. The number and type of arguments depends on the command, but these can be integer numbers, real numbers, text, or a filename.

Primary and secondary commands, as well as macros (see below), can be shortened to the minimum non-ambiguous abbreviation. They can be given to the program either from the keyboard, from the menu, or from any of the digitising table cursor puck buttons. The menu and puck buttons are simply quick ways of entering commonly used commands, and you can use LITES2 without issuing commands in this way.

Usually commands are terminated by end of line (carriage return), but several commands can be given per line if separated by # which is treated as end of line.

There are facilities in LITES2 for creating your own commands, called macro commands, from combinations of primitive commands, @file commands (see below) and other macros. Similarly, sequences of frequently used commands can be prepared in command files, which can be executed by typing @ followed by the filename. The commands within the files can be primitive commands, macros, or @file commands.

It is possible to alter the flow of control within macros or command files by means of special commands which allow conditional execution and jumps. These commands can be used in conjunction with LITES2 variables, which can be either system variables or ones defined by the user.

Although it is quite feasible to operate LITES2 using primitive commands alone (and this is the method employed in this guide), most installations will make full use of the facilities provided by the command language which is indeed a very powerful editing tool.

Full details of the command language facilities are described in the "LITES2 Reference Manual"

CHAPTER 2

STARTING A SESSION

2.1 Running the program

In order to run LITES2, you must be logged in to VMS at a terminal. See your system manager if you do not have the Username and Password required to do this.

On systems running on LITES1 workstations, you may have to run a special start-up sequence at the beginning of the day (or after the computer has been unavailable). If what you type at the terminal appears on both screens, then perform this procedure - it normally requires you to log on as a user "WOSP" or "MADES", with no password. This will perform various actions (including clearing the graphics screen), and will then log out again. The system is then ready for your use.

At some sites, you will be prompted for what you want to do, and will not have to give the actual command to start up LITES2. However, if you are presented with the "\$" (dollar) prompt, then you can start up the program by typing

LITES2

The program will announce itself as

```
LITES2 V[n] of [hh:mm:ss dd-mmm-yy],
```

where [n] is the version number, and [hh:mm:ss dd-mmm-yy] is the time and date of linking of this release. This is followed by a message indicating the licensed users of this version.

There will then be a pause while first stage initialisation is carried out, normally followed by execution of initialisation command files. Messages will appear whenever command files are opened or closed, and you will then be presented with a prompt:

*

This is an asterisk by default, but may have been changed during initialisation.

2.2 Initialisation

INITIAL state is the part of the session where you tell the program what maps you want to read in, what hardware (of that available) you want to use, and

generally set up the environment within which the rest of the session will be conducted.

Some of this work will already have been done for you; each site has a standard initialisation command file, which is read in and obeyed by LITES2 before it gives you that first prompt. This will do things like defining the commands attached to the puck buttons, setting up the various menus that you might use, and specifying options.

When you get the first prompt, the program will still be in INITIAL state, and you can then give some of your own initialisation commands. Since you are probably using LITES2 for the first time, try typing the command

ENABLE LEARNER

This option causes learner-aid messages to be output whenever errors occur, in addition to the normal error messages. The learner messages usually indicate the resulting program action, or give guidance as to what you should do next.

You can cancel options using the DISABLE command. Try

DISABLE STATUS

and

ENABLE STATUS

and note what happens to the status line on the terminal screen.

Before reading in any data, the program requires the FRT, TRI and SRI files to be defined. The necessary commands will probably have been set up for you in a command file, so you can simply type

@EXAMPLE.LCM

where EXAMPLE.LCM is the command file to be obeyed.

The IFF command gives the program the information required for reading in the map data. For a single map, simply type

IFF EXAMPLE.IFF

where EXAMPLE.IFF is the IFF file. It is possible to read in several maps, and to do this you would use the MAPS command followed by an IFF command for each map to be read in. Details can be found in the LITES2 Reference Manual.

2.3 Set Up

After the last IFF command has been issued, you may be prompted to set up your menu(s), map(s), and tracking area, in turn.

Whether or not you are prompted to set up menus depends on the number of MENU commands given during initialisation. If a menu is to be set up, you will be prompted to digitise each of the four corner points, in the order NW, SW, SE,

NE.

If you have not disabled map setup with the SETUP NONE command, you will be prompted to digitise the monument points on the map or maps attached to the table. For example, for a four point setup (SETUP FOUR), you will be prompted to digitise the corner points in the same order as for menus. If you have more than one map, you will be prompted to set each one up in the order that you requested them.

Lastly, you will be prompted to set up a tracking area on the table, if the TRACK command was given during initialisation. Again, the four corner points are digitised in the order NW, SW, SE, NE.

During set up, the last two buttons on the puck have special meaning. For example, on a 16 button puck with the following layout:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

these would be buttons 15 and 16 (if all 16 buttons are in use as a result of the PUCK command).

- * Pressing any button except for the last two (1 to 14 in the example) will digitise a corner point or monument point.
- * Pressing the last button (button 16) will abandon the setup of the current menu, map or tracking area. This facility is useful if you happen to make a mistake; the set up can be repeated using the SETUP AGAIN command (see below).
- * Pressing the last but one button (button 15) will use the values (for the current map, menu or tracking area) that were setup last time LITES2 was run at this workstation. This assumes that
 1. the same map, menu or tracking area was used during the last session, and was set up
 2. the map or menu has not been moved since then, and the map document has not warped

Note that these two items are not checked by LITES2, and rely upon your sense.

When the set up has been completed, the map data is read in to the workspace files and the graphics are drawn on the screen(s). (The workspace files are simply copies of the IFF files.) A message will appear at the terminal when reading in has finished, and the program will be in READY state.

If you made a major mistake during set up and abandoned it, you can now repeat the complete set up by typing

SETUP AGAIN

(which is valid only in READY state). The set up sequence will start again from the beginning; remember that you can press button 15 to use the previous values for the parts of the set up which were not abandoned.

You are now ready to start editing.

CHAPTER 3

SOME BASIC COMMANDS

3.1 *Command Information*

You can get information about commands using the VAX/VMS help facility. Type the command

HELP

to display a list of all of the LITES2 commands. Details of individual commands can be obtained by typing the HELP command with the command you want to know about as a text argument. Try

HELP ENABLE

and

HELP ENABLE STATUS

In order to find out all the commands available in the current state, type the command

SHOW COMMANDS

3.2 *Cursor Movement*

The refresh cursor cross on the screen can be moved in several ways. Type the command

POSITION 100 100

and note the position of the cross. Now move the digitising table cursor puck to a new position and press the appropriate button to give the command

NULL

(Although technically incorrect, your cursor button will probably be labelled with the command POSITION for this operation; this is for simplicity.) The screen cross should now appear at the corresponding position.

The digitising table tracking area corresponds to the area shown on the screen, although the area defined may be a different size to the screen. If you have a tracking area set up, move the cursor puck onto it, and give the command

NULL

Repeat this on different parts of the tracking area, and note the various positions of the cross on the screen.

3.3 *Windowing and Redrawing the Map*

If you wish to scale up a particular part of the map on the screen then the windowing facility can be used. This is useful if the map is cluttered or if greater accuracy is required. First position the cursor over the bottom left hand corner of the required window on the digitising table or screen and give the command

WINDOW or WINDOW MAP

Now position the cursor on the required top right hand corner and give the command

END

The scaled up window will then be drawn.

If you wish to scale up part of this window, position the cursor at the bottom left hand corner of the new window and issue

WINDOW SCREEN

Now position the cursor on the top right hand corner, give the command

END

and the new window will be drawn.

To redraw the current window after you have done some editing, use the command

DRAW or DRAW SCREEN

If you want the whole map back, use

DRAW MAP

CHAPTER 4

SIMPLE FEATURE OPERATIONS

4.1 Identifying a Feature

By positioning the refresh cursor cross on a feature to be edited and giving the command

FIND

the feature itself is highlighted in refresh. Once a feature has been found in this manner it is ready to be operated on. If there are many features very close together then repeatedly giving the FIND command (without moving the cursor) will find up to four features in turn. Try finding a few features, and note that the editor will move into LINE, CIRCLE, TEXT or SYMBOL state as appropriate.

4.2 Abandoning an Operation

Find a feature using

FIND

then give the command

ABANDON

and see what happens. If you wish to exit from the current state at any time then the ABANDON command will abort the current operation and return you to READY state.

4.3 Examining a Feature

The various attributes of a feature (eg. feature serial number, feature code, size etc) can be examined. First find a feature using

FIND

and then try

EXAMINE

This displays a summary of the feature attributes. Note that the program remains in the state it was in when the feature was found.

4.4 Deleting a Feature

In order to delete a feature, you first need to find it by positioning the cursor over it and giving the command

FIND

If the feature is highlighted in refresh then you have found it, if not, then you must try again. Now give the command

DELETE

The feature will be scrubbed out and the editor will return to READY state.

4.5 Recovering a Feature

If you delete a feature by mistake then it is possible to recover it at any time during the editing session. First position the cursor over the deleted feature and give the command

RECOVER

This command acts in a similar way to FIND, so if the feature does not appear refreshed, try again. When you have successfully recovered the feature, the program will be in RECOVER state and you can give the command

END

This command is used to complete many types of operation, and in this case it retrieves the feature, terminates the operation, and takes the program back to READY state.

4.6 Searching for Selected Features

In addition to FIND, feature identification can be achieved using the SEARCH command. SEARCH is like FIND, but is much more powerful in that it can carry out global searches rather than the local ones FIND does. Specified features can be selected for identification, and one of the criteria on which identification can be based is the feature code. You will get a list of feature codes for various types of feature by giving the command

SHOW FCS

Now issue the command

SELECT FCS 49,315

to select all scaled symbols. You can check these selections using the command

SHOW SELECTIONS

If you next give the command

SEARCH ALL

the program will start a spiral search from the current cursor position and will identify the nearest scaled symbol. Giving the command

SEARCH NEXT

will result in continuation of the spiral search and the next scaled symbol will be identified. The SEARCH NEXT command can be repeated until all scaled symbols on the map have been identified. To have all features selected again, issue

SELECT ALL

In many of the command sequences in this manual, the FIND command has been used to illustrate feature identification; using SEARCH would, of course, be equally good. It is also assumed that the appropriate selections have been made.

CHAPTER 5

EDITING LINES

5.1 *Making Changes to a Feature*

You can change a feature attribute, for example the feature code, by first of all issuing

FIND

on a line feature. The program will now be in LINE state, and you can change the feature code by giving the command

CHANGE FC n

where n is the new feature code. After changing the feature code, the program returns to READY state. Now find the feature again and use EXAMINE and DRAW to check the changes made.

It would become very tedious if you had to find the feature every time you wanted to make a change, and so the MODIFY command allows multiple changes. First you need to issue

FIND

and then give the command

MODIFY

Note that the program has entered ON state, and also that the cursor is constrained to move only on the feature. If, for example, you wanted to change the serial number and feature code, you would use the commands

CHANGE FSN n

CHANGE FC n

where n is the new serial number or feature code. The command

END

terminates the operation and takes the editor back to READY state.

5.2 *Moving a Feature*

In order to move a feature to a new position on the map, first give the command

FIND

followed by the commands

MODIFY

MOVE

The MOVE command attaches the feature to the cursor, although the feature will not be drawn in refresh on the screen. The program will now be in EDIT state, and you can move the feature around before issuing

END

to deposit it in its new position. Note that on some displays the feature will also still be shown in its old position, until the screen is redrawn using DRAW.

If the new feature is to have different attributes, you can use the CHANGE command as well as MOVE. If, however, the CHANGE command is not to be used, as in the sequence above, the MODIFY command can be omitted.

5.3 *Copying a Feature*

In order to make a copy of a line feature you must first issue

FIND

and then give the command

COPY

This creates a new copy of the feature in its original position. If you want to move the copy around, give the command

MOVE

followed by

END

when the feature is in the desired position. The copied feature will then be drawn in its new position. As with MODIFY, you can use the CHANGE command as well as or instead of the MOVE command. In fact, the MODIFY and COPY commands are very similar; the former has the effect of deleting the original feature, while the latter preserves it.

5.4 *Joining Features*

To join two line features together you must first position the cursor over the end of one of the lines and then issue

FIND

You then need to give the command

JOIN

which puts the program into EDIT state. Now move the cursor to the end of the other feature and issue

FIND

again. The cursor will now be attached to the ends of both features, and can be moved to the desired position. The operation is terminated by giving the command

END

By default, only line features with the same feature code can be joined together. However, this condition can be altered; if, for example, you want the feature serial numbers to be the same as well, you can use

MATCH FSN

before performing the join.

5.5 *Tying Features*

When edge matching two adjacent maps it is often necessary to tie two ends of the same feature together (not actually join them but line them up). This is done by positioning the cursor over one end and giving the command

FIND

followed by

TIE

Now position the cursor over the end of the other feature and issue

FIND

again. Both ends are now attached to the cursor which should be moved to the desired position on the boundary. The command

END

will record this point and terminate the operation.

5.6 *Closed Loops*

Many features need to have their start and end points joined up so as to form a closed loop. This is done by giving the command

FIND

followed by

LOOP or **LOOP EXTEND**

The former will move the first and last vertices to their mean position, and the latter will extend the first and last spans of the feature to a new meeting point. The command

END

should then be given to deposit the feature.

5.7 *Removing a Data Point*

To remove a single point from within a line feature (eg to smooth out a kink), first issue

FIND

The cursor will jump onto the feature, and you can use the commands

FIRST / LAST / PREVIOUS / NEXT / VERTEX n

where n is the vertex number, to put the cursor onto the vertex you want to delete. Now you can delete the point using

REMOVE

and you will find yourself back in READY state

5.8 *Inserting a Data Point*

To introduce a new point into a line feature you first use

FIND

to get the feature into hand. Now you can use the commands

FIRST / LAST / PREVIOUS / NEXT / VERTEX n

to position the cursor onto the vertex next to the new point. Giving the commands

MIDDLE / FRACTION r

where r is the fraction along the span you want the cursor to be, will position the cursor between vertices. Issuing

INSERT

creates a new vertex attached to the cursor, which can be moved to the desired position. The command

END

will terminate the operation and draw the feature with the new point.

5.9 *Editing a Data Point*

In order to reposition a particular point within a line feature you have to locate the cursor on the point using the command

FIND

followed optionally by

FIRST / LAST / PREVIOUS / NEXT / VERTEX n

The command

EDIT

attaches the current vertex to the cursor, which can be moved around until the command

END

is given.

5.10 *Deleting Part of a Feature*

In this and the remaining sections of this chapter, the words "position the cursor" are highlighted in a similar way to the commands, in order to clarify the sequences of operations.

In order to delete a section from within a line feature (resulting in two separate features) you must first issue

FIND

Now give the command

ON

which constrains the cursor to move only on the feature. Next

position the cursor

onto the point at one end of the part to be removed, and give the command

DELETE PART

Note that the program has entered ON state (which is not connected with the ON command). You should now

position the cursor

at the other end, and give the command

END

to delimit it and terminate the operation. The section indicated will be deleted and scrubbed out. Note that it can be recovered again as for a whole feature, and will be retrieved as a separate feature.

5.11 *Changing, or Copying and Changing Part of a Feature*

To change only part of a line feature, find the first end of the part to be changed using the sequence of operations

FIND

ON

position the cursor

Next issue

MODIFY PART or COPY PART

CHANGE FSN n

CHANGE FC n

to make the changes. Now

position the cursor

at the other end of the part to be changed and then issue

END

to delimit the part and terminate the operation. If, having issued MODIFY PART, the CHANGE commands were not given, the effect would simply be to divide the feature into three.

5.12 *Moving, or Copying and Moving Part of a Feature*

Part of a line feature can be modified or copied and then moved by first finding one end of the part to be copied using the sequence of operations

FIND

ON

position the cursor

Now issue

MODIFY PART or COPY PART

and note that the program is in ON state. To shift the new feature, first give the command

MOVE

You then have to

position the cursor

at the other end of the part being modified or copied, and define the end of the new feature by issuing the command

END

The program is now in EDIT state (as a result of the MOVE command), the new feature is attached to the cursor, and you can move it to the desired position. Giving the command

END

a second time will deposit the new feature and terminate the operation. As with whole features, you can alter the attributes of the new feature by using the CHANGE command along with the MOVE command. When using MODIFY PART and MOVE, the deleted part of the original feature will still be shown on some displays, until the screen is redrawn using DRAW.

5.13 *Extending a Feature*

You may wish to extend one end of a line feature in the direction defined by the last two points. To do this you have to issue

FIND

followed optionally by

FIRST / LAST / PREVIOUS / NEXT / VERTEX n

to position the cursor on the end point of the feature. Giving the command

EXTEND

will constrain the cursor to move only along the end segment of the feature and its extensions. The program will now be in EDIT state, and when the cursor is positioned, the command

END

will terminate the operation.

5.14 *Replacing Part of a Feature*

It is often necessary to add a new series of data points to a line and/or remove a series from a line, without splitting the line into two features. First issue

FIND

followed by

ON

and then

position the cursor

at one end of the section to be changed. Now give the command

BRIDGE

and note that the program moves into CONSTRUCT state. The command

START

should then be given for each new point to be inserted, and the new part of the feature (except for the first segment) will be drawn in refresh. If no new points are required, the START command need not be used. To locate the last point of the section to be changed, use

FIND

and then

position the cursor

The program remains in CONSTRUCT state until the command

END

which terminates the operation, is issued.

CHAPTER 6

EDITING TEXTS AND SYMBOLS

6.1 *Making Changes to a Text or Symbol*

Attributes of texts and symbols can be changed in a similar way to lines. You first have to identify a text or symbol using

FIND

Now try

CHANGE FSN n

to change the feature serial number. Note that unlike lines, the program automatically enters MODIFY state after you have issued the command; this happens with all of the text and symbol commands. Now, depending on whether your found-feature is a text or symbol, give either of the commands

CHANGE FC 399 or **CHANGE FC 49**

to change the feature code, and note that the editor remains in MODIFY state. This is the case for all of the operations on texts and symbols, and so several operations can be performed on these features without leaving MODIFY state. Editing of the feature is completed by issuing the command

END

This deposits the changed feature and returns the program to READY state.

For clarity, the FIND command and the last END command will not be illustrated in the command sequences for the remaining sections of this chapter. It is assumed that you will have an appropriate text or symbol feature in hand and be ready to try out each command in turn.

Note that there are three types of symbols, i.e. unoriented, oriented and scaled.

6.2 *Moving a Text or Symbol*

Assuming you are in TEXT, SYMBOL or MODIFY state, issue the command

MOVE

This attaches a refresh copy of the feature to the cursor, and you can move it to its new position before depositing it by issuing

END

Unlike lines, this command does not terminate the operation, and you remain in MODIFY state to allow for further editing of the feature. Note also that the refresh copy of the feature is no longer attached to the cursor.

6.3 Copying a Text or Symbol

Copying a text or symbol is similar to moving one, but you first have to issue

COPY

followed by

MOVE

Now you have a new copy of the feature attached to the cursor, and you can move the copied feature to its new position and issue

END

to deposit it. Again, you remain in MODIFY state.

6.4 Changing the Size of a Text or Scaled Symbol

A text or scaled symbol can be made larger using the command

LARGER

and smaller using the command

SMALLER

For texts with height in point-size units these commands use the next available point-size. Otherwise, the text or symbol is increased or decreased in size by 10%.

6.5 Rotating a Text, Oriented Symbol or Scaled Symbol

In order to rotate a text, oriented symbol or scaled symbol you can use the command

ROTATE CURSOR

Movement of the cursor will cause the text or symbol to rotate in that direction. When the feature is suitably oriented, the cursor can be disconnected by issuing the command

END

Note that you remain in MODIFY state.

6.6 *Editing a Text String*

Existing text can be completely replaced using the REPLACE command. For example, if the current feature in hand is a text and you want the text to be 'Trinity Street', issue the command

REPLACE Trinity Street

If you only want to replace part of a text string, you can use the SUBSTITUTE command. For instance, to substitute 'Sidney' for 'Trinity' in the above example, you would type

SUBSTITUTE Trinity Sidney

The text will now be 'Sidney Street'. If the text string contains spaces, it must be enclosed in quotes. In the above example, the command

SUBSTITUTE "Sidney" "St. Andrews"

results in the text 'St. Andrews Street'

6.7 *Changing the Size and Orientation of a Scaled Symbol*

You can use the cursor to define the size and orientation of a scaled symbol. Give the command

ALIGN

Now move the cursor to align the symbol; the distance between the symbol and the cursor is proportional to the symbol size, and the symbol is aligned along an imaginary line joining the symbol and cursor. When the symbol is suitably aligned, the cursor can be disconnected with the command

END

If you simply want to change the size of a scaled symbol, use the command

STRETCH

The cursor will be constrained to lie on the symbol orientation line, and you can change the symbol size by moving the cursor. When the size is satisfactory, disconnect the cursor with the command

END

CHAPTER 7

CREATING NEW FEATURES

7.1 *Creating a Line Feature*

Before creating a new line feature, you need to set up the feature attributes. There are sixteen feature attribute sets to enable construction of various line types and symbols, and the attributes used for feature construction are contained in the current attribute set. In order to copy a given attribute set, e.g. 1, into the current attribute set, you should give the command

GET 1

The attributes can now be modified, and to alter the feature code, for example to 30, you need to issue

SET FC 30

If the terminal status line is enabled, you will note that it displays the construction feature code. The actual construction can now be started. Position the cursor over the first point, and give the command

START

This puts the editor into CONSTRUCT state, and you should repeat the process for all the other points until the last point, where the command

END

should be given. This terminates the construction. If you now want to save the current attribute set, you can copy it into a specified attribute set, e.g. 2, by issuing

PUT 2

Note that the current attribute set remains unchanged, and so if the next new line feature is to have the same attributes, the GET and SET commands need not be used before starting the new construction.

7.2 *Closed Loops*

If the feature under construction is to be a closed loop, you would construct it as above, but just before giving the END command, issue

CLOSE or CLOSE SQUARE

The former will move the cursor onto the first vertex, so that the first and last vertices are in the same position. The latter, however, will extend the line joining the last digitised point and the current cursor position to a new point, such that the line joining the new point to the first vertex is a right angle. The cursor will finish up on the first vertex.

Note that the CLOSE command is analogous to the LOOP command used in line editing.

7.3 Creating a Text Feature

Some of the attributes used in feature construction are global; they do not belong to any particular attribute set, and can be set directly. One of these is the feature code used for new text features. To set this, for example to 28, give the command

SET TEXT 28

You can now construct a new text, for example 'Bridge Street', by giving the command

TEXT Bridge Street

The program will now be in MODIFY state, and you can use any of the text editing commands given in Chapter 6. Try

MOVE

and move the new text to the desired position. Giving the command

END

will deposit the feature, and if the text is the correct size and orientation, issue

END

again to terminate the operation.

7.4 Creating a Symbol Feature

Creating a symbol feature is a combination of the operations required for lines and texts. If you want to make one of the feature attribute sets, e.g. 3, current, issue

GET 3

Now (if it is not already set) set the feature code to a symbol, for example 11, by giving the command

SET FC 11

The symbol is created by using the command

START

which takes you into MODIFY state. Like texts, you can now use any of the symbol editing commands given in Chapter 6. To move your new symbol, issue

MOVE

and move it to the desired position. Giving

END

will deposit the symbol, and

END

again will terminate the operation. As with lines, you can copy the current attribute set into another, e.g. 4, by issuing

PUT 4

CHAPTER 8

FINISHING A SESSION

8.1 *Normal Completion*

The normal method of finishing a session is to use the command

EXIT

which creates new versions of the input IFF files and deletes the workspace files. If you have been editing a single IFF file and you want the new version to have a different name, for example MYMAP.IFF, you would issue

EXIT MYMAP.IFF

Note that the EXIT command has to be typed in full.

8.2 *Ignoring all Editing*

All editing will be ignored if you use the command

QUIT

which again should be typed in full. This does not create any new files and leaves the old file unchanged.

INDEX

- ABANDON command, 4-1
- Abandoning an operation, 4-1
- ALIGN command, 6-3
- Attributes of features, 4-1
- Audience, 1-1
- BRIDGE command, 5-8
- Button commands, 1-3
- CHANGE command, 5-1 to 5-2, 5-6, 6-1
- Changing
 - line features
 - part, 5-6
 - whole, 5-1
 - symbol features, 6-1
 - text features, 6-1
- CIRCLE state, 4-1
- CLOSE command, 7-2
- Closed loops, 5-4, 7-1
- Command files, 1-4
- Command states, 1-2
- Commands, 1-3
 - abbreviation, 1-3
 - ambiguity, 1-3
 - arguments, 1-3
 - conditional execution, 1-4
 - information, 3-1
 - jumps, 1-4
 - terminating, 1-4
 - validity, 1-2
- CONSTRUCT state, 5-8, 7-1
- Constructing
 - line features, 7-1
 - symbol features, 7-2
 - text features, 7-2
- COPY command, 5-2, 5-6 to 5-7, 6-2
- Copying
 - line features
 - part, 5-6 to 5-7
 - whole, 5-2
 - symbol features, 6-2
 - text features, 6-2
- Creating
 - line features, 7-1
 - symbol features, 7-2
 - text features, 7-2
- Cursor movement, 3-1
- Data, 1-2
- Data Points
 - inserting, 5-4
- Data points
 - editing, 5-5
 - removing, 5-4
- DEC, 1-1
- DELETE command, 4-2, 5-6
- Deleting
 - circle features, 4-2
 - line features
 - part, 5-5
 - whole, 4-2
 - symbol features, 4-2
 - text features, 4-2
- Description of program, 1-1
- Digital map data, 1-2
- DISABLE command, 2-2
- DRAW command, 3-2
- Edge matching, 5-3
- EDIT command, 5-5
- EDIT state, 5-2 to 5-3, 5-7 to 5-8
- Editing a data point, 5-5
- Editing text strings, 6-3
- ENABLE command, 2-2
- END command, 4-2, 6-2 to 6-3
- Enlarging
 - symbol features
 - scaled, 6-2 to 6-3
 - text features, 6-2
- Environment, 1-2
- Examining a feature, 4-1
- EXIT command, 8-1
- Exiting a session, 8-1
- EXTEND command, 5-8
- Extending line features, 5-7
- Feature attributes, 4-1
- Feature selection, 1-2
- Feature-in-hand, 1-2
- Features, 1-2
- @file command, 1-4, 2-2
- FIND command, 4-1
- Finding a feature, 4-1 to 4-2
- FIRST command, 5-4
- Found-feature, 1-2
- FRACTION command, 5-5
- FRT Guide, 1-2
- FRTLIB Reference Manual, 1-2

- GET command, 7-1
- GKS, 1-2
- HELP command, 3-1
- Help facility, 3-1
- Identifying a feature, 4-1 to 4-2
- IFF command, 2-2
- IFF files, 1-2
- IGES, 1-2
- Initialisation, 2-1
- Input data, 1-2
- INSERT command, 5-5
- Inserting a data point, 5-4
- JOIN command, 5-3
- Joining line features, 5-3
- Keyboard commands, 1-3
- LARGER command, 6-2
- LAST command, 5-4
- Layers, 1-2
- Learner messages, 2-2
- LINE state, 4-1, 5-1
- LITES, 1-2
- LITES1, 1-2
- LITES1 workstation, 2-1
- LITES2, 1-1
- LITES2 Reference Manual, 1-1
- LOOP command, 5-4
- Macro commands, 1-4
- MADES, 1-2, 2-1
- Map data, 1-2
- MAPS command, 2-2
- MATCH command, 5-3
- MENU command, 2-2
- Menu commands, 1-3
- MIDDLE command, 5-5
- MODIFY command, 5-1 to 5-2, 5-6 to 5-7
- MODIFY state, 6-1, 7-2 to 7-3
- MOVE command, 5-2, 5-7, 6-2
- Moving
 - line features
 - part, 5-7
 - whole, 5-2
 - symbol features, 6-1
 - text features, 6-1
- Moving the cursor, 3-1
- NEXT command, 5-4
- NULL command, 3-1
- ON command, 5-5
- ON state, 5-1, 5-6 to 5-7
- Orienting
 - symbol features
 - scaled, 6-3
- Password, 2-1
- POSITION command, 3-1
- PREVIOUS command, 5-4
- Primary commands, 1-3
- Program description, 1-1
- PUCK command, 2-3
- PUT command, 7-1
- QUIT command, 8-1
- Quitting a session, 8-1
- READY state, 4-1, 5-1
- RECOVER command, 4-2
- RECOVER state, 4-2
- Recovering
 - circle features, 4-2
 - line features
 - part, 5-6
 - whole, 4-2
 - symbol features, 4-2
 - text features, 4-2
- Redrawing, 3-2
- Reducing
 - symbol features
 - scaled, 6-2 to 6-3
 - text features, 6-2
- REMOVE command, 5-4
- Removing a data point, 5-4
- REPLACE command, 6-3
- Replacing part of a line feature, 5-8
- ROTATE command, 6-2
- Rotating
 - symbol features
 - oriented, 6-2
 - scaled, 6-2
 - text features, 6-2
- Running the program, 2-1
- SEARCH command, 4-2 to 4-3
- Searching for features, 4-2
- Secondary commands, 1-3
- SELECT command, 4-3
- Selection of features, 1-2
- SET command, 7-1 to 7-3
- Set up, 2-2
 - abandoning, 2-3
 - digitising, 2-3
 - mistakes, 2-3 to 2-4

puck buttons, 2-3
 repeating, 2-3
SETUP command, 2-3 to 2-4
SHOW command, 3-1, 4-2
SMALLER command, 6-2
SOLADI, 1-2
START command, 5-8, 7-1, 7-3
States, 1-2
 entering, 1-2
STRETCH command, 6-3
SUBSTITUTE command, 6-3
SYMBOL state, 4-1

TEXT command, 7-2
TEXT state, 4-1

TIE command, 5-3
TRACK command, 2-3
Tying line features, 5-3

Use of this manual, 1-1
Username, 2-1

Variables, 1-4
VAX, 1-2
VERTEX command, 5-4
VMS, 1-2

WINDOW command, 3-2
Windowing, 3-2
WOSP, 2-1