

Laser-Scan Ltd.

IFF

User Guide

Copyright (C) 2000 Laser-Scan Ltd
Science Park, Milton Road, Cambridge, England CB4 4FY tel: (01223) 420414

Document	"IFF User Guide"	Category	"User"
Document Issue	1.0 Tim Hartnall		05-Jan-1987
Document Issue	1.1 Tim Hartnall		23-Jan-1987
Document Issue	2.0 Clarke Brunt		14-Jan-1988
Document Issue	2.1 Ron Russell		17-Jun-1988
Document Issue	2.2 Ron Russell		26-Jul-1990
Document Issue	2.3 Clarke Brunt		11-Oct-1993

CONTENTS

1	IFF User Guide documentation change record	i
2	PREFACE	1
2.1	Intended audience	1
2.2	Structure of this document	1
2.3	Associated documents	2
2.4	Conventions used in this document	2
CHAPTER 1	INTRODUCTION	
1.0.1	General	1-1
1.0.2	IFF files within the VAX/VMS environment	1-1
1.0.3	Introduction of new IFF entries	1-1
1.0.4	Introduction of the CB entry	1-3
1.0.5	The IFF origin offset	1-4
1.0.6	IFF and three dimensional strings.	1-4
CHAPTER 2	IFF STRUCTURE	
2.0.1	Overview	2-1
2.0.2	Annotated File Listing	2-4
CHAPTER 3	FILE LEVEL ENTRIES	
3.0.1	General	3-1
3.0.2	File level entry order	3-1
3.0.3	EJ End of Job (data) marker	3-2
3.0.4	HI HHistory	3-3
3.0.5	RA RRange	3-5
CHAPTER 4	MAP LEVEL ENTRIES	
4.1	General	4-1
4.1.1	Map level entry order	4-1
4.1.2	EM End of Map marker.	4-2
4.1.3	MD Map Descriptor	4-3
4.1.4	MH Map Header	4-6
CHAPTER 5	SECTION LEVEL ENTRIES	
5.0.1	General	5-1
5.0.2	Sections - a warning	5-1
5.0.3	Section level entry order	5-2
5.0.4	CC Cubic Coefficients	5-3
5.0.5	CP Control Points	5-5
5.0.6	NS New Section	5-7

CHAPTER 6 LAYER LEVEL ENTRIES

6.0.1	General	6-1
6.0.2	Layer level entry order	6-2
6.0.3	EO End of layer marker	6-3
6.0.4	NO New layer (overlay) entry	6-4

CHAPTER 7 FEATURE LEVEL ENTRIES

7.0.1	General	7-1
7.0.2	Definition of an IFF feature	7-1
7.0.3	A note on IFF features and transmitted comments	7-3
7.0.4	AC Ancillary Code	7-5
7.0.5	CB Coordinate Block	7-8
7.0.6	EF End of Feature	7-10
7.0.7	FS Feature Status	7-11
7.0.8	NF Start of New Feature	7-14
7.0.9	RO ROTation entry	7-16
7.0.10	ST 2 dimensional point string entry	7-17
7.0.11	TH Text height / line THickness entry	7-18
7.0.12	TS Text Status entry	7-19
7.0.13	TX TeXt entry	7-21
7.0.14	ZS 3 dimensional point string entry	7-22

CHAPTER 8 MISCELLANEOUS AND OBSOLETE ENTRIES

8.0.1	General	8-1
8.0.2	CH Literal CHaracter entry	8-2
8.0.3	CS Character Size	8-3
8.0.4	SL Select Symbol Library entry	8-4
8.0.5	SS Symbol Select entry	8-5
8.0.6	TC Transmitted Comment	8-6
8.0.7	VO VOID	8-7
8.0.8	XX Unknown entry type	8-8

CHAPTER 9 IFF JUNCTION STRUCTURE

9.0.1	General	9-1
9.0.2	JB Junction Block	9-3
9.0.3	JP Junction Pointer	9-5
9.0.4	SH Sector Header	9-6

APPENDIX A THE SI AND SD COMMANDS

A.0.1	INTRODUCTION	A-1
A.0.2	Basic Use	A-1
A.0.3	Advanced Use	A-1

IFF User Guide documentation change record

Version 1.0 Tim Hartnall 05-January-1987

First public issue of IFF User Guide.

Version 1.1 Tim Hartnall 23-January-1987

Minor typing and presentation errors corrected. Interpretation of bit pattern within the status word of an FS entry now reflects that used by the converged Laseraid.

Version 2.0 Clarke Brunt 14-January-1988

Minor typing and presentation errors corrected. CB entry added. Description of IFF revision levels, and changes to AC entry to reflect new ACD mechanism.

Version 2.1 Ron Russell

17-June-1988

Include new default AC and ACD type 82 (Polygon_info)

Version 2.2 Ron Russell

26-July-1990

Correct typing error in above change

Version 2.3 Clarke Brunt

11-October-1993

Change chapter titles to be more meaningful, and get rid of the spurious Appendix A. Correct spelling in a number of places.

PREFACE

Intended audience

This manual is intended for all users of the Laser-Scan Automated Map Production System (LAMPS).

Structure of this document

This document is composed of 2 major sections.

The Introduction is an overview of IFF and is intended as a quick reference guide to the salient features of the Laser-Scan IFF (Internal Feature Format) file format.

There then follow detailed descriptions of the individual entries which comprise an IFF file. These descriptions are organised using the hierarchical structure of IFF format:

- o Entries at the file level
- o Entries at the map level
- o Entries at the section level
- o Entries at the layer level
- o Entries at the feature level
- o Obsolete entries
- o Entries for IFF junction structure.

Within these categories each individual IFF entry is described using the same basic framework. These are:

ENTRY MNEMONIC	- the name of the IFF entry and its definition
FORMAT	- the structure of the data which it contains.
DESCRIPTION	- a synopsis of what the entry is used for.
NOTES	- this includes any special usage, incompatibilities and restrictions.

Associated documents

For detailed information about how to create a specific IFF entry see the IFFLIB Reference Manual.

Conventions used in this document

Convention	Meaning
<CR>	The user should press the carriage return key on the terminal
<CTRL/x>	The phrase <CTRL/x> indicates that the user must press the key labelled CTRL while simultaneously pressing another key, for example, <CTRL/Z>.
\$ IINFO JIM<CR>	Command examples show all user entered commands in bold type.
\$ IFIXAREA JIM<CR> . . .	Vertical series of periods, or ellipsis, mean either that not all the data that a utility would display in response to the particular command is shown or that not all the data that the user would enter is shown.
file-spec...	Horizontal ellipsis indicates that additional parameters, values or information can be entered.
[logical-name]	Square brackets indicate that the enclosed item is optional. (Square brackets are not, however, optional in the syntax of a directory name in a file-specification, or in the syntax of a substring specification in a VMS assignment statement).
'byte'	A single byte (8 bits) is contained in the specified IFF entry field.
'byte-string'	Text held as a string of bytes (8 bits) is contained in the specified IFF entry field.
'integer'	An integer*4 (32 bit) number is contained in the specified IFF entry field.

Convention	Meaning
'real'	A real*4 (32 bit) floating-point number is contained in the specified IFF entry field.
'real*8'	A real*8 (64 bit) floating-point number is contained in the specified IFF entry field. The term "double precision" may be more familiar.
'word'	An integer*2 (16 bit) number is contained in the specified IFF entry field.
FSN 'integer' ('integer')	FSN followed by two integer arguments indicates an IFF feature serial number. The integer number enclosed in round brackets is the feature internal sequence number.
00003DE7	A hexadecimal address of a location within an IFF file. IMP modules express all IFF addresses using hexadecimal radix. The address is always padded with leading zeros to a standard field width of 8 characters.

CHAPTER 1

INTRODUCTION

INTRODUCTION

General

The acronym IFF stands for **I**nternal **F**eature **F**ormat. This is the Laser-Scan vector file format generated by LASERAID and other Laser-Scan mapping systems and used as the data structure throughout the Laser-Scan LAMPS system.

IFF files are binary and cannot be manipulated directly using a text editor. The IMP (**I**FF **M**ap **P**rocessing) component of the LAMPS system enables the user to manipulate the IFF file structure and contents and in addition enables the production of textual representations of an IFF file. This guide is designed to explain the component entries of an IFF file that the user is exposed to within the LAMPS environment.

Within an IFF file the component entries are built into a flexible hierarchical structure. A full understanding of this structure is essential for efficient cartographic production flowline planning and management.

This guide should enable a customer to interpret a textual representation of an IFF file as produced by the IMP ITOTEXT utility.

The software engineer designing IFF applications programs will need to refer to the IFFLIB Reference Manual which describes in detail the bits and bytes of IFF format and the routine calls to the IFF Library needed to generate and manipulate an IFF file.

IFF files within the VAX/VMS environment

Within the VAX/VMS system IFF files can be treated as any other file type for file management purposes. To enable the user to distinguish an IFF file from a file of another type IFF files have by default the file extension '.IFF'. To provide great flexibility in the LAMPS production environment IFF files are referenced by all the IMP modules using logical name LSL\$IF. (For an explanation of logical names see the VAX/VMS document set). Logical name LSL\$IF is assigned to a device and directory specification either using the VMS DEFINE command or the Laser-Scan SI utility, (see Appendix A).

Introduction of new IFF entries

In January 1987 Laser-Scan released the IMP (**I**FF **M**ap **P**rocessing) package. From that time IFF files are referred to in documentation as "old type" (or "historical") and "new type" (or "modern"). The distinguishing features of a "new type" IFF file are the introduction of four powerful new entries and a definitive statement on the preferred structure of an IFF file.

Downwards compatibility with "historical" IFF files is maintained within IMP. Many Laser-Scan customers have large archives of IFF files which do not contain the new IFF entries. IMP modules are

designed to treat IFF files of an historic nature in the same manner as the now obsolete DAMP utilities would. No new type IFF entries are placed into an IFF file if the IFF file specified as input to an IMP utility does not already have those entries. Compatibility between IMP and existing databases is thus achieved. "New type" IFF files which contain HI, type 2 MD, TS, and ZS entries may be generated using ISTART (or IFROMTEXT in conjunction with a suitable textual representation of an IFF file).

A problem arises for those IMP modules which take input from more than one IFF file when the input files are a mixture of "old type" and "new type" IFF files. Under these circumstances the input file which is specified first is considered to set the standard for the output file.

If the user deliberately introduces any of the new IFF entries into an "historical" IFF file using the IMP utilities downwards compatibility may be lost. Users worried about compatibility between the new standard and existing databases of IFF files should refer to Laser-Scan.

The four new IFF entry types available for use with LAMPS systems containing the IMP package are:

- (i) HI (HIstory) entry. This is a mechanism for automatically recording statistics in an IFF file each time it is updated, so that it may be determined which users and programs contributed to the final state of the file.
- (ii) MD (type 2 Map Descriptor) entry, upwards compatible with the existing type 1 MD. This entry contains projection, scale and spheroid information etc. It also contains an origin offset. This is an (X,Y) coordinate pair, each held as a REAL*8 (double precision) number, that is added to any pair of coordinates in the IFF file to give the true projection coordinates of the points. This allows the absolute size of the coordinates in the IFF file, which are only held as REAL*4 (single precision) numbers, to be reduced. This avoids problems of truncation and reduced accuracy.
- (iii) TS (Text Status) entry. Traditionally, IFF text features could contain only one text string, with associated location and descriptive data. The new TS entry allows IFF text features to be **composite** - that is composed of several sub-texts or text components, which may be manipulated independently or as a single entity. Each text component starts with a TS entry, and ends with the next TS entry, or the final EF of the feature. The first TS entry occurs immediately after the FS entry and any AC entries. Text components may not include FS or AC entries, but may contain any other entries that are legal within a normal text feature.

NOTE

Support for composite texts is a licensed option in the LITES2 cartographic editor, and is only supported by certain Laser-Scan software products. See product SPS for details.

- (iv) ZS (3 dimensional string) entry. The ZS 3 dimensional strings are held as x,y,z coordinates in a similar fashion to the existing IFF 2 dimensional ST (STring) entries. The X and Y coordinate values of a ZS entry are treated using the same rules as for ST entries. Therefore a ZS entry will be included in the XY range calculation for the IFF file.

NOTE

Support for 3D strings is only supported by certain Laser-Scan software products. See product SPS for details.

Introduction of the CB entry

In January 1988 the Coordinate Block (CB) entry was introduced. The CB entry provides the same functionality as ST and ZS entries, which it is intended to replace, but in addition allows the description of other per-point attributes in addition to the X, Y, and Z coordinates. A given IFF file may only contain either ST and ZS entries, or CB entries. The former case is IFF output revision level 0, while the latter is output revision level 1 - the output revision level of a given file is displayed by the IMP utility IPATCH in its status area. The output revision level is decided at the time the file is created, and may be controlled either by an application program (possibly using a command switch) or by assigning the desired output revision level "0" or "1" to the logical name LSL\$IFF_OUTPUT_REVISION. In the absence of either mechanism, a revision level 0 file will be created.

The IFF library provides compatibility for programs written before the introduction of CB entries. A program may set the 'IFF input revision level' to -1, 0, or 1, possibly controlled by a command switch as in the case of IMP modules IPATCH and ITOTEXT. Input revision level -1 allows all ST, ZS, and CB entries to be displayed exactly as they are in the file. At level 0, any CB entries will be presented to a program as ST or ZS entries, while at level 1, any ST or ZS entries will appear as CB entries. If an application program does not explicitly set the input revision level, 0 will be used, so old programs will still work with files containing CB entries provided that they are linked with the new IFF library.

Similarly for programs which create new entries, possibly by copying from one file to another, ST, ZS, or CB entries will be created as appropriate for the output revision level of the output file, independently of which type of entry the program actually tries to create. An IFF error message will be displayed when an output file is closed in the case when attributes other than X, Y, or Z have been lost due to CB entries being turned into ST or ZS entries in an output revision level 0 file.

The IFF origin offset

The origin offset is held in a type 2 MD (Map Descriptor) entry. The origin offset is an (X,Y) coordinate pair, each held as a REAL*8 (double precision) number, that is added to any pair of coordinates in the IFF file to give the true projection coordinates of the points. This allows the absolute size of the coordinates in the IFF file, which are only held as REAL*4 (single precision) numbers, to be reduced, thus avoiding problems of truncation and impaired accuracy. The origin offset entry may be set up using IFROMTEXT, ISTART or ITRANS/DESCRIPTOR. Generally the use of an origin offset results in the south west (bottom left) control point of the map being reduced to the coordinate (0.0,0.0). IMP modules are designed to utilise the origin offset only when reference to adjoining map sheets (as IFF files) or IFF file projection transformation is required. The user may manipulate the within-file data relative to the local (0.0,0.0) origin without reference to the origin offset. It is most important that the origin offset is correctly set. The IMERGE (file merging) and ITRANS (file transformation) utilities rely heavily upon the origin offset. Errors in its application may result in irrevocably scrambled data.

IFF and three dimensional strings.

Three dimensional string storage is now available within IFF files using the ZS or CB entry. All IMP modules handle 3 dimensional strings. Where appropriate the three dimensional strings are processed in the same manner as 2 dimensional strings.

CHAPTER 2

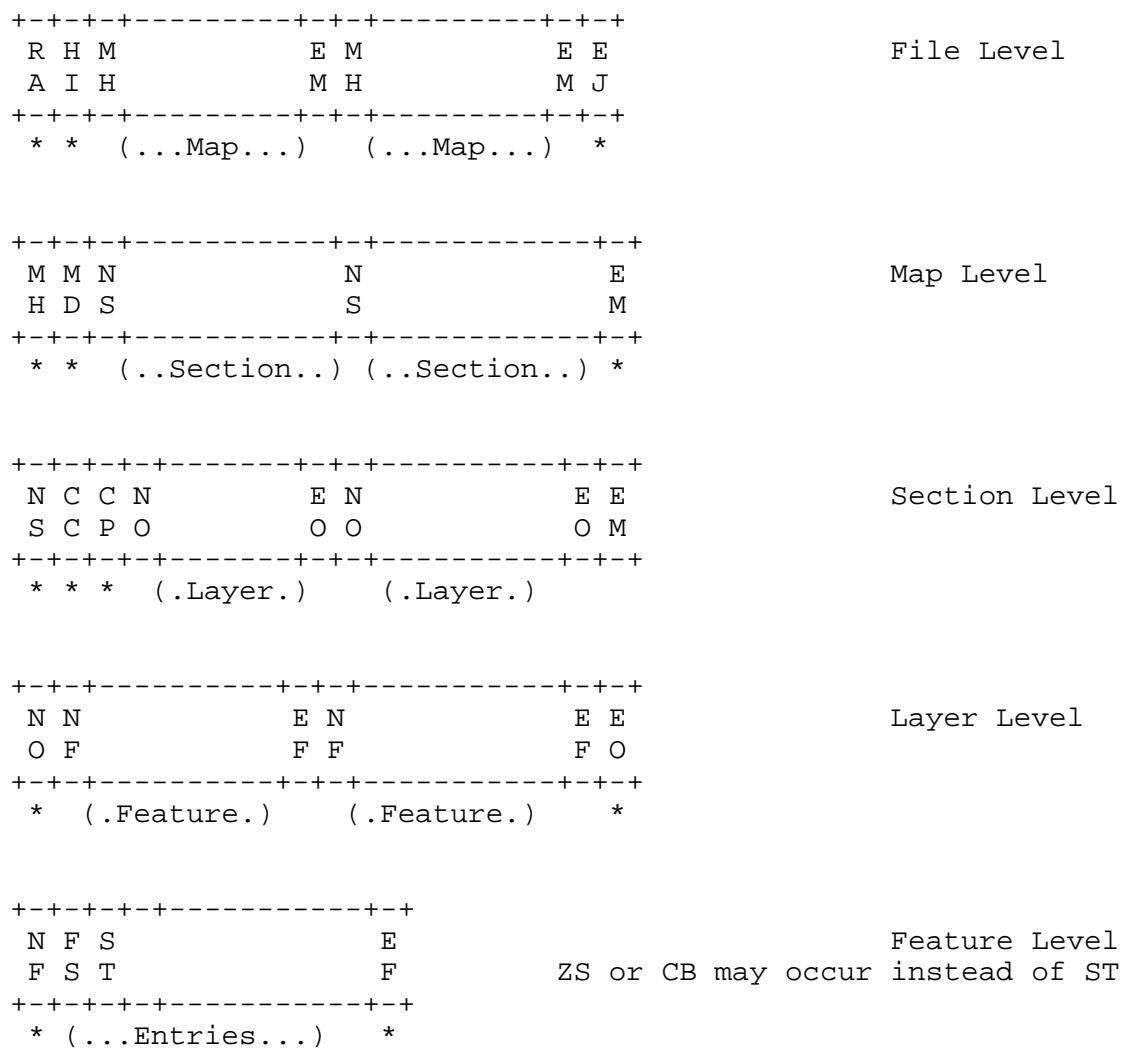
IFF STRUCTURE

IFF STRUCTURE

Overview

The IFF file structure is shown schematically below, broken down into Maps, Sections, Layers, Features, and Entries.

Asterisks are used to indicate entries at a particular level. Note that the IFF file structure can handle multiple maps per file, but this mechanism is no longer supported by LAMPS software.



The order in which IFF entries occur within each level is given below. The following display conventions are used. An entry printed in **bold** type is obligatory. Vertical ellipsis indicate that multiple occurrences of the preceding entry are permissible.

Entries at File Level

RA - coordinate RAnge information.
HI - HHistory of IFF file.
SH - Sector Header.
EJ - End Job marker (end of file).

Entries at Map Level

MH - Map Header (map-specific information).
MD - Map Descriptor (map projection information).
EM - End Map marker.

Entries at Section Level

NS - New Section identification.
CC - Cubic Coefficients for coordinate transformation.
CP - Corner Points for coordinate transformation.

Entries at Layer Level

NO - New Overlay (layer) number and status.
EO - End Overlay (layer) marker.

There is no specific order for the following entries, which historically occur within layers but outside features. They are regarded as obsolete.

TC - Transmitted Comment.
.
.
.
CH - Character data.
.
.
.
CS - Character Size and spacing.
.
.
.
SS - Symbol Select.
.
.
.
SL - Symbol Library select.

Entries at Feature Level

The precise content of features varies with the type of feature. IFF features can contain 2D or 3D strings and must contain at least one ST, ZS, or CB entry. Features may not contain a mixture of these entries.

Junction entries within IFF features are explained in the section devoted to the description junction structure.

All IFF features are built using a basic "building block".

NF	- start of new feature
FS	- feature status
ST/ZS/CB	- string of coordinates
EF	- end of feature

For a detailed description of the structure of IFF features see the section describing entries at the IFF feature level.

Other Entries The following can occur anywhere in a file :-

VO	- VOld
JB	- Junction Block

Annotated File Listing

The following is an annotated listing of a hypothetical example Ordnance Survey (UK) IFF file with extensions showing layout and order of entries.

```

RA  0.00    500.00    0.00    500.00    (coordinate RAnge)
HI - History                               (statistics for IFF file)
SH  0.0 0.0 500.0 500.0 100 100          (sector header)
MH - Map Header                           (non-graphic map info)
      1001      201000      101000      1250      0
      0      4097      0      1310722      ...
MD - Map descriptor                       (holds projection information)
NS IFF file layout demo sheet             (New Section identification)
CC - bicubic transform for non-linear corrections
      .000000000E 000      .000000000E 000
      .100000000E 001      .000000000E 000
      .000000000E 000      .100000000E 001
      .000000000E 000      .000000000E 000
      .000000000E 000      .000000000E 000
      .000000000E 000      .000000000E 000
      .000000000E 000      .000000000E 000
      .000000000E 000      .000000000E 000
      .000000000E 000      .000000000E 000
      .000000000E 000      .000000000E 000
      .000000000E 000      .000000000E 000
CP - 4 point transform                    (scale, translation, and rotation)
      0.0000 500.0000 0.0000 500.0000
      0.0000 0.0000 0.0000 0.0000
      500.0000 0.0000 500.0000 0.0000
      500.0000 500.0000 500.0000 500.0000
NO 0 0                                    (start layer 0 - grid)
NF 9980                                  (start New Feature)
FS 398                                  (Feature code is linear)
TH 0                                    (size)
ST 2 0                                  (String of coords)
      0.0000 0.0000
      500.0000 0.0000
EF                                      (End Feature)
..                                    (rest of grid)
..
EO                                      (End layer)
NO 1 0                                  (start New layer)
NF 1 1                                  (start New Feature)
FS 11 0 0 0                             (Feature code - 11 is linear)
AC 3 100.5                              (Ancillary Coding)
AC 4 34 Cambridgeshire                  (Ancillary Coding)
AC 5 34 Bedfordshire                   (Ancillary Coding)
TH 0                                    (line THickness unset)
ST 4 0                                  (STring of coords)
      137.2988 144.9971
      137.1202 156.9030
      150.4982 156.8733
      150.8999 146.3822
EF                                      (End Feature)
NF 2 2                                  (start New Feature)
FS 25                                  (Feature code -

```

TH 20	25 is an unoriented symbol)
ST 1 0	(size of symbol)
147.3486 257.3202	(locating point)
EF	(End Feature)
NF 3	(start New Feature)
FS 69 0 0 0	(Feature code -
	69 is an oriented symbol)
TH 40	(size of symbol)
ST 1 0	(locating point)
169.6900 252.4772	
RO 0.835	(ROtation angle)
EF	(End Feature)
NF 4	(start New Feature)
FS 49	(Feature code -
	49 is a scaled symbol)
TH 0	(size unset)
ST 2 0	(two points give angle and size)
149.4567 346.4330	
156.7132 355.5345	
EF	(End Feature)
NF 5	(start New Feature)
FS 28	(Feature code - 28 is a text)
TH 12	(point size)
ST 1 0	(locating point)
117.0385 144.7751	
RO 0.869	(ROtation angle)
TX Garden House	(the text string itself)
EF	(End Feature)
NF 6 6	(start New Feature)
FS 11 0 0 0	(Feature code - 11 is linear)
AC 4 0 Main drain	(Ancillary Coding)
TH 0	(line THickness unset)
ZS 4 0	(string of 3D coords)
137.2988 144.9971 12.78	
137.1202 156.9030 12.79	
150.4982 156.8733 12.93	
150.8999 146.3822 13.01	
EF	(End Feature)
NF 1 1	(start New Feature)
FS 28 0 0 0	(Feature code - 28 is text)
TS 14 0 0 0	(text status - text component code is 14
	Select font using this.)
TH 40	(Text Height)
ST 1 0	(STring of coords)
137.2988 144.9971	
RO 0.869	(ROtation angle)
TX Culvert, wood	(the 1st text string)
TS 14 0 0 0	(text status - text component code is 14
	Select font using this.)
TH 30	(Text Height)
ST 1 0	(STring of coords)
139.338 161.378	
RO 0.59	(ROtation angle)
TX Culvert, concrete	(the 2nd text string)
EF	(End Feature)
EO	(End layer 1)

EM	(End Map)
EJ	(End Job)

CHAPTER 3

FILE LEVEL ENTRIES

FILE LEVEL ENTRIES

General

The file level represents the highest division of data within the hierarchic IFF structure. All data within an IFF file share a common range entry which describes the coordinate minima and maxima in X and Y. The data in the file are also considered to share a common production history. The production history is detailed in the files HI (HIstory) entry. It is important for the user to recognise the significance of concatenating IFF files. The previous history of an IFF file is lost when it is merged with others and the concatenated file is considered to have been "created" and to have no previous history.

File level entry order

The order in which IFF entries occur at file level is given below. An entry printed in **bold** type is obligatory.

- RA - coordinate RAnge information**
- HI - HIstory of IFF file
- SH - Sector Header (see section describing IFF
Junction Structure)
- EJ - End Job (data) marker**

EJ End of Job (data) marker

FORMAT

EJ

The EJ entry does not have any contents.

DESCRIPTION

The EJ entry should always be the last entry in the IFF file. It signifies the end of data, and is used to provide a tidy end point (rather than relying on detecting the end of the file when a program cannot read any more data from the file).

NOTES

Problems may arise if the EJ entry is missing from an IFF file. Many IFF manipulation programs will terminate execution if the end of file is detected before an EJ entry is encountered.

The problem of a missing EJ entry may be overcome by using the IMP utility IMEND. Unlike other IFF manipulation programs, it is quite normal for IMEND to produce error messages when it reaches the end of sound data and attempts to read corrupted data. IMEND will truncate the IFF file after the last complete feature and will terminate the file with an EO, EM, EJ sequence of IFF entries.

HI HIsTory

FORMAT

```

HI
'date'   'time'   'username'   'program'   'function'   'elapsed'   'cpu'   'status'
.         .         .             .             .             .         .         .
.         .         .             .             .             .         .         .
.         .         .             .             .             .         .         .
  
```

Example

Date	Time	Username	Program	Function	Elapsed	CPU	STATUS
23-JUL-1985	12:22	CLARKE	TWOTVES	Output	01:31:34	00:09:05	00000001

DESCRIPTION

IFF files created after January 1987 provide an optional mechanism for automatically recording statistics each time they are updated, so that it may be determined which users and programs contributed to the final state of the file. The statistics are stored in an HI (HIsTory) entry within the IFF file. This entry is of fixed length (4001 words). The first word contains a count of the number of filled 'history records', and is followed by space for 100 80-byte ASCII history records.

NOTES

In order that the IFF history mechanism can work, a blank history entry must be inserted in files created from scratch. Clearly it would cause compatibility problems with customer databases which contain historic IFF files if HI entries were created in output files by default. To overcome this problem an HI entry is only written to the output file if (one of) the input files contained an HI entry. The IMP module ISTART creates a template IFF file containing an HI entry.

In addition to writing timing statistics into the HI entry, many IFF manipulation programs also set the 'function' field of the HI and also set the final status field to indicate the success (or failure!) of the run.

IMP provides a module (IINFO) which may be used to analyse the contents of an IFF file's HI entry and optionally produce timing statistics to indicate total elapsed and CPU time spent during the processing of the file.

When an IFF file is opened for write, a 'prototype' history record, with blank elapsed and CPU fields, and a status of 0, is written to the HI entry and also to the forepart of the file. In the event that the file is never properly closed, this record can be examined (possibly using the VMS DUMP utility) to determine which operation had failed. It will not be possible to open such a file with any IFF

library (IFFLIB) based utility (such as those of IMP) until the IMP module IMEND has been used.

RA RAnge

FORMAT

RA X-min X-max Y-min Y-max

where:

X-min is the minimum X-coordinate value (real*4) within the file

X-max is the maximum X-coordinate value (real*4) within the file

Y-min is the minimum Y-coordinate value (real*4) within the file

Y-max is the maximum Y-coordinate value (real*4) within the file

Example

RA 246.890 560.89 348.000 1000.000

DESCRIPTION

The RA entry records the maximum extent of the data in the IFF file. It is used by plot and display programs to work out whether to clip the file, and what scale it can be displayed at.

The range entry is always the first entry in the IFF file.

NOTES

The range entry will always be maintained by Laser-Scan IFF manipulation programs. The user should only need to alter the range if digitising using LITES2 results in data straying more than 10% outside the existing CP (Control Point) values. The range values may be altered using IPATCH.

The range entry values **do not reflect any origin offset** used in the IFF file. The range values only reflect the coordinate range of data held in ST (coordinate STring) and ZS (3-dimensional string) entries.

Note that only the recorded coordinates are used, so that the RA entry will not take into account the path of circle arcs or interpolated curves.

The contents of CP (Control Point) entries are not normally used in any calculation of data range.

The coordinate values of registration mark and grid features contained in layer 0 will be reflected in the range.

CHAPTER 4

MAP LEVEL ENTRIES

MAP LEVEL ENTRIES

General

In historical IFF files the map level was used when there is a need for holding different maps within the same file. The main reason for this was for edgematching of multiple maps in the (now obsolete) LITES cartographic editor. As the LITES2 cartographic editor can read multiple IFF files, this requirement is no longer valid.

Laser-Scan no longer support multiple maps within an IFF file, file which have this characteristic are considered to be 'historical' and some multi-map functionality may be unavailable with some modern IFF manipulation programs (eg those which comprise IMP).

With the exception of LASERAID patch files, all IFF files must contain at least one map within the hierarchic structure of the file.

Map level entry order

The order in which IFF entries occur at the map level is given below. All are obligatory.

- MH - Map Header (map-specific information)**
- MD - Map Descriptor (map projection, scale, etc)**
- EM - End Map marker**

EM End of Map marker.

FORMAT

EM

The EM entry does not have any contents.

DESCRIPTION

The EM entry is the last entry of a map and flags its end. It should be the last entry before the EJ entry.

Historically, IFF files were permitted to contain several maps, and the EM entry was then required to mark the end of each. Laser-Scan no longer support multiple maps within IFF files.

MD Map Descriptor

For historical reasons, IFF files may contain one of two Map Descriptor types:

FORMAT**Type 1 Map Descriptor**

The type 1 MD is now obsolete. It was designed for use with the IPR transformation module of the obsolete DAMP package. It has been superseded by the larger and more complex type 2 MD.

The type 1 MD has the following format:

MD	MA	'integer'	'real'	'real'	'real'	'real'
	GR	'integer'	'real'	'real'	'real'	'real'
	SC	'real'				
	PS	'integer'	'integer'			
	AG	'integer'	'real'	'real'	'real'	'real'

This contains information on the map projection, scale etc.

The alphabetic codes at the start of each line define the data content:

- o MA - Map Area,
- o GR - Grid Representation,
- o SC - SScale,
- o PS - Projection Status, and,
- o AG - Auxiliary Grid.

The grid representation (GR) and auxiliary grid (AG) fields are used to specify 2 independent projections. Which one is used as the source and which the destination will depend on the PS sub-entry of the type 1 MD entry, and on the commands given to the DAMP transformation module IPR.

The actual data contained in the various sub-entries is too specialised to describe here. The reader is referred to the now **obsolete** DAMP User Guide documentation.

MD	MA	-1	0.0	0.0	0.0	0.0
	GR	0	0.0	0.0	0.0	0.0
	SC			0.0		
	PS	0	0			
	AG	0	0.0	0.0	0.0	0.0

Type 2 Map Descriptor

It is not within the scope of this document to detail the options available within a type 2 MD. There are only 5 fields which may be of common interest to the ordinary user, these are interpreted for the user by the IMP utilities IPATCH, IINFO and IFROMTEXT.

This is an example of how the IMP patch editor IPATCH interprets and displays the contents of a type 2 MD. It shows the 5 commonly used fields:

```
- map descriptor version 2
- Local origin:      23200.0000,      680000.0000
- Map scale:         10000.0000
- Projection:         101  (Transverse Mercator)
- Spheroid:           9    (Airy)
- Units:              2    (Metres)
```

The remainder of the entry is stored as an array of longwords which is almost impossible to interpret by hand.

Example

[illegible]

0000 0000 0000 0000 0000

DESCRIPTION

The MD entry contains data describing the origin, projection and coordinate system of the IFF file.

It is not possible to edit the MD entry using IPATCH. The map descriptor describes the current origin, projection and units of the IFF file, and a change to the map descriptor must thus be accompanied by the appropriate changes to all point data in the file. The IMP utility ITRANS is provided to perform such transformations, and it will modify the map descriptor correctly.

The reader is referred to the following additional documentation:

DAMP User Guide

Although obsolete, it details the use of the IPR transformation utility and the IED IFF editor needed to set up the type 1 MD.

IMP Reference Manual

This describes the ITRANS transformation utility and contains appendices detailing the projections supported using a type 2 MD. This document also describes the use of the IPATCH editor which can be used to examine the contents of an MD entry.

NOTES

The MD entry occurs once in the file, after the MH entry. Note that historically IFF files could contain more than one map, and such maps each started with a separate map header, and ended with an EM entry. Each separate map contained its own MD. It was thus possible to have data pertaining to different projections within one IFF file. Multiple maps within one IFF file are no longer supported.

An unset map descriptor has its first word set to -1 (or FFFF in hex).

FORMAT

```

where:
'length'           is the length of the MH entry in words, and,
'customer_number'  is a number used to indicate which customer
                   format is used in the MH

```

[illegible]

DESCRIPTION

The IMP utility IPATCH supports editing of the map header's customer number. For editing the actual value within the map header itself, use the relevant customer specific map header editing program - that is:

- 0 map header is unset.
- 1 MCE specific map header. Use MCEHED to edit it.
- 2 OS specific map header. Use MHED to edit it.

Use the IMP utility IPATCH if you wish to change the customer number in the descriptor of the map header to a different value. Note that this will change the interpretation of any data in the map header by utility programs.

NOTES

The map header occurs once in the file, after the HI entry. Note that historically IFF files could contain more than one map, and such maps each started with a separate map header, and ended with an EM entry. Multiple maps within one IFF file are no longer supported.

CHAPTER 5

SECTION LEVEL ENTRIES

SECTION LEVEL ENTRIES

General

Sections are used to separate the data resulting from different digitising sessions. Such data may all be part of the same map, but the coordinate system used in one section will not necessarily be the same as that of another (due to the map being repositioned between sessions). Data sent to customers is unlikely to still be split up in this fashion, although at least one section must be present within a map.

Sections - a warning

Due to the nature of the digitising process, either using LASERAID or a manual table system, IFF files will usually contain data from several digitising sessions. Each time a digitising session is started the IFF file is extended with a new header sequence of NS (New Section), CC (Cubic Coefficients) and CP (Control Points) entries. A new layer is then opened and the additional digitising added.

It is essential that the multiple sections of an IFF file and the fragmentary layers be concatenated before further LAMPS processing may be conducted. LAPROCESS must first be used if the IFF file was digitised using LASERAID. Then either IMERGE or LITES2 may be used to consolidate all the layer parts from different sessions, before further IMP processing. If IMERGE is used the IFF file should be merged to itself, for example:

```
$ IMERGE/LOG SHEET274NE_CONTOURS.IFF SHEET274NE_CONTOURS.IFF
```

The consolidated file will be given the next highest version number to that of the input file.

NOTE

It is essential that IMERGE is NEVER used on files that were created using LASERAID and that have not been processed using LAPROCESS.

Failure to concatenate sections may result in unpredictable processing behaviour by some LAMPS utilities. Many IFF to customer format conversion utilities rely heavily upon the existence of a single section within the IFF file.

Section level entry order

The following entries are specific to sections, and they occur in the following order:

NS - New Section identification.

CC - Cubic Coefficients for coordinate transformation.

CP - Corner Points for coordinate transformation.

CC Cubic Coefficients

FORMAT

```

CC
      'real'  'real'
      'real'  'real'
      'real'  'real'
      'real'  'real'
      'real'  'real'
      'real'  'real'
      'real'  'real'
      'real'  'real'
      'real'  'real'
      'real'  'real'

```

The matrix contained within a CC entry defines a transformation between two coordinate systems.

Example

The following is an example of CC entry describing a unit transformation:

```

.000000E 000   .0000000E 000
.100000E 001   .0000000E 000
.000000E 000   .1000000E 001
.000000E 000   .0000000E 000
.000000E 000   .0000000E 000
.000000E 000   .0000000E 000
.000000E 000   .0000000E 000
.000000E 000   .0000000E 000
.000000E 000   .0000000E 000
.000000E 000   .0000000E 000
.000000E 000   .0000000E 000

```

DESCRIPTION

The CC entry occurs once per section, and is a matrix of real numbers of size (10,2), indexed as for Fortran. It defines a transformation between two coordinate systems to be used by a post-processing program to transform all points in the file into the same coordinate space. This is necessary when (for instance) data has been digitised on a Lasertrak.

The matrix may be represented as

a	k
b	l
c	m
d	n
e	o
f	p
g	q

h	r
i	s
j	t

which represents the transformation

$$\begin{aligned} X' &= a + bX + cY + dXX + eXY + fYY + gXXX + hXXY + iXYY + jYYY \\ Y' &= k + lX + mY + nXX + oXY + pYY + qXXX + rXXY + sXYY + tYYY \end{aligned}$$

The unit matrix specifies a unit transformation - ie no transformation is necessary. The unit matrix has all terms 0.0, except for those at (2,1) and (3,2) which are 1.0. In the convention adopted above these are the b and m terms.

NOTES

An IFF file which contains multiple CC entries should be treated with care. Check that the CC entries are set to describe the unit transform (examine the example CC entry above to see what a unit transform looks like). If they are not then examine the NS (New Section) or HI (HIstory) entry (if present) to determine whether the file was digitised using LASERAID. If LASERAID was used you may safely use LAPROCESS to apply the cubic coefficients. Use the IMPIMERGE utility to merge the file into itself. This will concatenate the sections and all layer fragments within the file. LITES2 may be used as an alternative to IMERGE.

If there is no evidence from the HI or NS entry that the file was digitised using LASERAID consult the flowline supervisor. The CC entry may have been set by special purpose software and thus the file should not be processed using LAPROCESS.

CP Control Points

FORMAT

	(Position)	(Current Space)		(Target Space)	
CP	NW	X-value	Y-value	X-value	Y-value
	SW	X-value	Y-value	X-value	Y-value
	SE	X-value	Y-value	X-value	Y-value
	NE	X-value	Y-value	X-value	Y-value

IFF control points are always used in the order NW, SW, SE and NE.

Example

CP	0.0	200.0	0.0	200.0
	0.0	0.0	0.0	0.0
	200.0	0.0	200.0	0.0
	200.0	200.0	200.0	200.0

DESCRIPTION

The CP entry occurs once for each section. It defines the control points for the section, in both original (current) and destination (target) space.

IFF control points are always used in the order NW, SW, SE and NE.

The SW control point is defined as the control point with the minimum X and minimum Y coordinates. The NE control point is defined as the control point with the maximum X and maximum Y coordinates.

IFF control points must be cyclic, that is to say that when plotted consecutively, the control points should define a four sided polygon in which none of the sides cross or touch each other.

CP values may be set in a template file prior to digitising by using the IMP utility ISTART.

CP values may be altered using the IMP patch editor IPATCH. (It is recommended that the user becomes familiar with the IPATCH documentation before attempting to change CP values).

Modern CP entries may be used in conjunction with the origin offset information contained in a type 2 MD (Map Descriptor).

NOTES

An IFF file which contains CP entries in which the "current" and "target" values are different needs processing using LAPROCESS. If it is an historical IFF file check with your flowline supervisor. Obsolete pre-converged versions of LASERAID (ELA, FLF and LAJ) require special purpose processing utilities, for example IPR, OPR and RAPATCH. All of these are now obsolete.

Historical IFF files may be irreparably damaged if processed with the wrong utility. Similarly a modern LASERAID IFF file will be ruined if processed with any of the obsolete processing/transformation utilities.

NS New Section

FORMAT

NS "text to identify the new digitising section"

The text written in an NS entry conventionally contains information describing how the following section of the IFF file was created.

Example

NS Template created by ISTART in default template mode

DESCRIPTION

The NS entry will follow either an MD entry (for the first section) or an EO entry (for later sections).

The NS entry is used to flag the start of a new digitising session. The text written in it conventionally contains the name of the digitising operator, the program being used and the date and time at which digitising was started.

The entry is necessary because IFF files are often digitised in several sessions. The NS entry will be followed by a CC and a CP entry, defining the transformations and control points applying from now on. Post-processing programs used to transform all data into the same coordinate system will normally also reduce the whole IFF file down to one section.

NOTES

The maximum number of characters in the text is 255.

Utility programs which accept IFF files containing multiple NS entries (eg IMERGE or LITES2) will copy only the first input NS entry to the output IFF file.

CHAPTER 6

LAYER LEVEL ENTRIES

LAYER LEVEL ENTRIES

General

The name of the IFF layer has evolved through time. Traditionally the division of IFF data provided by the NO "New Overlay" and EO "End Overlay" entries was referred to as either an "overlay", or alternatively, as a "layer". Laser-Scan have now standardised on the term "layer" and all modern IFF based documentation will use this term.

The role of the IFF layer has also varied through time. Historically, layers have been used for data division purposes which are no longer necessary in the modern LAMPS environment. For example, the replacement of SPM based plotting packages by the LAMPS FPP (Fast Plotter Program) has meant that plotter pens may now be selected by feature code rather than data division by layer. The replacement of the LITES editor by the powerful LITES2 editor enables IFF files containing many layers to be edited. Released from the constraints of plotting requirements and capacity restrictions in the IFF graphical editor layers offer the user great flexibility in data classification and division within a single IFF file. The feature codes contained in all IFF features may be used to further subdivide the data within the layers.

Laser-Scan reserve layer 0 for registration mark features and grids. All other layers may be assigned for any purpose. Historically, however, several layers were assigned special functions by both Laser-Scan and the Ordnance Survey (GB):

- o **Layer 0.** Although now reserved for registration mark features and grids this layer was at one time available for general data storage. Modern LAMPS utilities treat layer 0 as a special layer. The user should check the relevant documentation to determine the precise behaviour of each utility on encountering layer 0. The user should be aware that:
 1. data processing utilities often ignore layer 0,
 2. data selection utilities will often copy layer 0 to the output file even if it is not explicitly asked for.
- o **Layer 1.** Historically this is the Ordnance Survey data layer. All data were differentiated by feature code and only 3 predefined layers were used.
- o **Layer 11.** Historically this is the Ordnance Survey grid layer. All OS IFF files contain a grid which was defined by line features contained in layer 11. This layer is now available for general data storage.
- o **Layer 32.** Until 1986 the Laser-Scan LASERAID digitiser put registration ticks into layer 32. These are now placed in layer 0. Layer 32 is now used for general data storage.

All IFF features must lie within an IFF layer.

All layer numbers must lie within the range 0 to 32767.

Layer level entry order

The order in which IFF entries occur at the map level is given below. An entry printed in **bold** type is obligatory. Vertical ellipsis indicate that multiple occurrences of the preceding entry are permissible.

NO - New "Overlay" - includes layer number and status.

EO - End "Overlay" marker.

There is no specific order for the following optional (and obsolete) entries, which occur within layers but outside features. These entries are documented in the special "Miscellaneous and obsolete entries" section of this document.

CH - Character data.

.

.

.

CS - Character Size and spacing.

SL - Symbol Library select.

SS - Symbol Select.

TC - Transmitted Comment.

.

.

.

EO End of layer marker

FORMAT

EO

The EO entry does not have any contents.

DESCRIPTION

The EO entry flags the end of a layer, and balances the matching NO entry at the start of the layer. That NO entry should contain a pointer field which holds the address of this EO - this allows fast chaining through a file when particular layers are being ignored.

NOTES

Although primarily used to group IFF features, some non-feature based entries are legal between an EO and an NO - particularly the new section entries (NS,CC,CP) and also the (obsolete) CH entry.

NO New layer (overlay) entry

For historical reasons the user should be aware of 2 types of NO entry:

FORMAT - old pattern NO entry

NO layer-number status
where:

layer-number - is the layer number in the range 0 to 32767
status - is a status flag field. The status flag is
 not currently used, and should be set to zero.

FORMAT - modern pattern NO entry

NO layer-number status pointer

where:

layer-number - is the layer number in the range 0 to 32767
status - is a status flag field. The status flag is
 not currently used, and should be set to zero.
pointer - is the address of the matching EO entry which
 flags the end of the layer.

DESCRIPTION

The NO entry starts a layer. It contains the number identifying this layer, and a status word. It may also contain the address of the corresponding EO entry.

IFF files are generally divided up into multiple layers, where data of a common sort or source is grouped in the same layer. Note that a layer may be split into several parts, identified by all having the same layer number.

Layer 0 is conventionally reserved for 'non-essential' data - for instance a grid, symbols at control points, or MCE DFAD accuracy polygons. The assumption is that the IFF file would not be significantly degraded in terms of information content by throwing layer 0 away. If layer 0 is present, some programs expect it to be the first layer in the file.

Historically, layers 11 and 32 were also used for the same sort of purpose as layer 0. (See General introduction at the start of this section).

NOTES

If the EO pointer field is present, it should contain the address of the EO entry which matches this NO. This is used by programs which must ignore layers to 'jump' from the NO to the EO. Thus take great care when editing the EO pointer field, as an incorrect value could

cause a processing program to abort.

In the unlikely event that an NO entry contains an incorrect EO pointer address the file may be repaired using one of the following procedures:

- o Use the IMP patch editor IPATCH. For safety make a copy of the IFF file. Specify the /WRITE qualifier on the IPATCH command line. Use the find by entry command:

```
IPATCH> NO<CR>
```

to find the first NO entry in the file. Note the hexadecimal address of the NO entry. Does the NO entry have a pointer field? If not then try one of the other strategies listed below.

If the NO entry does contain an EO pointer field use the IPATCH find by entry command:

```
IPATCH> EO<CR>
```

to find the next EO entry. Note the address of the EO entry. If the address of the EO matches the address in the pointer field of the NO that you have just left then find the next NO. There is nothing wrong with the current NO-EO pair.

If the address of the EO entry does not match the address in the pointer field of the EO entry that you have just left, use the find by address command:

```
IPATCH> FIND 'address'<CR>
```

where 'address' is the hexadecimal address of the NO entry that you last found. Issue the NO entry edit command:

```
IPATCH> /POINTER<CR>
```

to edit the NO entry pointer field. Specify the address of the EO that you have just left. Repeat for any other NO-EO pairs within the IFF file.

- o As an alternative to IPATCH read the IFF file into LITES2 and then issue the "EXIT" command. LITES2 will correct any wrong pointer addresses and restore any missing ones. This strategy will succeed even if the NO entries contained no pointer address field. The IFF file will now have the modern pattern NO entries containing a pointer address field.
- o Use the IMP ITOTEXT utility followed by the IMP IFROMTEXT utility. Like the LITES2 strategy, this approach will succeed even if the NO entries contained no pointer address field. The IFF file will now have the modern pattern NO entries containing a pointer address field. Beware that this approach, while suitable for batch operation, may require a considerable amount of disk storage for the text file created by ITOTEXT.

CHAPTER 7

FEATURE LEVEL ENTRIES

FEATURE LEVEL ENTRIES

General

Features are the building blocks of IFF (Internal Feature Format) files.

Individual features are held at the feature level (a feature being synonymous with a cartographic feature such as a particular line, symbol or piece of text, etc).

Definition of an IFF feature

Although historically IFF files could contain empty features, all IFF features must now contain a FS entry and a minimum of one (X,Y) coordinate point. The (X,Y) coordinate point can be held either in an ST entry (2 dimensional string), in a ZS entry (3 dimensional string), or in a CB entry (coordinate block). A feature **cannot contain a mix of ST, ZS, and CB entries.**

The minimum structure of a single IFF feature may thus be defined:

NF	- start of new feature
FS	- feature status
ST/ZS/CB	- string of coordinates
EF	- end of feature

Using this basic building block other IFF feature types may be constructed. Ancillary codes and large numbers of coordinate points may be added, for example:

NF	- start of new feature
FS	- feature status
AC	- ancillary code
.	
.	
.	
ST/ZS/CB	- string of coordinates
.	
.	
.	
EF	- end of feature

Note that ACs are found immediately after the FS entry, before any other feature entries.

Simple text features may be defined using our feature building block. Text is located by a single point in an ST, ZS, or CB entry. Thus features with multi-point strings cannot have a TX entry within them, but they can, however, contain ACs which have an optional text field.

NF	- start of new feature
FS	- feature status
ST/ZS/CB	- locating point
TX	- text string
EF	- end of feature

And the text may be rotated counterclockwise relative to the X axis:

NF	- start of new feature
FS	- feature status
ST/ZS/CB	- locating point
RO	- rotation
TX	- text string
EF	- end of feature

The text size may be defined within the feature itself:

NF	- start of new feature
FS	- feature status
TH	- text height
ST/ZS/CB	- locating point
RO	- rotation
TX	- text string
EF	- end of feature

Multi-banked texts may be created by adding a text component for each piece of text. A text component takes the form:

TS	- text status
TH	- text height
ST/ZS/CB	- locating point
RO	- rotation
TX	- text string

Thus a feature with 2 text strings would take the form:

```
NF      - start of new feature
FS      - feature status
  TS      - text status
  TH      - text height
  ST/ZS/CB - locating point
  RO      - rotation
  TX      - text string
  TS      - text status
  TH      - text height
  ST/ZS/CB - locating point
  RO      - rotation
  TX      - text string
EF      - end of feature
```

NOTE

Note that although text features may contain ACs (Ancillary Codes), a text component may not (as ACs must occur immediately after the FS entry).

A note on IFF features and transmitted comments

Historically IFF structure definition has presented a liberal view of the use of TC (transmitted comment) entries. For several years some customers used TCs to carry the sort of information today contained in ACs. The TCs were allowed to lie outside of features but the context of the TC was considered to apply to any feature which followed the TC. Thus the following feature construction resulted:

2D feature only

```
TC - transmitted comment
NF - start of new feature
FS - feature status
ST - 2D string
EF - end of feature
```

As TCs lay outside features but inside the layer level a problem arose if a TC lay between the EF of the last feature in the layer and the EO (end of layer). There would be no feature to attach the TC to for editing purposes within the (now obsolete) SOL and LITES editors. To overcome this problem an empty feature having a feature serial number of zero was invented. This was used in the following manner:

2D feature only

.
EF - end of last real feature in layer
TC - transmitted comment
NF - start of empty feature with an FSN of 0
EF - end of feature
EO - end of layer

This construction is now obsolete and preservation of empty features is no longer guaranteed. For further details about TCs see the section devoted to obsolete IFF entries.

A single layer, or a single IFF file, may contain a maximum of 65535 features. In practice this figure need rarely worry the user. As file sizes increase file manipulation becomes less efficient. The i/o overhead for a LITES2 editing session, for example, may exceed the time taken to do the edits!

The (X,Y) coordinates from all features are used in the calculation of the RA (RAnge) entry values for the whole file.

AC Ancillary Code

FORMAT

AC type value ["text"]

Example

Two examples of ACs are given. The first illustrates use of the optional text field. The second shows a real value AC which does not have text.

```
AC      5      570      "GPO inspection pit, cast cover, depth 2.1m, 110v"
AC      3      384.2
```

DESCRIPTION

AC entries are used to hold miscellaneous information. They are composed of a type word, a longword value and an optional text.

The type of information held in an AC is determined by the AC type. Using the ACD section of an FRT file (see the FRT User's Guide), a name, datatype (integer, real, character, date, time), and range of legal values may be associated with the numerical AC type. Some default types are always defined, whether or not an FRT file is used:

Data type	AC type	Name	Minimum	Maximum
Integer	1	Secondary_FC	0	32767
Integer	2	Contour	-2147483647	2147483647
Real	3	Height	-1.0E37	1.0E37
Integer	4	LH_boundary	0	32767
Integer	5	RH_boundary	0	32767
Integer	6	Text	0	32767
Integer	7	DFAD_FADT	0	0
Integer	8	DFAD_ACC	0	0
Integer	9	Parent_FSN	0	65535
Integer	10	RELHT_START	0	100
Integer	11	RELHT_END	0	100
Real	80	Cliff_left	-1.0E37	+1.0E37
Real	81	Cliff_right	-1.0E37	+1.0E37
Real	82	Polygon_info	-1.0E37	+1.0E37
Real	91	X	-1.0E37	+1.0E37
Real	92	Y	-1.0E37	+1.0E37
Real	93	Z	-1.0E37	+1.0E37
Real	94	ZB	-1.0E37	+1.0E37
Real	95	ZC	-1.0E37	+1.0E37
Real	96	ZD	-1.0E37	+1.0E37
Real	97	Dheight	-1.0E37	+1.0E37

0 reserved for future assignment by LSL
1 secondary feature code. The AC value holds a subsidiary feature code. The AC text is not used.
2 integer contour height. The AC value holds a contour height, as an integer. The AC text is not used.
3 real contour height. The AC value holds a contour height, as a real number. The AC text is not used.
4 left hand boundary definition. The AC value holds a secondary feature code, the AC text holds a boundary text.
5 right hand boundary definition. The AC value holds a secondary feature code, the AC text holds a boundary text.
6 text. The AC value is not used.
7 MCE DFAD data. The AC value is not used, the AC text holds DFAD commands specifying typing information.
8 MCE DFAD accuracy data. The AC value is not used, the AC text holds DFAD commands specifying accuracy information.
9 ID number of parent feature. The AC value is a number which is the same for all features formed by splitting the parent feature. The AC text is not used.
10 'relative height' at start. The AC value is a sequence number (0 or higher) used to identify the height ordering of a series of features meeting at a junction. A type 10 AC refers to the ordering of the start of the feature. The AC text is not used.
11 'relative height' at end. The AC value is used as for type 10, but relates to the end of the feature. The AC text is not used.
12-79 reserved for future assignment by LSL
80 real height at left of cliff line. The AC value holds the height to the left of a cliff feature. The AC text is not used.
81 real height at right of cliff line. The AC value holds the height to the right of a cliff feature. The AC text is not used.
82 real area of a polygon. The AC text contains the polygon code.
83-90 reserved for future assignment by LSL. Note that these ACs will have real values in their AC values.
91-97 these definitions are not expected to occur in AC entries. They define columns (including the X, Y, and Z coordinates) which occur in CB entries.
98-99 reserved for future assignment by LSL. Note that these ACs will have real values in their AC values.
100-119 MCE specific codes. For internal use by MCE.
120-139 USGS specific codes. The AC type number and AC value are used for typing information. The AC text is not used.
140-999 reserved by LSL.
1000-32767 user-defined ACs specified by ACD records in the FRT file.

AC types may take values in the range 0-32767.

Note that AC type 3 and AC types 80 through 99 are currently the only default ACs to have a real value. Any AC type which is not defined by default or in the FRT file will be treated as having an integer value.

AC texts may be up to 255 characters in length.

NOTES

If the ancillary code type number is changed, the user should be most careful to ensure that undesirable alteration of the value field does not occur. ACs with type value 3 and ACs with type values in the range 80 to 99 have floating point numbers in the value field. If the AC type value is changed to a an AC type which has an integer value in the value field, the floating point value will be interpreted as an integer value. Similarly, if an AC type having an integer value in the value field is changed to an AC type having a floating point number in the value field the value field will be interpreted as a floating point number.

CB Coordinate Block

FORMAT

```
CB nrow flags gtype ncol natt
```

```
      (code value
natt(  .      .      fixed attributes
      (  .      .
```

```
          code_1 code_2 ... code_ncol
      (value_1 value_2 ... value_ncol
nrow(  .      .      ...      .
      (  .      .      ...      .
```

where:

```
nrow      - the number of coordinates in the CB entry
flags     - pen down (1) /pen up (0) control flag
gtype     - graphical type represented by this CB (1-12)
ncol      - number of attributes per point
natt      - number of fixed attributes
```

DESCRIPTION

The CB entry contains the point data defining the feature. There may be more than one CB entry in a feature. The CB entry allows attributes other than X, Y, or Z coordinates to be defined on a per-point basis. These attributes may be 'fixed' i.e. constant for all points in the CB, or 'varying' i.e. defined for each point. In either case, a particular attribute is identified by an integer code which is looked up exactly as for AC type in the ACD section of an FRT file to obtain a name, data type and range of permissible values. Even X, Y, and Z are dealt with in this way, being codes 91, 92, and 93 respectively. As with AC entries, some codes are defined by default, whether or not an FRT file is used.

If an attribute in a CB has the integer value '80000000' hexadecimal (-2147483648 decimal), then it is treated exactly as if the attribute was not present at all. This allows for the case when a particular attribute is only present at some of the points within a CB.

CB is the only possible coordinate string entry in an output revision level 1 IFF file.

Each CB should contain no more than 200 points, so a new CB with the flags set to 1 (ie keep pen down when drawing to the start of this new CB) will be started for the remaining points.

If an invisible line is to be coded in the feature, then this is represented by starting a new CB with the flags set to 0 (ie keep pen up when moving to the start of this new CB).

CB X and Y coordinate values may optionally be offset using the origin offset facility in a type 2 MD (Map descriptor).

NOTES

Note that the pen flag is always considered to be 0 for the first CB in a feature - that is the pen is always kept up to move to the start of a new feature.

A CB entry must contain at least X and Y columns in order to be sensible.

EF End of Feature

FORMAT

EF
The EF entry does not have any contents.

DESCRIPTION
The EF entry flags the end of a feature, and balances the NF entry at the start of the feature.

FS Feature Status

FORMAT

FS fc status pc/text user-word

where:

fc	- is the feature code
status	- flag data
pc/text	- feature type or process code
user-word	- ephemeral user dependent data.

Example

FS 23 0000 0000 0000

DESCRIPTION

The FS entry contains data which describes the feature containing it. It should be the first entry after the NF entry.

The FS entry has 4 word length fields, the contents of which may be summarised as follows:

1. The **feature code** is in field 1, and is the primary descriptive code for a feature.
2. The second, **status**, field is used by Laseraid and its post-processing programs.
3. The third **pc/text** field states whether this feature is text, symbol or line, and also holds either a process code or a description of the type of text.
4. The fourth **user-word** field is reserved for use by users. Beware that not all processing programs will preserve this word from input to output.

Within the modern LAMPS environment the first field of the feature status entry is called the feature code. Historically, this has not always been the case as the first field has been interpreted in one of two ways, depending on the degree of attribute coding required.

1. Interpolation Type - a direct indication of how the feature is to be drawn:

IT0	- straight lines
IT1	- single symbol at each point
IT5	- cubic interpolation
IT64	- text feature

Interpolation Type was sometimes historically referred to as Graphical Code (GC).

2. Feature Code (FC) - the value is looked up, either in a Legenda file, or an FRT file to yield a Graphical Type for the code, along with details of line style, symbol definitions, colour, line thickness and so on (see FRTLIB, FPP and LITES2 documentation).

Modern Laser-Scan usage and documentation always refers to "feature code".

The **status** field of the feature status contains status bits used by the Laser-Scan Laseraid digitiser. The status bits are defined currently as follows:

Bit number	Value	Meaning when set	Meaning when clear
0	1	feature is closed	feature is open
1	2	feature is a line	feature is an edge
2	4	REVersed feature	non-REVersed feature
3	8	two-part (FINd) feature	not a two-part feature
4	16	Paintout-Only feature	normal feature (keep)
5	32	squaring flag set	squaring flag clear
6	64	INVerse polarity used	normal polarity
15	32768	PaintOut Suppress used	no PaintOut Suppress

All other bits are undefined at present.

The **pc/text** word contains flags, interpreted according to the value of top two bits:

Bits	Values	Meaning
d14-15	0	line, circle, area or symbol string feature
	1	symbol feature
	2	text feature
	3	value reserved

For text features, the rest of the word is interpreted as follows:

d0-3	0-8	text position code
d4-5	0-3	type style (O.S. only)
d6-11	0-63	name category (O.S. only)

For other feature types, the rest of the word contains the process code.

NOTES

Words 2 and 3 of an FS entry are always decoded bitwise. The individual bits of these words may be edited using the IMP patch editor IPATCH.

The **user-word** word of the feature status contains user dependent data about the feature. The user should be aware that data stored in the **user-word** field of the feature status may be lost during LAMPS processing.

NF Start of New Feature

FORMAT

NF fsn isn

where:

- fsn - the feature serial number, duplicates are allowed.
- isn - feature internal sequence number. This should be
 unique and reflects digitising order.

Example

NF 38 38

DESCRIPTION

The NF entry starts a new feature. It contains two identifying numbers, both in the range 0-65535.

The feature serial number (FSN) is generally used as the 'name' of the feature. There are three main conventions about when or whether FSNs are unique:

1. FSNs unique within the IFF file. This mechanism has traditionally been used by OS, and may be used by other customers as well.
2. FSNs unique within a particular layer. Thus a feature could be identified by its layer and FSN. This mechanism has traditionally been used by MCE.
3. FSNs not unique. The main example of this now is in TRI and SRI files, where single symbols may be made up of multiple features, all with the same FSN. In these cases the FSN is the ASCII code of the text, or the identifying number of the symbol.

The internal sequence number (ISN) is unique within the IFF file, and can be used as a single unique identifier for a feature. It may, but need not, be the same as the FSN. The ISN is generally assigned starting at 1 and incremented as a file is digitised.

NOTES

Note that historically FSN 0 was considered special by many programs. It was used to flag an empty feature used as a place-holder in the IFF file, generally to hold TCs which applied to the entire layer. Thus it was possible to come across multiple FSN 0s even when FSNs were notionally unique within a file. In modern IFF files, the trailing TCs are no longer used.

Note that many programs use the internal sequence number as a unique identifier for features, and therefore care should be taken when editing this field using the IMP patch editor IPATCH.

RO ROtation entry

FORMAT

RO angle

where:

angle - is the angle of rotation.

Example

RO 0.72

DESCRIPTION

The RO entry defines the angle at which an oriented symbol or a text is to be drawn. The angle is in radians, measured counterclockwise from the positive horizontal axis.

The RO entry should occur after the ST, ZS, or CB entry of a feature, and before any TX entry.

ST 2 dimensional point string entry

FORMAT

ST npts pen
 x-coordinate y-coordinate
 .
 .
 .

where:

npts - the number of coordinates in the ST entry
pen - pen down (1) /pen up (0) control flag

DESCRIPTION

The ST entry contains the point data defining the feature. There may be more than one ST entry in a feature.

It is illegal to have both ST and ZS entries within the same IFF feature. ST entries only occur in IFF output revision level 0 files. In a revision level 1 file, the same effect is achieved by a CB entry with at least X and Y columns.

Each ST should contain no more than 200 points, so a new ST with the pen flag set to 1 (ie keep pen down when drawing to the start of this new ST) will be started for the remaining points.

If an invisible line is to be coded in the feature, then this is represented by starting a new ST with the pen flag set to 0 (ie keep pen up when moving to the start of this new ST).

An ST is suitable for data which does not have an inherent height, or which is all at one height (in which case the feature will also have a contour height AC). For truly three-dimensional data, a ZS entry is more appropriate.

ST coordinate values may optionally be offset using the origin offset facility in a type 2 MD (Map descriptor).

NOTES

Note that the pen flag is always considered to be 0 for the first ST in a feature - that is the pen is always kept up to move to the start of a new feature.

TH **Text height / line Thickness entry**

FORMAT

TH thickness

where:

thickness - is an integer defining line thickness or text size.

DESCRIPTION

The TH entry is used for the following purposes:

1. to hold the line thickness for displays which are capable of displaying multiple thickness lines
2. to hold the size of text for a text feature. In this case the integer is interpreted as either:
 - o the height of the text in hundredths of a millimetre
 - o the size of the text as a point size

The TH entry should occur before the ST or ZS entry, and after any ACs.

NOTES

On obsolete legenda based LAMPS systems a negative TH entry value determined line styles. The negative TH value was used to index a line pattern table. Negative TH values are no longer supported. Line styles are now determined by reference to an FRT (Feature Representation Table) which is indexed on the basis of the features feature code alone. The TH entry value now determines text height or line thickness only.

TS Text Status entry

FORMAT

TS tcc reserved textcode reserved

where:

tcc	- is the text component code
reserved	- reserved for future Laser-Scan allocation
textcode	- text word: describes the type of text

DESCRIPTION

The TS entry introduces a text component, and contains data which describes the text component following it.

Text features may contain one text string, with associated location and descriptive data, or they may be **composite** - that is composed of several sub-texts or text components, which may be manipulated independently or as a single entity. It is thus possible to have a "multi-banked text" or text paragraph contained within a single IFF feature.

Each text component starts with a TS entry, and ends with the next TS entry, or the final EF of the feature. The first TS entry occurs immediately after the FS entry and any AC entries. Text components may not include FS or AC entries, but may contain any other entries that are legal within a normal text feature.

Word 1 of the TS entry is the text component code (TCC), which is the primary descriptive code for a text component - it is effectively the feature code for this component of the composite text, and is used in the same manner.

Word 3 of the TS entry is the text word, and is identical in form to word 3 of a text's FS entry - it holds a description of the type of text component. Note that the top two bits (what would be the text/symbol bits in an FS) should always be set to '10' binary, as they would in an FS entry. These bits are referred to as redundancy bits, since they are not strictly necessary.

The second and fourth words are reserved for later definition, and should always be zero.

NOTES

The user should be aware that the banked text functionality provided by TS entries may not be available throughout a LAMPS system. Multi-banked text editing is a licensed option in the LAMPS LITES2 graphical editor. For details of LAMPS licensing arrangements please refer to Laser-Scan.

TX **Text entry**

FORMAT

TX "text string"

where:

text-string - is a text string

DESCRIPTION

The TX entry holds the text for a text feature. It should be the last entry before the EF entry.

The maximum number of characters in a TX is 255.

ZS 3 dimensional point string entry

FORMAT

```
ZS  npts  pen
    x-coordinate y-coordinate z-coordinate
      .          .          .
      .          .          .
      .          .          .
```

where:

npts - the number of coordinates in the ZS entry
pen - pen down/pen up control flag

DESCRIPTION

The ZS entry contains the three-dimensional point data defining the feature. There may be more than one ZS entry in a feature:

It is illegal to have both ST and ZS entries within the same IFF feature. ZS entries only occur in IFF output revision level 0 files. In a revision level 1 file, the same effect is achieved by a CB entry with at least X, Y, and Z columns.

1. Each ZS should contain no more than 200 points, so a new ZS with the pen flag set to 1 (ie keep pen down when drawing to the start of this new ZS) will be started for the remaining points.
2. If an invisible line is to be coded in the feature, then this is represented by starting a new ZS with the pen flag set to 0 (ie keep pen up when moving to the start of this new ZS).

A ZS entry is used when the data to be held is inherently three-dimensional - for instance in digitising a river bed, where the river flows downhill. For data at a constant height, an ST with a contour height AC is more appropriate.

NOTES

Note that the pen flag is always considered to be 0 for the first ZS in a feature - that is the pen is always kept up to move to the start of a new feature.

The user should be aware that 3D string manipulation may not be available throughout a LAMPS system. It is supported in the Basic LAMPS packages of MAPPING, IMP, LITES2, and PLOTTING, but other packages may not support ZS entries.

CHAPTER 8

MISCELLANEOUS AND OBSOLETE ENTRIES

MISCELLANEOUS AND OBSOLETE ENTRIES

General

IFF format has been extended and revised through time. New IFF entry types have been added and others have fallen from popular usage. This section describes those IFF entry types which do not fall neatly within the hierarchical structure of IFF format or which are no longer in general usage.

CH Literal CHaracter entry

FORMAT

CH "text"

DESCRIPTION

This entry is obsolete, and should not be used in new IFF files.

Its only current use is for storing the data in Laseraid patch files.

Note that CH entries must be outside features, and may also be outside layers (Laseraid patch files do not contain layers or features).

NOTES

In the event of a Laseraid patch file being improperly closed it is essential that the /NOTRUNCATE qualifier is specified on the IMEND utility command line. Although an IFF file (containing multiple CH, CC and CP entries), a Laseraid patch file has no map, layer or feature level entries. Without the /NOTRUNCATE qualifier IMEND will truncate the file at the very beginning thus losing all the file contents.

CS Character Size

FORMAT

CS height spacing

Where:

height - is the character height

spacing- is the character spacing

Example

CS 20 30

DESCRIPTION

This entry is obsolete, and should not be used in new IFF files.

SL Select Symbol Library entry

FORMAT

SL symbol-library

Where:

symbol-library is the numeric identifier of a symbol library.

Example

SL 5

DESCRIPTION

This entry is obsolete, and should not be used in new IFF files.

SS Symbol Select entry

FORMAT

SS symbol-selection

Where:

symbol-selection is the numeric identifier of a symbol within a symbol library

Example

SS 12

DESCRIPTION

This entry is obsolete, and should not be used in new IFF files.

TC Transmitted Comment

FORMAT

TC "text"

Where:

"text" is a text string

Example

TC "Remaining south support pier anchor pins"

DESCRIPTION

A TC entry is used to label the following feature. It is used by MCE to hold specialised plotting instructions.

Interpretation of a TC requires the IFF file to be regarded as a sequential file, since the TC applies either to the following feature, or to the layer as a whole (in which case it appears after the last feature - in old format IFF files before a special empty feature with FSN 0).

The TC entry is not allowed inside a feature, but must be inside a layer.

The maximum number of characters in a TC is 255.

VO VOid

FORMAT

VO nwords

Where:

nwords is the size of the void in words

DESCRIPTION

The VO entry is used to replace a series of deleted IFF entries. Since it is not possible to 'compress' an IFF file, a deleted entry or series of entries is overwritten with a void area of the requisite size.

XX Unknown entry type

FORMAT

XX

DESCRIPTION

The Laser-Scan IFF library (IFFLIB) returns the entry mnemonic XX when it has found an entry that it cannot identify as a valid IFF entry. This generally occurs because either the IFF file is corrupt in some manner, or because obsolete IFF utilities are being used on a "modern" IFF file containing new IFF entry types.

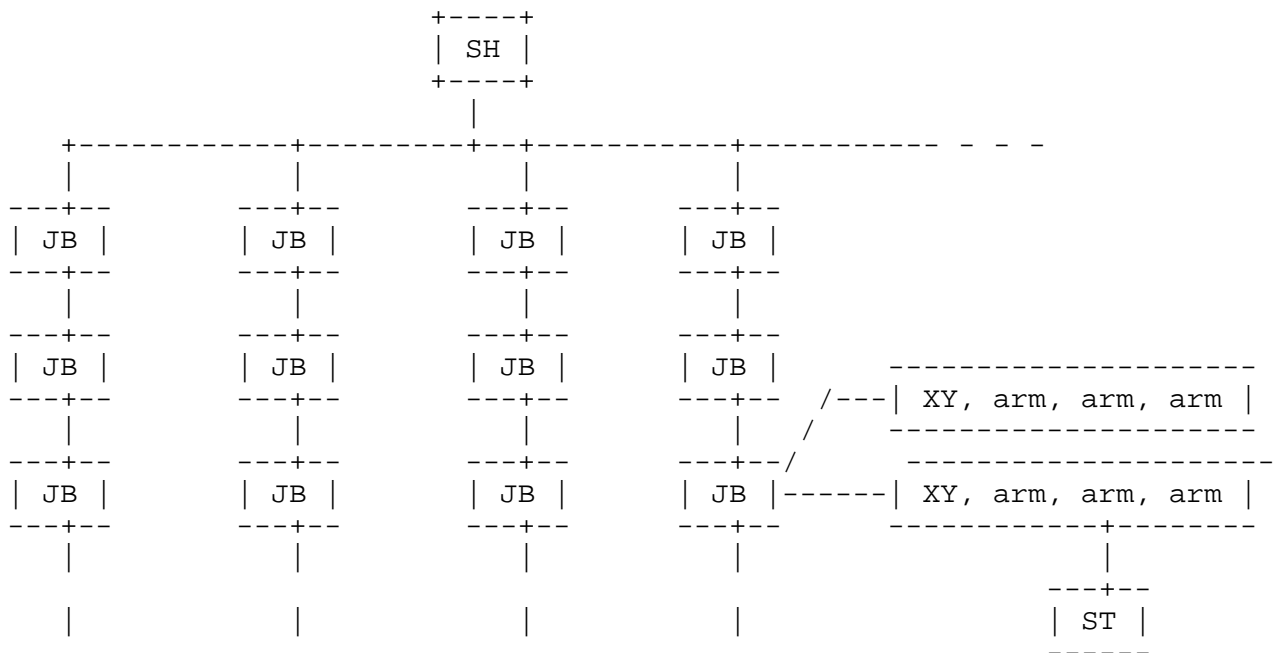
CHAPTER 9

IFF JUNCTION STRUCTURE

General

At present junction structure is available for use only with 2-dimensional coordinate strings held in ST (SString) entries. Junction structure for 3-dimensional strings held in ZS entries is not supported.

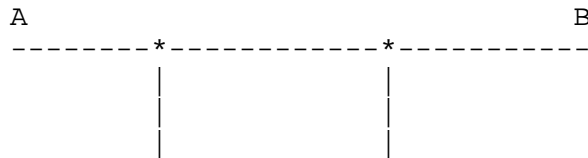
The sector concept may be visualised by means of the following diagram:



The ST pointers within the junction block enable the junction arms to be easily related back to the coordinate strings which form them (and also, although much less easily, to the features which contain the

arms). In order to refer to the junction data from within a feature, the Junction Pointer (JP) entry is provided (see below). Note that although the position of the JP entry within a feature is not formally defined in IFF, in order that processing programs should not need to be too complex a logical ordering should be adopted with JPs between STs as appropriate.

For example, the geometrical arrangement:



should have the horizontal feature AB represented by the IFF sequence:

```
NF FS ... ST JP ST JP ST EF
```

rather than, for example:

```
NF FS ... ST ST ST JP JP EF
```

Similarly, although JB entries may in theory occur anywhere in the IFF file, to accord with current practice they should not appear within features. This means that when the feature is created, all junctions implicated in that feature must be remembered until the EF has been generated.

It should be noted that this scheme preserves the integrity of the coordinate data and the file as a whole. The junction structure can be stripped out at any time and the resulting file will be geometrically complete; only the relationship between the junction arms will be lost.

JB Junction Block

FORMAT

```

JB  nsect next-jb
offset  narms  x-coordinate y-coordinate
      armno    pntno  string-address
      .        .        .        .
      .        .        .        .
      .        .        .        .
  
```

Where:

nsect	- is the sector number as defined within the SH entry
next-jb	- the pointer to the next JB in the sector (zero if none)
offset	- offset of the junction within the JB
narms	- number of arms at junction
x-coordinate	- the X coordinate of the junction
y-coordinate	- the Y coordinate of the junction
armno	- the arm ident
pntno	- string vertex number which lies at the junction
string-address	- address of the ST entry forming the arm

Example

```

JB  5 00000000
0005:  3 52838.900 159.628
      1:  1 000005D2
      2: 17 00000956
      3: 37 00000C7E
  
```

DESCRIPTION

The JB entry defines a series of junctions.

Within each sector of the IFF file, a chain of JB entries is maintained to hold the details of all junctions within that sector.

Each JB entry contains the number of the sector that it is in, a pointer to the next JB entry in this JB chain, and details of each junction.

For each junction, the JB contains

- o the number of arms at that junction
- o the X,Y position of the junction
- o the address of each ST corresponding to a junction arm
- o the number of the vertex (either 1 or the index of the last point in the ST) corresponding to the junction in that ST

See also the JP (Junction Pointer) and SH (Sector Header) entries.

JP Junction Pointer

FORMAT

JP jb-address jb-offset

Where:

jb-address - address of the JB entry describing this junction

jb-offset - start position of the data relating to present
 junction within the JB entry

Example

JP 0000056D 0005

DESCRIPTION

The JP entry is a pointer back to a JB entry. A JP entry is inserted before or after (as appropriate) each ST that starts or ends at a junction.

The point in the ST corresponding to the junction is either the first (in which case the JP occurs before the ST) or the last (the JP occurs after the ST).

Each JP entry contains the address of the relevant JB entry, and the sequence number of the junction within that JB.

SH Sector Header

FORMAT

The SH entry is from necessity a complex entry. It contains information defining the position and size of the sectors into which the data are divided. In addition it references information which is not directly accessible to the user (other than by IFF library common blocks). The contents of a typical SH entry is thus best explained by use of an example, using the format produced by the IMP utility IPATCH:

Example

SH
Sector Header entry

- size of entry = 200 words
- there are 100 sectors, 10 in X and 10 in Y
- sector origin: .5038E 005, .1558E 005
- sector size: 581.6, 459.4
- JB entry chains start at the following addresses:

00000000	00000000	00000000	00000000	00000568	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	0000031C	00000737	00000A15
00000B5E	00000DE5	00000000	00000000	00000000	00000000	00000F36	0000041B
00000836	000017EA	00001B80	00001F87	00001E56	00000000	00000000	000010CD
000024E0	00001350	00001603	000019E3	00001CF5	00000000	00003903	00000000
0000214A	000022B3	0000266B	00002858	00002AF9	000030F4	000032F5	0000362E
00000000	00000000	00000000	00004144	00003FF7	000043AB	0000490A	00004B7F
00004FE8	00003B28	00006F43	00000000	00000000	00003CC5	00005545	00005970
00005BD1	000061F0	000064F9	00006AFD	00006D1E	00000000	00003E86	0000708C
00007418	0000764F	0000602F	00005E84	00006798	000083F4	0000861B	00000000
00000000	00000000	00007261	00000000	00007B72	00007DE5	00007FA8	0000829F
00008B97	00000000	00000000	00000000	00000000	00008888	00000000	00000000
00000000	00000000	00000000	00000000				

Each hexadecimal value is the address of the JB entry for a sector within the file. JB entries may be 'chained'. Thus if a sector requires more than one JB entry then the address of the second and any subsequent JB entries for that sector will be held within the previous JB.

DESCRIPTION

The Sector Header entry is always placed immediately **after** the HI (HIstory) entry in the IFF file. If the file does not have an HI entry then the SH entry will come immediately after the RA (RAnge) entry.

The SH entry contains the addresses of the start of the JB (Junction Block) chain for each sector in the IFF file.

When an IFF file contains junction structure, the map is divided into rectangular sectors. A chain of JB entries is maintained for each sector, containing the junction information. The SH entry provides the address of the start of each of these chains.

NOTES

The origin of the sectors and the area covered by them need not reflect the coordinate range of the file expressed in the RA (Range) entry.

On many maps not all sectors will contain junctions. There will therefore be no JB entry start address for such sectors.

APPENDIX A

THE SI AND SD COMMANDS

The SI and SD commands

INTRODUCTION

IFF files are referenced by all the IMP modules using logical name LSL\$IF. Logical name LSL\$IF is assigned to a device and directory specification either using the VMS DEFINE command or the Laser-Scan SI utility. As a component of LAMPS (Laser-Scan Automated Map Production System) the Laser-Scan package "MAPPING" is supplied with all LAMPS component packages. Part of the MAPPING package is the SD (Set Default) and SI (Set IFF) utilities.

The SI command is not mandatory but is advisable for effective IFF file handling. Failure to use the 'SI' command prior to IFF file processing will result in the IFF files being accessed from (and output directed to) a default directory with logical name LSL\$IF. Within an automated map production environment it is often advantageous to put IFF files which represent different stages of development within a production flowline into different directories on magnetic disk. The SI utility enables the user to rapidly switch the device and directory defined by the logical name LSL\$IF using an easy to learn shorthand notation.

Users should also note a complementary Laser-Scan command 'SD' which directs VMS to the user's default directory on which can be run standard VMS commands. SD maintains logical name HERE for the current directory. Typical use of the SI and SD utilities may take the form:

Basic Use

The basic form of an SI or an SD command is shown in these two examples:

SI - assign logical name LSL\$IF to a device and directory
e.g. \$ SI DRA2:[BUREAU.COASTLINE]

SD - set default to a device directory
e.g. \$ SD DRA2:[BUREAU.COASTLINE]

The user will soon wish to abbreviate even further these commands. SI and SD offer a flexible and comprehensive shorthand notation for setting default directory and LSL\$IF assignments.

Advanced Use

There are a number of extensions to the SI and SD commands in which abbreviations can reduce considerably the amount of typing required. The list below with 'SD' and 'SI' compares the command syntax necessary with standard VMS equivalents.

SI Example	Equivalent to
SI	Show default assignment of LSL\$IF (also SI @ and SI ?)
SI TST	DEFINE LSL\$IF [TST]
SI .IFF.BILL	DEFINE LSL\$IF [.IFF.BILL]
SI DRA2:JOHN.TST	DEFINE LSL\$IF DRA2:[JOHN.TST]
SI ^	DEFINE LSL\$IF [-] (move up to parent directory)
SI ^^^.JOHN	DEFINE LSL\$IF [---.JOHN]
SI *	select your login directory
SI *.TST.BILL	select one of your sub-directories
SI *:TST	select login disc and directory [TST]
SI.	re-select the previous IFF directory (LSL\$OLDIF) This can be repeated ad nauseam (also SI <)
SI DRA1:	select same IFF directory name on device DRA1:
SI logical-name	select directory defined by given logical name
SI [TST]	same as SI TST
SD Example	Equivalent to
SD	SHOW DEFAULT (also SD @ and SD ?)
SD TST	SET DEF [TST]
SD .IFF.BILL	SET DEF [.IFF.BILL]
SD DRA2:JOHN.TST	SET DEF DRA2:[JOHN.TST]
SD ^	SET DEF [-] (move up to parent directory)
SD ^^^.JOHN	SET DEF [---.JOHN]
SD *	select your login directory
SD *.TST.BILL	select one of your sub-directories
SD *:TST	select login disc and directory [TST]
SD.	re-select the previous current directory (THERE) This can be repeated ad nauseam (also SD <)
SD DRA1:	select same current directory name on device DRA1:
SD logical-name	select directory defined by given logical name
SD [TST]	same as SD IFF

Users are referred to the VAX/VMS documentation for details of the SET DEFAULT command.

INDEX

- AC entry, 7-5
 - definition of types, 7-5
 - maximum text length, 7-6
 - type value range, 7-6
 - types of data storage, 7-5
- ACD, 7-5, 7-8
- CB entry, 1-3 to 1-4, 7-8
 - absent value, 7-8
 - definition of types, 7-8
 - types of data storage, 7-8
- CC entry, 5-2 to 5-3
- CH entry, 8-2
- Coordinate origin offset, 1-4
- Coordinate range, 3-1
- CP entry, 5-2, 5-5
- CS entry, 8-3
- 3D strings, 1-4
- DAMP, 4-5
 - IED, 4-5
 - IPR, 4-5
- DFAD, 6-4
- EF entry, 7-10
- EJ entry, 3-2, 4-2
- empty features, 7-3
- EO entry, 6-3
 - and section level entries, 6-3
 - historical legacy, 6-4
 - position in IFF file, 6-3
- FC, 7-12
- Feature Code, 7-12
- Feature level entries, 7-1
- File level entries, 3-1
- File level entry order, 3-1
- FPP, 6-1
- FRT
 - ACD, 7-5, 7-8
- FS entry, 7-11
 - definition of fields, 7-11
 - feature code, 7-11
 - historical IT code, 7-11
 - historical types, 7-11
 - pc/text word, 7-11
 - status word, 7-11
 - user word, 7-11
- GC, 7-12
- Graphical Code, 7-12
- HI entry, 1-2, 3-1, 3-3, 4-7, 9-6
- IF:, 1-1
- IFF
 - and 3D strings, 1-4
 - and origin offset, 1-4
 - and VMS environment, 1-1
 - definition of acronym, 1-1
 - revision levels, 1-3, 7-8, 7-17, 7-22
- IFF and multiple maps, 4-1
- IFF entries
 - AC, 7-5
 - CB, 1-3 to 1-4, 7-8
 - CC, 5-2 to 5-3
 - CH, 8-2
 - CP, 5-2, 5-5
 - CS, 8-3
 - EF, 7-10
 - EJ, 3-2
 - EO, 6-3
 - and section level entries, 6-3
 - historical legacy, 6-4
 - position in IFF file, 6-3
 - FS, 7-11
 - HI, 1-2, 3-3, 4-7, 9-6
 - JB, 9-1, 9-3, 9-5, 9-7
 - JP, 9-1, 9-3, 9-5
 - MD, 1-2, 4-3
 - and origin offset, 4-5
 - and projection data, 4-5
 - position in file, 4-5
 - type 1, 4-3
 - type 2, 4-4
 - MH, 4-2, 4-6
 - NF, 7-14
 - NO, 6-4
 - correction of bad EO address, 6-5
 - relationship to EO entry, 6-4
 - significance of layer 0, 6-4
 - NS, 5-2, 5-7
 - RA, 3-5, 9-6 to 9-7
 - RO, 7-16
 - SH, 9-1, 9-3, 9-6
 - SL, 8-4
 - SS, 8-5
 - ST, 7-17, 9-1, 9-5
 - TC, 8-6
 - TH, 7-18

- TS, 1-2, 7-19
- TX, 7-21
- VO, 8-7
- XX, 8-8
- ZS, 1-2, 1-4, 7-22, 9-1
- IFF features, 7-1
 - and ancillary codes, 7-1
 - and RA entry, 7-4
 - and TCs, 7-3
 - and text, 7-2
 - and text components, 7-2
 - as building blocks, 7-1
 - definition, 7-1
 - empty, 7-3
 - historical TC usage, 7-3
 - maximum number in file, 7-4
 - minimum structure, 7-1
 - multi-banked texts, 7-2
 - text and ancillary codes, 7-2
 - text rotation, 7-2
 - text size, 7-2
- IFF file structure
 - overview, 2-1
- IFF junction structure, 9-1
- IFF Reference manual, 1-1
- IFF revision levels, 1-3, 7-8, 7-17, 7-22
- IFFLIB, 3-3
- IFROMTEXT, 6-5
- IINFO, 3-3
- IMEND, 3-3, 8-2
- IMERGE, 5-1
- IMP, 1-1
 - and multiple maps in IFF files, 4-1
- IFROMTEXT, 4-4, 6-5
- IINFO, 3-3, 4-4
- IMEND, 3-3, 8-2
- IMERGE, 5-1
- IPATCH, 3-5, 4-4 to 4-7, 6-5, 9-6
- ITOTEXT, 1-1, 4-4, 6-5
- ITRANS, 4-5
- IPATCH, 3-5, 6-5, 9-6
- ITOTEXT, 6-5
- JB entry, 9-3, 9-5, 9-7
- JP entry, 9-3, 9-5
- Junction structure, 9-1
 - availability with 3D strings, 9-1
 - general, 9-1
 - IFF feature layout, 9-2
 - sectors, 9-1
 - SH entry, 9-1
- LAMPS, 1-1 to 1-2, 5-1, 6-1, 7-11
- LAPROCESS, 5-1
- Laseraid, 4-1, 5-1, 8-2
- Lasertrak, 5-3
- Layer, 6-1
- Layer 0, 6-1, 6-4
- Layer 11, 6-4
- Layer 32, 6-4
- Layer level entries, 6-1
 - general, 6-1
- Layer level entry order, 6-2
- Layers
 - reserved, 6-1
 - value range, 6-2
- LITES, 4-1, 6-1, 7-3
- LITES2, 4-1, 5-1, 6-1, 6-5, 7-4
- LSL\$IF:, 1-1
- LSL\$IFF_OUTPUT_REVISION:, 1-3
- Map level entries, 4-1
 - general, 4-1
- MCE, 8-6
 - and TC entries, 8-6
- MCE DFAD, 6-4
- MCEHED, 4-7
- MD entry, 1-2, 4-3
 - and origin offset, 4-5
 - and projection data, 4-5
 - position in file, 4-5
 - type 1, 4-3
 - type 2, 4-4
- MH entry, 4-2, 4-6
- MHED, 4-7
- Miscellaneous and obsolete entries, 8-1
 - CH, 8-2
 - CS, 8-3
 - general, 8-1
 - SL, 8-4
 - SS, 8-5
 - TC, 8-6
 - VO, 8-7
 - XX, 8-8
- Multiple maps in IFF, 4-1
- NF entry, 7-14
- NO entry, 6-4
 - correction of bad EO address, 6-5
 - relationship to EO entry, 6-4
 - significance of layer 0, 6-4
- NS entry, 5-2, 5-7
- Obsolete entries, 8-1
- Origin offset, 1-4

Overlay
 see layer

Patch files, 4-1, 8-2

RA entry, 3-5, 9-6 to 9-7
 coordinates used, 3-5
 position in file, 3-5

Reserved layers, 6-1

RO entry, 7-16

Section level entries, 5-1
 general, 5-1
 warning, 5-1

Section level entry order, 5-2

SH entry, 9-3, 9-6

SI, 1-1

SL entry, 8-4

SOL, 7-3

SPM, 6-1

SS entry, 8-5

ST entry, 7-17, 9-5

Structure of IFF file, 2-1

TC entry, 8-6
 size of text field, 8-6

Text Position, 7-12

TH entry, 7-18

TS entry, 1-2, 7-19

TX entry, 7-21

VMS, 1-1
 DEFINE, 1-1

VMS DUMP, 3-3

VO entry, 8-7

XX entry, 8-8

ZS entry, 1-2, 1-4, 7-22