Laser-Scan Ltd.

CONVERT PACKAGE

IFFSIF Reference

Issue 1.3 - 30-May-1995

Science Park, Milton Road, Cambridge, England CB4 4FY tel: (01223) 420414

Document "IFFSIF REFERENCE", Category "REFERENCE"
Document Issue 1.3   J Cadogan (modified 30-May-1995)

CONTENTS

CHAPTER 3 IFF2SIF UTILITY

CHAPTER 4 SIF2IFF UTILITY

IFFSIF - Change Record

Version 1.0   S Townrow 17-June-1991

Module IFFSIF - Reorganised package documentation.

Version 1.1   J Cadogan 21-August-1991

Module IFFSIF - SIF2IFF /LOG qualifier reinstated and the /UNITS qualifier now has a default value of 1000.

Version 1.2   J Cadogan 26-May-1994

Module IFFSIF - The Attributes section of the SIF data preparation is updated to state the limit for the number of ACs in a feature that can be translated in IFF2SIF.

Module IFFSIF - New warning message MAXAC for IFF2SIF to indicate that the maximum number of ACs in a feature that can be translated has been exceeded.

Version 1.3   J Cadogan 30-May-1995

Module IFFSIF - The SIF2IFF section of the SIF data preparation has been updated describe the use of the LOGICAL name LSL$SIF2IFF_MAX_PATTERN.

PREFACE

Intended audience

This manual is intended for users of a specific utility of the Laser-Scan CONVERT package running under the VAX/VMS operating system. Each manual contains the documentation for a particular CONVERT utility and a site will only receive new or updated documentation for those utilities which they have purchased.

Structure of this document

This document is composed of 2 major sections.

The Introduction is an overview of the CONVERT package and its purpose.

There then follow the User Guides for the individual modules which comprise CONVERT. Each individual module contains the same basic categories of information. These are:

MODULE - the name of the CONVERT module.

FORMAT DESCRIPTION - a description of the data format written or read by the utility programs in this conversion module.

DATA PREPARATION - guidance on how to digitise or prepare the IFF and other data required by the utility programs in this module.

For each utility program in the module, there will then be the following categories:

UTILITY - the name of the utility.

FUNCTION - a synopsis of what the utility does.

FORMAT - a summary of the utility command format and command qualifiers. Default qualifier settings are indicated.

PROMPT - how it prompts the user.

PARAMETERS - description of expected command parameters.

COMMAND QUALIFIERS - description of all command qualifiers. Qualifiers are ordered alphabetically and default argument values are indicated.

RESTRICTIONS - a summary of restrictions on the use of qualifiers

DESCRIPTION - the definitive description of the utility action.

COMMANDS - for interactive utilities only, a description of all commands. Commands are ordered alphabetically and default argument values are indicated.

EXAMPLES - annotated examples of utility useage.

MESSAGES - all classes of message are listed and described and suggested user action given. The messages are divided into sections according to message severity within which the messages are ordered alphabetically by message mnemonic.

Conventions used in this document

Convention	Meaning
<CR>	The user should press the carriage control 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>.
$ **IFF2SIF<CR>**	Command examples show all user entered commands in **bold** type.
$ **IFF2SIF<CR>** . . .	Vertical series of periods, or ellipsis, mean either that not all the data that CONVERT 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 elipsis 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 assisnment statement).

Convention	Meaning
'integer'	An integer number is expected in the specified input or output field. (See "Command line data types" below).
'real'	A real number is expected in the specified input or output field. (See "Command line data types" below).
'file-spec'	A VMS file specification is expected in the specified input or output field.
'device-name'	A VMS device specification (for instance, MTA0:) is expected in the specified input or output field.

CHAPTER 1

SIF FORMAT DESCRIPTION

FORMAT DESCRIPTION

Introduction

IFF2SIF provides a translation from Laser-Scan's Internal Feature Format (IFF) to a subset of Intergraph's Standard Interchange Format (SIF). For a brief comparison of the way IFF and SIF regard data, see the section on DATA PREPARATION - IFF2SIF below.

SIF2IFF provides the reverse translation.

IFF2SIF produces the ASCII text format, version 8.8.2, as described in the document **Standard Interchange Format (SIF) Command Language Implementation Guide (8.8.2)** of 12-May-1985 (Intergraph document DIXD4110).

If the /SCITEX qualifier of IFF2SIF is given, the format produced is the SCITEX version of SIF.

SIF2IFF receives the ASCII text format, version 8.8.2 and earlier.

Magnetic tape input/output

IFF2SIF writes to magnetic tape in standard SIF ASCII format. The output is at either 800 or 1600 bpi, in blocks of 1024 bytes. The record size defaults to 72 bytes, but may be specified at any size between 40 and 80 bytes (inclusive). Records are padded with spaces if necessary.

There are no header blocks, and the last block of each file is padded with spaces. Each file is separated from the previous by a tapemark, and there are two tapemarks after the last file. Files on the tape are synonymous with the original IFF files output.

SIF2IFF can read standard SIF ASCII format from tape. For users from CCG (Canadian Centre for Geomatics) SIF2IFF will read and ignore a CELL file preceding the SIF file if the /CELL qualifier given.

Disk file output

IFF2SIF output may also be made to disk file. Again, the record length defaults to 72 bytes, but may be changed to anywhere between 40 and 80 bytes (one character per byte). Records are filled with spaces if necessary.

Each disk file contains the output from one IFF file.

These files may then be output to magnetic tape using either of the VAX programs BACKUP or EXCHANGE - the former is intended for transfer between VAXes, and the latter provides an ANSI standard tape format (it replaces the RSX utility FLX on VAXes with VMS version 4.0 and up). Consult the relevant Digital manuals for more details of these utilities.

SIF2IFF can read standard SIF ASCII format from disk as well as tape, and writes the IFF file to disk.

The SIF subset supported - IFF2SIF

The data is output using only a subset of SIF commands. This section describes what may be expected in the output.

Summary of commands output

The following SIF commands are output:

```
      DID/NA=name,DA=date,MO=mode,
          RA=xmin,xmax,ymin,ymax,ident          drawing identification
      OVR/level                               specify overlay/level
      ACZ/LO=height,HI=height                 specify height value (Z)
      LAC/LS=style                            specify line style
      LAC/LT=weight                           specify line weight
      LAC/LC=colour                           specify line colour
      FNT/font                                specify text font
      TLC/CO=length,JU=just                   specify text length & justification
      BST/OP                                  begin complex string
      EST/                                    end complex string
      LST/OP,x1,y1,x2,y2,.._.                 line string - open
      LST/SO,x1,y1,x2,y2,.._.                 line string - solid (closed)
      LST/HO,x1,y1,x2,y2,.._.                 line string - hole  (closed)
      CIR/CE=x,y,P1=x1,y1,P2=x2,y2            circle
      ARC/CC,CE=x,y,P1=x1,y1,P2=x2,y2         arc, counterclockwise
      ARC/CL,CE=x,y,P1=x1,y1,P2=x2,y2         arc, clockwise
      ARC/P1=x1,y1,P2=x2,y2,P3=x3,y3          arc, circumcircle
      MTX/AN=rot,XS=xscale,YS=yscale          symbol cell transformation matrix
      SYM/OR=xo,yo,CELLNAME                   symbol cell
      CUR/x1,y1,x2,y2,...                     curved line (draw time
interpolation)
      TX3/OR=x,y,AN=rot,TH=ht,TW=wd           text line range block - 2d
      TXT/text                                text line - 2d
      TXT/OR=x,y,TH=ht,TW=wd,MA=matrix,text   text line - 3d
      IDE/AS=assoc,CO=class,ID=ident,KE=key
      ASV/DE=delim,ID=pent,KE=pocc,values
```

The following variations of SIF commands apply when the /SCITEX qualifer of IFF2SIF is specified:

```
      DID/NA=name,DA=date,MO=mode,
          RA=xmin,xmax,ymin,ymax,DU=munit,
          sunit,runit,munitid,sunitid,ident   drawing identification
      TXT/BL=x1,y1,BR=x2,y2,TL=x3,y3,
          MA=matrix,text                      text line - 3d
      SYM/OR=xo,yo,MA=matrix,CELLNAME         symbol cell
```

The following SIF commands are omitted when the /SCITEX qualifer of IFF2SIF is specified:

```
      MTX
      TX3
      LAC/LC
      IDE
      ASV
```

For lines and curves, continuation lines (flagged by 4 spaces at the beginning of the line) are also output.

The SIF subset supported - SIF2IFF

Not all of the SIF command set is applicable to the IFF format so SIF2IFF translates the following subset:-

DID LAC LST SYM TXT FNT IDE CIR ARC CUR OVR

EST PTN TPC PAR PLN CLP TX3 ACZ MTX PA3

Details of IFF2SIF command output

File Characteristics

Coordinate Format

Coordinate data is output in Units of Resolution (UORs). The size of one UOR is specified by the customer, preferably in metres on the ground.

All coordinates are output in unsigned 32 bit format - that is they fall within the range 0 to 4294967295. Coordinates are output in the minimum width possible.

Characters per record

By default, 72 characters are output per record. If requested, record sizes of 40 to 80 characters may be produced. The record size may not change within a single tape.

Records are padded with trailing spaces if necessary.

Graphic Mode

The files produced may be 2-d or 3-d files. The graphic mode used is stated in the DID command within the SIF file.

Spaces in the file

Spaces are not output between fields in commands, and are thus only present as

1. part of an ASCII string (text line or symbol cell name)

2. padding of a record or block

3. continuation characters at the beginning of a line

Non-repetition of commands

The OVR, ACZ and LAC commands (which set active characteristics) are only output when necessary to change a characteristic. Thus if a series of elements are in level 3, the OVR/3 command will be output before the first element, and not repeated before the rest.

Details of output

This section describes the commands output in more detail

Continuation lines

These are used for coordinate output (eg in LST), and may also be used to continue a command when the record length is such that the command must be split over several records.

Continuation lines are signalled by 4 spaces at the beginning of the record, followed by the continuing data. The following sequences are never split over record boundaries:

1. <SIF command> / <2 character data keyword>

2. <2 character data keyword> = <single data value>

3. <2 character data keyword> = <coordinate pair/triple>

Also, a field separation comma is never output at the start of a continuation record.

If a text string must be continued, then the text string is always split such that the split occurs after a non-space character. If the split would occur at a space character, then the text is split at the previous character.

MTX - Generate symbol cell transformation matrix

No mirroring is provided, and scaling is always the same in X and Y. The command is thus always output as

MTX/AN=rot,XS=scale,scale

where **rot** is the angle of alignment in degrees, and **scale** is the scale factor for the symbol cell - this defaults to 1.0. These are all output as positive floating point numbers.

If the /SCITEX qualifier of IFF2SIF is specified, the MTX command is omitted from the output file.

TX3 - Generate text line range block

The command is output as

TX3/OR=xo,yo,AN=rot,TH=height,TW=width

where **xo,yo** is the origin of the text (its position) in UORs, **rot** is the angle at which the text is placed in degrees, and is a positive floating point number, and **height** and **width** are the values (in UORs) defined for the relevant font.

If the /SCITEX qualifier of IFF2SIF is specified, the TX3 command is omitted from the output file.

This command is not used in 3-d files.

OVR - Define the active level

The command is output as

OVR/level

where **level** is an integer between 1 and 63. Level 63 is used as the error level for certain types of IFF2SIF error - unless otherwise stated, data on this level was probably improperly coded.

ACZ - Define the active Z value

The command is output as

ACZ/LO=height,HI=height

where **height** is the Z value for all following coordinates. The command

ACZ/

is used to unset the active Z level, when elements follow which do not have a defined Z value.

This command is not used in 3-d files.

LAC - Define the active line/area characteristics

The following commands are used

LAC/LS=style
LAC/LT=weight
LAC/LC=colour

where **style** is the line style, range 0 to 7, **weight** is the line weight, range 0 to 31, and **colour** is the line colour (or other coding), range 0 to 255. All of these arguments are integers.

If the /SCITEX qualifier of IFF2SIF is specified, the LAC/LC command is omitted from the output file.

FNT - Define the active text font

The command is output as

FNT/font

where **font** is an integer between 0 and 126.

TLC - Define text line characteristics

The command is output as

TLC/CO=length,JU=just

where **length** is the number of characters in the text line, between 1 and 255 (although the maximum is likely to be 80), and **just** is the text line justification, between 1 and 9, conventions as specified in the Intergraph SIF documentation. Both arguments are integers.

LST - Generate line string

The commands supported are

```
LST/OP,x1,y1,x2,y2,.....,xm,ym
LST/SO,x1,y1,x2,y2,.....,xm,ym
LST/HO,x1,y1,x2,y2,.....,xm,ym
```

where **x1,y1**, etc, are the coordinates in UORs (as integers - see "Coordinate Format" above). Any of these commands may be continued by continuation lines - for instance

```
LST/OP,x1,y1,x2,y2,.....,xm,ym,
    xn,yn,xo,yo,xp,yp,...
```

A single line string command will never contain more than 101 points. Line strings of length greater than 101 points are output as separate LST commands within a complex string - see the BST command.

For 3-dimensional data, each **xi,yi** becomes **xi,yi,zi**

CIR - Generate circle

This command is output as

CIR/OR=xc,yc,P1=x1,y1,P2=x2,y2

where **xc,yc** is the origin of the circle in UORs, and **x1,y1** and **x2,y2** are the coordinates of two points on the circumference of the circle.

For 3-dimensional data, each **xi,yi** becomes **xi,yi,zi**

ARC - Generate arc

This command may be output in one of three possible forms

```
ARC/CC,CE=xc,yc,P1=x1,y1,P2=x2,y2
ARC/CL,CE=xc,yc,P1=x1,y1,P2=x2,y2
ARC/P1=x1,y1,P2=x2,y2,P3=x3,y3
```

where the first is a counterclockwise arc, the second a clockwise arc, and the third a circumcircle arc.

For the first two forms, **xc,yc** is the centre of the arc, **x1,y1** the point defining its start, and **x2,y2** that defining its end. For the third form, **x1,y1** is the point defining the start of the arc, **x3,y3** the point defining its end, and **x2,y2** a point on the arc between these two. All of these coordinates are output in UORs.

For 3-dimensional data, each **xi,yi** becomes **xi,yi,zi**

SYM - Generate symbol cell

This command is output as

```
MTX/AN=rot,XS=1.0,1.0
SYM/OR=xo,yo,CELLNAME
```

where the MTX command is described above, **xo,yo** is the symbol origin, and **CELLNAME** is the name of the required symbol cell.

For 3-dimensional data, **xo,yo** becomes **xo,yo,zo**

Alternatively if the /SCITEX qualifier is specified this command is output as

```
SYM/OR=xo,yo,MA=matrix,CELLNAME
```

where **xo,yo** is the symbol origin, **matrix** is the transformation matrix and **CELLNAME** is the name of the required symbol cell.

SCITEX does not support 3D symbols.

CUR - Generate curve

If used, this command is output as

```
CUR/x1,y1,x2,y3,...,xm,ym
```

where **x1,y1**, etc, are the master points defining the curve. Continuation lines may be used as for the LST command, and curves of greater than 101 points are treated as complex strings in the same way as for the LST command.

For 3-dimensional data, each **xi,yi** becomes **xi,yi,zi**

Note that in general, all curves will be output as line strings using LST. Laser-Scan can produce curve strings using master points, but normally interpolate the curve at digitise time. This ensures that an acceptable curve is produced, and is visible on check plots, regardless of the interpolation being used.

TXT - Generate text line

This command is output as either

```
TX3/OR=xo,yo,AN=rot,TH=height,TW=width
TXT/text
```

for 2 dimensional data, or as

```
TXT/OR=xo,yo,zo,TH=height,TW=width,
    MA=matrix,
    text
```

for 3 dimensional data. **xo,yo** and **xo,yo,zo** are the origin of the text (its position) in UORs, **height** and **width** are the values (in UORs) defined for the relevant font, **matrix** is a rotation matrix defining the orientation of the text, and **text** is the text line required.

Alternatively if the /SCITEX qualifier is specified this command is output as

```
TXT/BL=x1,y1,BR=x2,y2,TL=x3,y3,MA=matrix,text
```

where **x1,y1** is the bottom left coord of the text, **x2,y2** is the bottom right and **x3,y3** is the top left, **matrix** is a rotation matrix.

SCITEX does not support 3D text.

BST - Begin complex string

The forms of this command supported are

```
BST/OP
BST/SO
BST/HO
```

which start a complex string, shape and hole respectively. The BST command is followed by a sequence of LST/OP or CUR commands, terminated by an EST command. The subsidiary linear commands are explicitly connected - that is, the last point of one is repeated as the first point of the next.

EST - End complex string

This command is output as

```
EST/
```

and ends a complex string sequence. Every BST command will be matched by an EST command.

DID - Drawing identification

This command is output as

```
DID/NA=name,DA=date,MO=dim,RA=xmin,xmax,ymin,ymax,ident
```

where **name** is the file name (in 1 to 12 characters) - this is the filename used for the original IFF file, uppercased, with any non-alphanumeric characters removed, and truncated to 12 characters. **date** is the current date, in the format

```
ddMMMyy
```

for instance being 18FEB85 for the 18th of February, 1985. **dim** is the dimensionality of the data in the file - either 2 or 3. **xmin,xmax,ymin,ymax** specifies the coord range of the file. **ident** is an identification string, which is currently always the text

Produced by Laser-Scan Ltd., Cambridge, England

Alternatively if the /SCITEX qualifier is specified this command is output as

```
DID/NA=name,DA=date,MO=dim,RA=xmin,xmax,ymin,ymax,
    DU=munit,sunit,runit,munitid,sunitid,
    ident
```

where **munit** is the number of master units, **sunit** is the subunits per master unit, **runit** is unit of resolution, **munitid** is the type of master units and **sunitid** is the type of sub units.

For SCITEX **munit** and **sunit** are always set to 1, and **munitid** and **sunitid** are set to M (metres) and MM (millimetres) respectively.

IDE - Identifier

This command is output as

```
IDE/AS=assoc,CO=class,ID=ident,KE=key
```

where **assoc** is the association type (always 1 in this version to signal new IGDS database entry). **class** is the linkage family class (always 0 for this version). **ident** is the identification number (always the current FSN). **key** is the occurrence number (always 0 in this version).

If the /SCITEX qualifier of IFF2SIF is specified, the IDE command is omitted from the output file.

ASV - Associative Values

This command is output as

ASV/DE=delim,ID=pent,KE=pocc,values

where **delim** is the delimiter character. **pent** is the parent entity number (always the current FSN). **pocc** is the parent occurrence number (always 0 for this version). **values** is the values field which contains attribute names and either values or text separated by the delimiter character.

If the /SCITEX qualifier of IFF2SIF is specified, the ASV command is omitted from the output file.

Typical IFF2SIF output clusters

The following clusters of records should be typical of those output to the SIF file:

* For a text line

```
OVR/level
FNT/font
LAC/LS=style
LAC/LT=weight
LAC/LC=colour
TLC/CO=length,JU=just
TX3/OR=x,y,AN=rot,TH=height,TW=width
TXT/text
```

* For a symbol cell

```
OVR/level
LAC/LS=style
LAC/LT=weight
LAC/LC=colour
MTX/AN=rot,XS=xscale,YS=yscale
SYM/OR=xo,yo,CELLNAME
```

* For an open line string

```
OVR/level
LAC/LS=style
LAC/LT=weight
LAC/LC=colour
LST/OP,x1,y1,x2,y2,....,xm,ym,
    xn,yn,xo,yo,....xz,yz
```

* For a contour of length greater than 101 points

```
OVR/level
ACZ/LO=height,HI=height
LAC/LS=style
LAC/LT=weight
LAC/LC=colour
BST/OP
LST/OP,x1,y1,x2,y2,....,xm,ym,
    xn,yn,xo,yo,....xz,yz
LST/OP,x1a,y1a,x2a,y2a,....,xma,yma,
    xna,yna,xoa,yoa,....xza,yza
EST/
```

CHAPTER 2

SIF DATA PREPARATION

DATA PREPARATION

Comparison of SIF and IFF

IFF is a feature oriented data format - data is separated into features, and each feature represents one 'thing' on the map. An integer feature code is used to say what type of thing it is that the feature represents. A feature might thus be a contour, or a house, or a river, depending on the type of map being digitised, and the way that the data is to be used.

In contrast, SIF has no concept of a 'feature', and the simple form of SIF we produce is essentially just a picture of the map - sequences of 'all following data looks like this' followed by the actual map data.

Of course, line colour, line thickness, text size and so on do describe the data, in the same manner that this is done on an ordinary paper map. Also, on the final Intergraph system, more linkages and non-graphical information can be inserted into the data, if the customer wishes.

Preparation for digitising data to be output to SIF

The customer should supply a list of how the various forms of data on the map(s) are to be output. This should describe what overlay (level) should be used for each data-type, what line weight, style and colour should be used, what text fonts and sizes are required, what symbols (cells) are to be used, and how to digitise them, and so on.

For digitising within IFF, feature codes must be assigned to all of these data-types, and an FRT file (together with SRI and TRI files) created.

Care should be taken in considering what graphical type the IFF features should have - see the section 'Comparison of SDF and FRT files' below.

In some cases, there may not be an obvious linkage between the way the data-type is specified in the customer specification, and the way it would most naturally be digitised. For instance, we might have

Description	Level	Weight	Code	Feature-type
Intermediate contour	1 (for 10,110,210,...) 2 (for 20,120,220,...) 3 (for 30,130,230,...) etc	0	0	LINE

In this case, there are two choices. The first is to treat all intermediate contours as being of one feature code, with a height AC, and process them in some way before outputting to SIF. The second is to provide a separate feature code for each contour height.

The sensible choice (taken in this particular case) is to choose one feature code, and use a small program to post-process the data before outputting to SIF. It is clearly much easier to digitise and edit the data using this approach. The small program is written purely for this application, and just takes the appropriate contour features and allocates a feature code to each according to its height.

Since customer requirements for SIF are likely to vary by quite a lot, care must be taken in designing the flowline for producing the data to be output to SIF, and small programs to process the data before output may often be required.

Three-dimensional and contour data

Three-dimensional SIF output

IFF2SIF is normally used to output two dimensional data, but it can also output SIF data in three dimensional mode. The data output uses true three dimensional coordinates - that is, an X,Y and Z value for each data point.

If three dimensional output is specified, then every IFF feature being output must contain explicit height data. If the feature contains ZS entries, then these will be used directly. If the feature contains ST entries, then a height must be supplied in a type 2 or 3 contour height AC.

Contour data

If there is only a single height associated with each feature, then the data can be output in two dimensional mode (the default). SIF provides an "active Z" (ACZ) command, which is similar in application to an IFF height AC. Thus IFF2SIF in two dimensional mode translates any height ACs found in the IFF file into ACZ commands. In this mode, the Z coordinates of any ZS entries are simply ignored.

Offsetting data

Within the IFF file, coordinates are stored relative to the bottom left hand corner of the sheet, the local origin. This allows more accuracy within the IFF file. In modern IFF files, the coordinates of that local origin are then stored in the map descriptor (MD) entry, and the absolute coordinates of a feature can be obtained by adding the value stored in the IFF file to the value taken from the map descriptor.

The default action of IFF2SIF is to output coordinates relative to the local origin of the sheet, as they are stored in the IFF file. However, in some cases the customer may want the absolute coordinates instead. If the MAPDESCRIPTOR OFFSET command is used, then IFF2SIF will add the local origin to each coordinate to produce the absolute coordinate. Alternatively, the user can specify a particular offset for each sheet, in IFF units, using the OFFSET command.

Z units can also need offsetting - here the most common reason is to make all Z values positive, as IFF2SIF does not output negative height values. The ZOFFSET command can be used for this purpose.

Digitising texts - composite text features

IFF2SIF will output composite text features (text features which contain more than one text component). It outputs such text components as if they were separate text features, using the feature code and descriptive data in the TS entry for each component, instead of the data in the FS entry for the feature as a whole.

Digitising texts - text size

There are three possible ways of handling text sizes. These are:

1. Each text font has a different feature code - no separate size coding

2. Each individual text has a size in points (as for OS text)

3. Each individual text has a size in millimetres on the sheet

Feature code separated text

The first method of handling text sizes is to assign each type or size of text a different feature code, using the SDF text definition. The size of a particular text is then determined by the size given in the SDF feature code definition for that feature code.

Text size in points

The second method requires that the SDF contain definitions of various point sizes. The actual size of a particular text is then determined by looking the point size up in the IFF TH entry. The size defined in the SDF feature code definition for the text is only used if the TH cannot be used.

This method has the advantage of allowing all texts to be digitised with a few feature codes, and is useful if the only difference between the texts is their size.

Text size in millimetres

The third method stores the text size in the IFF TH, in hundredths of a sheet millimetre. The size defined in the SDF feature code definition for the text is only used if the TH cannot be used.

This method is similar to the second method, but allows a very wide range of text sizes to be represented, with only a few feature codes.

Digitising texts - text level

There are three possible ways of handling text level. These are:

1. Use the level given by the SDF which corresponds to the text feature's own feature code, or

2. Use the level which corresponds to the feature code of the feature with which the text feature is to be associated.

3. Use the level given by the SDF for texts of that category.

Text level by own feature code

The first method is suitable when all text features with a particular feature code are to be output on the same level. The feature code is held in the IFF FS entry (or TS entry for composite texts) as usual, and is looked up in the SDF to determine the level required.

Text level by associated feature code

The second method is required whenever text features are to be output on the same level as some other feature. In this case, a secondary feature code AC (AC type 1) is inserted into the feature, with the AC code set to the feature code of the feature that the text is associated with. This secondary feature code will be looked up in the SDF to decide which level the text should be placed on.

Note that if there is more than one secondary AC in a text feature, the first one is the one used for determining text level, and the rest are used for repeating the text feature (see below).

Text level by category

The third method allows texts to be output to different levels according to their categories. The text category is a value between 0 and 63 which is held in the FS entry within the IFF file, and the SDF command CATEGORY can be used to specify that all texts with a particular category will be output on a particular level.

Repeated features

Sometimes it is necessary to output several different versions of an IFF feature. For instance, one linear feature might represent the path of an electricity line and also that of a telephone line. It is not desirable to store the IFF data several times, with only the feature code changing, so a mechanism is supplied to specify that a feature is to be output as if it had been so stored.

If IFF2SIF finds secondary feature code ACs (AC type 1) in a feature, it forms a "repetition list" of these feature codes. After the feature has been output to SIF (using the feature code from the FS entry), IFF2SIF then outputs the feature again for each feature code from the repetition list. Thus the same point data may be output in several different ways, depending on what the relevant line of the SDF specifies for each feature code.

Note that if the feature is a text feature, and the text level is being taken from the secondary feature code (see above), then the first AC type 1 is used for that purpose, and does not count towards repetition of the feature.

Attributes

If there are ACs in a feature, IFF2SIF translates them into SIF associative values by outputting the IDE command followed by the ASV command. IFF2SIF will use the standard ACD definitions in the translation if the is no ACD table in the FRT. For each attribute in a feature the attribute name followed by the attribute value is output in the ASV command. An IFF AC can have a value and an optional text string, but the SIF attribute value can only be either a number or a text string. If there is a value and an optional text string in the AC then the text string is ignored except if the value is set to either: 0 if its an integer, 0.0 if real, ' ' if a character string.

ACs are not translated if the /SCITEX qualifier is specified.

IFF2SIF is restricted to a limit of 200 ACs in a feature that can be translated to associative values. If a feature contains more than this a warning message is given and the excess ACs are not translated.

The SIF Definition File (SDF)

The SDF tells IFF2SIF what each feature code in the IFF file means in terms of SIF data. It may also contain point size and text category/level definitions.

Feature code definitions

The SDF contains one line for each feature code used in the IFF file, and these are composed as follows:

For ordinary feature types, we have:

fc type level style weight colour

For texts, we have

fc TEXT level style weight colour font height width

For symbol cells we have

fc CELL level style weight colour cell_name

where:

- **fc** is the feature code
- **type** is the type of feature - this may be any of
  - LINE - an open linear feature
  - SHAPE - a closed linear feature
  - HOLE - a closed linear 'hole' - not normally used
  - CURVE - interpolated curve - not normally used
  - TEXT - a text
  - CELL - a symbol cell
  - CIRCLE - a circle
  - ARC - a circle arc
- **level** is the SIF 'overlay' or level to place the feature on, from 1 to 63
- **style** is the line style to use - possible values are
  - 0 - solid
  - 1 - dotted
  - 2 - medium dashes
  - 3 - long dashes
  - 4 - dash, dot
  - 5 - short dashes
  - 6 - dash, dot, dot
  - 7 - long dash, short dash
- **weight** is the line weight to use, from 0 to 31
- **colour** is the line colour, or line code, from 0 to 255

If the style, weight or colour is not specified by the customer, then set it to zero (0).

For features of type TEXT, the additional values are

- **font** is the text font to use, from 0 to 126
- **height** is the text height, in metres on the ground

width is the text width, in metres on the ground

For features of type CELL, we have

cell_name is the name of the cell, used by the Intergraph system to identify the symbol. The cell name is ended by the end of the line, or a '!' character, and may not contain spaces. If the name is longer than 12 characters (or 10 for SCITEX), it will be truncated.

Point size definitions

It can be useful to determine text size by point size (as for OS text, for example). Point sizes 1 to 30 may be defined, although only those required need be given. A point size is defined by:

POINTSIZE size height width

where

size is the point size being defined (between 1 and 30)
height is the text height for that point size (a real number in the current units)
width is the text width for that point size (a real number in the current units)

Note that the sizes in the feature code definition of the text should still be given, as they will be used if the point size contained in the IFF file is unacceptable for some reason.

Choosing the size units

By default, all sizes in the SDF are in ground units - ie assumed to be metres on the ground. However, it can sometimes be more convenient to work in sheet millimetres, and the UNITS command allows the user to select what units sizes will be read in.

UNITS MM_ON_SHEET

After this command, all sizes will be read in sheet millimetres. This is generally more convenient for defining point sizes, where the text is measured on the sheet.

UNITS METRES_ON_GROUND

This command restores the default state. After this command, all sizes will be in read in ground metres.

Specifying level by category

If text features are to be written to a level according to their text category, then the CATEGORY command is used to indicate which level each category of text should go to.

```
CATEGORY  categ  level
```

where

categ is the text category, between 0 and 63
level is the level on which all texts with that category should be written, from 1 to 63

Comments

Comment lines are indicated by a '!' character - any line starting with this is ignored. Also, any data on the end of a line (except for CELL lines) is currently ignored, although it is still good practice to start a comment at the end of the line with a comment character.

Example SDF layout

For example:

```
!
! some example SDF lines:
!                                               comment
!fc     type    level   style   weight  colour  cell_name
!                                               font    height  width
!
10      line    2       2       1       0               ! line - roadway
11      cell    3       0       1       2       BLDG1   ! cell - house
12      text    4       0       1       27      2       3.4     2.6
!
! define our point sizes
!
UNITS  MM_ON_SHEET
!
POINTSIZE  4  4.5  4.0
POINTSIZE  8  8.5  8.0
```

In general the above three SDF graphical types are probably the only ones needed.

Comparison of SDF and FRT graphical types

There is a rough relationship between the feature types in the SDF file, and the graphical types in the FRT file. This is as follows:

LINE corresponds to FRT GT 1 (linear).

SHAPE corresponds to FRT GT 12 (fill area) or to GT 1 (linear).

Note that a SHAPE must be closed (first point and last point the same). This type may be requested for certain building features, but otherwise LINE is probably better.

HOLE basically as for SHAPE.

This type is similar to SHAPE. It must also be closed, but is unlikely to be required.

CURVE corresponds to FRT GT 6 (curve).

However, in general it is better to output LINE type, as we do not know what form of interpolation the customer will use on the curve points - this may mean that they get different results from the same data! Check with the customer whether LINE is acceptable instead of CURVE.

TEXT corresponds to FRT GT 10 (text).

Note that text size may be taken directly from the SDF definition, or from the IFF TH entry. In the latter case, it may be in points or in sheet millimetres.

CELL corresponds to FRT GTs 7,8,9 (symbols).

SIF symbols are always output with origin, angle, and scale. IFF2SIF uses the FRT GT to determine what data it requires from the IFF file. Check carefully how each symbol should be digitised, and set the FRT GT appropriately. If scaled symbols are to be output, then IFF2SIF will need to calculate the size of the symbol, and will thus require an SRI file.

CIRCLE corresponds to FRT GT 5 (circle).

It may be easier to digitise these using LINE (ie as a circle interpolated when digitised). Check if the customer will accept this. IFF2SIF will not convert a 'linear' circle to CIRCLE type.

ARC corresponds to FRT GTs 2,3,4 (arc).

The direction of arc for GTs 3 and 4 is deduced from the FRT GT. Comments are as for CIRCLE.

SIF2IFF data preparation

The SIF2IFF program requires that the correlation between IFF feature codes and SIF graphical types and characteristics must be defined. For this an FRT must be created. The FRT should be consulted when creating a parameter file containing a Feature code Description table, which defines IFF feature codes in terms of combinations of SIF graphical types and graphical characteristics. The user must have already prepared an FRT file defining the feature codes to be used. The associated TRI and SRI files must also be prepared.

Note: text features will contain TH entries holding the text height. All features will have the text/symbol bit in the FS entry set to the appropriate value to indicate the feature type.

The SIF2IFF parameter file

Before creating the parameter file the following must be deduced:

1. All the line styles and/or line patterns used in the SIF file. A SIF line or pattern must correspond to an appropriate secondary code for graphical types 1 (line), 2 (clockwise arc), 3 (anticlockwise arc), 4 (circumcircle arc), 5 (circle), 6 (interpolated curve) , or 12 (area fill) in the FRT. Note that a SIF file can use either LAC or the PTN commands to specify line or area patterns. For the LAC command a number specifies the pattern, whereas for the PTN command a name string specifies the pattern.

2. All the SIF line weights used in the SIF file. For every line weight there should be a corresponding width value in millimetres in the FRT.

3. All the SIF line colours used. For every colour there should be a corresponding colour value in the FRT.

4. All the SIF symbols used. The name of each symbol must be known and whether it is scaled or oriented. All SIF symbols have an orientation point so there are no unoriented symbols. Those symbols which have a scale factor should correspond with graphical type 9 in the FRT while those that do not should correspond with graphical type 8.

5. All the SIF text fonts used. For every text font there should be a corresponding combination of size and secondary code in the FRT relating to a font in the TRI file.

6. All the types of arc used. SIF arcs are either clockwise, anticlockwise or circumcircle. An anticlockwise arc corresponds with graphical type 3, a clockwise arc with graphical type 2 and a circumcircle arc with graphical type 4 in the FRT.

7. If it is required that the IFF 'TH' entry should contain pointsizes, then all the combinations of SIF text heights and widths must be known so that each combination can be assigned to a pointsize.

Ideally the FRT should be prepared first so that the above criteria can be deduced.

When the above has been deduced, appropriate combinations of the SIF graphical types must be arranged in the parameter file to correspond with the feature codes in the FRT. Also if pointsizes are required, all the pointsize definitions must be set up in the parameter file.

Note: When displaying in LITES2 the IFF file which SIF2IFF produces, HEIGHT must be ENABLEd so that text sizes are taken from the IFF file instead of the FRT. If the heights in the IFF file are pointsizes the

PSIZE must also be ENABLEd.

Each line of the parameter file is prefixed with the FDF command, except when text pointsizes are specified where the prefix is POINTSIZE. The order of entries on a line is in a fixed format, but the actual order of lines in the file is not important. Any text starting with the "!" character is regarded as a comment, and is ignored. Comments can appear on the same lines as commands, or on separate lines. Note that commands may be in upper or lower case, but cannot be abbreviated.

A line has the form :-

**FDF gtype [lsty lcol lwgt lpat apat patn
adir symt symn font] fc**

and for specifying pointsizes:-

POINTSIZE size height width

where the fields are :-

FDF - the command FDF which states that this line is an entry in the Feature code Description table.

gtype - a keyword indicating a type of graphic. The valid keywords are: LINE, CIRCLE, CURVE, ARC, SYMBOL, AREA, TEXT
lsty - a SIF line style, an integer in the range 0 to 7.
lcol - a SIF line colour, an integer in the range 0 to 255.
lwgt - a SIF line weight, an integer in the range 0 to 31.
lpat - a SIF line pattern, an integer in the range 0 to 100.
apat - a SIF area pattern, an integer in the range 0 to 100. The upper limit for both the lpat and apat range can be extended by defining the logical name LSL$SIF2IFF_MAX_PATTERN, but note that upper limit of pattern numbers in a standard SIF file is 100.
symt - type of symbol, a value of 1 for a scaled symbol or 0 for an oriented symbol.
symn - a SIF symbol name, an unquoted string of up to 12 characters.
patn - a SIF line or area pattern name, a unquoted string of up to 6 characters. If there is to be no pattern name, the string must be **NOPAT**. This is to be used only if the SIF file contains the PTN command. When it is used the **apat\lpat** field should be set to 0.
adir - direction specifier for a a SIF arc, a value of 1 for clockwise, -1 for anti-clockwise or 0 for a circumcircle arc.
font - a SIF font number, an integer between 0 and 255.
fc - the IFF feature code, a value in the range 0 to 32767.

POINTSIZE- the command POINTSIZE which states that this line gives a pointsize specification.
size - the pointsize value which will represent the SIF height and width combination of a text.
height - the SIF height of a text in UORs
width - the SIF width of a text in UORs

The **FDF** and **gtype** entries are required on every line of the table (except lines beginning with POINTSIZE), but thereafter the required entries depend on the **gtype** entry. The required entries for each **gtype** are as follows:-

LINE - lsty lcol lwgt lpat patn fc

CIRCLE - lsty lcol lwgt lpat patn fc

CURVE - lsty lcol lwgt lpat patn fc

ARC - lsty lcol lwgt lpat patn adir fc

SYMBOL - symt symn fc

AREA - lsty lcol lwgt apat patn fc

TEXT - font lcol fc

The **POINTSIZE** entry is required for pointsizes only and must be followed by the **size**, **height** and **width** entries as follows:-

POINTSIZE - size height width

Note that the required entries for each **gtype** or **POINTSIZE** must be present and in the order specified above. If more than the required entries are present, the surplus will be ignored. If less than the required entries are present an error message will be generated. Every entry on a line must be separated by at least one blank or tab.

When the program is run, if it finds any combinations of SIF graphical types which have not been assigned to a feature code in the parameter file, then it will output error messages which show the missing combinations. Also if the program finds any combinations of SIF text height and width which have not been assigned to a pointsize in the parameter file, then it will output error messages which show the missing combinations. The user must put these missing combinations in the parameter file, then re-run the program.

An example showing a parameter file and a corresponding FRT follows :-

1. FRT file:

```
!
!     FC      GT      Colour  Width   Size    SC      Description
FRT   0       1       1       0.2     0       0       Unknown feature
type
FRT   1       1       3       0.25    0       0       B Class Road
FRT   3       1       2       0.2     0       0       Main Through
Road
FRT   11      8       3       0.5     0.5     1       Parking
FRT   25      8       2       0.5     0.5     1       School
FRT     27    8       3       0.5     0.5     1       strange symbol
!
```

```
!symbols
!
FRT   57      8       4       0.5     0.5     1       Telephone
!
!     TEXT
!     FC      GT      Colour  Width   Size    SC      Description
!
FRT   60      10      1       0.15    1.4     1        B Class Road
FRT   62      10      1       0.15    1.0     1        Minor Road
FRT   64      10      1       0.15    6       1        Grid
FRT   65      10      1       0.2     6       1        District
!
!fill
!
FRT   98      12      1       0       0.25    -1      Playing Field
FRT   321     8       3       0.1     0.1     1       Playing Field
Seed
!
!SYMBOL COMPONENTS TABLE
!     FC      GT      Colour  Width   Size    SC      Description
SCT   0       1       1       1       0       0
SCT   1       1       1       1       0       0
SCT   2       2       1       1       0       0
SCT   3       3       1       1       0       0
SCT   4       4       1       1       0       0
SCT   12      12      1       1       0       0
!
!
```

2. Parameter file:

```
!
!            lsty lcol lwgt lpat  patn    fc
!
FDF     LINE    0    0    0    0     NOPAT   0
FDF     LINE    0    0    1    0     NOPAT   1
FDF     LINE    0    0    0    0     NOPAT   3
!
!            lsty lcol lwgt apat  patn    fc
!
FDF     AREA    0    0    1    0     NOPAT   98
!
!            font lcol                    fc
!
FDF     TEXT    1    0                       60
FDF     TEXT    2    0                       62
FDF     TEXT    3    0                       64
FDF     TEXT    4    0                       65
!
!              symt  symn                 fc
!
FDF     SYMBOL  0   Cross                    27
FDF     SYMBOL  0   Telephone                57
FDF     SYMBOL  0   Parking                  11
FDF     SYMBOL  0   Sch                      25
FDF     SYMBOL  0   Fieldseed                321
```

```
!
POINTSIZE    6  60000     50000
```

In the above example line widths 0.2 and 0.25 in the FRT have been designated to correspond with SIF line weights 0 and 1 respectively in the parameter file. For texts there are four combinations of width, size, and secondary code, and these have been designated to correspond with SIF fonts 1, 2, 3, and 4. Secondary code -1 for graphical type 12 in the FRT has been designated to correspond with SIF area pattern number 0. For symbols width and size values in the FRT cannot be used to influence the feature code of a symbol in the parameter file. The parameter file allows only the symbol type (scaled or oriented) and the SIF symbol name to determine the feature code.

Note: CCG (Canadian Centre for Geomatics) users must prepare a version of the parameter file in their own special format using CCSM codes. The following is an example of the format of a CCG parameter file:

```
25    BA 31758 490    ! BARN SYMBOL
20    BR 03555 290    ! BUILDING SYMBOL
30    BM 15607 090    ! CHURCH POINT SYMBOL
31    BM 35500 020    ! CHURCH CROSS
59    BF 11330 040    ! FIRE STATION SYMBOL
35    BA 12000 190    ! GREENHOUSE SYMBOL
50    BR 11123 213    ! DOME SYMBOL
46    BG 88750 390    ! POWER HOUSE SYMBOL
53    EA 99400 490    ! TANK
51    EA 87126 111    ! TOWER
55    AQ 18347 230    ! SEA PLANE ANCHORAGE
58    AL 94389 010    ! FORT SYMBOL
!
```

CHAPTER 3

IFF2SIF UTILITY

UTILITY IFF2SIF

FUNCTION

IFF2SIF provides a translation from IFF to a subset of Intergraph's Standard Interchange Format (SIF). It produces SIF ASCII text format, to magnetic tape or to a disk file. The actual translation is guided by a SIF Definition File (SDF) and a normal FRT file. Commands are provided to allow the scaling and shifting of data as it is output.

FORMAT

```
$ IFF2SIF
```

Command qualifiers
/ SCITEX

PROMPTS

IFF2SIF is an interactive, command driven utility. Command input is expected when the following prompt is given:

```
iff2sif>
```

COMMAND QUALIFIERS

/SCITEX

- enables output for the SCITEX version of SIF. This is a licensed option. To use this qualifier the logical name **LSL$IFF2SIF_SCITEX** must point to the shared image supplied by Laser-Scan which checks that a licence has been obtained. If this image is not present an error occurs.

DESCRIPTION

Files required

IFF2SIF requires that the following files be specified:

- o IFF file(s)
  These contain the data to be translated.

- o FRT file
  This is the Feature Representation File used to describe the meaning of the feature codes within the file. It will be the same FRT that was used when the file was created, or when it was edited with LITES2.

- SDF file
  This is the SIF Description File, which defines how each feature code should be translated to SIF.

In addition, two other forms of file may also be used

- SRI file
  This is a Symbol Representation IFF file, which defines the graphic used for each IFF symbol. It is only required if scaled symbols are being output via IFF2SIF.

- Input list file
  This is a text file containing the list of IFF files to be output, one per line. It may be used instead of giving a succession of IFF commands.

Output units and scale

The data is output in SIF Units of Resolution (UORs). The customer should have defined what size one UOR should be - perhaps one millimetre on the ground. In order to convert the units in the IFF file into UORs in the output, IFF2SIF must be told the scale of the IFF units with respect to ground units, and the size of UORs in ground units.

The coordinates read from the IFF file may be offset and scaled to place them into the correct coordinate range in the destination SIF file.

The transformations applied are:

x in UORs = (x in IFF + x offset) * IFF scale / UOR unit
y in UORs = (y in IFF + y offset) * IFF scale / UOR unit

If height data is being scaled, then

z in UORs = (z in IFF + z offset) * z scale / UOR unit

or if height data is being magnified, then

z in UORs = (z in IFF + z offset) * z multiplication

otherwise

z in UORs = (z in IFF + z offset)

x offset and **y offset** are either taken from the map descriptor of each file, after the MAPDESCRIPTOR OFFSET command has been given, or are explicitly specified with the OFFSET command. **z offset** is specified with the ZOFFSET command. If offsets are not requested, then the offset addition is not performed.

Offsets should be used in the following cases:

1. when the local origin of the IFF file is to be added to the x,y coordinates so that the absolute coordinates may be output.

2. when a constant offset is to be added to the z values.

3. when negative coordinates are to be output, to make the coordinates be greater than or equal to zero (since IFF2SIF outputs unsigned data). If this is to be done for x,y coordinates, the map descriptor's local origin may not currently be used.

IFF scale is specified with the IFFSCALE command, and is the scale of the data in the IFF file. This will generally be 1:1 - that is one IFF unit represents one ground unit. We assume that ground units are metres.

z scale is specified with the ZSCALE command, and is the scale of the height data in the IFF file. This command is only given if Z data is present and to be scaled in a similar manner to the X,Y coordinates. The default action is not to scale any height data.

z multiplication is specified with the ZMULTIPLY command, and is the value to multiply all heights by before outputting them. This command is only given if Z data is present and to be multiplied by a constant, for instance to retain the precision of fractional heights. The default action is not to multiply any height data.

UOR unit is specified with the UORUNITS command, and is the size of a UOR in metres on the ground. Thus if UORs are ground millimetres, one UOR is 0.001 ground metres, and **UOR unit** will be 0.001

Finally, IFF2SIF also requires the scale of the digitised document with regard to the ground. This is needed in case the user specifies text sizes in millimetres on the sheet. Since all data is output in UORs, which are in ground units, IFF2SIF requires the sheet scale so that it can guarantee to be able to output any texts correctly.

COMMANDS

IFF2SIF is told what to do by various commands. One command may be given on each prompt line - anything after that command is ignored.

Commands may be abbreviated, so long as the abbreviation is not ambiguous (the program will complain if it is, and ignore the abbreviation).

Summary of commands

In summary, the available commands are:

APPEND	append data to end of tape
DEBUG	for debugging
DENSITY number	choose magtape density
DIMENSION dim	specify 2 or 3 dimensional output
FRT filespec	specify FRT file
HELP command	give help on a command
IFF filespec	specify next file to output
IFFSCALE scale	specify scale of IFF units (default 1:1)
IGNORE layer	ignore an IFF layer
INPUT_LIST filespec	read the IFF filespecs from this file
JUSTIFICATION posn	specify a text justification
MAGTAPE tapedrive	choose magtape drive
MAPDESCRIPTOR what	choose to use information from the MD entry
NOMAGTAPE	don't output to magtape
NOOUTPUT	undo OUTPUT
NOREFLECT	for debugging
NOZSCALE	heights not to be scaled (default)
OFFSET X,Y	specify offset of map (in IFF units)
OUTPUT filespec	output next IFF file to disk
RECORD size	specify output record size (default 72)
REFLECT	for debugging
SDF filespec	specify SIF definition file
SHEETSCALE scale	specify scale of sheet
SHOW	show the current state for next output
SRI filespec	specify a symbol definition file
TEXTLEVEL what	where to get text level from
TEXTSIZE what	where to get text size from
UORUNIT metres	specify size of UORs (output units)
ZMULTIPLY multiply	specify amount to multiply height units
ZOFFSET Z	specify offset of heights (in IFF units)
ZSCALE scale	specify scale of height units

Alphabetical list of commands

APPEND

This command tells the program that, when it initialises the magnetic tape, it must find the end of data and prepare to add more SIF files to the tape. Thus the command may only be given before the tape has been initialised.

Note that the command will only work sensibly if there are already SIF data files on the magtape.

The default action on initialising the tape is to rewind it, ready to start from scratch.

DEBUG

This command is intended for debugging only - it is equivalent to giving the sequence of commands

```
REFLECT
NOMAGTAPE
OUTPUT  TT:
SHOW
```

DENSITY 'integer'

This command tells the program to output SIF data to the magtape at the specified density. The value may be either 800 or 1600.

Note that this command may only be given before the magtape has been initialised - it is not possible to change the tape density after the tape has been written to.

The default density is 1600 bpi.

DIMENSION 'integer'

This command tells the program that all output from now on will be either two or three dimensional, according to whether 'integer' is 2 or 3. Two dimensional output is the default.

If two dimensional output is selected, then the Z coordinates of ZS entries are ignored, and heights encoded in height ACs are output using the ACZ command. If three dimensional output is selected, then all IFF features must contain an explicit height value, either from a height AC or from the Z values in a ZS entry.

FRT 'file-spec'

This command tells the program which FRT definition file to use from now on. The file chosen will be used for all further IFF files output unless the FRT command is given again.

The FRT file is used alongside the SDF to check that features are coded correctly. Note that if the SDF command is not given, then an SDF of the same directory and name as the FRT will be assumed (the extension is just changed from .FRT to .SDF)

The FRT file-spec is filled out from the default LSL$FRT:.FRT

HELP 'command'

This command asks for help on the given IFF2SIF command. If no command is specified, then it gives a list of the commands available.

IFF 'file-spec'

This command tells the program which IFF file to output. After it has been given, the program will attempt to output the file. Therefore any other commands affecting output must have been given before the IFF command. In particular, at least one of the SDF or FRT commands must have been given to determine what SIF and FRT definition files are to be used when translating. Also, the mapscale and UOR unit must have been given - see MAPSCALE and UORUNIT below.

The IFF file-spec is filled out from the default LSL$IF:.IFF;0

See also the INPUT_LIST command, which allows multiple IFF files to be specified.

IFFSCALE 'real'

This command tells the program what scale the IFF units are in, with respect to ground units. The program will assume a scale of 1.0 (ie 1:1), and this command need only be given if that assumption is incorrect.

The scale of IFF units is used with the UORUNIT command, to allow the program to decide on the multiplication factor to use in going from IFF map units to SIF UORs.

The IFF scale is assumed the same for all further IFF files output, unless this command is given again.

Note that this scale is only applied to X and Y coordinates - to scale height values in the same manner, use the ZSCALE command.

IGNORE 'integer'

This command tells the program to ignore the contents of layer 'integer' in the IFF files being output. Only one layer may be ignored. By default, the contents of layer 0 will be ignored.

If you want to output all layers, then ask the program to ignore a negative layer number.

INPUT_LIST 'filespec'

This command tells the program to output the IFF files specified in the file 'filespec'. The input list file must contain one IFF filespec on each line. Blank lines and lines starting with an exclamation mark (!) will be ignored. When the end of the input list file is reached, IFF2SIF will exit.

JUSTIFICATION 'integer'

By default, the text justification position is taken from the text code entry in the third word of the IFF FS entry. However, this is normally only set in the IFF file if OS text conventions are in use. Therefore this command is provided to set the text justification for all text features from now on to be 'integer'.

The justification positions are numbered 1 to 9, where the normal position (bottom left hand corner of the text) is justification 3.

The command JUSTIFICATION 0 will cause the program to look in the FS entry again.

MAGTAPE 'device-name'

This command tells the program to write to the specified magnetic tape unit. If the argument is omitted, then MTA0 is assumed. All following IFF files will be output to this tape unit.

The command may only be given before the magnetic tape has been initialised, since it is not possible to change the tape drive being used partway through the program.

Note that the UNIT, DENSITY and APPEND commands also enable output to magtape, and that when the program starts up, it assumes that it will be outputting to magnetic tape (on MTA0).

MAPDESCRIPTOR 'what'

This command tells the program that it should read the map descriptor (MD) entry of each IFF file from now on. The 'what' tells it what information to use from the MD entry. The following are supported:

MAPDESCRIPTOR OFFSET

From now on, the local origin of each file will be taken from the map descriptor and added to all X and Y coordinates before they are output. Any use of the OFFSET command will be ignored.

NOMAGTAPE (applies to next IFF file only)

This command tells SIF not to use the magnetic tape drive for the next IFF file being output. The default action is to write all files to magnetic tape.

NOOUTPUT (applies to next IFF file only)

This command is provided as the opposite of OUTPUT. It tells the program not to output the SIF data file to a text file for the next IFF file. Note that this is the default state.

NOREFLECT (applies to next IFF file only)

This command is intended for use during debugging only - it undoes the effect of a preceding REFLECT command.

NOZSCALE

This command switches off scaling or multiplying of Z (height) coordinates. Any offset specified by the ZOFFSET command will be added to height data, but the resulting quantity will be output to the SIF file unchanged (apart from rounding to an integer quantity).

NOZSCALE is the default.

OFFSET 'Xreal', 'Yreal' (applies to next IFF file only)

This command tells the program that all data in the next IFF file is to be shifted by 'Xreal' in X and by 'Yreal' in Y. The offsets are specified in IFF units. This command is ignored if the MAPDESCRIPTOR OFFSET command has been given.

For shifting the Z (height) data, see the ZOFFSET command.

This command only affects the next IFF file.

OUTPUT 'file-spec' (applies to next IFF file only)

This command tells the program to write the SIF text data to the specified text file. It is possible to output data to magtape and text file at the same time. Use the NOMAGTAPE command to suppress output to the magtape.

Only the next IFF file will be output to the text file.

The default extension for the text file is .TXT

RECORD 'integer'

This command tells the program that data should be output in records of length 'integer' (bytes or characters). 'integer' must be between 40 and 80.

Note that this command may only be given before data has been written to the magnetic tape - it is not possible to change the record size after the tape has been written to.

The default record size is 72 bytes.

REFLECT (applies to next IFF file only)

This command is intended for use during debugging only - it tells the program to reflect the data in the next IFF file to the terminal as the file is read.

SDF 'file-spec'

This command tells the program which SIF Definition File (SDF) to use from now on. The file chosen will be used for all further IFF files output unless the SDF command is given again.

The SDF describes the SIF attributes associated with each feature code, and is thus similar in concept to FRTLIB's FRT file.

Note that if the FRT command is not given, then an FRT of the same name as the SDF will be assumed (the extension is just changed from .SDF to .FRT)

The SDF file-spec is filled out from the default LSL$FRT:.SDF;0

SHEETSCALE 'real'

This command tells the program what scale the sheet is at, with respect to ground units. Thus if the next IFF file represents a sheet that was at scale 1:500, the command would be

```
SHEETSCALE  500
```

This command is used to define the transformation from sheet millimetres to ground units, for use in text sizing. The command is always required before reading the SDF, even if the SDF doesn't use sheet mm - the program cannot tell that until after it has read the file.

The sheet scale is assumed the same for all further IFF files output, unless this command is given again. If the sheet scale is changed, and the SDF did use sheet mm, then the SDF will be read again (and incidentally, so will the FRT file).

SHOW

This command asks the program for a status report - what the program believes its current state to be. It will tell you the current FRT and SDF filenames, whether the magtape is in use, and if so which one, and what transformations are being applied to IFF map units.

SRI 'file-spec'

This command specifies an SRI file to be used when determining the size of scaled symbols. An SRI file is only necessary when scaled symbols are being output. The program will read the SRI file immediately.

The SRI file-spec is filled out from the default name LSL$FRT:.SRI;0

TEXTLEVEL 'what'

This command tells the program where it should find the level on which texts should be output. The following are supported:

TEXTLEVEL SDF

This is the default, and means that the text level should be that defined in the SDF entry for the relevant text feature code.

TEXTLEVEL AC

This command states that the text level is defined in the SDF entry for the feature code held in the IFF AC 1 (secondary feature-code AC) entry.

TEXTLEVEL CATEGORY

This command states that the text level is to be deduced from the text category, as specified in the SDF. This requires that the SDF contains a sequence of CATEGORY commands to relate each text category to a SIF level.

TEXTSIZE 'what'

This command tells the program where it should find the size of texts. The following are supported:

TEXTSIZE SDF

This is the default, and means that the text size should be that defined in the SDF entry for the relevant text feature code.

TEXTSIZE POINT

This command states that the text size is defined as a point size in the IFF TH entry, and that this point size will have been defined in the SDF (using the POINTSIZE command).

TEXTSIZE MILLIMETRES

This command states that the text size is defined in the IFF TH entry, in hundredths of a millimetre on the sheet.

UORUNIT 'real'

This command tells the program what size a UOR is, in metres on the ground. Thus if the next SIF data file is to be output with UORs of size 1 millimetre on the ground, the command given would be

```
UORUNIT  0.001
```

This command is used with the IFF units scale, to allow the program to decide on the multiplication factor to use in going from IFF units to SIF UORs.

The UOR unit will remain the same for all further IFF files, unless this command is given again.

ZMULTIPLY 'real'

This command switches on multiplication of the height data by a constant. All height values will be multiplied by 'real' before they are output.

The Z multiplication is assumed the same for all further IFF files output, unless this command is given again.

By default, Z values are not multiplied by anything before output.

The NOZSCALE command is provided to switch off scaling and multiplying of Z coordinates. The ZSCALE command allows scaling of height data in a similar manner to X,Y data.

ZOFFSET 'Zreal' (applies to next IFF file only)

This command tells the program that all height data in the next IFF file is to be shifted by 'Zreal'. The units are the same as the height units in the IFF file.

For shifting the X,Y data, see the OFFSET command.

This command only affects the next IFF file.

ZSCALE 'real'

This command switches on scaling of the height data, and tells the program what scale the IFF units of height are in, with respect to the required units. By default, Z values are not scaled at all.

The Z scale is assumed the same for all further IFF files output, unless this command is given again.

The NOZSCALE command is provided to switch off scaling and multiplying of Z coordinates. The ZMULTIPLY command allows multiplication of height data by a constant value.

EXAMPLE

```
$ MOUNT/FOREIGN MTA0:<CR>
%MOUNT-I-MOUNTED, FRED mounted in MTA0:
$ IFF2SIF<CR>
IFF2SIF of 10:23:05 25-NOV-87

iff2sif> IFFSCALE 1<CR>
iff2sif> SHEETSCALE 2000<CR>
iff2sif> UOR 0.001<CR>
iff2sif> SDF EXAMPLE<CR>
iff2sif> FRT EXAMPLE<CR>
iff2sif> JUST 3<CR>
iff2sif> SHOW<CR>
SIF definition file     LSL$FRT:EXAMPLE.SDF;0
FRT definition file     LSL$FRT:EXAMPLE.FRT;0
Magtape unit MTA0 (density 1600) will be used
(Tape will be rewound)
Output record size is 72
Output data is 2 dimensional

IFF data is at scale    1:1
Sheet is at scale       1:2000
UOR units are   0.0010 metre
coord in UORs = coord in IFF units   1000.000
This means maximum map range is  -2147483.75 to  2147483.75

Height values (Z coordinates) will not be scaled

Ignoring contents of layer 0
Text justification overridden - set to 3
iff2sif> IFF OV1.OUT<CR>
SIF definition file     LSL$FRT:EXAMPLE.SDF;0
FRT definition file     LSL$FRT:EXAMPLE.FRT;0
Magtape unit MTA0 initialised (density 1600)

RANGE in UORs is       -36194,    18401382         -6337,    13900000
Drawing name:   OV1     Date:   15MAR85

iff2sif> <CTRL/Z>
Total of 25 blocks now on magtape

 ELAPSED: 00:00:26.23  CPU: 0:00:02.22  BUFIO: 16  DIRIO: 9  FAULTS: 154
$
```

This example shows a user outputting a single IFF file to tape in SIF. Note the use of the SHOW command to confirm that the correct values have been set up. $STATUS will be set to SS$_NORMAL after this run of the program.

MESSAGES (INFORMATIONAL)

These messages normally follow a more serious error message.

ABANDON, Abandoning processing of this IFF file

Explanation: A severe error has occurred, which leaves no other choice. This message will follow the message describing that error.

User action: Depends upon the previous message.

FORCELEVEL, Forcing feature 'integer' to level 'integer'

Explanation: This message is output after an associated message indicating that the correct level for the feature could not be worked out. The feature will be output on a SIF error level, normally 63.

User action: The data as produced is incorrect, so the input data should be corrected and reprocessed.

FORCETOLINE, Forcing feature 'integer' to line, on level 'integer'

Explanation: This message is output following the FRTSIFFC error message. The feature will be output on a SIF error level, normally 63.

User action: The data as produced is incorrect, so the input data should be corrected and reprocessed.

SYMNAMLONG, Symbol name %S in SDF file is too long. Truncating to 12 characters (or 10 for SCITEX).

Explanation: This message is output when a symbol name in the SDF file is too long. The limit is 12 characters but for SCITEX output the limit is 10.

User action: The name will be truncated and the program will continue processing, but the user may prefer to change the names in the SDF file then continue processing.

TRYTOGOON, Attempting to continue outputting feature 'integer'

Explanation: Appears after some of the other errors.

User action: Depends upon the previous message.

MESSAGES (WARNING)

These messages give warning of non severe errors

MAXAC, Too many ACs in feature FSN 'integer' - maximum number is 'integer'. Rest of ACs ignored.

Explanation: Number of ACs in a feature that can be translated is limited. The feature will be translated with up to the maximum allowed ACs.

User action: Contact Laser-Scan to raise the limit if necessary.

MESSAGES (ERROR)

These messages reflect errors in the data, or mistakes in giving commands.

ABANDONED, Circle feature 'integer' abandoned

Explanation: As it says - a previous message should clarify this. The program will attempt to continue.

User action: Depends upon the previous message.

BADARC, Arc feature 'integer' abandoned - two points identical

Explanation: Two of the points defining an arc are identical. The feature will not be output. The program will attempt to continue.

User action: Correct the original IFF file and reprocess the data.

BADCATARG, Can't read level argument in CATEGORY 'integer' command

Explanation: The CATEGORY command for category 'integer' in the SDF is incomplete or incorrect. The command will be ignored.

User action: Edit the SDF to correct the command.

BADCATNUM, Can't read category in CATEGORY command

Explanation: A CATEGORY command in the SDF does not have a legible category number. The command will be ignored.

User action: Edit the SDF to correct the command.

BADDIM, Data may only be 2 or 3 dimensional

Explanation: The DIMENSION command was given, with a value other than 2 or 3 following it. The command will be ignored.

User action: Give the command again, with a correct argument.

BADENSE, Tape density 'integer' is illegal - use either 800 or 1600

Explanation: This message is output if you try to specify DENSITY with other than 800 or 1600. The command will be ignored.

User action: Give the command again, with a correct argument.

BADFCNUM, Unable to read feature code in SIF definition file

Explanation: The feature code number in the SDF is garbled.

User action: Correct the SDF and then run the program again.

BADFONT, Illegal font 'integer' in feature 'integer' - using font 0

Explanation: As it says. The program will attempt to continue.

User action: Correct the SDF, or specify appropriate commands to define the font, and reprocess the data.

BADJUST, Illegal text justification - 1-9 allowed, 0 unsets

Explanation: As it says. This message is given if the JUSTIFICATION command was given with an incorrect argument. The command is ignored.

User action: Give the command again with a correct argument.

BADLEVEL, Illegal level 'integer' in feature 'integer' - using level 63

Explanation: As it says. The program will attempt to continue.

User action: Correct the SDF and run the program again.

BADPTARG, Can't read size argument in POINTSIZE command

Explanation: Either the height or width argument of the POINTSIZE command in the SDF file was non-numeric, or otherwise badly formed. The program will ignore the command.

User action: Correct the SDF and run the program again.

BADPTSIZNUM, Can't read point size in POINTSIZE command

Explanation: Whilst reading the SDF, a POINTSIZE command has been encountered with a non-numeric first argument. It is ignored.

User action: Correct the SDF and run the program again.

BADRANGE, Bad map range - value 'real' can not be output

Explanation: The given map range value is too large, or too small to be represented in UORs at the defined transformation. The program will ask if you want to continue.

User action: Either allow the program to continue, in which case data will be truncated at the edge of the sheet, or respecify the UOR units and try again.

BADRECSIZ, Record size 'integer' not allowed - must be from 40 to 80

Explanation: The argument to the RECORD command was out of range - it is ignored.

User action: Give the command again with a correct argument.

BADST, 'integer' points in ST - only 'integer' used - in feature 'integer'

Explanation: There were too many points (more than 200) in an ST in the named feature. The maximum number allowed in an ST will be read and used.

User action: Check why an IFF file was produced with too many points in an ST. The file can be corrected by outputting it with ITOTEXT and then recreating it with IFROMTEXT.

CATLEVEL, Bad level 'integer' for CATEGORY 'integer' - must be between 1 and 63

Explanation: The CATEGORY command for the specified category number indicates an illegal level - SIF level numbers must be between 1 and 63.

User action: Edit the SDF to correct the CATEGORY command.

CATRANGE, Illegal category 'integer' ignored (must be in range 0 to 63)

Explanation: A CATEGORY command in the SDF attempts to define the level for an illegal category - category numbers must be between 0 and 63.

User action: Edit the SDF to correct the CATEGORY command.

CIRSLANT, Circle does not have constant Z value, in feature 'integer'

Explanation: A three point circle with differing Z values for the points has been found. Circles must have constant Z. The circle will be output with the Z value from the first point.

User action: The circle is almost certainly an error. Correct the input data and try again.

CLIPPED, Map coordinate 'real' clipped in feature 'integer'

Explanation: The given map coordinate is too large, or too small, to be represented in UORs at the defined transformation. It is clipped to the maximum/minimum value, and the program attempts to continue.

User action: This message should only normally occur after the program has issued the BADRANGE error, and the user has elected to continue. If this is **not** the case, then the range in the IFF file is incorrect, and should be corrected (for instance by reading the file into LITES2 and exiting).

CLOSECIRCLE, Points too close to define circle in feature 'integer'

Explanation: The points defining a circle must not be too close. The circle cannot be output. The program will attempt to continue.

User action: Correct the circle in the IFF file and reprocess the data.

DEFNAMERR, Error in SIF definition file name

Explanation: The file-spec given to the SDF command was not acceptable. A further message may help with what was wrong with the name.

User action: Give the command again with a correct file-spec.

EMPTYIFF, IFF file is empty - no entries found

Explanation: As it says. Nothing will be output.

User action: Check why the IFF file is empty.

FCNOTFND, Feature code 'integer' not found in SIF definition file

Explanation: As it says. Other messages should indicate the action taken by the program.

User action: Correct the SDF to define the action for that feature code, or if the feature code is incorrect, change the IFF file and reprocess it.

FILCLOSE, Error closing SIF definition file 'file-spec'

Explanation: As it says. The program should output other messages to indicate what went wrong.

User action: Dependent upon the other messages.

FRTNAMERR, Error in FRT definition file name

Explanation: The file-spec given to the FRT command was not acceptable. A further message may help with what was wrong with the name.

User action: Give the command again with a correct file-spec.

FRTSIFFC, FC 'integer' is a SIF 'type', but a FRT 'type'

Explanation: The FRT graphical type for this feature code does not agree with the equivalent description in the SDF. The two 'type's are the names of the types (for instance "line" or "circumcircle arc"). The feature will be output as if it were a linear feature.

User action: Check the FRT and SDF, correct them and reprocess the data.

GOT_MT, Cannot change 'text' after tape has been initialised

Explanation: As it says. 'text' will be one of "magtape unit number" or "tape density" or "record size", depending upon which command was given.

User action: Give the appropriate command again, with a correct argument.

IFFEOF, Unexpected end of IFF file after feature 'integer' (no EJ)

Explanation: The IFF file ended abruptly. The data on tape will be ended as if an EJ had been found.

User action: Investigate why the IFF file was incorrectly terminated.

IFFNAMERR, Error in IFF file name

Explanation: The file-spec given to the IFF command was not acceptable. A further message may help with what was wrong with the name.

User action: Give the command again with a correct file-spec.

IFFOPN, Error opening IFF file 'file-spec' on unit 'integer'

Explanation: As it says, the IFF file could not be opened. Other messages should define the problem.

User action: Dependent upon the other messages. Likely causes are that the file does not exist, that it is being accessed by someone else, or that it is protected against you.

INLINE, In line 'integer' of SIF definition file 'file-spec'

Explanation: All of the errors in reading the SDF file are followed by this supplementary message to allow the user to locate the offending line.

User action: Depends upon the preceding message.

LACCOLOUR, Illegal line colour 'integer' in feature 'integer' - using 0

Explanation: The line colour found for this feature (by looking up the feature code) is not an allowed value.

User action: Correct the SDF and reprocess the data.

LACSTYLE, Illegal line style 'integer' in feature 'integer' - using 0

Explanation: The line style found for this feature (by looking up the feature code) is not an allowed value.

User action: Correct the SDF and reprocess the data.

LACWEIGHT, Illegal line weight 'integer' in feature 'integer' - using 0

Explanation: The line weight found for this feature (by looking up the feature code) is not an allowed value.

User action: Correct the SDF and reprocess the data.

LEZERO, Number must be greater than zero

Explanation: This message is given when a bad numeric argument is found for a command. The command is ignored.

User action: Give the command again, with a correct argument.

LINEARCIRCLE, Circle points collinear in feature 'integer'

Explanation: The three points defining the circle are in a straight line. The circle cannot be drawn. The program will attempt to continue.

User action: Correct the feature in the IFF file and reprocess the data.

MAXFCS, Too many secondary feature code ACs - maximum number is 'integer'

Explanation: This message is output if there are too many secondary feature codes in the current feature to fit into the internal tables. The limit is currently 25. The later ones will be ignored.

User action: If this is really a problem, consult Laser-Scan about increasing the limit.

MTBLOCK, Error writing block 'integer' to magtape

Explanation: This should not happen. A further message will be given, which may help explain the problem.

User action: Consult your system manager.

MTINITERR, Error initialising magtape unit 'file-spec' (density 'integer')

Explanation: The program could not get the magtape unit. A further error message should clarify the problem.

User action: Depends upon the other messages. The likeliest cause is that the magnetic tape has not been mounted (foreign) before using the program.

MTOBCK, Error backspacing magtape - around block 'integer' on tape unit 'device'

Explanation: The APPEND command causes the program to move to and fro searching on the tape - an error has occurred during this.

User action: Consult your system manager.

MTOREAD, Error reading magtape - around block 'integer' on tape unit 'file-spec'

Explanation: As it says. This may occur after the APPEND command.

User action: Consult your system manager.

NEEDTHREE, Need three points to define an arc in feature 'integer'

Explanation: The feature has too few points for the arc definition. It will not be output. The program will attempt to continue.

User action: Correct the IFF file and reprocess it.

NOAC, No AC for text level in feature 'integer' - error level ('integer') used

Explanation: The TEXTLEVEL AC command was given, but this text feature does not contain a relevant AC. The text is output to the error level instead.

User action: Edit the IFF file using LITES2 to insert the AC, and reprocess the file.

NODXDY, Points coincident for scaled symbol in feature 'integer'

Explanation: The two points defining a scaled symbol should be different. The symbol is output at scale 1.0, and horizontally.

User action: Edit the IFF file with LITES2 to correct the symbol, and reprocess the data.

NOFACTOR, Need UORUNIT (and possibly IFFSCALE) to translate coordinates to UORs

Explanation: The program will not output an IFF file until the coordinate transformation has been fully defined. This requires that at least the UORUNIT command be given.

User action: Use the UORUNIT command to specify the UOR units. Also give the IFFSCALE command if necessary.

NOFRTSIF, Must have SIF and FRT definition files before outputting file

Explanation: Self-explanatory - the SDF and FRT commands have not been given.

User action: Specify the FRT file, and the SDF.

NOHEIGHT, No height AC in feature 'integer' - assuming height 0.0

Explanation: Three dimensional data is being output, and this feature does not contain any height data. Note that if it had a ZS entry instead of an ST, then this error would not occur.

User action: Edit the IFF file with LITES2 to insert a height AC, and reprocess the data.

NOPTSIZ, Point size 'integer' not defined - using size from SDF definition

Explanation: The IFF TH entry for this text specifies a point size that was not set up in the SDF - the size values defined in the feature code definition will be used for this text.

User action: Either correct the SDF to define this size, or edit the IFF file to reference a size that is defined.

NORANGE, First entry in IFF file must be RAnge - found <entry>

Explanation: As it says. The file will be abandoned.

User action: Investigate why the first entry in the IFF file is not the range.

NORO, Two identical points for orientation, assuming feature 'integer' horizontal

Explanation: As it says - symbols with two orientation points must have two different points.

User action: Edit the IFF file with LITES2 to define the symbol correctly.

NOSHTSCALE, Must have SHEETSCALE before reading the SDF

Explanation: The SDF may use sheet millimetres, so the SHEETSCALE command must be given before it is read, in order to define how sheet mm may be changed to ground metres.

User action: Specify a sheet scale with the SHEETSCALE command.

NOST, No ST found in feature 'integer' - no points to output

Explanation: The current feature does not contain either an ST or ZS entry. There is therefore no data to output from it, and it will be ignored.

User action: Find out why such a feature was generated in the IFF file.

NOTCLOSED, Shape or hole in feature 'integer' not closed - forcing to line

Explanation: SHAPE and HOLE type features must be closed (first and last points identical). The feature is output as a line.

User action: Edit the IFF file with LITES2 to close the feature, and reprocess the file.

NOTH, No TH entry found for text size - using SDF values

Explanation: The text size was meant to be in POINTs or MILLIMETREs, but this text feature has no TH to look it up in. The size values defined for this feature code in the SDF will be used instead.

User action: Edit the IFF file to correct the feature, or specify TEXTSIZE SDF.

NOTMD2, Map descriptor entry is not type 2 - use ITRANS/DESCRIPTOR

Explanation: The MAPDESCRIPTOR OFFSET command has been given, and the program has found the map descriptor entry for this file. However, it is either a type 1 (old style) map descriptor, or it is unset.

User action: Use the IMP utility IPATCH to inspect the map descriptor, and use ITRANS/DESCRIPTOR to set it correctly, before attempting to output the file again.

OPENLIST, Error opening INPUT_LIST file 'file-spec'

Explanation: As it says, the input list file could not be opened. Other messages should define the problem.

User action: Depends upon the other messages.

OPENSIF, Error opening SIF definition file 'file-spec' on unit 'integer'

Explanation: As it says, the SDF could not be opened. Other messages should define the problem.

User action: Depends upon the other messages.

OUTNAMERR, Error in text output file name

Explanation: A further message may help with what was wrong with the file-spec.

User action: Give the OUTPUT command again with a correct file-spec.

OUTOFORDER, SIF definition entry out of order - FC 'integer' found after FC 'integer'

Explanation: As it says. It is not safe to continue the program using this SIF.

User action: Sort the SDF into feature code order first, and then try again.

OUTOPEN, Error opening text output file 'file-spec'

Explanation: As it says, the text output file could not be opened. Other messages should define the problem.

User action: Depends upon the other messages.

PARSE, Error parsing filename 'file-spec'

Explanation: Some problem occurred in checking the file-spec to make sure that it was sensible, or to abstract some part of the spec. Other messages should tell you what action is being taken by the program.

User action: Depends upon the other messages.

PTSIZNEG, Height and width must both be positive in POINTSIZE command

Explanation: As it says. The relevant POINTSIZE command in the SDF will be ignored.

User action: Correct the SDF and try again.

PTSIZRANGE, Illegal point size 'integer' ignored (must be in range 1 to 'integer')

Explanation: The point size specified in the POINTSIZE command in the SDF is out of the valid range. It is ignored.

User action: Correct the SDF and reprocess the file.

RDCOMM, Error reading command

Explanation: The program was unable to read the command just typed for some reason. Other messages should indicate the problem.

User action: Depends upon the other messages.

SCANSYM, Error in scanning scaled symbol size, FC 'integer' in feature 'integer'

Explanation: This should not occur, but means that the symbol will be output with unit scale, instead of the digitised size.

User action: Depends upon the other messages output with this. A possible cause is that no SRI file was specified.

SIFLINE, Error reading line from SIF definition file

Explanation: A further message should clarify what the problem was.

User action: Depends upon the other message.

SYMSCALE, Symbol cell matrix 'name' scale 'real' out of range in feature 'integer'

Explanation: The X or Y scale is invalid for this symbol. A scale of 1.0 is used.

User action: Check that the original scaled symbol in the IFF file is sensible.

TAPEMK, Error outputting tapemark at about block 'integer'

Explanation: This should not happen. A further message may help with what was wrong.

User action: Consult your system manager.

TLCCOUNT, Illegal text length 'integer' in feature 'integer' - outputting 1 character

Explanation: The text length was too great, zero, or negative.

User action: Inspect the feature using IPATCH or LITES2, and correct it.

TLCJUST, Illegal text justification 'integer' in feature 'integer' - using 3

Explanation: The text justification field in the FS entry for this feature was not valid, and has been ignored.

User action: Correct the field using IPATCH or LITES2, and reprocess the file.

TOOBIG, Cannot represent 'real' in UORs - maximum is 'integer'

Explanation: The given value is too large to represent in UORs at the given transformation.

User action: Check that any offset given by the OFFSET command is correct.

TOOMANYSPACES, Too many spaces found in string to be output - some ignored

Explanation: When outputting a text string, the program will split the string over more than one record if necessary. It will not end a text string fragment with a space - the space is always carried over to the next record. If there are too many spaces in the text string for this to work (ie of the order of the record size!), then this message will be given, and some of the spaces ignored.

User action: Modify the text string and try again.

TOOSMALL, Cannot represent 'real' in UORs - must be positive

Explanation: The given value is less than zero, and therefore cannot be output in UORs.

User action: Check that any offset given by the OFFSET command is correct.

TRUNCSTR, String too long - truncated to 'text'

Explanation: Given if a name is too long - notably if the magnetic tape unit name is longer than eight characters. The truncated version is the one that would be used, if the user does not correct this.

User action: Either accept the truncated version, or give the relevant command again with a correct argument.

TXHEIGHT, Text height 'real' clipped in feature 'integer'

Explanation: The given height value is too large, or too small to be represented in UORs at the defined transformation.

User action: Either correct the defined transformation, or edit the text feature using LITES2, and try again.

TXTLEN, 'integer' characters in text - using 'integer' - in feature 'integer'

Explanation: There were too many characters in the text string in the named feature. The maximum number allowed (the second number given) will be output.

User action: None.

TXTNOFC, Level not defined for text feature

Explanation: No output level is defined for this text feature. Other messages should clarify the problem.

User action: Depends upon the other messages.

TXWIDTH, Text width 'real' clipped in feature 'integer'

User action: The given width value is too large, or too small to be represented in UORs at the defined transformation.

User action: Either correct the defined transformation, or edit the text feature using LITES2, and try again.

UNABLE, Unexpected end of line, or unexpected entry, for FC 'integer'

Explanation: This occurs if a line in the SDF is garbled. The line is ignored.

User action: Correct the SDF.

UNSETCAT, Category 'integer' does not have a level defined

Explanation: Text features are being output with TEXTLEVEL CATEGORY, but the current text feature has a category that has not been described in the SDF with a CATEGORY command. The text will be output on level 63.

User action: The data output is incorrect, so a new CATEGORY command should be added to the SDF to defined the level for this category, and the data should be processed again.

MESSAGES (FATAL)

Although these errors are classified as fatal, the program may still attempt to continue. However, any results produced should be regarded with great suspicion.

When it finishes, the program will set $STATUS to be SS$_ABORT.

BADEOPTR, Corrupt pointer to EO entry, in NO for layer 'integer'

> **Explanation:** This should not occur. The program will continue, ignoring the corrupt pointer. However, the IFF file is technically incorrect, and other problems may occur.
>
> **User action:** Investigate why the pointers are corrupt. They can be corrected by editing the file with LITES2 and exiting from it.

BADSTPTR, Corrupt pointer to first ST entry in feature 'integer'

> **Explanation:** When multiple feature codes are used, the program repeats the feature for each secondary feature code. To do this, it remembers the position of the first ST or ZS entry. This message indicates that the remembered position is wrong. This should never occur.
>
> **User action:** Report the error to Laser-Scan with as many details as possible.

HLPERR, Error using help file 'file-spec'

> **Explanation:** As it says. A further message should explain more.
>
> **User action:** Depends upon the other messages.

NOLICENCE, No licence obtained for /SCITEX qualifier.

> **Explanation:** The shared image which checks that the licence for using the /SCITEX qualifier has been obtained does not exist.
>
> **User action:** Report to Laser-scan if you require a licence or if you have a licence, but for some reason do not have the shared image.

NOLOGNAM, Logical name pointing to shared image not set up

> **Explanation:** The logical name which points to the SCITEX licence checking shared image has not been set up.
>
> **User action:** Execute the command file 'CONVERTINI.COM'

NOMD, No map descriptor in IFF file

> **Explanation:** The MAPDESCRIPTOR OFFSET command has been used, and this requires the program to find the map descriptor (MD) entry in each IFF file. All normal IFF files must have an MD entry, but this file does not.

User action: Use the IMP utility IPATCH to inspect the IFF file. The most probable cause of this error is an attempt to output a patch file. Alternatively, trace back the history of the file to determine what program might have produced it.

TOOMANYFC, Too many feature codes to fit in internal table (more than 'integer')

Explanation: This is given if there are too many feature codes defined in the SDF.

User action: If this is a problem, consult Laser-Scan to have the table size increased.

MESSAGES (BUGS)

These messages are also flagged as fatal, and should never occur. They represent problems with the coding of the program itself, and should be reported to Laser-Scan at once, with as many details as possible.

BADCMD, BUG - Unexpected command number 'integer'

BADLSTOP, BUG - Illegal line type 'integer' in feature 'integer' - using 1 (line)

BADMDARG, BUG - Unexpected MAPDESCRIPTOR subcommand - cmd no 'integer'

BADTXTLVARG, BUG - Unexpected TEXTLEVEL subcommand - cmd no 'integer'

BADTXTSZARG, BUG - Unexpected TEXTSIZE subcommand - cmd no 'integer'

BADUNITCMD, BUG - Unexpected UNITS subcommand found in SDF - cmd no 'integer'

BUG_1, BUG - Internally inconsistent feature type in feature 'integer'

LACTYPE, BUG - Illegal type of line characteristic 'integer' in feature 'integer'

MTXMIRROR, BUG - Illegal symbol cell matrix mirror flag 'integer' in feature 'integer'

NOCIRCLE, BUG - Impossible circle radius 'real' - feature 'integer' abandoned

PUTFLN, BUG - Error parsing filename 'file-spec', to abstract drawing name

SIFCNTNEG, BUG - Corrupted SIF definition table length 'integer' (searching for FC 'integer')

SYMMIRROR, BUG - Illegal symbol cell mirror flag 'integer' in feature 'integer'

TOOMANYCUR, BUG - 'integer' points (using 'integer') given for curve in feature 'integer'

TOOMANYPTS, BUG - 'integer' points (using 'integer') given for line in feature 'integer'

TXTMIRROR, BUG - Illegal text mirror flag 'integer' in feature 'integer'

CHAPTER 4

SIF2IFF UTILITY

UTILITY SIF2IFF

FUNCTION

SIF2IFF extracts data from a SIF command file and outputs it to an IFF file. The program must use a parameter file which must assign IFF feature codes to combinations of SIF graphical output commands and their associated characteristics. Users from CCG (Canadian Centre for Geomatics) can use a parameter file containing a lookup table of CCSM codes.

FORMAT

```
$ SIF2IFF SIF-file-spec IFF-file-spec
```

Command qualifiers	**Defaults**
/PARAMETER=file-spec	See text
/[NO]CCG	/NOCCG
/[NO]CELL	/NOCELL
/[NO]LOG	/NOLOG
/UNITS=integer	See text
/DIMEN=integer	See text
/[NO]OFFSET[=(coordinates)]	/NOOFFSET
/SCALE=integer	See text

PROMPTS

```
_SIF-file:          SIF-file-spec

_IFF-file:          IFF-file-spec
```

PARAMETERS

SIF-file-spec

- specifies the input SIF file or magnetic tape device. There is no default file specification.

IFF-file-spec

- specifies the output IFF file. Any part of the file-spec which is not supplied will be taken from the default specification 'LSL$IF:IFF.IFF'.

COMMAND QUALIFIERS

/CCG
/NOCCG

- Specifies that input SIF file and parameter file is in the format which CCG (Canadian Centre for Geomatics) uses.

/CELL
/NOCELL

- If a SIF file in CCG format is to be read from tape, specify /CELL

/LOG
/NOLOG

- When /LOG is present, the SIF2IFF utility will output diagnostic messages about any file creation and what is happening as it processes the input SIF file so that any errors may be traced.

/UNITS=integer

- specifies how many SIF UORs per metre on the ground, eg if 1000 UORS represents 1 metre then UNITS = 1000. The default value is 1000.

/DIMEN=integer

- specifies whether the map is in 2 dimensions or 3 dimensions. If no value is given the program looks for the dimension in the header field of the SIF file. If its not found there the default value is 2.

/OFFSET[=(X:real,Y:real)]
/NOOFFSET

- specifies that all the data in the IFF SIF file is to be shifted by 'Xreal' in X and by 'Yreal' in Y. Both keywords must be present if supplied. /NOOFFSET implies an offset of (0,0). The default is /NOOFFSET with the default keyword values.

/PARAMETER=file-spec

- specifies the IFF parameter file, which defines the IFF feature codes in terms of SIF graphic output commands and graphic characteristic commands. The default parameter file-spec is taken from 'LSL$IF:IFF.IFF'.

/SCALE=integer

- specifies the scale of the map to be entered in the MD (Map Descriptor) entry in the IFF file. The default value is 1.

DESCRIPTION

SIF2IFF reads ASCII SIF commands either from file or from magnetic tape, and writes an IFF file. It reads in a parameter file which must contain information which associates SIF graphical types and characteristics with IFF feature codes.

Parameter file

This file contains translations of SIF graphical types and characteristics into IFF feature codes. The format of the parameter file is described in the DATA PREPARATION - SIF2IFF section under the heading 'The SIF2IFF parameter file'.

Using the program

Normal use of the program is in batch mode - that is, running it via a batch command file, submitted with the SUBMIT command. It is also possible to run the program interactively.

It is probable that the symbol SIF2IFF will have been set up to include at least the /PARAMETER qualifier, so you may not need to specify this explicitly.

If /OFFSET is specified on the command line, care should be taken to ensure that there will be no loss of precision of the data on transfer to IFF; coordinates in IFF are accurate to approximately 7 significant figures. The keyword values specified (either explicitly or by default) with /OFFSET are the origin offset values which will be stored in the Map Descriptor (MD) entry, the primary purpose of the origin offset in IFF being to maintain accuracy over large areas.

How the program works

Once the program has been invoked and the command line successfully read, a summary of the inputs and outputs is typed. If a magnetic tape device has been specified as input, then the tape is read directly. The parameter file is then read and if read successfully, the IFF file is opened.

The data is then processed to produce the IFF file. The feature code for a particular combination of SIF graphical types and graphical characteristics is obtained from the parameter file.

For the translation the various SIF graphical types are treated as follows :-

- **Line strings (LST)** are output as linear features.

- **Arcs and circles (ARC,CIR)** are output as clockwise arcs, anti-clockwise arcs circumcircle arcs and full circumcircles.

- **Smooth curves (CUR)** are output as interpolated curves. The conditions for starting a new feature are the same as for lines.

- **Areas** are output as fill areas with fill patterns.

- **Symbols** are output as oriented or scaled symbols.

- **Text (TXT)** is output as text with a specified font.

When all applicable data has been output to the IFF file, the program will exit.

Errors which result in termination of the program will cause the output of appropriate messages. Unless nothing at all has been written to the IFF file, it will be closed and preserved.

IFF files

SIF2IFF produces a new style IFF file containing IFF HIstory (HI) and type 2 Map Descriptor (MD) entries. The latter will have its origin offset and scale fields set.

SIF OVR levels are translated into corresponding IFF layers with the same numerical value.

The program does not produce a layer containing a grid or corner ticks. If either of these are desired, the file should be merged using the IMP utility IMERGE, with a template IFF file produced by the IMP utility ISTART.

RESTRICTIONS

- The SIF continuation line which always starts with four blank characters is supported, but the CON/ continuation command is not supported.

- ACZ commands are interpreted as contour heights only and written to IFF as ancillary codes.

- Text in paragraphs is output as separate text lines.

- SIF point strings (PST) are not implemented.

- The IDE command is processed only when the /CCG qualifier is used.

- Trailing blanks in text strings are truncated.

- BST commands are ignored. Any commands within the BST command are output as individual features.

- UNITS of measurement are in metres/millmetres only.

EXAMPLES

```
$ SIF2IFF/PAR=SIF.FDF LSL$SIF:SIF.SIF SIF.IFF
Parameter file         : LSL$LOOKUP:SIF.FDF
SIF file               : LSL$SIF:SIF.SIF
IFF output file        : LSL$IF:SIF.IFF
%SIF2IFF-I-READPAR, reading parameter file
%LSLLIB-I-IFFOPENED, LSL$DATAROOT:[LSL.IFF]SIF.IFF;2 opened for write
%SIF2IFF-I-READSIF, reading SIF file
%SIF2IFF-I-IFFCLOSED, IFF file closed

 ELAPSED:    0 00:00:06.08  CPU: 0:00:01.00  BUFIO: 20  DIRIO: 34  FAULTS: 194
$
```

This is a typical example showing the use of SIF2IFF to produce an IFF file. The parameter file directory has been defaulted. Note that if the parameter file is fixed at a site, the symbol SIF2IFF could be set up to include the /PAR qualifier. A directory-spec has been given for the SIF file-spec as there is no default. A default directory-spec has been added to the IFF file-spec.

MESSAGES (INFORMATIONAL)

These messages give information only, and require no immediate action by the user. They are used to provide information on the current state of the program, or to supply explanatory information in support of a warning or error message.

IFFCLOSED, IFF file closed

> **Explanation:** This message appears if /LOG was specified on the command line, and confirms that the IFF file has been successfully closed.
>
> **User action:** None.

IFFDELETED, IFF file deleted

> **Explanation:** This message appears if /LOG was specified on the command line, and confirms that the IFF file has been closed and deleted. This will only happen if execution is terminated after the IFF file has been opened but before the header data has been successfully written.
>
> **User action:** None.

LINEINPAR, line 'number' of parameter file

> **Explanation:** This message always appears after an error has occurred while decoding a line in the parameter file, and it informs the user of the offending line number. The previous message will have been generated either by LSLLIB or by the program itself, and the line in question will be ignored.
>
> **User action:** If the error message was generated by LSLLIB, it most likely indicates an error in the parameter file which should be amended. Otherwise, see the relevant error message explanation.

LINEINSIF, line 'number' of SIF file

> **Explanation:** This message always appears after an error has occurred while decoding a line in the SIF command file, and it informs the user of the offending line number. The previous message will have been generated either by LSLLIB or by the program itself, and the line in question will be ignored.
>
> **User action:** If the error message was generated by LSLLIB, it most likely indicates an error in the SIF command file which should be amended. Otherwise, see the relevant error message explanation.

MAXPAT, user defined max line pattern number is 'number'

> **Explanation:** This message appears if the user has defined the logical name LSL$SIF2IFF_MAX_PATTERN as the maximum number of line patterns to allow. useraction
>
> **User action:**

NULLTXTSTR, blank text string found

Explanation: A null (blank) text string for the SIF TXT record was found in the SIF file. The LINEINSIF message indicating the SIF file line number will follow, and the program will continue to read the file, ignoring blank text.

User action: none.

READMAG, reading from tape

Explanation: This message appears if /LOG was specified on the command line, and informs the user that the tape drive is being read.

User action: None.

READPAR, reading parameter file

Explanation: This message appears if /LOG was specified on the command line, and informs the user that the parameter file is being read.

User action: None.

READSIF, reading SIF file

Explanation: This message appears if /LOG was specified on the command line, and informs the user that the SIF command file is being read.

User action: None.

UNSUPCMD, "'command'" commands are not supported

Explanation: This command is output if the /LOG qualifier is specified. The given command in the SIF file is not supported. The LINEINSIF message indicating the SIF file line number will follow, and the program will continue to read the file, ignoring this line. Only the first occurance of the unsupported command will be given.

User action:

MESSAGES (WARNING)

These messages are output when an error has occurred that can be corrected immediately by the user or that the program will attempt to overcome.

BADACZ, ACZ LO and HI unequal - using HI value

Explanation: The high and low values of active 'z' read from the SIF file are unequal, so the high value is taken as the contour height.

User action:

BADARCDIR, arc direction 'arcdir' out of range

Explanation: An invalid arc direction specifier has been read from the parameter file. Only values 0 and 1 are valid.

User action: Amend the arc direction in the parameter file, and if necessary run the program again.

BADAREA, area pattern 'pattern number' out of range

Explanation: An invalid area pattern specifier has been read from the parameter file. Only values 0 - 100 are valid.

User action: Amend the area pattern in the parameter file, and if necessary run the program again.

BADCOLOUR, line colour 'colourtype' out of range

Explanation: An invalid line colour specifier has been read from the parameter file.

User action: Amend the line colour in the parameter file, and if necessary run the program again.

BADFC, feature code 'fc' out of range

Explanation: An invalid feature code (outside the range 0 to 32767) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the feature code in the parameter file, and if necessary run the program again.

BADFONT, text font 'fonttype' out of range

Explanation: An invalid SIF fonttype (outside the range 0 to 255) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the SIF fonttype in the parameter file, and if necessary run the program again.

BADHEIGHT, text height 'textheight' out of range

Explanation: An invalid SIF text height (outside the range 1 to 100.0) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the text height in the parameter file, and if necessary run the program again.

BADLINE, line style 'linetype' out of range

Explanation: An invalid SIF linestyle (outside the range 1 to 7) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the SIF linetype in the parameter file, and if necessary run the program again.

BADPATTERN, line or area pattern 'pattern number' out of range 0 to 'max pattern number'

Explanation: An invalid SIF line pattern (outside the allowed range) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the SIF line pattern type in the parameter file, and if necessary run the program again.

BADPATTYP, wrong pattern type 'patterntype'

Explanation: An invalid SIF pattern type specifier (outside the range 0 to 5) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the SIF pattern type specifier in the parameter file, and if necessary run the program again.

BADPSIZE, text size 'textsize' out of range

Explanation: An invalid SIF text pointsize (outside the range 1 to 30) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the text pointsize in the parameter file, and if necessary run the program again.

BADSIFCMD, unknown command "'command'"

Explanation: The given command in the SIF file was not recognised. The LINEINSIF message indicating the SIF file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Check the documentation for a list of the supported commands. Some unsupported commands are not relavant to IFF. If you require any unsupported commands to be implemented report to Laser-Scan.

BADSYMNAM, invalid symbol name "'symnam'"

Explanation: A symbol name longer than 12 characters was read in the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the symbol name in the parameter file, and if necessary run the program again.

BADSYMTYPE, invalid symbol type "'symtype'"

Explanation: An invalid symbol type, neither 0 or 1 was specified. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the symbol type in the parameter file, and if necessary run the program again.

BADWEIGHT, line weight 'lineweight' out of range

Explanation: An invalid SIF line weight (outside the range 1 to 31) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the line weight in the parameter file, and if necessary run the program again.

BADWIDTH, line width 'textwidth' out of range

Explanation: An invalid SIF text height (outside the range 1 to 100.0) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the text height in the parameter file, and if necessary run the program again.

COORDNOTFOU, coordinate not found

Explanation: An X, Y, or Z coordinate was expected in the SIF file, but not found.

User action: Check that the SIF file has not been corrupted or incorrectly formatted.

COORDOUTRA, coord out of range in feature 'fsn'

Explanation: An X, Y, or Z coordinate was is outside the map range specified

User action: Check the coordinates in the SIF file.

EOLEXP, end of line was expected

Explanation: The end of a SIF line was expected but not found. The LINEINSIF message indicating the SIF file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Check that the SIF file has not been corrupted or incorrectly formatted.

NOARCDIR, arc direction not found

Explanation: An invalid arc direction specifier (outside the range -1 to 1) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the arc direction specifier in the parameter file, and if necessary run the program again.

NOARCFC, No FC for ARC feature with: lsty ''number'', lcol '%N', lwgt '%N', lpat '%N', patn ''string'', adir '%N'

Explanation: The current SIF ARC feature with the above combination of characteristics has not been assigned to a feature code in the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Assign a feature code for the above combination in the parameter file, and run the program again.

NOAREAFC, No FC for AREA feature with: lsty ''number'', lcol '%N', lwgt '%N', apat '%N', patn ''string''

Explanation: The current SIF AREA feature with the above combination of characteristics has not been assigned to a feature code in the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Assign a feature code for the above combination in the parameter file, and run the program again.

NOCIRCFC, No FC for CIRCLE feature with: lsty ''number'', lcol '%N', lwgt '%N', lpat '%N', patn ''string''

Explanation: The current SIF CIRCLE feature with the above combination of characteristics has not been assigned to a feature code in the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Assign a feature code for the above combination in the parameter file, and run the program again.

NODATA, IFF file does not contain any map data

Explanation: The IFF file produced by SIF2IFF does not contain any map data. Previous warning messages may indicate more plainly the reason.

User action: Check the SIF input file has all the correct data.

NOLINEFC, No FC for LINE/CURVE feature with: lsty ''number'', lcol '%N', lwgt '%N', lpat '%N', patn ''string''

Explanation: The current SIF LINE, or CURVE, feature with the above combination of characteristics has not been assigned to a feature code in the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Assign a feature code for the above combination in the parameter file, and run the program again.

NOOVRNUM, overlay number was not found

Explanation: Overlay number for the OVR command is either missing or corrupted in the SIF file.

User action: Amend the OVR command in the SIF file, and if necessary run the program again.

NOPATLNM, logical name %S ill defined

Explanation: The logical which defines the maximum number of line patterns allowed was badly defined. useraction

User action:

NOPSIZE, No POINTSIZE with: height ''number'', width '%I10'. Default POINTSIZE value of 1 used.

Explanation: The current SIF text with the reported combination height and width has not been assigned to a point size value in the parameter file. The default pointsize will be 1. The LINEINPAR message indicating the parameter file line number will follow.

User action: Assign a pointsize for the above combination in the parameter file, and run the program again.

NOSYMFC, No FC for SYMBOL feature with: symt ''number'', symn ''string''

Explanation: The current SIF SYMBOL feature with the above combination of characteristics has not been assigned to a feature code in the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Assign a feature code for the above combination in the parameter file, and run the program again.

NOSYMNAM, cannot read symbol name

Explanation: Symbol cannot be read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the symbol name in the parameter file, and if necessary run the program again.

NOTEXTFC, No FC for TEXT feature with: font ''number'', lcol '%N'

Explanation: The current SIF TEXT feature with the above combination of characteristics has not been assigned to a feature code in the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Assign a feature code for the above combination in the parameter file, and run the program again.

NUMNOTFOU, real or integer expected, not found

Explanation: A real or integer number was expected in the SIF file, but not found. The LINEINSIF message indicating the SIF file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Check that the SIF file has not been corrupted or incorrectly formatted.

RADNOTFOU, radius not found

Explanation: A radius value was expected when the CIR command in the SIF file The LINEINSIF message indicating the SIF file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Check that the SIF file has not been corrupted or incorrectly formatted.

SEPNOTFOU, seperator expected, not found

Explanation: A seperator was expected in the SIF file, but not found The LINEINSIF message indicating the SIF file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Check that the SIF file has not been corrupted or incorrectly formatted.

TEXNOTFOU, text expected, not found

Explanation: A text string was expected in the SIF file, but not found The LINEINSIF message indicating the SIF file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Check that the SIF file has not been corrupted or incorrectly formatted.

UNEXPCH, unexpected character ''char''

Explanation: An alphabetic character has been read from the parameter file where a digit was expected. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Edit the parameter file to correct the offending character, and if necessary run the program again.

UNKCMD, unknown command "'command'"

Explanation: The given command in the parameter file or SIF file was not recognised. The LINEINSPAR or LINEINPSIF message indicating the parameter or SIF file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Check that the parameter file or SIF file has not been corrupted. If not then this message could be due to an error in the program and should be reported to Laser-Scan.

UPCASE, error converting "'string'" to upper case

Explanation: An error occurred while converting the given string to upper case. A system message will follow, most likely indicating that the string has been truncated. Execution will continue, but problems may occur later.

User action: This message probably indicates a minor programming error, and should be reported to Laser-Scan.

MESSAGES (ERROR)

These messages indicate an error in processing which will cause the program to terminate. The most likely causes are a corrupt or otherwise invalid input file, or an error related to command line processing and file manipulation.

ERRCLOPAR, error closing parameter file 'file-spec'

Explanation: The parameter file could not be closed. An additional LSLLIB message will follow, giving the reason for the failure. If this is due to a system error, the system error message will also appear. The program exits immediately.

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed.

ERRCLOSIF, error closing SIF file 'file-spec'

Explanation: The SIF file could not be closed. An additional LSLLIB message will follow, giving the reason for the failure. If this is due to a system error, the system error message will also appear. The program exits immediately.

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed.

ERRINIMAG, error initialising tapedrive 'dev-spec'

Explanation: The magtape device specified could not be iniialised. An additional LSLLIB message will follow, giving the reason for the failure.

User action: Ascertain the cause of the failure from the messages given. There may be a a serious problem such as a system failure, and in that case the System Manager may have to be informed.

ERROPNPAR, error opening parameter file 'file-spec'

Explanation: The parameter file could not be opened. An additional LSLLIB message will follow, giving the reason for the failure. If this is due to a system error, the system error message will also appear. The program exits immediately.

User action: Ascertain the cause of the failure from the messages given. There may be a simple remedy, e.g. a change in the directory specification. Alternatively, this error could be due to a more serious problem such as a system failure, and in that case the System Manager may have to be informed.

ERROPNSIF, error opening SIF file 'file-spec'

Explanation: The SIF file could not be opened. An additional LSLLIB message will follow, giving the reason for the failure. If this is due to a system error, the system error message will also appear. The program exits immediately.

User action: Ascertain the cause of the failure from the messages given. There may be a simple remedy, e.g. a change in the directory specification. Alternatively, this error could be due to a more serious problem such as a system failure, and in that case the System Manager may have to be informed.

ERRRDMAG, error reading tape 'dev-spec'

Explanation: The magtape device could not be read. An additional LSLLIB message will follow, giving the reason for the failure. If this is due to a system error, the system error message will also appear. The program exits immediately.

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed.

ERRRDPAR, error reading from parameter file 'file-spec' at line 'number'

Explanation: The given line could not be read from the parameter file. An additional LSLLIB message will follow, giving the reason for the failure. If this is due to a system error, the system error message will also appear. The program exits immediately.

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed.

ERRRDSIF, error reading from SIF file 'file-spec' at line 'number'

Explanation: The given line could not be read from the SIF file. An additional LSLLIB message will follow, giving the reason for the failure. If this is due to a system error, the system error message will also appear. The program exits immediately.

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed.

ERRRWDMAG, error rewinding tape 'dev-spec'

Explanation: The magtape device specified could not be rewound. An additional LSLLIB message will follow, giving the reason for the failure.

User action: Ascertain the cause of the failure from the messages given. There may be a a serious problem such as a system failure, and in that case the System Manager may have to be informed.

IFFABORTED, cannot write to IFF file

Explanation: There has been an error while attempting to write to the IFF file, which has therefore been aborted. The cause may be a system error, or it may be a programming error. The message will be preceded by an IFF library message which will give more details, including a two-character code.

User action: Look up the IFF error code in the IFFLIB Reference Manual (part of the MAPPING package documentation) and take the appropriate action.

IFFCLOERR, cannot close IFF file

Explanation: The IFF file could not be closed. This may be due to a previous IFF error, or it may indicate a more serious system error. The message will be preceded by an IFF library message which will give more details, including a two-character code.

User action: Look up the IFF error code in the IFFLIB Reference Manual (part of the MAPPING package documentation) and take the appropriate action.

MISSKEY, missing keyword with /'qualifier'

Explanation: The given qualifier has a keyword missing which should have been specified. The program exits immediately.

User action: Reissue the SIF2IFF command again, this time supplying the appropriate keyword.

NODEVICE, no magtape device specified 'dev-spec'

Explanation: A magtape device was expected but not specified.

User action: Re-run the program making sure that a valid magtape device is specified.

NOTMNTFOR, magtape not mounted foreign 'dev-spec'

Explanation: The magtape device was specified is not mounted with /FOR.

User action: Make sure that the magtape device is mounted, then re-run the program.

MESSAGES (FATAL)

These messages indicate a severe error in processing, or some form of system failure, which has caused the program to terminate.

NORA, unable to patch RAnge - internal pointer corrupted

Explanation: After all of the data for an area or drawing has been output, the program attempts to patch the RAnge entry in the IFF file, to reflect the true range of the data. This error indicates that it was unable to find the RA entry again. This means that either the program or the IFF file has become corrupted. The program will abort.

User action: Please report this problem to Laser-Scan.

MESSAGES (OTHER)

In addition to the above messages which are generated by the program itself, other messages may be produced by the command line interpreter (CLI) and by Laser-Scan libraries. In particular, messages may be generated by the IFF library and by the Laser-Scan I/O library, LSLLIB. IFF library messages are introduced by '%IFF' and are documented in the IFF library users' guide. In most cases IFF errors will be due to a corrupt input file, and this should be the first area of investigation. If the cause of the error cannot be traced by the user, and Laser-Scan are consulted, then the output file should be preserved to facilitate diagnosis. LSLLIB messages are introduced by '%LSLLIB' and are generally self-explanatory. They are used to explain the details of program generated errors.