Laser-Scan Ltd.

STRUCTURE

Reference Manual

Issue 5.6 - 24-Dec-1997

Science Park, Milton Road, Cambridge, England CB4 4FY tel: +44 (0) 1223 420414

Document "STRUCTURE Reference"		Category "REFERENCE"
Document Issue 1.0	Andrew Morris, Tim Hartnall	03-Apr-1987
Document Issue 2.0	Tim Hartnall	15-Jun-1987
Document Issue 3.0	Andrew Morris	16-Sep-1987
Document Issue 3.1	Andrew Morris	18-Dec-1987
Document Issue 4.0	Andrew Morris	15-Apr-1988
Document Issue 4.1	Andrew Morris	23-Jun-1988
Document Issue 4.2	Andrew Morris	12-Dec-1988
Document Issue 4.3	Jon Barber	30-Aug-1989
Document Issue 4.4	Jon Barber	12-Mar-1991
Document Issue 4.5	Steve Townrow	09-Aug-1991
Document Issue 4.6	Jon Barber	21-Aug-1991
Document Issue 4.7	Jon Barber	07-Nov-1991
Document Issue 4.8	Steve Townrow	30-Mar-1992
Document Issue 4.9	Jon Barber	07-Aug-1992
Document Issue 5.0	Jon Barber/Bill James	10-Mar-1993
Document Issue 5.1	Jon Barber/Paul Hardy	25-May-1993
Document Issue 5.2	Jon Barber	15-Sep-1993
Document Issue 5.3	Jon Barber	13-Oct-1993
Document Issue 5.4	Jon Barber	07-Feb-1995
Document Issue 5.5	Jon Barber	22-Feb-1996
Document Issue 5.6	Matt Wenham	24-Dec-1997

CONTENTS

CHAPTER 3 MODULE RELHT

CHAPTER 4 MODULE ICASE

STRUCTURE documentation change record

Version 0.0 Various 02-Mar-1987

Provisional issue of STRUCTURE documentation.

Version 1.0 Andrew Morris, Tim Hartnall 09-Apr-1987

First customer issue of STRUCTURE documentation.

Version 2.0 Tim Hartnall 16-Jun-1987

ILINK /MAXPOINTS, /MAXPPF, /NFEATURES and /SAVED_PARAMETERS qualifiers removed. /LITES2 and /ONEARM qualifiers added. All references to ILINKSTAT.DAT files removed. Examples now reflect Version 2.0 ILINK log output. New message definitions added. /FCP and /FCC qualifier descriptions expanded.

Version 3.0 Andrew Morris 16-Sep-1987

ILINK chapter changed. New /MERGE qualifiers added. /CONTINUOUS qualifier removed. Examples now reflect Version 3.0 ILINK log output.

Version 3.1 Andrew Morris 18-Dec-1987

ILINK /VERIFY, /VRTOL and /LIST qualifiers added. Examples now reflect Version 3.1 ILINK log output.

Version 4.0 Andrew Morris 24-Mar-1988

ILINK (initial) CB handling and new /LCP qualifier added.

Version 4.1 Andrew Morris 23-Jun-1988

Several ILINK LLJOIN and CB handling bugs fixed, and new /ACP qualifier added.

Version 4.2 Andrew Morris 12-Dec-1988

New ILINK /PAC, /TOL and /FREE_ENDS qualifiers added.

Version 4.3 Jon Barber 30-Aug-1989

New ILINK message definition for ILINK_TOOMNYPTS.

QUADSCAN error messages changed.

Version 4.4 Jon Barber 12-Mar-1991

New ILINK message definition for ILINK_TOOMNYLAYERS.

New ILINK message ILINK_MISSOUTFILE to signal missing output file in command line when it is required.

Usage of the ILINK /FCP file feature code entries clarified.

Messages added to the RELHT chapter.

Handling of ST/ZS or CB coordinate strings in ILINK and RELHT described in the Introduction chapter of the STRUCTURE Reference Manual.

Version 4.5 Steve Townrow 09-Aug-1991

New ILINK qualifier /MINTOL documentated which enables the user to define the minimum distance between two points after which they are considered to be duplicate.

Version 4.6 Jon Barber 21-Aug-1991

ILINK qualifiers /STRUCTURE and /FREE_ENDS disallowed together.

Version 4.7 Jon Barber 07-Nov-1991

New ILINK qualifier /ABSOLUTE documentated which enables the user to output all coordinates as absolute values.

Version 4.8 Steve Townrow 30-Mar-1991

Documentation for new utility, ICASE, added to reference manual. ICASE is suited to the production of large scale schematic road casings and `area fills as often seen in road atlases. Given a junction structured IFF file containing road centrelines, a table of feature codes, priorities and road widths, the program will create an output IFF file containing road casings and area fills for the selected features. It is important to realise that the input road centreline data must be geometrically clean to prevent spurious results occurring.

Version 4.9 Jon Barber 07-Aug-1992

New ILINK qualifier /KEEP = FSN to enable the input Feature Serial Numbers to be used in the output features in the /BREAK and /MERGE processes.

Version 5.0 Jon Barber/Bill James 10-Mar-1993

New ILINK messages NOTMERGE and NOMERGE to signify places where there were either coordinate differences, or different numbers of points, in the sections suitable for merging (ie. starting and ending at the same node and emerging at the same angle). These sections are now being kept where they should possibly have been merged, and it is up to the user to check the data at the specified positions and to alter the input file accordingly (in particular problems have been known to arise from double-digitising).

The minimum resolution distance, normally set by /MINTOL if required, is now by default equal to the maximum coordinate value divided by 2.0E6, rather than by 0.5E6, resuting in fewer valid line sections being considered identical and subsequently lost.

Version 5.1 Jon Barber/Paul Hardy 25-May-1993

New ILINK qualifier /WARNING to output extra diagnostic messages, in particular NOMERGE, NOTMERGE and ORIGPNTDEL.

New ILINK message BADMINTOL to warn that the value given with the /MINTOL qualifier is below the dataset floating point arithmetic resolution, possibly leading to unpredictable results.

New ILINK message ORIGPNTDEL to indicate the deletion of an original point. This is only output with the /WARNING qualifier.

Version 5.2 Jon Barber 15-Sep-1993

New ILINK qualifier /SYMBOL to enable symbol (point) features to be moved on to line features. This is set by default, and /NOSYMBOL will restore the original ILINK behaviour. This qualifier may be used for the /LPJOIN process only.

Version 5.3 Jon Barber 13-Oct-1993

New ILINK qualifier /KEEP = PARENT to enable the input type 9 AC entries to be transferred from the input IFF file to the output.

Version 5.4 Jon Barber 07-Feb-1995

The use of the /FREE_ENDS qualifier is described better within the context of other ILINK modules.

The number of layers allowed has been increased to 500, along with the maximum number of fc pairs allowed in the FCP file.

Version 5.5 Jon Barber 22-Feb-1996

The restrictions within ILINK on the handling of invisible moves within features are described in the ILINK Reference Manual and the SPS.

Version 5.6 Matt Wenham 24-Dec-1997

A new flag, /KEEP=COLINEAR, has been added to ILINK. It is described in the ILINK Reference Manual.

PREFACE

Intended audience

This manual is intended for all users of the Laser-Scan STRUCTURE package running under the VAX/VMS operating system.

Structure of this document

This document is composed of 2 major sections.

The Introduction is an overview of STRUCTURE and is intended as a quick reference guide to the salient features of the STRUCTURE package.

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

MODULE - the name of the STRUCTURE module.

REPLACES - which of any older Laser-Scan programs the new STRUCTURE module replaces.

FUNCTION - a synopsis of what the modules does

FORMAT - a summary of the module 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.

DESCRIPTION - the definitive description of the module action.

EXAMPLES - annotated examples of module 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.

Where applicable, additional categories are available for some modules. Some modules, for example, have a "RESTRICTIONS" category.

Associated documents

For summary information about a specific STRUCTURE module see the STRUCTURE User Guide.

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>.
$ **ILINK JIM<CR>**	Command examples show all user entered commands in **bold** type.
$ **ILINK JIM<CR>** . . .	Vertical series of periods, or circles, mean either that not all the data that STRUCTURE 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 a file-specification, or in the syntax of a substring specification in a VMS assisnment statement).
'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).

Convention	Meaning
FSN 'integer' ('integer')	FSN followed by two integer arguments indicates an IFF feature serial number. The integer number enclosed in round brackets is the feature internal sequence number.
00003DE7	A hexadecimal address of a location within an IFF file. STRUCTURE modules express all IFF addresses using hexadecimal radix. The address is always padded with leading zeros to a standard field width of 8 characters.

Command line data types

STRUCTURE utilities use the VMS Command Line Interpreter (CLI) to get and parse the program command line. STRUCTURE utilities thus offer a VMS emulating user interface. Unfortunately the VMS Digital Command Language (DCL) does not support the real (or "floating point") data type. Many STRUCTURE utilities require real value arguments for the specification of tolerances and distances etc. To meet this requirement, Laser-Scan have developed an enhanced CLI based command line decoding mechanism. This enables the interpretation of numbers as either "real" or "integer". Throughout this document the number types are differentiated by the words 'integer' for integer numbers and 'real' for real (or "floating point") numbers.

STRUCTURE command line decoding operates in decimal radix.

CHAPTER 1

INTRODUCTION

INTRODUCTION

STRUCTURE is the Laser-Scan structured IFF processing package. IFF stands for **I**nternal **F**eature **F**ormat and is the Laser-Scan vector file format generated by LASERAID and other Laser-Scan mapping systems and used as the data structure throughout the Laser-Scan LAMPS system. IFF files are binary and cannot be manipulated directly using a text editor. The STRUCTURE package enables the user to perform a wide range of data structuring tasks related to the requirements of the automated mapping industry.

STRUCTURE - FEATURES

The STRUCTURE package consists of independent modules which together form a structured data manipulation system within an automated mapping environment. The modules which form the STRUCTURE package offer:

- common command syntax. Module command lines are decoded using the Command Line Interpreter as used by the VAX/VMS utilities.

- VMS format messages referenced using 32 bit condition code symbols.

- VMS DCL symbol $STATUS on image exit.

- comprehensive documentation in this reference manual using a style consistent with that used by Digital Equipment Corporation in their VMS utility manuals. The STRUCTURE User Reference documentation includes an explanation of the messages output by the modules together with suggested user action.

STRUCTURE and Existing LSL Utilities.

STRUCTURE is designed to replace the existing Laser-Scan utilities IFJ and CBA, as well as to create new facilities for the creation and manipulation of structured data.

STRUCTURE and IFF

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

STRUCTURE and DCL symbol $STATUS

Like VMS utilities, all STRUCTURE modules generate VMS format messages and set VMS DCL symbol $STATUS on image exit. This is a valuable feature as a non-interactive process can test the success of a preceding STRUCTURE module before proceeding. $STATUS will always be set to a VMS 32 bit condition code. Successful program execution will result in $STATUS being set to SS$_NORMAL. If an error occurred during STRUCTURE processing, SS$_ABORT of varying severities, or a VMS System or CLI (Command Line Interpreter) condition code will be used. The user may simply test $STATUS for TRUE or FALSE within a DCL command procedure. If $STATUS is TRUE then processing was successful. If it is FALSE, an error occurred during processing. For a detailed description of the uses of $STATUS see the VAX VMS document set.

STRUCTURE and three dimensional strings.

Structure module ILINK handles all data internally as CB entries and therefore can cope with IFF files containing ST, ZS or CB entries. Module RELHT (another STRUCTURE module) expects ST or ZS entries, and hence can handle ST's and ZS's, or CB entries. For both modules, the Z coordinates are ignored and just copied unchanged.

Getting started with STRUCTURE

Once logged in the user must give two commands to initialise the STRUCTURE package. Before the STRUCTURE package can be used DCL symbols and logical names must be assigned to enable the user to invoke the modules. This is done using a command procedure STRUCTUREINI.COM which is supplied as part of the STRUCTURE package. STRUCTUREINI itself will be defined as a DCL symbol at your site and should be invoked thus: (see PREFACE for explanation of presentation conventions)

```
      $ STRUCTUREINI<CR>
```

The STRUCTUREINI command invokes a command procedure which defines a DCL symbol (the module name) for each of the STRUCTURE modules. After using STRUCTUREINI the user need only type the symbol name to activate the selected module.

As an alternative to explicitly typing the STRUCTUREINI command each time the user wishes to use the STRUCTURE package, the STRUCTUREINI command may be placed in the user's login file, or in the site dependent default login file.

The second command which must be given before using the STRUCTURE package is the SI command. The SI command assigns the logical name LSL$IF: (or IF: for short) to the device-directory specification which contains the IFF file(s) that are to be manipulated. For example:

```
$ SI DUA3:[BUREAU.TRIALS.DIGITISING]
```

This will assign logical name LSL$IF: to the device and directory specification DUA3:[BUREAU.TRIALS.DIGITISING]

How to specify STRUCTURE command qualifier arguments

STRUCTURE utilities use the VMS Command Line Interpreter (CLI) to get and parse the program command line. STRUCTURE utilities thus offer a VMS emulating user interface. As many STRUCTURE utilities require floating point arguments to command qualifiers, Laser-Scan has developed an enhanced CLI based command line decoding mechanism. This enables the interpretation of numbers as either "real" or "integer".

The CLI allows the user to specify single and lists of integer qualifier arguments. If a list of arguments is specified, each argument must be separated by a comma and the whole list enclosed within parentheses, for example:

Single argument:

```
$ EXAMPLE/QUALIFIER=7<CR>
```

Where "EXAMPLE" is the command and /QUALIFIER is a qualifier to that command. There is one qualifier argument - 7

Argument list:

```
$ EXAMPLE/QUALIFIER=(2,5,8,9,10,11,12,13,14)<CR>
```

Where "EXAMPLE" is the command and /QUALIFIER is a qualifier to that command. There are 9 qualifier arguments within the argument list.

Integer value ranges

While developing the floating point command line data type (see Preface) it was recognised that there is a need for numeric range decoding within a VMS emulating command line. Argument ranges are specified with the syntax:

n:m

Where n is the lower limit of the range and m is the upper limit of the range (inclusive).

Such ranges are expanded in full. A maximum of 1024 arguments can be specified to any one command qualifier.

If we take our example argument list used above, i.e:

```
$ EXAMPLE/QUALIFIER=(2,5,8,9,10,11,12,13,14)<CR>
```

and now use the Laser-Scan argument range decoding mechanism:

```
$ EXAMPLE/QUALIFIER=(2,5,8:14)<CR>
```

we see that a more compact command line results but yields the same arguments. This is clearly an advantage in an IFF map processing environment where a single file could contain hundreds of attributes which the user may wish to reference via command line arguments.

Other examples are:

```
$ EXAMPLE/QUALIFIER=2:9<CR>
```

This yields 8 integer arguments: 2, 3, 4, 5, 6, 7, 8, and 9

```
$ EXAMPLE/QUALIFIER=:8<CR>
```

This yields 9 integer arguments: 0, 1, 2, 3, 4, 5, 6, 7, and 8

If when ranges are decoded, a qualifier has more than 1024 arguments the Laser-Scan LSLLIB library issues the error message:

```
%LSLLIB-E-RESPARSOVF, result of parse overflowed buffer
```

and program execution is terminated.

Floating point value ranges

Floating point value ranges are decoded in a different manner to integer value ranges. Instead of expanding the range to yield all its component integer values the command decoder merely leaves the range as a lower limit and an upper limit. Processing then takes account of any possible value lying between these limits (inclusive).

For example:

```
$ EXAMPLE/HEIGHT=(23.5:110.2)<CR>
```

Select all features having a height which lies within the range 23.5 to 110.2 inclusive.

CHAPTER 2

MODULE ILINK

MODULE ILINK

REPLACES IFJ, CBA

FUNCTION

ILINK is an IFF link-node structuring and manipulation utility. ILINK also incorporates powerful geometry tidying capabilities.

ILINK provides the following functions:

- **link-node structure** generation.
- **alignment of lines with lines** (ie. feature alignment).
- **merging of duplicate line sections into single features.**
- **joining of line ends to lines** (viz. joining feature ends to any point along a feature, including points along line segments between original feature vertices).
- **joining of feature ends to ends**.
- **breaking features into separate features where they cross.**

One-point features are treated as zero length linear features.

Structured data can be used for many network processing operations, such as polygon generation, route planning, map colouring and data-base compilation.

FORMAT

```
$ ILINK input-file-spec  [output-file-spec]
```

Command qualifiers	**Defaults**
/[NO]ABSOLUTE	/NOABSOLUTE
/ACP=file-spec	/ACP=SYS$DISK:[]ACP.ACP;0
/BPF=real	/BPF=2
/BREAK	See text.
/EXTOL=real	/EXTOL and /JNTOL values made equal.
/FCC=file-spec	/FCC=SYS$DISK:[]FCC.FCC;0
/FCP=file-spec	/FCP=SYS$DISK:[]FCP.FCP;0

/FREE_ENDS=(FC:n,LAYER:n)	See text.
/FRT=file-spec	/FRT=LSL$FRT:FRT.FRT;0
/JNTOL=real	/JNTOL=0.0
/KEEP=COLINEAR	See text.
/KEEP=FSN	See text.
/KEEP=PARENT	See text.
/LAYERS=integer[,...]	All layers.
/LCP=file-spec	/LCP=SYS$DISK:[]LCP.LCP;0
/[NO]LIST[=file-spec]	/NOLIST
/[NO]LITES2[=file-spec]	/NOLITES2
/LLJOIN	See text.
/LPJOIN	See text.
/MERGE	See text.
/MINTOL=real	See text.
/[NO]MONITOR	/NOMONITOR
/PAC=file-spec	/PAC=SYS$DISK:[]PAC.PAC;0
/[NO]PARENT	/NOPARENT
/PPJOIN	See text.
/[NO]PROJECT	/PROJECT
/SHRFC=integer	/SHRFC=999
/SORTARMS	See text.
/STRUCTURE	See text.
/[NO]SYMBOL	/SYMBOL
/TOL=file-spec	/TOL=SYS$DISK:[]TOL.TOL;0
/[NO]VERIFY	/NOVERIFY
/[NO]VERTEX	/NOVERTEX
/VRTOL=real	/VRTOL=0.0
/[NO]WARNING	/NOWARNING

PROMPT

```
_Input_file:             input-file-spec
_Output_file:            output-file-spec
```

PARAMETERS

Input-file-spec

- specifies the IFF file which is to be tidied or structured. Any part of the file specification which is not supplied will be taken from the default specification 'LSL$IF:IFF.IFF'. The input file is required.

Output-file-spec

- specifies the IFF file which is to be output. Any part of the file specification which is not supplied will be taken from the input file specification with the version number incremented by one, unless the output file is to be Junction Structured, in which case the default output file extension will be '.IFJ'. If the output file is not given when required by the particular application, an error message will result and processing will stop.

COMMAND QUALIFIERS

/ABSOLUTE
/NOABSOLUTE (default)

- specifies that any coordinates are to be output as absolute values, with ABSOLUTE x y commands throughout any LITES2 guidance file if requested with the /LITES2 qualifier. By default, all coordinates are the usual LITES2 values, with POSITION x y commands in any guidance file.

/ACP=file-spec

- Specifies a text file which defines Ancillary Code pairs for swapping when merged features run in the opposite direction to the feature they are merged into. Missing components of the /ACP file specification are taken from the default specification SYS$DISK:[]ACP.ACP;0.

  In the absence of the /ACP qualifier, all AC entries are copied into the shared feature exactly as they are found in the feature being merged. If the /ACP qualifier is given, AC pairs which are found in the /ACP file are swapped.

  The /ACP qualifier may only be used in conjunction with the /MERGE process qualifier.

The AC pair definition file is an ASCII text file with the following record format, in which no two ACs are equal:

```
       'AC1'      'AC2'
```

where:

'AC1' is the AC code (between 1 and 32767) with which AC2 is to be swapped when AC2 occurs in a feature which is merged into some other feature which runs in the opposite direction (only the AC code is swapped, AC values are unchanged).

'AC2' is the AC code with which AC1 is to be swapped when AC1 occurs in a feature which is merged into some other feature which runs in the opposite direction.

Note that ILINK does not check that the value types in each AC pair are equal. It is up to the user to ensure that these are consistent.

For example, an ACP file containing the lines:

```
!
! Comment  - example ACP file
!
!     swap:     with:
         4       5
         7       6
!
```

/BPF=real

- specifies a maximum for the average number of breaks that will be introduced into a feature during processing. The value supplied with the /BPF qualifier is needed to calculate memory requirements as ILINK memory is allocated dynamically.

  By default the number of breaks is assumed to be 2.0.

  The /BPF qualifier may be used with process qualifiers /BREAK, /LLJOIN, /LPJOIN and /MERGE only.

/BREAK

- breaks features into separate features at every feature intersection point.

  Features will be broken only where existing features already touch or cross (see Description section, Fig. 1).

/EXTOL=real

- specifies maximum vector EXtension TOLerance. This is the maximum distance by which one vector may be extended or truncated to meet another. If the /EXTOL distance is exceeded then a vector end may be moved off the original line of the vector in any direction, to a point within /JNTOL tolerance. This will cause the end of the vector to rotate (see Figs 3 and 5).

  The /EXTOL qualifier may be used with process qualifiers /LPJOIN and /PPJOIN only.

  The /EXTOL value must be expressed in IFF units.

/FCC=file-spec

- specifies a text file which specifies replacement feature-codes which are to be given to shared features with particular feature-code combinations. Missing components of the /FCC file specification are taken from the default specification SYS$DISK:[]FCC.FCC;0.

  The file has the following format for each FC-combination:

```
        FCC     'shared feature code'   [FC list]
        [FC list]
```

  where:

  FCC signals the start of a new FCC definition

  'shared feature code' is the shared feature code to be used to identify shared features having a combination of the listed feature codes.

  FC list is a list of the feature codes. Feature codes must lie in the range 0 to 32767 inclusive.

  For example:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Example FCC file
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! FCC record      Shared feature     Feature Codes to
! identifier      code               be combined
     FCC          101                11,13,35:39 ! note that ranges are allowed
!
! FCC record      Shared feature     Feature codes to
! identifier      code               be combined
     FCC          102                2:6,91,108,220:230
!
! Feature codes to be combined to make shared feature with FC 101 (continued)
     240:242,307                ! notice that continuation lines are permitted
!
! FCC record      Shared feature
! identifier      code

     FCC          103           ! new FCC definition
     14,77,90,210               ! feature codes to be combined
     214,239                    ! (continued)
```

Note that comments may be delimited by a '!' character.

ILINK would decode the FCC definitions in the above example file as follows:

Shared feature code 101 will be applied when shared features are made of a combination of features having FC's 11, 13, 35, 36, 37, 38 and 39

Shared feature code 102 will be applied when shared features are made of a combination of features having FC's 2, 3, 4, 5, 6, 91, 108, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229 and 230

Shared feature code 103 will be applied when shared features are made of a combination of features having FC's 14, 77, 90, 210, 214 and 239

Shared features with feature-code combinations not specified in the file specified with the /FCC qualifier will be given the feature code specified by the /SHRFC qualifier (default value 999).

The /FCC qualifier may be used with process qualifier /MERGE only.

/FCP=file-spec

- specifies a text file which defines feature-code pairs for processing. Missing components of the /FCP file specification are taken from the default specification SYS$DISK:[]FCP.FCP;0.

If a /FCP file is specified, only combinations of features which have feature-codes which form one of the defined feature code pairs are processed. All other features in the input file are simply copied to the output file.

Selection of features for processing may also be done by IFF layer and by layer-pairs (see Command Qualifiers /LCP and /LAYER). An example of feature selection is given in the Examples section.

In the absence of the /FCP, /LCP or /LAYER qualifiers, all features are processed, except those in layer 0.

If the /FCP qualifier is given in conjunction with /LAYER, then only features from within the layers specified with feature codes corresponding to /FCP feature code pairs will be processed.

The /FCP qualifier may not be used in conjunction with the /LCP qualifier.

During feature alignment, positioning will be from first code onto the second, unless codes are equal, in which case it will be from lower internal feature sequence number to higher.

Note that due to the internal re-arrangement of features, features with the first column FC will be pulled onto other features with the same FC.

The feature code pair definition file is an ASCII text file with the following record format:

```
      'FC from'      'feature-code'
```

where:

'FC-from' is a feature code

'feature-code' is either an individual feature code, or a feature code range. Feature codes or feature code ranges are separated by commas, tabs or spaces. Feature codes must lie in the range 0 to 32767 inclusive.

For example, an FCP file containing the lines:

```
!
! Comment  - example FCP file
!
!      Move
!      from:      to:
        20      10:12,30
        10      12,40:43,60
!
```

This file specifies FC pairs:

```
Move
from  to
( 20, 10)
( 20, 11)
( 20, 12)
( 20, 30)
( 10, 12)
( 10, 40)
( 10, 41)
( 10, 42)
( 10, 43)
( 10, 60)
```

/FREE_ENDS[=(FC:integer,LAYER:integer)]

- specifies that all free-ends (1-arm nodes) are to be found and reported.

If /LITES2 is specified a message is also sent to the LITES2 command file.

If either of the FC or LAYER process qualifiers are given then all free-end nodes will be output as point features into a separate final IFF layer. The layer number, and the feature code for each free-end feature, may be specified. The layer will default to the highest layer number in the file before this layer, plus one. The feature code will default to 0.

/FRT=file-spec

- specifies an FRT (Feature Representation Table) file from which ILINK can determine which feature codes have a graphical type which is suitable for ILINK processing.

  Missing components of the /FRT file specification are taken from the default specification LSL$FRT:FRT.FRT;0.

  Graphical types which are unsuitable for ILINK processing are:

  o 2 - clockwise arc

  o 3 - anticlockwise arc

  o 4 - circumcircle arc

  o 5 - full circumcircle

  o 6 - interpolated curve

  o 11 - symbol string - (processed only with /LLJOIN - see Description sub-section "ILINK processing options - detailed operation").

  If no FRT file is specified then it is up to the user to select out features having unsuitable graphical types using the /FCP and/or /LAYERS qualifiers as necessary.

  For a detailed description of FRT file contents and layout see the "FRT User Guide".

/JNTOL=real

- specifies JoiN TOLerance - the distance apart below which points may be moved together during alignment or junction formation, (default 0.0).

  With the /LPJOIN process, if both points are from the same feature then they will only be joined if they are separated by more than the /JNTOL tolerance along that feature.

  The /JNTOL value must be expressed in IFF units.

  The /JNTOL qualifier may be used with process qualifiers /LLJOIN, /LPJOIN and /PPJOIN only.

/KEEP=COLINEAR

- by default, points which are colinear, or which are within 1/10 of MINTOL of being colinear are deleted from the output of ILINK. When using /BREAK, this can result in points from the original data being deleted, rather than those which have been inserted. /KEEP=COLINEAR prevents colinear or near-colinear points from being deleted, whatever their origin.

/KEEP=FSN

- specifies that the parent Feature Serial Numbers (FSN's) are to be used in the output features in the /BREAK and /MERGE processes. Without this qualifier, the new output FSNs will be created sequentially.

/KEEP=PARENT

- specifies that the parent AC type 9 entries are transferred from the input to the output IFF file. Without this qualifier, these entries are ignored in the transfer in preparation for the creation of new type 9 ACs.

/LAYERS=integer[,...]

- specifies layers from which features are to be taken for processing. Features which lie in unselected layers are simply copied from the input file to the output file.

  /LAYERS values must lie in the range 0 to 32767.

  By default ILINK will process features in all layers except layer 0 which is reserved for registration features and grids. Layer 0 is always copied, unprocessed, to the output file unless explicitly included in the /LAYER value list.

/LCP=file-spec

- specifies a text file which defines layer-pairs for processing. Missing components of the /LCP file specification are taken from the default specification SYS$DISK:[]LCP.LCP;0.

  If a /LCP file is specified, only combinations of features from layers which are from one of the defined layer-pairs are processed. All other features in the input file which do not come from layers which form one of the defined layer-pairs are simply copied to the output file.

  Selection of features for processing may also be done by feature-code-pairs or by IFF layer (see Command Qualifiers: /FCP, /LAYER).

In the absence of the /LCP, /FCP or /LAYER qualifiers, all features are processed, except those in layer 0.

The /LCP qualifier may not be used in conjunction with the /FCP or /LAYER qualifiers.

During feature alignment, positioning will be from first layer onto second, unless layers are equal in which case it will be from lower internal feature sequence number to higher.

The layer-pair definition file is an ASCII text file with the same format as for the /FCP file, but with feature-codes replaced by layers.

/LIST[=file-spec]
/NOLIST (default)

- directs ILINK output to file-spec instead of to SYS$OUTPUT. By default the LIST file specification is parsed against that of the input IFF file but with the substitution of the current directory and extension '.LIS'. For example, if the input IFF file is called LSL$IF:TST.IFF then the default LIST file is SYS$DISK:[]TST.LIS.

LITES2 command sequences produced by ILINK take the following typical form:

```
%MESSAGE  End nodes lie within /JNTOL tolerance - link deleted
%POSITION     765.3     456.89
%PING
%RESPOND
%ABANDON
%ABANDON
```

/LITES2[=file-spec]
/NOLITES2 (default)

- creates a LITES2 command file to take the user to potential errors. Messages generated to report a potential error are incorporated in the file. By default the LITES2 command file specification is parsed against that of the input IFF file but with the substitution of logical name LSL$LITES2CMD: and the extension '.LCM'. For example, if the input IFF file is called LSL$IF:TST.IFJ then the default LITES2 command file is LSL$LITES2CMD:TST.LCM.

/LLJOIN

- brings parts of features together where any part of one feature (or point feature) comes within /JNTOL tolerance of any part of any other feature (or point feature) (see Description section, Fig. 2).

No feature will be joined with any part of itself.

If /LLJOIN is used with very small /JNTOL tolerance values, "pockets" or "sausage strings" may result between aligned sections. If too large a tolerance is used with /LLJOIN, an excessive processing time will result. It is important to give careful consideration to the tolerance that will be used with /LLJOIN.

/LPJOIN

- forms junctions where feature ends and/or point features almost touch lineal features anywhere along their length (see Description section, Fig. 3).

/MERGE

- merges duplicate feature sections into single features (see Description section, Fig. 4).

/MINTOL=real

- specifies a minimum distance tolerance. Any points closer than this tolerance will be considered duplicate and will be merged. This will have the effect of filtering the data as points can be lost in the processing. If the qualifier is not specified on the command line, the tolerance will be the maximum range (RA) coordinate divided by 2000000.

NOTE

> This minimum tolerance can effect the other tolerances EXTOL, JNTOL and VRTOL. These tolerance values can be set to be larger than MINTOL but if they are defined to be less, they will default to the value of MINTOL.

The /MINTOL value must be expressed in IFF units.

/MONITOR
/NOMONITOR (default)

- activates a progress display for the more time consuming operations. Even if /MONITOR is specified on the command line, its action is automatically disabled when the program is not being run in interactive mode.

/PAC=file-spec

- specifies that new points inserted by ILINK between original points are to receive point attribute codes according to information given in the specified text file, missing components for the specification of which are taken from the default SYS$DISK:[]PAC.PAC;0.

  The file has format as in the following example:

```
! text from '!' to end of line is ignored
! attribute name
   CAPTUREXY
! original attribute code   new attribute code
    56                            72
    52                   ! default = original code
    54                            76
```

  where:

  The code used for a new point inserted between two original points will be the new-code corresponding to the original code which comes first in this table. The order of lines in the /PAC file is therefore significant. For example, if lines are in order of increasing accuracy associated with each attribute code, then the new code used will correspond to the original code for the least accurate neighbour.

  The /PAC qualifier may be used with the process qualifiers /BREAK, /LLJOIN and /MERGE only, and must be accompanied by the /FRT qualifier, because the capture name code must be looked up in the FRT file.

  Any point attribute code encountered in the IFF file which was not in the /PAC file will be given new code = original code.

/PARENT
/NOPARENT (default)

- specifies that type 9 ACs (Ancillary Code entries) should be used in shared features to identify the particular feature from which the shared feature is derived. A type 9 AC is output to the shared feature for each original feature which shares the vector string in the shared feature. For further details of shared features see the Description sub-section "ILINK and shared or broken features".

  The /PARENT qualifier may be used with process qualifiers /BREAK and /MERGE only.

/PPJOIN

- forms junctions where lineal feature ends and/or point features almost touch (see Description section, Fig. 5).

/PROJECT (default)
/NOPROJECT

- if there are just two feature end-points that are to be joined, they are projected together. Where there are more than two, they are moved to their "centre of gravity" or mean position.

  If /NOPROJECT is specified, all feature ends at a junction are moved to the position of the feature end with the highest feature priority.

  The /PROJECT qualifier may be used with process qualifier /PPJOIN only.

/SHRFC=integer

- specifies the feature-code to be used to distinguish features shared by more than one input feature. The /SHRFC feature code value is only used when an explicit shared feature code is not specified in the /FCC file. The /SHRFC value must lie in the range 0 to 32767. The default /SHRFC value is 999.

  The /SHRFC qualifier may be used with process qualifier /MERGE only.

/SORTARMS

- specifies that the arms in the IFF JB (Junction Block) node-to-link pointer entries in the Junction Structured input IFF file should be sorted in order of increasing orientation angle (measured anticlockwise from the positive x-direction). ILINK works on the input IFF file in situ and does not alter it in any way, other than to reorder the contents of JB entries.

/STRUCTURE

- specifies that feature end-point connectivity (link/node structure) should be determined and that this should be written into IFF Junction Structure entries in the IFF output.

  An implicit /PPJOIN process is carried out with a near-zero join-tolerance to determine end-point connectivity. The nodes in the link/node structure will be at existing feature-ends and at point-features only.

  Every feature in a valid Junction Structured IFF file must be a part of that structure. Therefore, **input features not selected for structuring (through use the of /FCP, /FRT or /LAYERS qualifiers) will be omitted from the output file.**

/SYMBOL (default)
/NOSYMBOL

- specifies that in the process specified by the /LPJOIN qualifier, symbol or point features should be moved onto the line features, subject to the feature selections operating and other qualifiers such as /VERTEX.

  If /NOSYMBOL is specified, symbol features are fixed and line features moved onto them.

  The /SYMBOL qualifier may be used with process qualifier /LPJOIN only.

/TOL=file-spec

- Specifies that a tolerance is to be applied to each point according to its point attribute code. These attribute codes, the JNTOL (join) and extension (EXTOL) tolerances which are to be applied to them, and the attribute name, are all contained in the specified text file, missing components for the specification of which are taken from the default SYS$DISK:[]TOL.TOL;0.

  The file has format as in the following example:

```
! text from '!' to end of line is ignored
! attribute name
   CAPTUREXY
! attribute code   join tolerance   extension tolerance
    56                   1                2.5
    52                   2         ! default EXTOL = JNTOL
    54           ! default JNTOL from /JNTOL qualifier
    64                   5.5              0
    63                   0
```

  The /TOL qualifier may be used with the process qualifiers /LPJOIN and /PPJOIN only, and must be accompanied by the /FRT qualifier, because the capture name code must be looked up in the FRT file.

  Any point attribute code encountered in the IFF file which was not in the /TOL file will be given tolerances from the /JNTOL and /EXTOL qualifiers.

/VERIFY
/NOVERIFY (default)

- specifies that a message should be sent to the LITES2 command file for each /LPJOIN or /PPJOIN join which is across a distance within the range VRTOL to JNTOL.

  No output file will be created.

  The /VERIFY qualifier may be used with the process qualifiers /LPJOIN and /PPJOIN only.

/VERTEX
/NOVERTEX (default)

- specifies that in the process specified by the /LPJOIN qualifier, priority should be given to joining feature ends to original feature vertices over joining to any inter-vertex join position along a vector.

  The /VERTEX qualifier may be used with process qualifier /LPJOIN only.

/VRTOL=real

- specifies VeRification TOLerance (default 0.0) - the lower bound of the join distance verification range (VRTOL to JNTOL) within which each /PPJOIN or /LPJOIN join will be counted. The distribution of join distances between ten equal subintervals over this range is finally displayed as a histogram.

  The /VRTOL value must be expressed in IFF units.

  The /VRTOL qualifier may be used with process qualifiers /LPJOIN and /PPJOIN only.

/WARNING
/NOWARNING (default)

- outputs extra diagnostic messages, in particular NOTMERGE, NOMERGE and ORIGPNTDEL.

RESTRICTIONS

o ILINK requires that a record for every feature in the input file which has been specified for processing must share virtual memory simultaneously.

  For the /PPJOIN and /STRUCTURE processes, this record has a short fixed length for every feature, but for the /LLJOIN, /MERGE, /LPJOIN and /BREAK processes, it must also include the coordinates of every feature point.

  The maximum size of IFF file which can be processed by ILINK is therefore dependant on the process required and the size of the user's virtual-page-count limit.

- o Exactly one process must be specified on each run. The process qualifiers which may be specified with each process are given in the table below:

'*'=Legal '.'=Illegal	PROCESS QUALIFIER																					
PROCESS	ACP	BPF	EXTOL	FCC	FCP	FRT	JNTOL	LAYERS	LCP	LIST	LITES2	MINTOL	MONITOR	PAC	PARENT	PROJECT	SHRFC	TOL	VERIFY	SYMBOL	VERTEX	VRTOL
BREAK	.	*	.	.	*	*	.	*	*	*	*	*	*	*	*	.	.	.	.	.	.	.
FREE_ENDS	.	.	.	.	*	*	.	*	*	*	*	*	*	.	.	.	.	.	.	.	.	.
LLJOIN	.	*	.	.	*	*	*	*	*	*	*	*	*	*	.	.	.	.	.	.	.	.
LPJOIN	.	*	*	.	*	*	*	*	*	*	*	*	*	.	.	.	.	*	*	*	*	*
MERGE	*	*	.	*	*	*	.	*	*	*	*	*	*	*	*	.	*	.	.	.	.	.
PPJOIN	.	.	*	.	*	*	*	*	*	*	*	*	*	.	.	*	.	*	*	.	.	*
SORTARMS	.	.	.	.	.	.	.	.	.	*	.	*	.	.	.	.	.	.	.	.	.	.
STRUCTURE	.	.	.	.	*	*	.	*	*	*	*	*	*	.	.	.	.	.	.	.	.	.

- o ILINK will not retain invisible moves within features, but instead will create independent features for each visible section.

DESCRIPTION

ILINK is an IFF geometry tidying and structuring utility.

The following geometry tidying operations are available:

- o Feature alignment. For example, where a polygon map has been digitised as a series of closed polygons and these are required to be aligned.

- o Feature merging. For example, where a series of closed polygons which are exactly aligned are required to have duplicate sections merged into single shared-features. joining

- o Feature-end to feature joining. This process is useful when feature ends fall just short of, or project just over other features, or where features are required to move onto point features which are close to them. For example, railway lines being moved so as to pass exactly through railway stations. joining

- o Feature-end to feature-end joining. For example, where a digitised network such as a road map needs to have feature ends within some small tolerance brought together to arrive at a unique point

- o Breaking features where they cross. For example, where a wire-frame perspective projection is required to be broken into separate line features between every point where lines cross, so that hidden line sections can subsequently be edited out manually.

- o Junction structuring. This option is useful when the link-node structure resulting from the ends with ends joining operation described above, is required for subsequent processing. For example, the compilation of a structured data base or route planning,

- o Junction Structure 'arm' sorting. The arms in the IFF JB (Junction Block) node-to-link pointer entries are sorted (in situ) into order with orientation angle (measured anticlockwise from the positive X-direction) increasing.

ILINK may be used to operate on a subset of the features in an IFF file by use of the /LAYERS and /FCP qualifiers. Only features which fall within specified layers and which have a specified feature code will be processed. All other features will be transferred from input to output IFF files unchanged. All IFF Junction Structure entries present in the IFF input will be ignored. These qualifiers may not be used with the /SORTARMS process.

The default ILINK action is to operate on **ALL** features in the IFF file except those which lie in IFF layer 0. Layer 0 is reserved for registration features and grids, but will be processed if it is explicitly included in the /LAYERS qualifier value list. Note that there is an upper maximum to the number of different layers of 500.

Any NO entry defining the same layer as the current one being read is ignored in order not to use up this layer limit too soon, but this will only work for sequential identical layers, and later reappearance of a previously encountered layer after another new layer has been started will count as a different layer.

ILINK will only process "line-strings". Features whose feature-code, as given in the FRT table specified with the /FRT qualifier, do not have graphical type 1 ("line-string"), will normally be rejected from processing (if selected) and transferred from the input file to the output file unchanged.

The only exception to this rule is that when the /FRT qualifier is used with /LPJOIN, features with graphical type 11 (symbol-string) **will** be processed and will be treated correctly as symbol-strings (ie. strings of point-features), rather than as line-strings.

Compound features (ie. features containing more than one pen-up ST entry), will be separated into independent parts on input, and will not be reassembled before output. The reason for this is that these parts themselves may be broken into further parts during processing, so that it is not clear which parts, if any, should be reassembled with which.

It is important to realise that although ILINK is a very powerful geometry idealisation utility, it cannot totally replace the skills of an experienced cartographer.

The quality of the ILINK output IFF file is directly proportional to the quality of the data in the input IFF file. If the input file quality is so poor that an excessively large /JNTOL tolerance has to be used, no guarantee of the aesthetic appeal of the output data can be made. Line ends may be aligned quite correctly in relation to a set of mathematical rules, but if ILINK is given large tolerances, the line end combinations may reflect cartographic nonsense. The user should determine and rectify the cause of poor input data early in a production flowline, to save unnecessary frustration and wasted ILINK processing time.

This page to be replaced by Fig. 1

This page to be replaced by Fig. 2

This page to be replaced by Fig. 3

This page to be replaced by Fig. 4

This page to be replaced by Fig. 5

ILINK processing options - detailed operation

OPTION 1 Feature breaking - /BREAK

- The ILINK /BREAK process breaks features into separate features at every feature intersection point.

Junctions are created and features broken only where existing features already touch or cross (see Description section, Fig. 1).

OPTION 2 Feature Free Ends - /FREE_ENDS

- The ILINK /FREE_ENDS process reports feature free ends. This is a separate ILINK process, similar to the use of /BREAK or /STRUCTURE, involving the use of very small tolerances, and therefore changes in the feature geometry are possible.

The data file resulting from /FREE_ENDS processing is created from the input data with the specified /MINTOL value. As a result it treats as 'not free' any end points which would get pulled to a common junction, and any points lying within MINTOL of each other may be moved.

This option, like /BREAK or /STRUCTURE, should only be used after other ILINK options, over which there is more control over the output.

OPTION 3 Feature alignment - /LLJOIN

The /LLJOIN qualifier causes ILINK to bring parts of features together where any part of one linear feature (or point feature) comes within tolerance of any part of any other linear feature (or point feature). (See Fig. 2).

The tolerance used to determine whether a feature should be aligned is specified using the /JNTOL qualifier. (See Command Qualifiers: /JNTOL).

If /LLJOIN is used with a very small /JNTOL tolerance value, "pockets" or "sausage strings" will result between aligned sections. However, if too large a tolerance is used with /LLJOIN, an excessive processing time will result. It is important to give careful consideration to the tolerance that will be used with /LLJOIN.

Feature alignment priority, which determines which feature is moved onto which, may optionally be set using a definitions file specified using the /FCP qualifier. (See Command qualifiers: /FCP).

In the absence of an /FCP file, feature alignment priority is based on the internal sequence numbers of each pair of features under consideration. Feature sequence numbers normally increase monotonically through the IFF file. The feature with the lower internal sequence number will be moved onto the feature with the higher internal sequence number.

If an /FCP qualifier is supplied, only feature pairs having the specified feature codes will be aligned.

The /FCP file contains pairs of feature codes which define feature alignment priority. During feature alignment, positioning will be from the first feature code onto the second. If feature codes are equal, the feature with the lower internal sequence number will be aligned onto the one with the higher internal sequence number.

The alignment process is deliberately constrained so that in the vicinity of isolated junctions where lines cross or meet at corners, junctions are formed at a single point only and line sections are not "pinched together" along a continuous length.

Any junctions formed during the alignment process between feature ends are incidental to the main purpose of this process which is to align continuous line sections. The /LPJOIN and /PPJOIN processes are specifically designed for line-end to line and line-end to line-end junction formation and will generally produce "better" junctions.

OPTION 4 Feature-end to feature joining - /LPJOIN

The /LPJOIN qualifier causes ILINK to move the free ends of features onto other line or point features. (See Fig. 3).

The selection of the feature to which a free end is to be joined is controlled by 2 tolerances:

- o the JoiN TOLerance - specified using /JNTOL, and,

- o the EXtension TOLerance - specified using /EXTOL.

The join tolerance sets the distance below which the end point may be moved onto the target feature. If both the free end point and the intersection point on the target feature are from the same feature then they will only be joined if they are separated by more than /JNTOL distance along the feature.

The extension tolerance sets the maximum distance by which one vector may be extended or truncated to meet another before a vector end is moved off the original line of the vector, to a point within the /JNTOL tolerance in any direction, causing it to rotate (see Figures 3 and 5).

Some user applications require that priority should be given to joining free ends to original feature vertices on the "target" linear feature. The /VERTEX qualifier ensures that free feature ends are moved to original feature vertices which are within the tolerance specified by /JNTOL. This priority weighting overrides a choice of an inter-vertex join position along a vector.

/LPJOIN will not join feature ends to any position along the first or last vector of any feature that is within the /EXTOL tolerance of the feature end. This is because it is difficult to allow such joins without introducing the possibility that the end vector moved onto will not subsequently be moved itself. This would leave hanging in space the end which was moved in the earlier operation. As a result it is recommended that /PPJOIN is **always** applied after using /LPJOIN.

Note that /PPJOIN should not be used before /LPJOIN, as feature ends pulled together by /PPJOIN may be pulled apart again by /LPJOIN.

OPTION 5 Feature merging - /MERGE

The /MERGE qualifier causes ILINK to merge duplicate feature sections (i.e. parts of features which have identical coordinates with parts of other features) together into single "shared features" (See Fig. 1).

If an /FCP qualifier is supplied, only feature pairs having the specified feature codes will be merged.

The feature code given to a "shared" feature is taken from the value supplied with the /SHRFC qualifier (default 999), unless a Feature Code Combination (.FCC) file is specified with the /FCC qualifier. In this case the feature code given to a shared feature which results from the merging of features with any given set of feature codes, will be taken from the FCC file definition for the feature code which is to correspond to this feature code combination (if this combination is present in the FCC file).

Using this /FCC mechanism, it is possible to assign a particular feature code to every possible category of shared features, such as house-fence; river-road; river-railway-road; forrest-road, etc.

The feature codes of each of the component features which formed the shared feature are not lost. They are output in type 1 AC (Ancillary Code) entries within the shared feature. (See Description sub-section "ILINK IFF output file structure and content" - AC entry).

The code field of a type 1 AC is used to transmit the parent feature's feature code. If the /PARENT qualifier is selected, the code field of a type 9 AC is also used to transmit an identifier which is unique to the parent feature.

Single feature - link not shared:

```
NF 1 1          - start of feature with FSN 1
FS 24 0 0 0       feature code 24
ST 2 0          - 2 point link
90.8   100.8
110.7  109.4
EF              - end of feature
```

Shared feature - link shared by 3 features in input IFF file

```
NF 21 21        - start of feature with FSN 21
FS 999 0 0 0    - feature code 999 (/SHRFC was not specified
                  and so the default has been used)
AC 1 30         - type 1 AC carrying the feature code (30)
                  of first feature sharing this link
AC 9 112        - type 9 AC carrying the FSN (112) of
                  first feature sharing this link. This AC
                  is only present as the /PARENT qualifier
                  was used
AC 1 67         - type 1 AC carrying the feature code (67)
                  of second feature sharing this link
AC 9 115        - type 9 AC carrying the FSN (115) of
                  second feature sharing this link. This AC
                  is only present as the /PARENT qualifier
                  was used
AC 1 2          - type 1 AC carrying the feature code (2)
                  of third feature sharing this link
AC 9 325        - type 9 AC carrying the FSN (325) of
                  third feature sharing this link. This AC
                  is only present as the /PARENT qualifier
                  was used
ST 2 0          - the shared 2 point link
90.8   100.8
110.7  109.4
EF              - end of shared feature
```

It is (theoretically) possible to reconstruct the original (non-shared) data structure using the information contained in the type 1 and type 9 ACs, even though, as in the example above, one feature now represents the location of three features in the input file.

OPTION 6 Feature-end to feature-end joining - /PPJOIN

The ILINK /PPJOIN process forms junctions where lineal feature ends and/or point features almost touch (see Fig. 5).

The selection of the feature-end to which a free end is to be joined is controlled by 2 tolerances:

- the JoiN TOLerance - specified using /JNTOL, and,
- the EXtension TOLerance - specified using /EXTOL.

The join tolerance sets the distance below which the end point may be moved onto the target feature.

The extension tolerance sets the maximum distance by which one vector may be extended or truncated to meet another before a vector end is moved off the original line of the vector, to a point within the /JNTOL tolerance in any direction, causing it to rotate (see Fig. 3 and 5).

The precise action of the /PPJOIN process is controlled by the /PROJECT qualifier. By default the /PROJECT qualifier is active and the action of the ILINK /PPJOIN process is as follows:

- Where there are just two feature end-points to be joined, they are projected together. The new junction between the two features lies at the intersection point of the two extended vectors.
- Where there are more than two feature ends within the /JNTOL tolerance, all the feature ends are moved to their common centre of gravity.

This action is changed if /NOPROJECT is specified. All feature ends at a junction are moved to the position of the feature end with the highest feature priority.

Feature alignment priority, which feature is moved relative to another, may optionally be set using a definitions file specified using the /FCP qualifier. (See Command qualifiers: /FCP).

In the absence of a /FCP file, feature alignment priority is based on the internal sequence numbers of each pair of features under consideration. The feature with the lower internal sequence number will be moved relative to the feature with the higher internal sequence number.

If a /FCP qualifier is supplied, only features having the specified feature codes will be aligned.

The /FCP file contains pairs of feature codes which define feature alignment priority. During feature alignment, positioning will be from the first feature code onto the second. If feature codes are equal, the feature with the lower internal sequence number will be aligned with the one with the higher internal sequence number.

OPTION 7 **Arm sorting - command qualifier /SORTARMS**

Some user applications require that the arms which radiate from a junction must be in a consistent order. The ordering of the junction arms which radiate from a junction may be specified using the /SORTARMS qualifier. ILINK /SORTARMS specifies that the arms in IFF JB (Junction Block) entries should be sorted with orientation angle increasing, (measured anticlockwise from the positive X-direction).

This option, if used alone, will operate on a junction structured IFF file in situ (although you are still required to provide ILINK with a dummy output filespec).

OPTION 8 **The IFF junction structuring process - /STRUCTURE**

A "link/node" structure is a data structure which contains information connecting every link record to the node records corresponding to each of its ends, and also connecting every node record to all of the link records which hold links ending at this node.

A powerful ILINK feature is the ability to create an IFF link/node data structure from unstructured IFF vector input. This IFF link node structure is referred to as "IFF Junction Structure"

When the ILINK /STRUCTURE qualifier is given, the output IFF file will contain a link/node structure in the form of IFF "Junction-Structure" entries. For a description of IFF junction structure see the "IFF User Guide".

The nodes in the link/node structure will all be at existing feature-ends and at point-features only. An implicit /PPJOIN process is carried out with a near-zero join-tolerance to determine end-point connectivity.

ILINK and LITES2

ILINK is designed to complement the LAMPS graphic editor LITES2. No automatic geometry correction software can correct all geometric errors in digitised data and some tasks can only be performed manually by a skilled cartographic operator. To facilitate such editing ILINK will optionally generate a LITES2 command file. This can be used to direct the user to the position of the suspected error within LITES2. The positions supplied in the LITES2 command file locate the suspected error and so provide a mechanism for semi-automatic editing.

An entry is made in a LITES2 command file if ILINK detects one of the following potential error situations:

1. The nodes defining the two ends of a link are found to be within the /JNTOL tolerance of each other during processing. This results in the deletion of the link from the output IFF file and the following LITES2 commands are generated:

```
%MESSAGE End nodes lie within /JNTOL tolerance - link deleted
%POSITION    10.000    10.000
%TEST $CURSINWIN
%ELSE %ZOOM 1
%PING
%RESPOND
%ABANDON
%ABANDON
```

2. A node is found to be connected to only one link. This is potentially disruptive to polygon formation utilities. If the data produced by ILINK are to be used for polygon formation, it is strongly recommended that the ILINK/FREE_ENDS process is used to check for unwanted free-ends. This will result in the generation of the following LITES2 commands.

```
%MESSAGE Node has only one arm.
%POSITION    1890.030    1627.923
%TEST $CURSINWIN
%ELSE %ZOOM 1
%PING
%RESPOND
%ABANDON
%ABANDON
```

3. If the /VERIFY qualifier has been given then each /LPJOIN or /PPJOIN join which is across a distance within the range VRTOL to JNTOL will result in the generation of the following LITES2 commands.

```
%MESSAGE Point moved to (   242.157,    123.456)
%POSITION   241.726   125.824
%TEST $CURSINWIN
%ELSE %ZOOM 1
%PING
%RESPOND
%ABANDON
%ABANDON
```

To enable the LITES2 operator to identify the contents of a LITES2 command file and to facilitate the use of the LITES2 command file sorting utility LCMORG, ILINK outputs a standard header at the start of the LITES2 command file. The following is a typical example:

```
%POSITION     0.000     0.000
%POSITION   100.000   100.000
%ABANDON
%MESSAGE
%MESSAGE               L I T E S 2   C O M M A N D   F I L E
%MESSAGE
%MESSAGE                             created by
%MESSAGE
%MESSAGE ============================= I L I N K ==============================
%MESSAGE
%MESSAGE ILINK invoked by TIM using terminal LTA1: at 12-JUN-1987 14:41:00.13
%MESSAGE
%MESSAGE Command line:
%MESSAGE
%MESSAGE ILINK SOILS SOILS2/PPJOIN/JNTOL=50/LAYER=1/LITES2
%MESSAGE
%MESSAGE ======================================================================
%MESSAGE
```

ILINK directs all LITES2 command file output to the file directory defined by the logical name LSL$LITES2CMD. As ILINK processing tends to be iterative, it is strongly recommended that this directory is regularly purged of unwanted LITES2 command files.

Using ILINK in place of IFJ

ILINK is designed to replace the obsolete Laser-Scan IFJ utility. IFJ generates IFF junction structure with the restriction that junctions can only be created at feature ends. Features cannot be split to form a junction at intersection points. With IFJ the features must be manually prepared for junction formation using the LITES2 graphical editor. Clearly ILINK offers a very cost effective alternative to IFJ as features can be automatically split without recourse to expensive operator interaction.

This section is designed to guide Laser-Scan customers who are replacing IFJ within their production flowlines with ILINK.

Qualifiers available with IFJ are:

1. /NS: nx ny

   The 2 integer values nx and ny specify that IFJ is to split the input IFF file into (nx) by (ny) (maximum 10x10) node-sectors to aid searching for matching junctions.

   ILINK has no equivalent qualifier, but will always divide the file into (nx) by (ny) square node-sectors, with a maximum of 10x10.

2. /TO:real

Specifies the minimum distance between junctions that are to be considered separate. The default tolerance is 0.0.

The equivalent qualifier for ILINK is:
/JNTOL=real

3. /AR:integer

   Specifies the maximum number of arms allowed in a single junction. An error will occur if a junction is found with more than the specified number of arms. By default, IFJ allows a maximum of 4 arms.

   ILINK has no equivalent qualifier. ILINK imposes no limit on the number of arms per junction.

The user should specify the following IFJ command line to create a junction structured IFF output file using a junction tolerance of 1.5 IFF units:

```
$ IFJ out-file-spec=in-file-spec/TO:1.5<CR>
```

This is equivalent to running ILINK twice as follows:

```
$ ILINK in-file-spec work-file-spec/PPJOIN/JNTOL=1.5<CR>

$ ILINK work-file-spec out-file-spec/STRUCTURE<CR>
```

Using ILINK in place of CBA

ILINK is designed to replace the Laser-Scan CBA (Common Boundary Alignment) utility. CBA is a utility for common-boundary alignment and/or end-point joining.

This section is designed to guide Laser-Scan customers who are replacing CBA within their production flowlines with ILINK.

Qualifiers available with CBA are:

1. [-]AA[:tol1 (default:50)] (default:/-AA)
   Perform alignment of areals with areals with a tolerance of tol1.

2. [-]AL[:tol2 (default:50)] (default:/-AL)
   Perform alignment of areals with lineals with a tolerance of tol2.

3. [-]LL[:tol3 (default:50)] (default:/-LL)
   Perform alignment of lineals with lineals with a tolerance of tol3.

4. [-]LEE[:tol4 (default:50)] (default:/-LEE)
   Perform alignment of lineal-ends with lineal-ends with a tolerance of tol4.

5. [-]LEP[:tol5 (default:50)] (default:/-LEP)
   Perform alignment of lineal-ends with points with a tolerance of tol5.

6. <OVERLAYS,LAYERS>[: ovn1[-ovn2]] (default: 1-3)
   Restricts all alignment to features within specified overlays.

7. [-]MONITOR (default:/-MONITOR)
   Write information to the terminal at intervals to indicate the proportion of the process completed.

8. [-]VERIFY (default:/-VERIFY)
   Write a list of the locations of all the nodes found (no points, including line end points, are considered to be nodes unless they are shared by more than one feature), and also a list of the node indices of nodes encountered running along the length of each feature.

9. [-]AC (default:/-AC)
   Get type from a type 7 AC (Ancillary Code) entry, not from the FS (Feature Status) entry.

ILINK has no qualifiers equivalent to the /AC qualifier above, but every other option can be effected using ILINK as follows:

ILINK will only perform one /LLJOIN, /LPJOIN or /PPJOIN process per run. The /MONITOR and /LAYERS qualifier is the same in ILINK as for CBA (except that the standard DCL list-format is somewhat different from the "m:n" range format used by CBA).

Assuming point-features to have feature-code 100; lineal-features 101 and areal features 102, the following examples illustrate how ILINK can be used to emulate CBA:

(1) The CBA command line:

```
$ CBA out-file-spec=in-file-spec/AA:real<CR>
```

is equivalent to the ILINK command line:

```
$ ILINK in-file-spec out-file-spec/LLJOIN -<CR>
_$ /JNTOL=tol/FCP=file-spec<CR>
```

with the /FCP file containing the line:

```
   102     102             ! areals onto areals
```

(2) The CBA command line:

```
$ CBA out-file-spec=in-file-spec/AA:real/AL:real/LEE:real<CR>
```

is equivalent to the two-stage ILINK process defined by the commands:

```
$ ILINK in-file-spec temp-file-spec/LLJOIN -<CR>
_$ /JNTOL=real/FCP=fcp-file-spec<CR>
```

with the /FCP file containing the lines:

```
      102     102             ! areals onto areals
      102     101             ! areals onto lineals
```

followed by:

```
$ ILINK temp-file-spec out-file-spec/PPJOIN -<CR>
_$ /JNTOL=real/FCP=fcp-file-spec<CR>
```

with the /FCP file containing the line:

```
      101     101             ! lineals onto lineals
```

(3) The CBA command line:

```
$ CBA out-file-spec=in-file-spec/LEP:real<CR>
```

is equivalent to the ILINK command line:

```
$ ILINK in-file-spec out-file-spec/LPJOIN/JNTOL=real -<CR>
_$ /FCP=fcp-file-spec/FRT=frt-file-spec<CR>
```

where the fcp-file-spec specifies a file containing line and symbol-string (DFAD "point") feature code selections for processing. The frt-file-spec specifies an FRT (Feature Representation Table) file. For a description of FRT files see the "FRT User Guide".

ILINK IFF output file structure and content

IFF files produced as output from ILINK take their header entries from the input IFF file. For a detailed description of the content and structure of an IFF file see the "IFF User Guide".

Every feature, or part of original feature, will be output to the same layer in the output file as it was found in the input file.

If /BREAK is specified, and a feature is broken into parts, these parts will be contiguous and ordered within the file as they are along the original feature. (see Description sub-section "ILINK and shared or broken features"). The qualifier /KEEP=FSN may be used to specify that the parent feature FSNs will used for the output features, otherwise the output FSNs will be created sequentially.

ILINK output IFF file characteristics, by IFF entry, are:

RA - reflects the range of the data
HI - present only if available in the input IFF file
MH - copied from the input IFF file
MD - copied from the input IFF file
NS - contains the text "Created by ILINK on 'date' at 'time'"
CC - copied from the input IFF file
CP - copied from the input IFF file

Layer 0 - simply copied from input IFF file, (unless /LAYER=0 used to explicitly include layer 0 for ILINK processing)

NO - copied from input IFF file.
NF - feature FSNs (Feature Serial Numbers) are resequenced by ILINK.
FS - by default this contains the feature-code from the input file feature. In the case of shared features, the feature code is taken from that specified by values with the /SHRFC or /FCC qualifiers. If no feature code is specified using the /SHRFC or /FCC qualifiers a default value of 999 is assumed.
[AC] - Any ACs present in the input IFF file are are copied to the output IFF file, except type 1 and 9 ACs.

ILINK reserves these ACs for special coding, see Description sub-section "ILINK and shared or broken features".
ST - there may be several STs in a feature. Together they contain the coordinates which define the feature.
EF - end of feature

.
.
Other IFF features ...
.
.

EO - end of layer
EM - end of map
EJ - end of file

ILINK and shared or broken features

The ILINK /BREAK and /MERGE processes both result in input features being broken into parts and output as separate features.

It is sometimes required to know which original feature each part feature has come from. For this purpose, the /PARENT qualifier is provided to specify that AC type 9 entries should be put into every part feature of a feature that is split during the /BREAK process, or into every compound feature produced during the /MERGE process, so that the information is there in principle from which all original "parent" features can be reconstituted.

The qualifier /KEEP=FSN may be used to specify that the parent feature FSNs will used for the output features, otherwise the output FSNs will be created sequentially.

Broken Features Produced By The /BREAK Process

The /BREAK process will break features where they touch or cross. The parts of broken features will be contiguous in the output file, and ordered as they were along the original feature.

The qualifier /KEEP=FSN may be used to specify that the parent feature FSNs will used for the output features, otherwise the output FSNs will be created sequentially.

If the /PARENT qualifier is used then an AC type 9 entry is written into each part feature to hold a number which can be used to identify the original "parent" feature which this feature belongs to (see also /MERGE).

Shared Features Produced By The /MERGE Process

The /MERGE process (used, for example, where a series of exactly aligned polygons are required to have duplicate sections merged into single "shared" or "compound" features) will break features at junctions where aligned features separate. Two or more exactly aligned part features are then merged into a single compound feature.

For each feature merged into a compound feature, an AC type 1 entry is written to hold the feature code of the feature being merged, followed by every other AC entry from this feature. If the /PARENT qualifier is used then an AC type 9 entry will follow the AC type 1 entry, to hold a number which can be used to identify the original "parent" feature which this feature belongs to (see also /BREAK).

The qualifier /KEEP=FSN may be used to specify that the parent feature FSN's will used for the output features, otherwise the output FSNs will be created sequentially.

It may be required to give a feature code to compound features which allows them to be distinguished from other features according to their graphical representation when displayed in LITES2.

All shared features are given the same special feature code, which is 999 by default, but may be specified using the /SHRFC qualifier (see also /SHRFC)

If it is required to distinguish graphically between compound features arising from the merging of different combinations of feature codes, the /FCC qualifier mechanism can be used to assign a particular feature code to each possible feature code combination, such as house-fence; river-road; river-railway-road; forrest-road, etc. Combinations not found in the /FCC file are given the /SHRFC code. (see also /FCC).

Sometimes the AC coding for each component feature in a compound feature may be dependent on the direction of the component relative to the compound point string, such as left/right AC codes, for example, which give the code for the area to the left and right of a feature. Whereas normally all AC entries are copied into the compound feature exactly as they are found in the feature being merged, the /ACP qualifier can be used to specify that AC-type pairs within a compound feature which are found in the /ACP file should have their AC types swapped (see also /ACP).

Note that ILINK reserves type 1 and type 9 ACs for special coding. Type 9 ACs found in the input IFF file will be ignored and will not be transmitted to the output IFF file, because the parent feature identifiers which they carry are unique only within the run of ILINK in which they are produced.

EXAMPLES

In most production flowlines ILINK processing is done in the batch environment. Due to the very nature of the geometry idealisation problem, ILINK processing can take quite a long time. By default, ILINK outputs copious run-time diagnostics as an aid to processing progress analysis. This is the opposite behaviour to that exhibited by those LAMPS (Laser-Scan Automated Map Production System) utilities which are designed primarily for quick interactive applications. At any time during the execution of a batch job the VMS TYPE utility may be used to type the contents of the batch process log file.

In the ILINK processing examples, presented in this section, no attempt is made to summarise the diagnostic output and so it may be regarded as typical of what a user will see when interpreting a batch log.

```
$ ILINK DUA3:[BUREAU.SSEW]S106SEF S106SEF.JOINED_IFF/PPJOIN
:-/JNTOL=1.5/VRTOL=0.5<CR>

 ILINK - Version  2.0

 ILINK COMMAND INTERPRETATION
*******************************************************************************

 Input from                    : DUA3:[BUREAU.SSEW]SSEW106SEF.IFF;5
 Output to                     : DUA3:[BUREAU.SSEW]S106SEF.JOINED__IFF;2
 Process                       : PPJOIN : Ends with ends junction formation
                               : ends projected
 Join tolerance                :    1.500
 Max vector extension          :    1.500
 Min verification tolerance    :    0.500
 Layers to be processed        : All layers, (excluding layer 0).
 Output structure              : Normal IFF structure.
 LITES2 command file           : None specified.
 FRT file                      : None specified.
 FCP file                      : None specified.

 ILINK LOG
*******************************************************************************

 Scanning IFF file for dynamic memory allocation
 Reading data from IFF input
 Creating link/node structure
        Setting up a node for each feature end
        Merging nodes separated by less than     1.000000
        Creating node arms
 Joining ends to ends
 Writing to IFF output
 Counting arms-per-node

 ILINK STATISTICS
*******************************************************************************

 Input features processed      : (Point 0),  (Open 716),  (Closed 21)
 Processed features output     : (Point 0),  (Open 715),  (Closed 22)
 Total features in IFF input  : 740
 Total features in IFF output : 740

 Nodes found joined to 0 arms  (single point features): 0
                       1 arm   (unattached links)     : 0
                       2 arms  (link joined to link)  : 20
                       3 arms  (link joined to links) : 450
                       4 arms  (link joined to links) : 21
                      5+ arms  (link joined to links) : 0

                                           Total      : 491
 -----------------------------------------------------------------------------
 Join separations over the range   0.500  to   1.500

 Range    Count | Proportion
 ---------------|-------------------------------------------------------------
  0.500+    161 | ********************
  0.600+    154 | *******************
  0.700+     70 | *********
  0.800+     41 | *****
  0.900+     23 | ***
  1.000+     22 | **
  1.100+     10 | *
  1.200+      8 | *
  1.300+      2 | *
  1.400+      0 |
 ---------------|-------------------------------------------------------------

 ELAPSED:    0 00:01:08.79  CPU: 0:00:39.34  BUFIO: 8  DIRIO: 1697  FAULTS: 423
$
```

This example demonstrates the use of the /PPJOIN and /JNTOL qualifiers for joining all feature ends to feature ends, within a (general)join and extension /truncation join tolerance of 1.5, and a minimum join verification tolerance of 0.5 IFF units.

The user has specified that input is to be from the file DUA3:[BUREAU.SSEW]S106SEF.IFF;0. ILINK has taken the missing extension and version fields for the file-spec from the default input file specification 'LSL$IF:IFF.IFF;0'. Output is to be to the file DUA0:[BUREAU.SSEW]S106SEF.JOINEDIFF;3. ILINK has taken the missing output file device and directory fields from the parsed input file-spec. The output file version number defaults to the highest existing version number of the file, incremented by one.

Notice that in the absence of an explicit /EXTOL qualifier, the maximum vector extension had been set to have the same value (1.5) as /JNTOL. Vector free ends will be projected, not merely moved to the centre of gravity of all the free ends within the /JNTOL tolerance.

The /VRTOL qualifier has been used to give the minimum join verification tolerance a value greater than the default of 0.0 so that the histogram produced will show the distribution of joins made over the more critical end of the join tolerance range rather than including all joins including those across zero distance.

No /FRT qualifier has been specified and so, in the absence of any explicit /LAYER layer selections, it must be assumed that the user believes that all the features in DUA3:[BUREAU.SSEW]S106SEF.IFF;12 are suitable for ILINK processing. If an FRT (Feature Representation Table) file is specified, ILINK is able to look up the graphical type of each feature to determine whether processing is valid for that graphical type. Features with inappropriate graphical types are copied, unprocessed, to the output file, (see Command Qualifiers: /FRT).

No /LAYER selections or /FCP file is specified and so all features will be processed. Feature alignment priority, (which feature is moved relative to another), is defined by default using the feature internal sequence number, (see Description).

The "ILINK STATISTICS" section of the log summarises the results of the processing run. The log shows that 1 feature that was open on input is closed by the ILINK processing (i.e. first and last points are now coincident).

The "nodes found" section of the log shows how many nodes are present in the output file. The nodes are classified by node degree (i.e. by the number of arms radiating from the node).

If the user subsequently ran ILINK to produce an IFF junction structure (/STRUCTURE), the zero arm junctions would be omitted from the IFF junction structure, because the concept of a zero arm junction is (unfortunately) foreign to IFF format. Instead the zero arm junctions would be output as single point features which do not contain any IFF junction entries.

The "join separation" section shows the distribution of join distances between ten equal subintervals over range VRTOL to JNTOL. Here it can be seen from the way the histogram tails off that the join tolerance as used has probably found all of the joins which the user intended it to. It it had not tailed off then the /JNTOL value might have been too small. It it had tailed off and then picked up again, the /JNTOL value might have been too large, so that some joins were made which the user had not intended. Of course, if the digitising is very rough and feature density is very high, there may be no tolerance value which can separate good from bad joins in this way.

At the end of the run the standard LAMPS timing statistics summary is output. The inexperienced user may find it instructive to try successive runs on the same data, altering between each run only the /EXTOL and /JNTOL tolerance values. Significant variations in processing time will result if wide variations in tolerance values are used. It cannot be over-stressed that ILINK is not a substitute for accurate source digitising. The submission of inaccurate digitising to ILINK requires bigger processing tolerances and longer run times, and is also likely to introduce various other problems.

The run has completed successfully and DCL symbol $STATUS is set to SS$_NORMAL - "normal successful completion".

```
$ ILINK POLYSEG.IFF;5 POLYSEG.A/LPJOIN/JNTOL=4.0/LAYER=1/LITES2<CR>

$ ILINK POLYSEG.A POLYSEG.B/BREAK/LAYER=1<CR>

$ ILINK POLYSEG.B POLYSEG.IFJ/STRUCTURE/LAYER=1<CR>

(Output to SYS$OUTPUT not shown here).
```

This example demonstrates the use of the /LPJOIN and /JNTOL qualifiers for joining feature ends to features, within a /JNTOL tolerance of 4.0 coordinate units.

The user has specified that input is to be from the file DUA1:[PACKAGE.POLYGONS]POLYSEG.IFF;5. ILINK has taken the missing device and directory fields for the file-spec from the default input file specification 'LSL$IF:IFF.IFF;0'. Output is to be to the file DUA1:[PACKAGE.POLYGONS]POLYSEG.IFJ;1 via two intermediate stages. ILINK has taken the missing output file device and directory fields from the parsed input file-spec. The output file version number defaults to the highest existing version number of the file, incremented by one.

The user has used the /LITES2 qualifier with the /JNTOL process to specify that messages resulting from the deletion of features which are less than the /JNTOL tolerance in length will be sent to a LITES2 command file. As the optional /LITES2 qualifier file-spec value has been omitted, the default LITES2 command file-spec will be used. This uses logical name LSL$LITES2CMD and the filename from the input IFF file specification and the extension '.LCM'.

Notice that in the absence of an explicit /EXTOL qualifier, the maximum vector extension has been set to have the same value (4.0) as /JNTOL.

Although /LAYER=1 is specified, no /FRT qualifier has been specified. It must therefore be assumed that the user believes that all the features layer 1 of DUA3:[BUREAU.SSEW]S106SEF.IFF;12 are suitable for ILINK processing. If an FRT (Feature Representation Table) file is specified, ILINK is able to look up the graphical type of each feature to determine whether processing is valid for that graphical type. Features with inappropriate graphical types are copied, unprocessed, to the output file, (see Command Qualifiers: /FRT).

No /FCP file is specified and so all features in the specified (/LAYER=1) layer will be processed. Feature alignment priority, (which feature is moved relative to another), is defined by default using the feature internal sequence number, (see Description).

The /BREAK qualifier results in all features being broken at feature intersections. These intersections will be represented by junctions in the junction structured IFF output file.

Had any input features been shorter than the /JNTOL tolerance or had resulted in the formation of one arm nodes, messages would have been output to the LITES2 command file.

The run has completed successfully and DCL symbol $STATUS is set to SS$_NORMAL - "normal successful completion".

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.

BADFTR, input feature with FSN 'fsn' ('isn') in layer 'integer' rejected

> **Explanation:** The indicated feature is invalid for ILINK processing.
>
> **User action:** Use LITES2 or IPATCH to examine and edit the indicated feature. Re-run ILINK if necessary.

TOOMNYPPF, too many points per feature - maximum allowed is 'integer'

> **Explanation:** The maximum number of points allowed per feature has been exceeded.
>
> **User action:** Break excessively large features into smaller ones so that the number of points does not exceed the maximum absolute limit.

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.

BADMINTOL, mintol value 'real' below dataset resolution 'real'

Explanation: The value given with the /MINTOL qualifier is below the dataset floating point arithmetic resolution, and unpredictable results may occur.

User action: Increase the /MINTOL value to exceed the given dataset resolution.

BADST, ST entry has 0 points and was ignored

Explanation: Every ST entry in the input IFF file should contain 1 or more points. ILINK will continue processing but the cause of the empty ST entry should be investigated.

User action: Examine the input IFF file HI entry (if present) to determine which utilities have been used to manipulate the file. If a Laser-Scan supplied utility is suspected, please preserve the data and submit an SPR to Laser-scan.

DELETE, end nodes lie within /JNTOL tolerance - link deleted at ('real', 'real')

Explanation: A link is shorter than the /JNTOL tolerance at the specified position and so has been shrunk to zero length and has been deleted. If the /LITES2 qualifier is specified this warning will also be written to the LITES2 command file.

User action: Check that the link at the specified position should really be that short. If not, either re-submit the ILINK command line with a smaller /JNTOL tolerance, or use LITES2 to correct the error.

FREEENDS, free-end node detected at ('real', 'real')

Explanation: This message is only generated if the /FREEENDS qualifier is specified. A node has been detected which is connected to only one link. If the /LITES2 qualifier is specified this warning will also be written to the LITES2 command file.

User action: Check that the node at the specified position should have only one arm. If not, either re-submit the ILINK command line with larger /JNTOL and /EXTOL tolerances, or use LITES2 to correct the error.

MDDEFAULT, MD error: origin defaulted to (0,0)

Explanation: The /ABSOLUTE qualifier has been given, but there was some error in the MD (map descriptor) entry, either not type 2 or not a valid MD type 2 entry. The origin offset has been set to a default value of (0,0) and the /ABSOLUTE qualifier ignored.

User action: Check the MD entry, and check that it is type 2 and of the correct length.

NOFCFRT, feature code 'integer' not found in FRT file - features copied unprocessed to output file

Explanation: This message is only generated if the /FRT qualifier is specified. ILINK uses the FRT file to check that the graphical types of features with given feature codes are actually suitable for ILINK processing. A feature code has been detected in the input IFF file which cannot be found in the FRT file. ILINK will copy all features having the specified feature code to the output file, without making any attempt to include those features in geometrical processing.

User action: Check that the correct FRT file was specified. If it was, then add a new entry for the missing feature code. Re-run ILINK using the updated FRT file.

NOMERGE, merge rejected (coordinates differ at ('real', 'real')

Explanation: The features at the specified location were not merged because there were coordinate differences found near the nodes belonging to the two features. This message is given only if the /WARNING qualifier was given.

User action: Check the input and output files at this location for consistency, and alter the output file if necessary.

NOPRFTRS, no features for processing

Explanation: Either the input IFF file contains no valid features, or they have all been excluded through use of the /FCP and/or /LAYERS qualifiers.

User action: Submit corrected command line.

NOTMERGE, merge rejected (different number of points near ('real', 'real')

Explanation: The features at the specified location were not merged because a different number of coordinates was found between the nodes in the two features. This message is given only if the /WARNING qualifier was given.

User action: Check the input and output files at this location for consistency, and alter the output file if necessary.

OBSOLETE, the /'string' qualifier is now obsolete; please check documentation

Explanation: There was an obsolete qualifier in the ILINK command line.

User action: Check the offending line and resubmit it.

ORIGPNTDEL, an original point at ('real' 'real') is being deleted

Explanation: ILINK is deleting an original point at the indicated location. This message is given only if the /WARNING qualifier was given, and is only meant to be a debugging aid.

User action: Check the input and output files at this location for consistency, and alter the output file if necessary.

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.

BADACPLIN, error reading line 'integer' from /ACP file

Explanation: ILINK is unable to read or interpret the specified line in the /FCP lookup file.

User action: Check the offending line in the FCP file against the definition of FCP file syntax given in the description section.

BADFCCLIN, error reading line 'integer' from /FCC file

Explanation: ILINK is unable to read or interpret the specified line in the /FCC lookup file.

User action: Check the offending line in the FCC file against the definition of FCC file syntax given in the description section.

BADFCPLIN, error reading line 'integer' from /FCP file

Explanation: ILINK is unable to read or interpret the specified line in the /FCP lookup file.

User action: Check the offending line in the FCP file against the definition of FCP file syntax given in the description section.

BADIFFNO, IFF NO entry has bad EO address in file 'filename'

Explanation: The address given in an NO entry does not correspond to the start of a valid EO entry.

User action: Process the IFF file using the IMEND utility which will automatically correct all incorrect NO-EO addresses.

BADJNSTR, junction structure is invalid

Explanation: In a valid junction structure, every link will address a valid node at both ends, and every node will address a valid list of zero or more links. If link L addresses node N then node N must address link L, and vice versa. There are also rules which apply to the number of links and nodes and the degree of each node. If any of these rules is broken then the junction structure is invalid.

User action: If the input file has not been changed in any way since the formation of junction structure using ILINK, please report the problem to Laser-Scan. Include the input file and run information in the software performance report.

BADLCPLIN, error reading line 'integer' from /LCP file

Explanation: ILINK is unable to read or interpret the specified line in the /LCP lookup file.

User action: Check the offending line in the LCP file against the definition of LCP file syntax given in the description section.

BADPAC, attribute code out of range, valid range is 'integer' to 'integer'

Explanation: The valid attribute code range is specified for each attribute code in the accompanying FRT file.

User action: Correct the attribute code in /TOL or /PAC file to lie within its valid range.

BADPACLIN, error reading line 'integer' from /PAC file

Explanation: Illegal file format in /PAC file.

User action: Correct error in specified line of /PAC file.

BADPACRANGE, attribute code range is too large, maximum allowed is 'integer'

Explanation: The attribute code range specified by the FRT file for one of the attributes given is greater than that allowed by ILINK.

User action: Check that the code range given by the FRT file is in fact correct. If it is correct, then please submit an SPR and ILINK will have its maximum attribute code range increased.

BADRANGE, bad range, xmin='real', xmax='real', ymin='real', ymax='real'

Explanation: A maximum X or Y coordinate range value was less than or equal to a minimum coordinate range value.

User action: Investigate the cause of the incorrect RA entry. Read the file into LITES2 and then use the LITES2 'EXIT' command to correct the range entry.

BADSHFCC, error reading shared feature code at line 'integer' of /FCC file

Explanation: Illegal file format in /FCC file.

User action: Correct error in specified line of /FCC file.

BADTOLLIN, error reading line 'integer' from /TOL file

Explanation: Illegal /TOL file format.

User action: Correct error on specified line of /TOL file.

EQUALACS, not all ACs in /ACP file are different

Explanation: The ACP file contained some equal AC codes.

User action: Check the ACP file for duplicate AC entries.

ERRSCANCB, error scanning feature for CB's, status was 'integer'

Explanation: An attempt to amalgamate all CB's within a feature has failed. One possible cause is that the amalgamation contains too many columns.

User action: Check that the input data does not contain features with multiple CB's whose columns vary significantly. Failing this, treat this error as INTRNLERR.

FAILTRLG, failed to translate logical name 'string'

Explanation: ILINK is unable to translate the specified logical name.

User action: Use the VMS ASSIGN or DEFINE commands to correctly define the logical name. Re-run ILINK. Normally the logical names required by ILINK are defined at user login time. See the "LAMPS Environment Guide" for details of these logical names.

FCCRANGE, error reading feature code range "'integer':'integer'" at line 'integer' of /FCC file

Explanation: ILINK is unable to read or interpret the indicated feature code range on the specified line in the /FCC lookup file.

User action: Check the offending line in the FCC file against definition of FCC file syntax given in the description section. Ensure that in all range specifications of the form N:M that N < M.

ILLEGALAC, AC codes outside range 0-32767, or equal to 1 or 9 not allowed in /ACP file

Explanation: The ACP file contained AC codes which are either reserved by ILINK for its own purposes or are illegal (ie. not in the range 0 to 32767).

User action: Check the ACP file AC entries.

ILLEGALCYCLE, 'integer' illegal cycle(s) in /FCP or /LCP file

Explanation: The FCP or LCP file contains illegal cycles within the pairs given. It is not possible to move feature A onto feature B, and then later to indirectly move feature B onto A without the internal algorithms becoming corrupted.

User action: Check the file entries for illegal cycles, and resubmit.

INVALBPF, invalid /BPF argument - value must lie in the range 1.000 to 100.000

Explanation: Invalid argument given to the /BPF qualifier.

User action: Correct error in the command line and resubmit.

INVALEXTOL, invalid /EXTOL argument - value must be positive

Explanation: Invalid argument given to the /EXTOL qualifier.

User action: Correct error in the command line and resubmit.

INVALFC, 'integer' is an invalid feature code argument - value must lie in range 1-32767

Explanation: IFF feature codes lie in the range 0 to 32767.

User action: Ensure that all feature code specifications lie within the range 0 to 32767. Re-run ILINK.

INVALJNTOL, invalid /JNTOL argument - value must be positive

Explanation: Invalid argument given to the /JNTOL qualifier.

User action: Correct error in the command line and resubmit.

INVALLAY, 'integer' is an invalid layer argument - value must lie in range 0-32767

Explanation: IFF layer numbers must lie in the range 0 to 32767.

User action: Ensure that all IFF layer specifications lie within the range 0 to 32767. Re-run ILINK.

INVALMINLEN, invalid /MINLEN argument - value must be positive

Explanation: Invalid argument given to the /MINLEN qualifier.

User action: Correct error in the command line and resubmit.

INVALMINTOL, invalid /MINTOL argument - value must be positive

Explanation: Invalid value supplied with the /MINTOL qualifier.

User action: Re-run ILINK with a valid number.

INVALVRTOL, invalid /VRTOL argument - value must be between 0 and JNTOL

Explanation: Invalid argument given to the /VRTOL qualifier.

User action: Correct error in the command line and resubmit.

INVFCCARG, invalid feature code 'integer' at line 'integer' of /FCC file - values must lie in the range 0-32767

Explanation: ILINK is unable to interpret the indicated feature code which most probably lies outside of the specified range.

User action: Check the offending line in the FCC file against definition of FCC file syntax given in the description section. Ensure that all feature codes lie within the specified range.

MDABSENT, MD entry missing

Explanation: the /ABSOLUTE qualifier has been given, but there was no MD (map descriptor) entry.

User action: check the IFF file for a valid MD entry.

MISSFC, missing or invalid feature code at line 'integer' of /FCC file

Explanation: ILINK expected but was unable to read a feature code on the specified line of the FCC file, and which most probably lies outside the specified range.

User action: Check the offending line in the FCC file against the definition of FCC file syntax given in the description section. Ensure that all feature codes lie within the range 0 - 32767, that there are no non-numeric characters within feature codes and that all feature code range specifications of the form M:N do have an integer value.

MISSOUTFILE, required output file missing

Explanation: ILINK expected an output file in the command line.

User action: Supply the output file and rerun the program.

NOATTCOD, capture method attribute name 'string' is not in FRT file

Explanation: Error finding attribute in ACD section of the /FRT file.

User action: Check that the FRT file contains the correct ACD entries.

NOATTNAM, no capture method attribute name in 'filename'

Explanation: Error finding capture attribute name in the specified file.

User action: Check the file contains the correct capture attribute name.

NOIFFRA, IFF RA entry not found

Explanation: The input IFF file did not contain a valid RA entry.

User action: Investigate the cause of the missing RA entry. Read the file into LITES2 and then use the LITES2 'EXIT' command to correct the range entry.

NOJNSTR, no junction structure in file 'filename'

Explanation: An ILINK operation, such as /SORTARMS, has been requested which requires a junction structure to be present in the input IFF file, but this was not present.

User action: Use ILINK to create a junction structure first. The /SORTARMS qualifier can be used either together with /STRUCTURE, or on its own.

NOPROCESS, no process specified

Explanation: No ILINK process qualifier given on command line.

User action: Correct error in the command line and resubmit.

OPNACP, error opening /ACP file 'filename'

Explanation: Error opening /ACP file.

User action: Check file exists and supply the correct filespec.

OPNFCC, error opening /FCC file 'filename'

Explanation: The specified file cannot be opened for reading. This message will usually be followed by another giving the reason for the failure.

User action: The accompanying message should indicate the cause of the error. Check the file specification for errors, check that the file is in the correct directory, check that the file and directory are not protected against read access. If the problem continues, notify the system manager.

OPNFCP, error opening /FCP file 'filename'

Explanation: The specified file cannot be opened for reading. This message will usually be followed by another giving the reason for the failure.

User action: The accompanying message should indicate the cause of the error. Check the file specification for errors, check that the file is in the correct directory, check that the file and directory are not protected against read access. If the problem continues, notify the system manager.

OPNFRT, error opening /FRT file 'filename'

Explanation: Error opening /FRT file.

User action: Check file exists and supply the correct filespec.

OPNLCP, error opening /LCP file 'filename'

Explanation: Error opening /LCP file.

User action: Check file exists and supply the correct filespec.

OPNPAC, error opening /PAC file 'filename'

Explanation: Error opening /PAC file.

User action: Check file exists and supply correct filespec.

OPNTOL, error opening /TOL file 'filename'

Explanation: Error opening /TOL file.

User action: Check file exists and supply the correct filespec.

OUTOFRANGE, coordinate(s) specified out of IFF range

Explanation: A point coordinate is outside the coordinate range specified in the RA entry in the same file. The RA entry is therefore incorrect.

User action: Read the file into LITES2 and then use the LITES2 'EXIT' command to correct the range entry.

PROCFRT, error while processing FRT file

Explanation: ILINK has detected an error while analysing the FRT file specified with the /FRT qualifier.

User action: The accompanying message should indicate the cause of the error.

SPACELIM, insufficient workspace - maximum workspace available is 'integer' bytes

Explanation: ILINK has attempted to allocate memory for more points and features than is permitted by the user's virtual-page-count limit.

User action: Either reduce the number of points or features which ILINK is attempting to process, or if this is not possible because the file to be processed is simply too large, then the VMS system virtual page count limit must be increased. If this is not possible, you will have to divide the IFF file into two or more separate files, process these separately and then merge them together again. If, however, this course of action is taken it will not be possible to obtain junction structure for the whole file by merging together each of its structured parts. If a junction structure is required over the final whole area then this must be produced using ILINK on the whole file. This will often be possible, even though ILINK could not run the /LLJOIN, /LPJOIN or /BREAK processes on the whole file due to lack of virtual memory space, because these require address space for every feature point whereas the PPJOIN and STRUCTURE operations do not require address space for feature points but only for a fixed small amount of data per feature.

TOOMNYACPRS, too many AC pairs in /ACP file - maximum allowed is 'integer'

Explanation: The ACP file contains more than the maximum number of different AC pairs permitted.

User action: Try using less AC pairs, or by running ILINK several times with separate ACP files.

TOOMNYAPJ, too many arms per IFF junction - maximum allowed is 'integer'

Explanation: An attempt has been made to output a node with too many arms to an IFF junction file.

User action: As for TOOMNYAPN.

TOOMNYAPN, too many arms per node - maximum allowed is 'integer'

Explanation: A node has been found with more arms than the data structure used by ILINK will allow. This number is very high so this error should never arise, but if it does it is probably due to the specification of an excessive /JNTOL tolerance.

User action: Check the /JNTOL tolerance value is correct relative both to the coordinate system defined in the IFF file and to the length of the shortest features in the IFF file. If it is found that the input geometry does actually possess nodes with higher degree than ILINK has allowed for, this should be reported and the maximum number of arms per node will be increased within reason.

TOOMNYARMS, too many arms - maximum allowed is 'integer'

Explanation: Processing has formed a junction with more arms than the data structure used by ILINK will allow. This number is very high so this error should never arise, but if it does it is probably due to the specification of an excessive /JNTOL tolerance.

User action: Check the /JNTOL tolerance value is correct relative both to the coordinate system defined in the IFF file and to the length of the shortest features in the IFF file. If it is found that the input geometry does actually possess junctions with higher degree than ILINK has allowed for, this should be reported and the maximum number of arms per node will be increased within reason.

TOOMNYFCARG, too many feature code arguments at line 'integer' of /FCC file - max allowed is 'integer'

Explanation: The specified line of the /FCC file contains more than the permitted maximum number of feature codes.

User action: Try using fewer feature codes, possibly by making maximum possible use of the /LAYERS layer selection qualifier, or by running ILINK several times with separate FCC files.

TOOMNYFCPRS, too many different FC pairs in /FCP file - maximum allowed is 'integer'

Explanation: FCP file contains more than the maximum number of different feature code pairs permitted.

User action: Try using less feature code pairs, or by running ILINK several times with separate FCP files.

TOOMNYFCS, too many different FC's in /FCP file - maximum allowed is 'integer'

Explanation: FCP file contains more than the maximum number of different feature codes permitted.

User action: Try using less feature codes, possibly by making maximum possible use of the /LAYERS layer selection qualifier, or by running ILINK several times with separate FCP files.

TOOMNYFTRS, too many features - maximum allowed is 'integer'

Explanation: The maximum number of features allowed for when memory was allocated by ILINK has been exceeded. There are either too many features in the input file, or the error has resulted from the creation of new features during ILINK processing. New features are generated when features are broken into separate parts where they cross. This happens during several of the available ILINK operations, not just in /BREAK. Even when /BREAK is not specified, features are always broken where they cross and are reconstituted on output as necessary.

User action: Increase the number of features for which ILINK should allocate memory using the /BPF qualifier (average number of Breaks Per Feature during processing). A large /BPF value value will cause more memory to be allocated to receive the expected increase in the number of output features.

TOOMNYLAYERPRS, too many different layer pairs in /LCP file - maximum allowed is 'integer'

Explanation: The LCP file contains more than the maximum number of different layer pairs permitted.

User action: Try using less layer pairs, or by running ILINK several times with separate LCP files.

TOOMNYLAYERS, too many layers - maximum allowed is 'integer'

Explanation: The maximum number of layers allowed by ILINK has been exceeded, even after ignoring sequential NO entries for the same layer.

User action: Rearrange the layer structure within the IFF file, or use IMERGE to compress the layers.

TOOMNYNDS, too many nodes - maximum allowed is 'integer'

Explanation: The maximum number of nodes allowed for when memory was allocated by ILINK has been exceeded. This is because the number of nodes allowed for is equal to twice the number of features allowed for, which may have been too low.

User action: Increase the number of features for which ILINK should allocate memory using the /BPF qualifier (average number of Breaks Per Feature during processing). A large /BPF value value will cause more memory to be allocated to receive the expected increase in the number of output features.

TOOMNYPACS, too many point attribute code lines in TOL or PAC file - max allowed is 'integer'

Explanation: Number of lines is greater than ILINK allows for.

User action: Please submit an SPR and ILINK will have its maximum line count increased.

TOOMNYPRFCS, too many different feature codes suitable for processing - maximum allowed is 'integer'

Explanation: ILINK has analysed the FRT file specified with the /FRT qualifier. There is insufficient memory to hold all the feature codes which have been read from the FRT file and have been flagged as suitable for ILINK processing. The input data must have very complex feature coding.

User action: Ensure that there are not redundant feature codes in the data and the FRT file. Try using less feature codes, possibly by making use of the /LAYERS qualifier, in conjunction with running ILINK several times with separate FRT files.

TOOMNYPTS, too many points - maximum allowed is 'integer'

Explanation: The maximum number of points allowed for when memory was allocated by ILINK has been exceeded.

User action: Increase the number of features which ILINK should allocate memory for, using the /BPF qualifier (average number of Breaks Per Feature during processing). A large /BPF value value will cause more memory to be allocated to receive the expected increase in the number of output features.

TOOMNYRJFCS, too many different feature codes rejected for processing - maximum number allowed is 'integer'

Explanation: ILINK has analysed the FRT file specified with the /FRT qualifier. There is insufficient memory to hold all the feature codes which have been read from the FRT file and have been flagged as unsuitable for ILINK processing. The input data must have very complex feature coding.

User action: Ensure that there are not redundant feature codes in the data and the FRT file. Try using less feature codes, possibly by making use of the /LAYERS qualifier, in conjunction with running ILINK several times with separate FRT files.

TOOMNYSHFTRS, too many shared features - maximum allowed is 'integer'

Explanation: The maximum number of shared features allowed for when memory was allocated by ILINK has been exceeded.

User action: Increase the number of features which ILINK should allocate memory for, using the /BPF qualifier (average number of Breaks Per Feature during processing). A large /BPF value value will cause more memory to be allocated to receive the expected increase in the number of output features.

UNEXPEOF, unexpected end of IFF file 'filename'

Explanation: This message indicates there is something seriously wrong with the IFF file which has caused immediate termination of the program. ILINK has detected the end of the IFF file, but has not detected an IFF 'EJ' entry.

User action: Use IMEND on the file, which will correctly position the EOF marker and insert an EJ entry at the end of the file. Re-run ILINK on the corrected file.

MESSAGES (FATAL)

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

BADADDR, bad IFF address

> **Explanation:** A IFF address is pointing to an invalid IFF location. There is a fault in the ILINK routine which generated this address.
>
> **User action:** Please report the problem to Laser-Scan. Include the input file and run information in the software performance report.

INTRNLERR, internal consistency error 'integer' detected in routine 'string' - please submit an SPR to Laser-Scan

> **Explanation:** ILINK has detected an internal error in the routine named. This has caused an error which invalidates any further processing, so execution is terminated.
>
> **User action:** Please report the problem to Laser-Scan. Include the input file and run information in the software performance report.

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.

CHAPTER 3

MODULE RELHT

MODULE RELHT

FUNCTION

Given a junction structured IFF file and a table of feature code priorities, the program assigns relative heights to the ends of all the links. The junction structure is not preserved in the output file.

FORMAT

```
$ RELHT file-spec file-spec
```

Command qualifiers	**Defaults**
/[NO]LITES2	/NOLITES2
/[NO]LOG	/NOLOG
/[NO]NODE_LIST	/NONODE_LIST

PROMPT

From: input-file-spec
To: output-file-spec

PARAMETERS

input-file-spec

- Specifies the IFF file which is to be processed. Any part of the filename which is not specified will be taken from the default specification 'LSL$IF:.IFF;0'.

output-file-spec

- Specifies the IFF file which is to be created to contain all the map data from the input file with relative heights for the end of each link. Any part of the filename which is not explicitly given will be taken from the parsed input specification. Note that a version number must **not** be specified for the output file. If a file with the specified name already exists, a new file will be created with the version number incremented by one.

COMMAND QUALIFIERS

/LITES2[=file-spec]

- allows the user to specify a LITES2 guidance file which will contain commands to position the cursor on all junctions with doubtful relative heights. If there are no doubtful junctions, a guidance file will not be created. Any part of the filename which is not explicitly given will be taken from LSL$LITES2CMD:<output IFF file name part>.LCM - this is the actual filename used if none is specified with the qualifier. If a file with the specified name already exists, a new file will be created with the version number incremented by one.

/LOG

- this will result in supplementary messages being sent to SYS$OUTPUT. Supplementary messages are generated when an IFF file is successfully opened or closed and a reassuring message output each time a new IFF layer is encountered in the input IFF file. Other supplementary messages will be output to indicate what stage the program has reached, and if the /LITES2 or /NODE_LIST qualifiers were given, whether or not these files have been created.

/NODE_LIST[=file-spec]

- allows the user to specify a text file which will contain a list of coordinates and number of arms for all junctions in the input IFF file. Any part of the filename which is not explicitly given will be taken from <output IFF file name part>.LIS - this is the actual filename used if none is specified with the qualifier. If a file with the specified name already exists, a new file will be created with the version number incremented by one.

DESCRIPTION

The program first reads the file containing the feature code relative height priorities from LSL$RELHT_PRIO, so this logical name must already have been set up.

Each line in the file is of the form

```
<n1>    <n2>    <FC range>
```

where

n1 is a relative priority starting at 0

n2 is set to 0 if there is no doubt about the relative priority, and 1 if there is likely to be a conflict with other features

FC range is a range of feature codes, e.g. 101-104,111,114,116-118

A relative priority of 0 indicates that the height is not relevant; the same is assumed for any feature codes not in the table.

After reading in the priority table, all of the map data is copied from the input file into the output file. The junction structure is ignored, and dummy AC type 10 and type 11 entries are created for the relative heights of the start and end of each link respectively, with values of 0. At the same time, lists are made of :-

a) pointers to all FS entries in the input file

b) feature codes of all features in the input file

c) pointers to all AC entries in the output file

d) doubt flags for each feature in the output file (see below)

Common boundaries are normally denoted by feature code 999, and the feature codes of the original features which were amalgamated to form the common boundary are given in type 1 AC entries. For such features the program simply checks the number of type 1 AC entries; if there are more than one, the feature is assumed to be a common boundary. The height priorities of these shared features can conflict as there may be more than one feature with a priority greater than zero. If this is the case, then we note that this feature is in doubt. If not, the feature code used is that for the first in the list if all priorities are zero; otherwise it is the first in the list with a non-zero priority.

The second stage of processing is to set the relative heights for all of the links in the output file. Use is made of the IFF junction handling routines to go through each junction in the input file in turn, by scanning every sector.

For each arm of a junction, the following procedure is followed :-

- get the ST pointer and vertex number

- get the feature code by comparing the ST pointer with the list of FS pointers to find the relevant FS entry

- get the height priority from the priority table

- use the index from the FS pointer list in conjunction with the vertex number to get the AC pointer from the AC pointer list

Once this information has been obtained for each arm, the relative heights are reassigned, starting at 0. These values are then written into the relevant AC entries in the output file. Only features that cross another have their AC types 10 and 11 changed from the default value of zero.

The junction positions for any doubtful link ends are recorded in the LITES2 guidance file if one is being produced. A junction is in doubt if either

i) the feature code of one of the arms has a doubt value of 1 in the priority table, or

ii) one of the arms is a common boundary with conflicting height priorities

If a node list is being produced, all junction positions are written to it.

Note that there will be no visible difference to the display of any RELHT output file in LITES2 or other display mechanisms.

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.

COPYING, copying features over to output IFF file

CREATELCM, LITES2 guidance file 'filename' created

CREATELIS, node list file 'filename' created

IFFINCLO, input IFF file closed

IFFINOPN, IFF file 'filename' opened for input

IFFOUTCLO, output IFF file closed

IFFOUTOPN, IFF file 'filename' opened for output

LINENUM, line 'integer' in priority file

NEWLAYER, new layer 'integer' encountered

NOLCM, heights OK, no LITES2 guidance file created

READPRIO, reading feature code priority file 'filename'

SETTING, setting relative heights in output IFF file

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.

ABAOUTFIL, output IFF file abandoned

FEATINCOMP, incomplete feature 'fsn' found in input file, not copied

LINETRUNC, line truncated

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.

BADARMNUM, bad JB arm count 'integer' (max. 'integer') at 'integer' 'integer'

BADJBARM, bad JB pointer 'integer' (LPOSE='integer') at ('integer','integer'), arm 'integer'

BINCHOPERR, internal binary chop error for feature at 'integer'

EOIINFEAT, unexpected end of input IFF file within feature

ERRCLOIN, cannot close input IFF file

ERRCLOLCM, cannot close LITES2 guidance file on unit 'integer'

ERRCLOLIS, cannot close node list file on unit 'integer'

ERRCLOOUT, cannot close output IFF file

ERRCLOPRIO, cannot close feature code priority file on unit 'integer'

ERRDELLCM, cannot delete LITES2 guidance file on unit 'integer'

ERROPNIN, cannot open input IFF file 'filename'

ERROPNLCM, cannot open 'filename' on unit 'integer'

ERROPNLIS, cannot open 'filename' on unit 'integer'

ERROPNOUT, cannot open output IFF file 'filename'

ERROPNPRIO, cannot open 'filename' on unit 'integer'

ERRPOSAC, error positioning to AC entry in output file

ERRREADPRIO, cannot read feature code priority file on unit 'integer'

ERRSELLCM, cannot select LITES2 guidance file on unit 'integer'

ERRSELLIS, cannot select node list file on unit 'integer'

EXPDIGIT, expecting digit in feature code range

EXPFCRANGE, expecting range of feature codes

FIRSTTOOBIG, first number of range greater than maximum

IFFLIBERR, IFF library error

INVALFC, invalid feature code, must be positive

INVALRANGE, first number in range greater than second

MISSINT, range has missing integer

SECONDTOOBIG, second number of range greater than maximum

TOOMANYFEAT, too many features in input file (max. 'integer')

TOOMANYSHARED, too many features in shared feature (max. 'integer')

UNEXPEOF, unexpected end of file

UNEXPEOI, unexpected end of input IFF file

UNEXPEOL, unexpected end of line

UNEXPEOO, unexpected end of output IFF file

MESSAGES (FATAL)

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

SOFTERR, software error

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.

CHAPTER 4

MODULE ICASE

MODULE ICASE

FUNCTION

ICASE is suited to the production of large scale schematic road casings and area fills as often seen in road atlases. Given a junction structured IFF file containing road centrelines, a table of feature codes, priorities and road widths, the program will create an output IFF file containing road casings and area fills for the selected features. It is important to realise that the input road centreline data must be geometrically clean to prevent spurious results occurring.

Note that for small scales work, it is often easier to generate road casings 'on the fly' during display and plotting, using the prioritised multiple representation capabilities of FRTLIB now used in LITES2 and FPP.

FORMAT

```
$ ICASE file-spec file-spec
```

Command qualifiers	**Defaults**
/[NO]CASINGS	/CASINGS
/[NO]FILLS	/FILLS
/FRT=file-spec	No FRT
/PARAMETER=file-spec	Parameter compulsory
/TOLERANCE=real	Minimum casing width

PROMPT

From:	Input-IFF-file
To:	Output-IFF-file

PARAMETERS

Input-IFF-file

- Specifies the IFF file which is to be processed. Any part of the filename which is not specified will be taken from the default specification 'LSL$IF:IFF.IFF;0'.

Output-IFF-file

- Specifies the IFF file which is to be created to contain all the road casings and area fills corresponding to the selected input centreline features. The input road centrelines are not copied to the output file. Any part of the filename which is not explicitly given will be taken from the parsed input specification.

COMMAND QUALIFIERS

/[NO]CASINGS

- this qualifier is used to select the output of road casings. Each input feature declared in the parameter file will be used to create two casings in the output file with the specified width and feature code.

/[NO]FILLS

- this qualifier is used to select the output of fill areas. Each input feature declared in the parameter file will be used to create a corresponding area feature in the output file with the specified width and feature code. Appropriate road ends at junctions will be generated depending on the priorities attached to the input road centreline features.

/FRT=file-spec

- allows the user to specify an FRT file which is used to verify the graphical types of the feature codes given in the parameter file. An error message will occur if any input road centreline or output casing feature codes are not of graphical type 1 or if any output fill areas are not of graphical type 12.

/PARAMETER=file-spec

- this compulsory qualifier specifies a file containing a Road Casing Table. This table declares which features of the input file will be processed and what their respective casings and fills will look like. An example parameter file follows with an explanation of its contents.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Example parameter file containing the                          !
! Road Casing Table for use with ICASE                           !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                                                                !
!                                                                !
!     FC   Priority  CaseFC  FillFC  CaseWid  FillWid            !
!                                                                !
RCT   404     5       1       800     105.0    100.0             !
RCT   405     1       3       802      55.0     50.0             !
!                                                                !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

The columns of the table are all compulsory and are as follows:

RCT - Denotes a line as belonging to the Road Casing Table.

FC - The feature code of input road centrelines. If /FRT is used to verify the parameter file, this feature code must be of graphical type 1.

Priority - The priority of the input road centrelines which is used to determine how the roads should be terminated when roads of differing feature codes meet. The values in this field can be any positive integer and only needs to reflect the priority of roads relative to each other. The road with the highest priority will have all other roads give way to it and will never be broken at junctions unless it terminates.

CaseFC - The feature code of output road casings. If /FRT is used to verify the parameter file, this feature code must be of graphical type 1.

FillFC - The feature code of output area fills. If /FRT is used to verify the parameter file, this feature code must be of graphical type 12.

CaseWid - The width (in IFF units) of the output road casings. This value is the total road width from casing to casing. It should typically be slightly larger than the fill width so that the casing touches the area fill.

FillWid - The width (in IFF units) of the output area fills and should typically be slightly less than the casing width taking into account the width of the casing line.

Lines starting with exclamation marks are comment lines and will be ignored.

/TOLERANCE=real

- allows the user to specify the warning tolerance. If part of a line segment is smaller that this tolerance, a warning message will be given to help identify the feature as potentially causing a problem. If this qualifier is not given, the tolerance will default to the minimum casing width of all entries held in the parameter file.

RESTRICTIONS

The parameter file specified using /PARAMETER is compulsory.

ICASE will not simulate bridges and so heighted data cannot be used to determine if a road crossed another.

DATA PREPARATION

It is **ESSENTIAL** that at least the following three ILINK processes are performed (in the order given) before the program is run:

```
ILINK/BREAK      TEST.IFF TEST.IFJ
ILINK/STRUCTURE  TEST.IFJ TEST.IFJ
ILINK/SORTARMS   TEST.IFJ TEST.IFJ
```

Other ILINK processes (such as /PPJOIN, /LLJOIN or /LPJOIN) may be useful in order to snap points together or onto other lines. Refer to the ILINK reference manual for details.

Remember that the reported FSNs of any input features causing spurious results will be in the junction structured IFJ file, not in the original IFF file. In order to eradicate these short line segments, it is recommended that the **structured** file be edited using LITES2. This is because the original file may not have the erroneous points since ILINK may have introduced them.

When saving the file from LITES2, any structure in the file will be corrupted. Therefore it is vital to **REMEMBER TO RE-STRUCTURE THE IFF FILE USING THE 3 COMPULSORY STAGES OF ILINK** mentioned above.

Even though post editing is likely to be necessary anyway it is probably quicker and more efficient to correct as many problems in the input file as possible. This way, the program can do most of the work and the amount of post-editing can be kept to a minimum.

Until experience is gained in how 'clean' the input data should be, it is likely that spurious result will occur. The first time the program is run on a data set, it is recommended that road fills are disabled to make it easier to see any problems with the casings. This could be done using LITES2 so that the area features are not present to obscure the casings. Once satisfied that all spurious errors have been removed from the casings, the program should be run again to generate both casings and area fills. Once the casings are error free it is likely that the road fills will also be clean since they are narrower than the casings.

In order to prevent the omission of any steps, an example command file is given below. Clearly, the tolerances in the example should be adjusted to reflect the resolution of the particular data set.

```
$!  Example command file for use with ICASE
$!
$       SI LSL$DATA_ROOT:[TEMP]
$!
$       ILINK/BREAK                       TEST.IFF TEST.IFJ
$       ILINK/LPJ/JNTOL=2.0/EXTOL=2.0  TEST.IFJ TEST.IFJ
$       ILINK/PPJ/JNTOL=2.0/EXTOL=2.0  TEST.IFJ TEST.IFJ
$       ILINK/STRUCTURE                   TEST.IFJ TEST.IFJ
$       ILINK/SORTARMS                    TEST.IFJ TEST.IFJ
$!
$       ICASE /PARAMETER=LSL$DATA_ROOT:[TEMP]TEST.PAR -
              /CASINGS -
              /FILLS -
              /FRT=LSL$FRT:TEST.FRT -
              TEST.IFJ -
              TEST_CASED.IFF
$!
$       PURGE TEST.*
$!
```

DESCRIPTION

When the program is invoked, the command line is interpreted and the parameter and FRT file (if specified) are read in. After decoding the command line, an initial pass of the IFF file is performed to check the file contents and store its junction structure internally.

A second pass of the input file is then performed to produce casing and/or area fills corresponding to the input road centrelines specified in the parameter file.

When producing casings, each line segment is considered in turn and casing features generated for each side. To do this, the centrelines joining the junction nodes at each end are used to determine the casing intersection points. It is therefore essential that the node arms have first been sorted using ILINK/SORTARMS in order to determine which centrelines are adjacent to the current one.

Producing area fills is more involved. Priorities assigned to features are used to determine how roads fills will join others that they meet at junction nodes.

Roads joining others with equal priority will have area fills that are 'spliced'. (ie both road area fills taper to a point and join at the junction node). This will look acceptable if the two road fills are of the same colour. An example of this is shown in Figure 1 with road fills of different colours to show the effect. This effect will not occur if all road priorities in the parameter file are different.

When two roads of differing priorities meet, the fill of the lesser priority road will give way to the other and be butted up to it. This is shown by the diagram in Figure 2. This is the more acceptable way of joining road fills and can be achieved if all the priorities in the parameter file are different.

Figure 3 shows examples of erroneous road casings generated because of the presence of short line segments, often at road junctions. It is not always enough for the input road centreline data to be geometrically clean, but must also be devoid of line segments which are shorter than at least the minimum casing width. Usually, this only applies to short line segments at road junctions, but the first diagram in Figure 3 shows and example of a sharp angle containing a short line.

These short lines cause problems because every point in a feature is used to generate the casing and area fill, and if the perpendicular distance of a point from a junction node is less than an adjacent road width, the casing generated from this erroneous point will overshoot the intersection point. Roads joining at an acute angle often cause this effect. An example of this can seen in the right hand diagram of Figure 3.

The output file can now be examined in LITES2 to check for any problems. If these are relatively minor, the output file could be edited to remove them. Severe errors may involve examining the input file for the source of the problem(s) as described above.

FIGURES

Figure 1.

Figure 2.

Figure 3.

EXAMPLES

```
$ ICASE/NOFILLS/CASINGS/PAR=TEST4.PAR/FRT=HERE:TEST TEST4.IFJ TEST4_CASED.IFF
FRT file:          LSL$DATA_ROOT:[TEMP]TEST.FRT;1
Parameter file:    LSL$DATA_ROOT:[TEMP]TEST4.PAR;14
Warning Tolerance: 5.0
Road Casings:      Selected
Road Fills:        Deselected
Performing initial pass of IFF file...
Generating output features...
%ICASE-W-SHORTSEG, short line segment in FSN 23 may give spurious results
%ICASE-W-SHORTSEG, short line segment in FSN 54 may give spurious results
%ICASE-W-SHORTSEG, short line segment in FSN 56 may give spurious results
Operation successful.
 ELAPSED:    0 00:00:11.14  CPU: 0:00:02.10  BUFIO: 7  DIRIO: 166  FAULTS: 121
$
```

This examples shows the warning messages produced because three features contained line segments which were shorter that the warning tolerance. It is recommended that the output file be examined around these features to see if any spurious result occurred. This could be done by reading both the input and output file into LITES2 to see what effect these features had. If there is an obvious problems the input file, TEST4.IFJ, it should be edited and saved. It must then be restructured before running the program again.

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.

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.

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.

BADFC, Feature code 'code' out of range

> **Explanation:** An invalid IFF 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.

IGNORED, FSN 'number' with FC 'code' not in parameter file. Feature ignored

> **Explanation:** An entry for the reported feature code was not present in the parameter file and the feature was ignored.
>
> **User action:** If the feature is not required then no user action is necessary. If it is a line feature of GT 1 and is required to be cased, an entry should be made in the parameter and FRT files and the program run again.

INVALTOL, invalid tolerance. Default value used

> **Explanation:** The tolerance specified on the command line was unacceptable. The tolerance will default to the minimum road or casing width in the parameter file.
>
> **User action:** If the default tolerance is not suitable, specify a correct tolerance and run the program again.

NOPAR, no parameter file specified - any attributes ignored

> **Explanation:** The parameter file was not specified on the command line and any attributes found in the EXPORT file will not be included in the translation but will be ignored.
>
> **User action:** If the EXPORT attributes are desired in the translation, a parameter file must be created and specified on the command line. See the relevant documentation on the parameter file.

SHORTSEG, short line segment in FSN 'number' may give spurious results

> **Explanation:** A line segment (any part of a line between two points) was found to be less than the warning tolerance and so this message occurred.
>
> **User action:** It would be prudent to check the input IFF file containing the road centrelines to make sure that line segments in or near the offending feature are not smaller than the casing width.

UNKCMD, unknown command "'command'"

Explanation: The given command in the parameter file was not recognised. 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: Check the parameter file against the documentation to find the source of the problem and correct the file accordingly.

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.

BADADDRESS, error positioning to ST entry in FSN 'code'

Explanation: This error occurred because the address in a JB block does not correspond to that of an ST entry.

User action: This error may indicate that the IFF file is corrupt. Check that it was successfully structured by ILINK. If there is still a problem, it should be reported to Laser-Scan.

CASEORFILL, illegal combination of qualifiers

Explanation: The qualifiers /NOCASINGS and /NOFILLS were specified on command line. One or both of these should be present but must not both be negated.

User action: Run the program again with the correct combination of command line qualifiers.

ERRCLO, error closing file "'file-spec'"

Explanation: Some form of error occurred in closing one of the input or output files.

User action: Depends upon the associated LSLLIB message.

ERRINNODE, Node point is not consistent with ST entry in FSN 'number'

Explanation: An internal error was detected when checking that node point exists at one end of the point list in the feature. This means that the junction node structure in the file is corrupt.

User action: Ensure that the IFF file has been successfully structured by ILINK. See the documentation for details. If there is still a problem, it should be reported to Laser-Scan.

ERROPN, error opening file "'file-spec'"

Explanation: Some form of error occurred in opening one of the input or output files.

User action: Depends upon the associated LSLLIB messages.

ERRRDFRT, error reading FRT file 'file-spec'

Explanation: An error occurred while reading in the FRT file. A message output by the Feature Representation Library (FRTLIB) will appear before this message, giving further information as to what has gone wrong. For example, the specified FRT file may not exist, or it may be in another directory.

User action: Dependent upon the associated error messages.

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.

IFFCORRUPT, entries of FSN 'number' are missing or in the wrong order

Explanation: IFF entries in or around the offending feature are in the wrong order or missing indicating that the IFF file is corrupt in some way.

User action: Use IPATCH to check the entries in or around the offending feature. Also check the history (HI) entry of the IFF file and try to determine which (if any) process or program may have corrupted the file. If no problem can be found, report this problem to Laser-Scan.

NOFRT, specified FRT file does not exist

Explanation: The FRT file specified with the qualifier on the command line does not exist.

User action: Ensure that the file-spec given with the /FRT qualifier is for a file that exists.

NOIFF, specified IFF file does not exist

Explanation: The IFF file specified on the command line does not exist.

User action: Ensure that the file-spec given on the command line is for a file that exists.

NOPARFILE, specified parameter file does not exist

Explanation: The parameter file specified on the command line does not exist.

User action: Ensure that the file-spec given with the /PARAMETER qualifier is for a file that exists.

NORA, no RA entry could be found in IFF file

Explanation: The RA entry could not be found in the output IFF file in order to update the range information.

User action: This error should not occur under normal conditions and should be reported to Laser-Scan.

UNSTRUCTURED, the input file is unstructured

Explanation: No Junction Pointer (JP) entries were found in the input file which indicates that the IFF file has not been structured using ILINK/STRUCTURE.

User action: Ensure that the IFF file has been processed using the 3 critical stages of ILINK that are necessary before using this utility. See the DATA PREPARATION section for details.

WRONGGT, in parameter file, FC 'code' has an output feature with wrong GT

Explanation: The FRT is used to check that the features specified in the parameter fill are of the correct graphical type. The road centrelines and casing feature codes should be of GT 1 and the road fills should be of GT 12. The FC reported is that of the input centreline in order to easily locate which line of the parameter file has an incorrect value.

User action: Ensure that the correct parameter and FRT files are being used or amend the feature entry in the parameter file to contain a feature with the correct graphical type and run the program again.

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.

INDEX