

*Laser-Scan Ltd.*

*CONVERT PACKAGE*

*IFFDXF Reference*

*Issue 2.0 - 30-September-1994*

Copyright (C) 1994 Laser-Scan Ltd  
Science Park, Milton Road, Cambridge, England CB4 4FY tel: (01223) 420414

Document "IFFDXF REFERENCE", Category "REFERENCE"  
Document Issue 2.0 J Barber (modified 30-Sep-1994)

CONTENTS

IFFDXF - Change Record . . . . .	i
PREFACE . . . . .	1
Intended audience . . . . .	1
Structure of this document . . . . .	1
Conventions used in this document . . . . .	2

CHAPTER 1           DXF FORMAT DESCRIPTION

FORMAT DESCRIPTION . . . . .	1-1
Introduction . . . . .	1-1
The DXF subset supported . . . . .	1-2
HEADER section . . . . .	1-2
TABLES section . . . . .	1-2
BLOCKS section . . . . .	1-2
ENTITIES section . . . . .	1-3
Example of DXF file . . . . .	1-5
Comparison of DXF and IFF . . . . .	1-9

CHAPTER 2           DXF DATA PREPARATION

DATA PREPARATION . . . . .	2-1
The I2DXF and DXF2I parameter file . . . . .	2-1
The layer description table . . . . .	2-1
The entity description table . . . . .	2-2
The text size description table . . . . .	2-4
Mapping restrictions between DXF entities and IFF features . . . . .	2-6
IFF LINEAR graphical type and DXF LINE/POLYLINE . . . . .	2-8
Example of parameter file . . . . .	2-9
Reading of parameter file . . . . .	2-9
MESSAGES (SUCCESS) . . . . .	2-10
MESSAGES (INFORMATIONAL) . . . . .	2-11
MESSAGES (WARNING) . . . . .	2-13
MESSAGES (ERROR) . . . . .	2-15
MESSAGES (OTHER) . . . . .	2-17

CHAPTER 3           I2DXF UTILITY

UTILITY I2DXF . . . . .	3-1
FUNCTION . . . . .	3-1
FORMAT . . . . .	3-1
PROMPTS . . . . .	3-1
PARAMETERS . . . . .	3-1
COMMAND QUALIFIERS . . . . .	3-2
DESCRIPTION . . . . .	3-3
EXAMPLES . . . . .	3-6
MESSAGES (SUCCESS) . . . . .	3-7
MESSAGES (INFORMATIONAL) . . . . .	3-8
MESSAGES (WARNING) . . . . .	3-10

MESSAGES (ERROR) . . . . .	3-14
MESSAGES (OTHER) . . . . .	3-16

CHAPTER 4            DXF2I UTILITY

UTILITY DXF2I . . . . .	4-1
FUNCTION . . . . .	4-1
FORMAT . . . . .	4-1
PROMPTS . . . . .	4-1
PARAMETERS . . . . .	4-1
COMMAND QUALIFIERS . . . . .	4-2
DESCRIPTION . . . . .	4-3
EXAMPLE PARAMETER FILE . . . . .	4-8
EXAMPLES . . . . .	4-11
MESSAGES (SUCCESS) . . . . .	4-12
MESSAGES (INFORMATIONAL) . . . . .	4-13
MESSAGES (WARNING) . . . . .	4-16
MESSAGES (ERROR) . . . . .	4-22
MESSAGES (OTHER) . . . . .	4-29

---

**IFFDXF - Change Record**

---

---

**Version 1.0    S Townrow    17-June-1991**

Module IFFDXF        - Reorganised package documentation.

---

---

**Version 1.1    J Barber    27-June-1991**

Module IFFDXF        - New DXF2I qualifiers /HEADER to create an area feature outlining the DXF file extents in layer 0 (FSN 1, FC 0) to contain some DXF HEADER variables in AC texts, and /QUIET to turn off certain warning messages.

- DXF2I New messages RANGEUPDATE, DEFLAYNAM, DEFTEXHGT, DIFFEXPOFF, DIFFEXPZOFF, DIFFEXPSCALE and DIFFEXPZSCALE, and some messages changed from error to warning (UNDEFBLK, UNDEF LIN, UNDEF LAY and UNDEFTEX).

- DXF2I Substantial changes to the description of DXF2I processing, and limitations, and to the parameter file mechanism, in particular the new EXPAND entry specifying that any entities in the BLOCKS section are to be output to IFF features as they are encountered there. There are also major changes and additions to the way the program works (eg. the use of more DXF header variables to set up missing defaults and also colouring or line types by layer).

---

---

**Version 1.2    J Barber    22-July-1991**

Module IFFDXF        - New I2DXF qualifiers /HEADER to use AC's in the area feature (FSN 1, FC 0 in layer 0) outlining the IFF file extents for the DXF header variables, and /QUIET to turn off certain warning messages.

- I2DXF New messages RDHDRACS to signal (when /LOG is given) that the header feature is being read for the AC entries, and error BADHDRTR when this feature could not be found, or there was some reading error, when /HEADER is given.

- I2DXF New message BADENDOFLAY to signal that the end of layer EO entry could not be found when the current IFF layer number was not in the parameter file for a match to DXF layer name, and the IFF layer is to be skipped over.

- DXF2I New message WRTHDRFTR to signal (when /LOG is given) that the header feature is being written with the AC entries from the DXF HEADER section.

---

**Version 1.3    J Barber    12-September-1991**

Module IFFDXF    - DXF2I messages ERRLAYCOL and ERRLAYLTYPE changed from error to warning.

---

**Version 1.4    J Barber    26-June-1992**

Module IFFDXF    - DXF2I description of the default MD2 entries being unset, with a MD2SCL field of 1.0 and MD2LOC local origin of (0,0).

---

**Version 1.5    J Barber    13-July-1992**

Module IFFDXF    - I2DXF new qualifier /ABSOLUTE to enable the output of absolute coordinates. New messages MDDEFAULT and MDABSENT if the MD entry is not found if /ABSOLUTE is given.

---

**Version 1.6    J Barber    20-October-1992**

Module IFFDXF    - DXF2I new qualifier /ABSOLUTE to enable the output of coordinates relative to the origin contained in the \$EXTMIN header variable.

---

**Version 1.7    J Barber    8-July-1993**

Module IFFDXF    - DXF2I new qualifier /SCAN = scan-file to create a default parameter file from the DXF file, and which may be used as the parameter file for subsequent runs of DXF2I.  
- DXF2I New message TOOMNYENTS to warn that there are too many unique entity, style, colour and feature code combinations (1000 at present) in the DXF file to store while checking for a previous occurrence before output to the scan file in the /SCAN option.  
- DXF2I New message TOOMNYLAYS to warn that there are too many unique layers (1000 at present) in the DXF file to store while checking for a previous occurrence before output to the scan file in the /SCAN option.  
- DXF2I New message ERRSCANLAY to warn that there was an unreferenced layer in the DXF file which will not have been output to the SCAN file, and so must be edited before being used as the PARAMETER file for this DXF file.

- DXF2I New message TOOMNYPNTS to warn that there are too many points (more than 30000 at present) in a POLYLINE feature, which will be abandoned.
- DXF2I New message DEFIFFHDR to warn that a DXF file has been given without a HEADER section, and default IFF header entries will be written for later update.

---

**Version 1.8    J Barber    12-January-1994**

- Module IFFDXF
- I2DXF new qualifier /DECIMAL\_PLACES = integer to specify the number of decimal places to output in the x,y coordinates in the case of IFF files with small absolute magnitude coordinate values where resolution is to be preserved.
  - I2DXF New message SINGPNTDEL to warn that a potential single point line feature has been deleted from a POLYLINE entity. These may arise from the closing invisible moves in nested polygons.

---

**Version 1.9    J Barber    23-February-1994**

- Module IFFDXF
- I2DXF, DXF2I and Preparation
  - DXFPAR New severity for message ERRFNDFC from error to just warn that a feature code referenced in the parameter file is not in the FRT file.

---

**Version 2.0    J Barber    30-September-1994**

- Module IFFDXF
- I2DXF, DXF2I and Preparation
  - DXFPAR New TEXT SIZE facility to specify the mapping from mm's to point sizes for text heights in the programs DXF2I and DXF2I, used if the /POINT\_SIZE qualifier option is used.
  - DXFPAR New ERRTEXTFONT message to warn the user that there was an error in the font number field of the ENTITY TEXT entry of the parameter file.
  - DXFPAR New ERRRDTEXT message to warn the user that there was an error in the TEXT SIZE entries of the parameter file.
-

---

## PREFACE

---

### Intended audience

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

---

### Structure of this document

This document is composed of 2 major sections.

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

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

- |                    |  |
|--------------------|--|
| MODULE             | - the name of the CONVERT module.  |
| FORMAT DESCRIPTION | - a description of the data format written or read by the utility programs in this conversion module.            |
| DATA PREPARATION   | - guidance on how to digitise or prepare the IFF and other data required by the utility programs in this module. |

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

- |                    |   |
|--------------------|---|
| UTILITY            | - the name of the utility.  |
| FUNCTION           | - a synopsis of what the utility does.  |
| FORMAT             | - a summary of the utility command format and command qualifiers. Default qualifier settings are indicated.               |
| PROMPT             | - how it prompts the user.  |
| PARAMETERS         | - description of expected command parameters.   |
| COMMAND QUALIFIERS | - description of all command qualifiers. Qualifiers are ordered alphabetically and default argument values are indicated. |
| RESTRICTIONS       | - a summary of restrictions on the use of   |

	qualifiers
DESCRIPTION	- the definitive description of the utility action.
COMMANDS	- for interactive utilities only, a description of all commands. Commands are ordered alphabetically and default argument values are indicated.
EXAMPLES	- annotated examples of utility useage.
MESSAGES	- all classes of message are listed and described and suggested user action given. The messages are divided into sections according to message severity within which the messages are ordered alphabetically by message mnemonic.

---

## Conventions used in this document

Convention	Meaning
<CR>	The user should press the carriage control key on the terminal
<CTRL/x>	The phrase <CTRL/x> indicates that the user must press the key labelled CTRL while simultaneously pressing another key, for example, <CTRL/Z>.
\$ <b>IFF2SIF</b> <CR>	Command examples show all user entered commands in <b>bold</b> type.
\$ <b>IFF2SIF</b> <CR> . . .	Vertical series of periods, or ellipsis, mean either that not all the data that CONVERT would display in response to the particular command is shown or that not all the data that the user would enter is shown.
file-spec...	Horizontal elipsis indicates that additional parameters, values or information can be entered.
[logical-name]	Square brackets indicate that the enclosed item is optional. (Square brackets are not, however, optional in the syntax of a directory name in a file-specification, or in the syntax of a substring specification in a VMS assisment statement).

---



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

## CHAPTER 1

### DXF FORMAT DESCRIPTION

---

**FORMAT DESCRIPTION**

---

**Introduction**

The Digital eXchange Format (DXF) is the format produced by several CAD packages, including Auto-CAD and ZMAP. DXF is produced by Auto-CAD program DXFOUT and is read back in by Auto-CAD program DXFIN. For a brief comparison of the way IFF and DXF regard data, see the section on COMPARISON OF DXF AND IFF below.

The format produced is an ASCII text format, as described in the Auto-CAD Release 10 Reference Manual, Appendix C.

For more information on the IFF file structure see the **IFF User's Guide** of the Laser-Scan MAPPING package documentation.

-----  
**The DXF subset supported**

There are four sections in a DXF files: HEADER, TABLES, BLOCKS and ENTITIES. A file consists of a series of groups, each of which has a code and a value.

**HEADER section**

The HEADER section contains various settings of variables associated with the drawing. Several variables from this section are used in the conversion to and from IFF. They include \$EXTMIN, \$EXTMAX (to give the IFF range), \$CECOLOR giving the colour derivation; whether BYLAYER (or BYBLOCK, not supported), \$CELTYPE giving the default line type derivation, or BYLAYER or BYBLOCK (not supported), \$TEXTSTYLE (default text style), \$TESTSIZE (default text height) and \$CLAYER (default layer name). If the entities are to be coloured by layer, the value of \$CECOLOR will be set to 256. A value of zero signifies colour by block (not supported).

**TABLES section**

The TABLES section contains three supported tables: LTYPE, LAYER and STYLE. They define various basic characteristics of the drawing and are referenced by the ENTITIES section (see ENTITIES section).

- o The LAYER table defines a set of transparent areas where the entities logically lie. The only items supported are the layer name (2 group), layer line type (6 group) and the layer colour index (62 group). The layer colour index or line type items will only be used when the entities in the drawing are to be grouped by layer (that is, when \$CECOLOR in the HEADER section is set to 256, or \$CELTYPE is BYLAYER). If the entities are to be grouped by layer, the colour entries or line types encountered in the LAYER table will override those defined in individual entities in the ENTITIES section. Users should note that all entities in the ENTITIES section must reference a valid layer defined in the LAYER table, either given explicitly, or the default from \$CLAYER.
- o The LTYPE table defines the linetypes to be used for DXF linear entities. The only item supported is the linetype name (2 group).
- o The STYLE table defines the text style of DXF text entities. The only item supported is the text style name (2 group).

**BLOCKS section**

The BLOCKS section contains all the block definitions. A block, in DXF terms, is defined as a group of graphical entities which represent a symbol definition in the IFF context. A block is referenced by the INSERT entities in the ENTITIES section. Normally, only the block name entry (2 group) is supported and treated as a symbol representation of the block. However, if there is an EXPAND entry in the parameter file, the constituent elements of the BLOCK (symbol) are

output to IFF with DXF2I, modifying the internal coordinates with the offsets and scalings given in the parameter file representing the final INSERTion point and scale. There is no reverse translation from IFF to DXF BLOCKS.

### ENTITIES section

The ENTITIES section contains all the graphical data in the drawing. The entity types supported are: LINE, POINT, CIRCLE, ARC, TRACE, SOLID, TEXT, INSERT, POLYLINE and VERTEX. The entity types not currently supported are: SHAPE, ATTDEF and ATTRIB. Entities are defined by a series of group values which indicate the type of value they represent, followed by the actual value. Each entity begins with a 0 group identifying the entity type. Every entity contains an 8 group which gives the name of the layer in which the entity resides. A 6 group gives the linetype name (if not by layer). Note that the layer names and linetype names are defined in the LAYER table and LTYPE table of the TABLES section respectively. The default linetype name is 'CONTINUOUS', and default text style 'STANDARD'. A 62 group gives the colour index number (if not by layer). It is to be used only when the entities are not grouped by layer, otherwise the colour index number defined in the LAYER table of the TABLES section will be used instead.

The following groups may be included in any entity if it has non-default values for the properties:

6	LINE TYPE NAME
38	ELEVATION
39	THICKNESS
62	COLOUR NUMBER

The rest of the groups that make up an entity depend upon the entity type. These are:

LINE	10, 20 and 30 (start point coords), 11, 21 and 31 (end point coords). Line features in DXF are restricted to two point segments. More complex lines are defined by POLYLINE features (see below).
POINT	10, 20 and 30 (x, y and z coordinates)
CIRCLE	10, 20 and 30 (centre) and 40 (radius)
ARC	10, 20 and 30 (centre), 40 (radius), 50 (start angle) and 51 (end angle)
TRACE	four points defining the corners of the trace: 10, 20 and 30; 11, 21 and 31; 12, 22 and 32; 13, 23 and 33
SOLID	as for TRACE. If only three points were entered, the third and fourth points will be the same.

TEXT	10, 20 and 30 (insertion point), 40 (height), 1 (text value), 50 (rotation angle) and 7 (text style name - default is 'STANDARD'). The default linetype is 'CONTINUOUS'. Note that the style name is defined in the STYLE table of the TABLES section.
INSERT	2 (symbol/block name), 10, 20 and 30 (insertion point), 41 (X scale factor - taken to be the overall scale factor), 42 (Y scale factor) and 50 (rotation angle). Note that the block name is defined in the BLOCKS section.
POLYLINE	70 (closure flag - specifies whether feature is closed, DXF does not repeat the first coordinate of a closed feature), 66 (vertices follow flag). The coordinate information of a POLYLINE entity is provided by following VERTEX entities.
VERTEX	10, 20 and 30 (location) VERTEX information is the positional data for the preceding POLYLINE entity.

-----  
**Example of DXF file**

```
0                ! (begin HEADER section)
SECTION
2
HEADER
9                ! (map maximum limits)
$LIMMAX
10               ! (X max limit)
1200.00000
20              ! (Y max limit)
840.00000
9                ! (map minimum limits)
$LIMMIN
10               ! (X min limit)
0.00000
20              ! (Y min limit)
0.00000
9                ! (entities grouping method)
$CECOLOR
62
  256            ! (grouped by layer)
0
ENDSEC          ! (end HEADER section)
0                ! (begin TABLES section)
SECTION
2
TABLES
0                ! (begin LTYPE table)
TABLE
2
LTYPE
70              ! (number of LTYPE tables - 2)
  2
0                ! (first LTYPE)
LTYPE
2                ! (linetype name - CONTINUOUS)
CONTINUOUS
0                ! (second LTYPE)
LTYPE
2                ! (linetype name - Dashed)
Dashed
0
ENDTAB          ! (end LTYPE table)
0                ! (begin STYLE table)
TABLE
2
STYLE
70              ! (total number of STYLE tables - 2)
  2
0                ! (first STYLE)
STYLE
2                ! (style name - STANDARD)
STANDARD
```

```

0          ! (second STYLE)
STYLE
2          ! (style name - American)
American
0
ENDTAB    ! (end STYLE table)
0          ! (begin LAYER table)
TABLE
2
LAYER
70        ! (total number of LAYER tables - 3)
3
0          ! (first LAYER)
LAYER
2          ! (layer name - Road)
Road
62        ! (layer colour - 1)
1
0          ! (second LAYER)
LAYER
2          ! (layer name - Culture)
Culture
62        ! (layer colour - 10)
10
0          ! (third LAYER)
LAYER
2          ! (layer name - Pipeline)
Pipeline
62        ! (layer colour - 11)
11
0
ENDTAB    ! (end LAYER table)
0
ENDSEC    ! (end TABLES section)
0          ! (begin BLOCKS section)
SECTION
2
BLOCKS
0          ! (first BLOCK)
BLOCK
6          ! (linetype name - CONTINUOUS) (default)
CONTINUOUS
2          ! (block name - Church)
Church
70        ! (reference flag)
64
10        ! (X position)
0.0
20        ! (Y position)
0.0
0
ENDBLK    ! (end of first block)
0
ENDSEC    ! (end BLOCKS section)
0          ! (begin ENTITIES section)
SECTION

```



```

  2
ENTITIES
  0                ! (first ENTITY - ARC)
ARC
  8                ! (resident layer - Road)
Road
  6                ! (linetype - Dashed)
Dashed
  62               ! (colour - 1)
    1
  10               ! (X centre)
100.000
  20               ! (Y centre)
100.000
  40               ! (arc radius)
50.000
  50               ! (arc starting angle)
20.000
  51               ! (arc ending angle)
30.000
  0                ! (second ENTITY - TEXT)
TEXT
  8                ! (resident layer - Pipeline)
Pipeline
  6                ! (linetype - CONTINUOUS) (default)
CONTINUOUS
  7                ! (text style - STANDARD) (default)
STANDARD
  62               ! (colour - 11)
    1
  10               ! (X text position)
100.000
  20               ! (text position)
100.000
  40               ! (text scale)
3.000
  50               ! (text rotation)
20.000
  1                ! (text value)
example
  0                ! (third ENTITY - INSERT)
INSERT
  8                ! (resident layer - Culture)
Culture
  2                ! (name of block to be used - Church)
Church
  62               ! (colour - 10)
    10
  10               ! (X insertion position)
150.000
  20               ! (Y insertion position)
150.000
  42               ! (block scale)
3.000
  50               ! (block rotation)
21.000

```

```
0
ENDSEC                !(end ENTITIES section)
0
EOF                   !(end of file)
```

---

### Comparison of DXF and IFF

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

DXF format shares this concept of a 'feature', but the features are not uniquely feature coded. Each individual feature has coding relating to its graphical appearance only such as colour and line type. Control over these parameters may be exercised individually for each entity or by defining overall values for each layer - the features are notionally coded by grouping similar features in a layer which has a name describing its contents.

The method of coding to be applied is determined by the setting of the \$CECOLOR variable in the HEADER section. If the value of the variable is set to 256, all the entities of the drawing will be grouped by the layer they reside in and will appear with the colour defined in the LAYER table of the TABLES section. If the value is set to 0, all the entities will be grouped by block (note that this option is not supported).

## CHAPTER 2

### DXF DATA PREPARATION

---

## DATA PREPARATION

In order to facilitate a flexible transfer of data between IFF and DXF formats, lookup tables are used. These determine how IFF features are translated into DXF entities and vice versa. It will be possible to use the same lookup table for both programs if a one-to-one translation is maintained.

It is also possible to use the same lookup table to define a conversion between DXF text height (in mms.) and point sizes.

---

### The I2DXF and DXF2I parameter file

The parameter file contains two lookup tables :-

- a) the layer description table
- b) the entity description table
- c) the text size description table

Each line of a particular table is prefixed with a command which identifies the line as belonging to that table. The lines are free-format, so the order of entries matters, but the actual position of entries on the line is not important. Commands may be in upper or lower case.

A maximum of 10000 entries will be allowed per parameter file.

Any text starting with the "!" character is regarded as a comment, and is ignored. Comments can appear on the same lines as commands, or on separate lines.

### The layer description table

The layer description table defines the mapping between DXF layers and IFF layers. The DXF layer is identified by a character string representing the layer name while the IFF layer is identified by a unique integer. Each line is prefixed with the LAYER command.

A line has the following form:-

**LAYER    name    number**

where the fields are:-

- name**    - the DXF layer name, a character string, as defined in the LAYER table of the TABLES section in the DXF file.
- number** - the IFF layer number, an integer, as defined in the IFF file.

The character string format does not allow the use of the space character and is case sensitive.

All numerical entries must be non-negative, and all fields must be present.

### The entity description table

The entity description table describes the translations required to produce DXF entities from IFF features or vice versa. Each line is prefixed with the ENTITY command, which has one of the secondary commands illustrated below. The secondary commands correspond to DXF entity types, except for the EXPAND secondary command, which signifies that any entities constituting a BLOCK (symbol) in the BLOCKS section of the DXF file should be output to IFF using the current entity conversions present. This expansion will use the specified offsets and scalings to alter the internal symbol coordinates to get the world coordinates (using  $x_{new} = Xscale \cdot x + Xoff$ ). These offsets and scalings will be in the final INSERT for the symbol, unless there is a hierarchy of BLOCKS within BLOCKS each with different scalings/offsets, in which case the final scaling and offset must be calculated and used in the parameter file. This EXPAND entry is only used for DXF2I, as there is no mechanism to reconstruct a BLOCK structure from IFF.

A line has one of the following forms :-

ENTITY	LINE	line_type	colour	fc
ENTITY	POINT		colour	fc
ENTITY	CIRCLE	line_type	colour	fc
ENTITY	ARC	line_type	colour	fc
ENTITY	TRACE	line_type	colour	fc
ENTITY	SOLID	line_type	colour	fc
ENTITY	TEXT	text_style	colour	fc
ENTITY	POLYLINE	line_type	colour	fc
ENTITY	INSERT	block_name	colour	fc

ENTITY	EXPAND	block_name	Xoff	Yoff	Zoff	Xscale	Yscale	Zscale
--------	--------	------------	------	------	------	--------	--------	--------

where the fields are :-

**line\_type** - the line type name, a character string, as defined in the LTYPE table of the TABLES section in the parameter file.

**text\_style** - the text style name, a character string, as defined in the STYLE table of the TABLES section in the parameter file.

**block\_name** - the block (symbol) name, a character string, as defined in the BLOCKS section in the parameter file.

**colour** - the colour index, integer.

**fc** - the IFF feature code, integer.

Note that the colour index specified in the parameter file overrides those in the FRT file if there is any difference.

The character string format does not allow the use of the space character and is case sensitive.

All numerical entries must be non-negative, (except for the colour, group 62 entry, which can use negative values) and all fields must be present.

There is no individual entry field for VERTEX as VERTEX is considered as part of the definition of POLYLINE.

Text heights will be read from/written to IFF TH entries, (ENABLE HEIGHT will be required in LITES2). DXF uses mms. in the text height fields, whereas IFF TH entries may represent mms./100, or point sizes, and the translation to use is set by a /POINT\_SIZE qualifier in both programs DXF2I or I2DXF.

If the point size option is enabled in either program, there is a facility to translate between mms. and point sizes. The ENTITY TEXT line may have an optional font number assigned to the FC/text style/colour combination. This font number must lie between 1 and 100, and refers to the font number used in the TEXT SIZE entries which set up assignments between mms. and point sizes for each font. If the font exists on the entity line, the TEXT SIZE entries are read and used, otherwise some program defaults are used.

The following reserved names must be specified in the parameter file when converting from DXF to IFF format, if their relevant fields are not explicitly specified in the DXF file:

- o LTYPE 'CONTINUOUS' for all DXF linear entities, LINE, CIRCLE, ARC, TRACE, SOLID and POLYLINE.
- o STYLE 'STANDARD' for DXF TEXT entities.

Note that upper case should be used for these values in order to denote their default nature (i.e. 'CONTINUOUS' and 'STANDARD').

There are nine default line types supplied by I2DXF. They are:

- o 'CONTINUOUS' \_\_\_\_\_
- o 'DASHED'     — — — — — — — — — —
- o 'HIDDEN'     - - - - - - - - - -
- o 'CENTRE'     \_\_\_\_\_ - \_\_\_\_\_ - \_\_\_\_\_
- o 'PHANTOM'    \_\_\_\_\_ - - - \_\_\_\_\_ - - - \_\_\_\_\_
- o 'DOT'        .....
- o 'DASHDOT'    — . — . — . — . — .
- o 'BORDER'     — — . — — . — — . —
- o 'DIVIDE'     — . . — . . — . . — .

There are also three default text styles supplied by I2DXF. They are:

- o 'STANDARD'
- o 'ROMANC'
- o 'ROMANT'

For users who wish to output a line type other than the nine provided by I2DXF (see above), they can do so by specifying their user-specific line type in the appropriate column of the parameter file. However, as the graphical representation for this user-specific line type is not supported by I2DXF, the graphical representation of the default line type CONTINUOUS is used instead. Any DXF entities with this user-specific line type are therefore graphically indistinguishable from those entities with CONTINUOUS line type. Users who are interested in customising their line types are advised to refer to Section B5 of the AutoCad Reference Manual.

Similarly, any DXF text entities with text style other than the three provided by I2DXF (see above) are graphically indistinguishable from those entities with STANDARD text style. Users who are interested in customising their text styles are advised to refer to Section B7 of the AutoCad Reference Manual.

#### **The text size description table**

The text size description table defines the mapping between DXF text heights (in mms.) and point sizes. It requires the /POINT\_SIZE qualifier to be used, so that any IFF TH entries are interpreted as points, instead of mms./100.

A line has the following form:-

<b>TEXT SIZE</b>	<b>font_number</b>	<b>point_size</b>	<b>mm_size</b>
------------------	--------------------	-------------------	----------------

where the fields are:-

<b>font_number</b>	- the font number, a number between 1 and 100, representing the particular font assigned to the particular FC/text style/colour combination in the ENTITY TEXT line.
<b>point_size</b>	- the IFF point size number, an integer between 1 and 100, used in the IFF TH entry.
<b>mm_size</b>	- the DXF mm. size, a real number used in the DXF text height entry.

All numerical entries must be positive and in their stated ranges, and all fields must be present. Any mm. text heights will be interpolated to the nearest point value given in the table. If none exists, the last mm. size read is used as a default, with an appropriate message given.

An example use of these fields is given below:

!	type	style	col	fc	font
---	------	-------	-----	----	------



ENTITY	TEXT	STANDARD	2	28	3
!	font	point	mm.s		
TEXT SIZE	3	1	0.12		
TEXT SIZE	3	2	0.33		
TEXT SIZE	3	3	0.52		
TEXT SIZE	3	4	0.92		
TEXT SIZE	3	5	1.00		
TEXT SIZE	3	6	1.25		
TEXT SIZE	3	7	1.55		
TEXT SIZE	3	8	2.00		

-----  
**Mapping restrictions between DXF entities and IFF features**

The mapping between DXF entities and IFF features is restricted by the nature of the DXF entity types and their associated IFF graphical types.

It is possible to define a many-to-one mapping for the conversion between some IFF graphical types and DXF entity types, i.e. when converting from IFF to DXF format, allowing more than one IFF graphical type to map onto only one DXF entity type, and when converting from DXF to IFF format, allowing more than one DXF entity type to map onto only one IFF graphical type.

For example, when converting from IFF to DXF format, an IFF UNORIENTED SYMBOL feature can either map onto a DXF POINT entity or a DXF INSERT entity. However, when converting in the reverse direction, a DXF POINT entity can only map onto an IFF UNORIENTED SYMBOL feature.

The following is the summary of the mapping restrictions from IFF graphical types to DXF entity types:-

IFF Graphical Types	DXF Entities
1 = LINEAR	LINE TRACE SOLID POLYLINE
2 = CLOCKWISE ARC	ARC
3 = ANTICLOCKWISE ARC	ARC
4 = CIRCUMCIRCLE ARC	ARC
5 = FULL CIRCUMCIRCLE	CIRCLE
6 = INTERPOLATED CURVE	POLYLINE
7 = UNORIENTED SYMBOL	POINT INSERT
8 = ORIENTED SYMBOL	INSERT
9 = SCALED SYMBOL	INSERT
10 = TEXT	TEXT
11 = SYMBOL STRING	POLYLINE
12 = FILL AREA	POLYLINE TRACE SOLID

The following is the summary of the mapping restrictions from DXF entity types to IFF graphical types:-

DXF Entities	IFF Features(GT)
LINE	LINEAR (1)
POINT	UNORIENTED SYMBOL (7)
CIRCLE	FULL CIRCUMCIRCLE (5)
ARC	CLOCKWISE ARC (2)
	ANTICLOCKWISE ARC (3)
	CIRCUMCIRCLE ARC (4)
TRACE	LINEAR (1)
	FILL AREA (12)
SOLID	LINEAR (1)
	FILL AREA (12)
TEXT	TEXT (10)
INSERT	UNORIENTED SYMBOL (7)
	ORIENTED SYMBOL (8)
	SCALED SYMBOL (9)
POLYLINE	LINEAR (1)
	INTERPOLATED CURVE (6)
	SYMBOL STRING (11)
	FILL AREA (12)

It is also possible to define a many-to-one mapping for the conversion between the IFF features and DXF entities of the same type, i.e. when converting from IFF to DXF format, allowing more than one IFF feature code to map onto only one DXF entity, and when converting from DXF to IFF format, allowing more than one DXF entity to map onto only one IFF feature code.

For example, given the following ENTITY lookup table :-

ENTITY	DXF_type	line_type	Colour	FC
ENTITY	POLYLINE	DASHED	3	20
ENTITY	POLYLINE	DASHED	3	21

If feature code(FC) 20 and feature code (FC) 21 both represent fill areas (IFF graphical type 12), the conversion from IFF to DXF format is unambiguous since they bear a many-to-one mapping relationship, i.e. more than one IFF feature code is mapped onto only one DXF entity. The only resulting DXF entity is POLYLINE bearing linetype 'DASHED' and a colour index of 3.

The use of this many-to-one mapping method does allow a certain degree of flexibility but it is not without a problem. In the above example, problems will occur if we try to convert data from DXF format back to IFF format using the same parameter file. Since the two POLYLINE entities have the same characteristics, i.e. linetype 'DASHED' and colour index 3, the program will not be able to uniquely identify their corresponding feature code entries. When ambiguities of this kind occur, the program will use the first feature code entry, i.e. FC 20 in this particular example.

In summary, when converting from IFF to DXF format, the feature code (FC) entry should be unique, otherwise the first of the duplicated feature codes will be used. Similarly, when converting from DXF to IFF format, the entities' characteristics combination, i.e. DXF entity type, DXF entity name (depends on entity type) and colour should be unique, otherwise only the first of the duplicated combinations will be used.

#### **IFF LINEAR graphical type and DXF LINE/POLYLINE**

There is no distinction in IFF format between line features with only two points and those with more than two. In DXF format, line entities with only two data points are stored as LINE while those entities with more than two data points are stored as POLYLINE. Problems will occur when converting from IFF to DXF format since the IFF feature code(FC) entries in the parameter file should be unique, i.e. the user cannot specify both DXF LINE and POLYLINE entity types for the same IFF feature code.

As a result of this, for I2DXF, the following rules then apply :-

- o If the input IFF feature has only two data points, I2DXF will convert the feature into DXF LINE format provided the entity type specified by the user in the parameter file is either LINE or POLYLINE.
- o If the input IFF feature has more than two data points, I2DXF will convert it into DXF POLYLINE format provided the entity type specified by the user in the PARAMETER file is either LINE or POLYLINE.

For example, given an IFF file with only two linear features, FSN 1 with only two data points and FSN 2 with three data points (both with feature code 30), and the following parameter file:

<b>ENTITY</b>	<b>LINE</b>	<b>DASHED</b>	<b>3</b>	<b>30</b>
---------------	-------------	---------------	----------	-----------

When converting from IFF to DXF format, I2DXF will convert FSN 1 into DXF LINE and FSN 2 into DXF POLYLINE.

Problems will occur when converting the so-generated DXF file back into IFF format using the same parameter file. Errors will be encountered since the DXF POLYLINE entity type is not specified in the original parameter file. To account for the DXF POLYLINE, a new command line has to be added to the parameter file to define the mapping between DXF POLYLINE and IFF feature code (FC) 30. The new parameter file will be in the following form:

<b>ENTITY</b>	<b>LINE</b>	<b>DASHED</b>	<b>3</b>	<b>30</b>
<b>ENTITY</b>	<b>POLYLINE</b>	<b>DASHED</b>	<b>3</b>	<b>30</b>

Users are advised to design the mapping in the parameter file carefully if the parameter file is to be used in both directions and the integrity of the data is to be preserved.

---

**Example of parameter file**

```

!
! Layer lookup table
!
!           DXF_LAYER_NAME           IFF_LAYER_ID
!
LAYER      DXF_Road                   1
LAYER      DXF_Culture                2
LAYER      DXF_Pipeline               3
!
! Entity lookup table
!
!           ENTITY_TYPE      NAME           Colour      IFF_FC
!
ENTITY     LINE              DASHED         3            20
ENTITY     POLYLINE          DASHED         3            20
ENTITY     LINE              CONTINUOUS     4            1
ENTITY     POLYLINE          CONTINUOUS     4            1
ENTITY     TRACE             DASHED         5            40
ENTITY     SOLID             HIDDEN        8            21
ENTITY     POLYLINE          DASHED        16           12
ENTITY     ARC               CONTINUOUS     2            49
ENTITY     ARC               DASHED         2            55
ENTITY     CIRCLE            CENTRE         7            67
ENTITY     POINT             6            189
ENTITY     INSERT            petrol_stn     3            201
ENTITY     INSERT            church         2            223
ENTITY     TEXT              STANDARD      10           78
ENTITY     POLYLINE          DASHED        11           34
ENTITY     ARC               User_defined  10           48
ENTITY     POINT             13           180
ENTITY     TEXT              ROMANC         3            89

```

Note that the ARC with 'User\_defined' line type is graphically indistinguishable from those ARC with 'CONTINUOUS' line type.

Note that the INSERTs (symbols) 'petrol\_stn' and 'church' are represented by two single points as I2DXF/DXF2I do not support the conversion of symbol definitions.

---

**Reading of parameter file**

I2DXF and DXF2I utilise the same library, DXFPARLIB, for reading the parameter file. Messages output from this library are prefixed by '%DXFPAR\_'. These messages are documented at the end of the DATA PREPARATION section.

---

**MESSAGES (SUCCESS)**

These messages are used to indicate that the program has succeeded in performing some action, and do not require any user action.

NORMAL, successful function return

**Explanation:** This message is used internally by the program for debugging purposes and will not normally appear to users.

**User action:** None.

---

**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.

BLOCKFND, number of BLOCK(symbol)s found: 'number'

**Explanation:** This message informs the user of the total number of BLOCKs(symbols) found in the parameter file. This message will be displayed when the /LOG qualifier is selected.

**User action:** None.

ENTITYFND, number of ENTITYs found: 'number'

**Explanation:** This message informs the user of the total number of ENTITYs found in the parameter file. This message will be displayed when the /LOG qualifier is selected.

**User action:** None.

LAYERFND, number of LAYERs found: 'number'

**Explanation:** This message informs the user of the total number of LAYERs found in the parameter file. This message will be displayed when the /LOG qualifier is selected.

**User action:** None.

LTYPEFND, number of LTYPEs found: 'number'

**Explanation:** This message informs the user of the total number of LTYPEs found in the parameter file. This message will be displayed when the /LOG qualifier is selected.

**User action:** None.

OPNPAR, PARAMETER file 'name' opened for read

**Explanation:** This message informs the user that the parameter file has been successfully opened for read. This message will be displayed when the /LOG qualifier is selected.

**User action:** None.

RDFRTERR, error reading FRT data

**Explanation:** There was a problem reading the FRT data in the parameter file. This message is used internally and not seen by the the user.

**User action:** Check the FRT or parameter file and correct as necessary.

STYLEFND, number of STYLEs found: 'number'

**Explanation:** This message informs the user of the total number of STYLEs found in the parameter file. This message will be displayed when the /LOG qualifier is selected.

**User action:** None.



-----  
**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.

COLNOTMAT, colours in PARAMETER and FRT files do not match, FC 'number', colour 'number'

**Explanation:** The colour specified by the entry in the parameter file does not match that found in the FRT file. The FC and colour number will be given in the message. The program will use the colour index in the parameter file (i.e. the colour index explicitly specified by the user) for that particular entity. This message may be turned off by use of the /QUIET qualifier.

**User action:** Check the consistency of colour fields in both PARAMETER and FRT file.

ERRFNDFC, error finding FC 'number' in FRT file

**Explanation:** Error finding IFF feature code (FC) in the FRT file having, having read the FC from the parameter file. This error may be caused either by having the wrong FC entry in the parameter file, or specifying the wrong FRT file. The program will continue upon the detection of this error, but may incur other errors later when trying to translate this FC on encountering it in the IFF file.

**User action:** Check the FC in the ENTITY entry in the parameter file, and the specification of the FRT file.

UNKPRICMD, unknown primary command parameter at line 'number'

**Explanation:** The primary command read from the parameter file is not recognised. Valid primary parameter entries are LAYER and ENTITY. The program will continue ignoring the current line.

**User action:** Check the parameter file.

UNKSECCMD, unknown secondary command parameter at line 'number'

**Explanation:** The secondary command read from the parameter file is not recognised. Valid secondary parameter entries are LINE, POINT, CIRCLE, ARC, TRACE, SOLID, TEXT, INSERT, and POLYLINE. The program will continue ignoring the current line.

**User action:** Check the parameter file.

UNKTEXTCMD, unknown second TEXT command parameter at line 'number'

**Explanation:** The secondary command read from the parameter file for the TEXT command is not recognised. Valid secondary parameter entries are SIZE.

**User action:** Check the parameter file, and rerun the program.

-----  
**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.

ERROPNPAR, error opening parameter file: 'filename'

**Explanation:** The parameter file could not be opened. The program will not be able to continue upon the detection of this error.

**User action:** Check the existence and status of the parameter file.

ERRRDENT, error reading ENTITY entry at line 'number'

**Explanation:** Error encountered when reading ENTITY entry in the parameter file. This may be caused either by a missing field or a negative numerical entry. The program will not be able to continue upon the detection of this error.

**User action:** Check the ENTITY entry in the parameter file.

ERRRDLAY, error reading LAYER entry at line 'number'

**Explanation:** Error encountered when reading LAYER entry in the parameter file. This may be caused either by a missing field or a negative IFF LAYER number. The program will not be able to continue upon the detection of this error.

**User action:** Check the LAYER entry in the parameter file.

ERRRDPAR, error reading parameter file at line 'number'

**Explanation:** The indicated line number of the parameter file could not be read. The program will not be able to continue upon the detection of this error.

**User action:** Check the parameter file at the indicated location.

ERRRDTEXT, error reading TEXT entry at line 'number'

**Explanation:** Error encountered when reading TEXT SIZE entry in the parameter file. This may be caused either by a missing field or a negative number. The program will not be able to continue upon the detection of this error.

**User action:** Check the TEXT SIZE entry in the parameter file.

ERRTEXTFONT, error in font entry in TEXT entry at line 'number'

**Explanation:** There was an error in the font supplied in the ENTITY TEXT entry. This may be caused either by a negative or unreadable number. The program will not be able to continue upon the detection of this error.

**User action:** Check the TEXT entry in the parameter file.

PARFILERR, error reading DXF parameter file

**Explanation:** There was an error in the specified parameter file rendering it unreadable. Other messages will give further information and the line number.

**User action:** Check the parameter file and correct the 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 3

### I2DXF UTILITY

---

## UTILITY I2DXF

---

### FUNCTION

I2DXF is a Laser-scan IFF format to Auto-CAD DXF ASCII format converter. It reads from a standard IFF file and produces a standard DXF text file. Output is directed to disk file.

---

### FORMAT

\$ I2DXF input-file-spec output-file-spec

#### Command qualifiers

/[NO]ABSOLUTE  
/[NO]DECIMAL\_PLACES  
/FRT=file-spec  
/[NO]HEADER  
/[NO]LOG  
/PARAMETER=file-spec  
/[NO]POINTSIZE  
/[NO]QUIET

#### Defaults

/NOABSOLUTE  
/NODECIMAL\_PLACES  
No defaults  
/NOHEADER  
/NOLOG  
No defaults  
/NOPOINTSIZE  
/NOQUIET

---

### PROMPTS

Input IFF filename : input-file-spec  
Output DXF filename : output-file-spec

---

### PARAMETERS

input-file-spec

- specifies the IFF file which is to be converted into DXF format. Any part of the file name which is not specified will be taken from the default specification 'LSL\$IF:IFF.IFF'.

output-file-spec

- specifies the DXF file which is to be created. Any part of the file name which is not explicitly given will be taken from the default specification 'LSL\$DXF:DXF.DXF'.

Note that both logical names, 'LSL\$IF:' and 'LSL\$DXF' must be defined, or overridden.

---

## COMMAND QUALIFIERS

### /[NO]ABSOLUTE

- enables the output of absolute coordinates to DXF. The default action is to output the local IFF coordinates. Error messages will be output if the MD entry is missing, or does not contain a local origin, if this qualifier is given.

### /[NO]DECIMAL\_PLACES

/DECIMAL\_PLACES = integer

- enables the user to specify the number of decimal places output in the x,y coordinates. This may be useful for the output to DXF from IFF files which contain coordinates of small absolute magnitude where resolution is to be preserved. The supplied number should be between 1 and 9. If this qualifier is not given, the default output format is with 3 decimal places.

### /FRT = file-spec

- causes an FRT file, containing definitions of feature codes to be read in. These definitions will then be used to identify the features read in the IFF file. The default is LSL\$FRT:DXF.FRT. This qualifier is mandatory.

### /[NO]HEADER

- enables the text fields of type 6 AC's attached to the area feature (FSN 1, FC 0 in LAYER 0) to be read and used for some of the DXF HEADER variables.

### /[NO]LOG

- enables a range of informational messages concerning the progress of the data translation to be displayed on the terminal.

### /PARAMETER = file-spec

- specifies the parameter file which defines the transformation of feature codes in the IFF data to entity types in DXF. The parameter file-spec is parsed from the default LSL\$LOOKUP:DXF.PAR. Any part of the file name which is not specified by the /PARAMETER qualifier will be taken from the above default specification. This qualifier is mandatory.

### /[NO]POINTSIZE

- signals that the contents of any TH (text height) entry is treated as a point size, instead of the usual mm./100 for the default of /NOPOINTSIZE, and converted into mm. in DXF.



/[NO]QUIET

- enables the suppression of a range of messages during the data translation to be displayed on the terminal, including COLNOTMAT, LINEUSED and POLYUSED.

---

## DESCRIPTION

After the command line is decoded, relevant files will be opened. The program then scans through the parameter file to extract the definitions of the lookup mapping between the two formats. The mapping of IFF/DXF layers and that of IFF features/DXF entities will then be defined and stored. If a /LOG qualifier is included, informational messages will be displayed to inform the user of the number of layers, features/entities, symbols/blocks, linetypes and text styles found in the parameter file. Any illegal command entry will cause a warning message to be issued. Other errors detected in this stage will cause the program to abort.

After the mapping between the two formats is successfully defined, the program then accesses the Feature Representation Table (FRT) file to check for the validity of Feature Code (FC) and colour entries in the ENTITY lookup table. Again, the program will warn if any of the feature codes specified in the parameter file cannot be found in the FRT file. Otherwise, the Graphical Type (GT) of the feature code will be stored for later matching purposes. The program will also cross-check the colours specified in the parameter file and those found in the FRT file. In case of any difference, a warning message (COLNOTMAT) will be displayed (which may be suppressed with the /QUIET qualifier). Users should note that colours explicitly specified in the parameter file will have a higher priority over those in the FRT file for a given feature code.

The next stage is the construction of the HEADER section of the output DXF file. The numerical values of the 'RA' entry in the IFF file are used for the translation into the DXF map extents (\$EXTMIN and \$EXTMAX). I2DXF does not support the colour or linetype by layer formats in I2DXF and thus the \$CECOLOR or \$CELTYPE entries in the DXF HEADER section will be filled with standard defaults (\$CECOLOR = 0 (BYBLOCK), \$CELTYPE = CONTINUOUS and \$TEXTSTYLE = STANDARD). This means that all entities in the output DXF file will have a colour index (62 group) and line style (6 group) attached to them. The colour index entry in DXF LAYER table will be undefined.

If the /HEADER qualifier is given, an area feature in layer 0 is expected (with FSN 1, FC 0), containing type 6 AC's with DXF HEADER variables and their values in the text fields. The program will search for this feature, ignoring any others preceding it, in order to fill the HEADER section, so ideally this feature should be the first in the IFF file. An error message will result if this feature could not be found, or there was an error reading it. A list of the supported variables and their format is given in the DXF2I section of this manual. These variables will override the default ones mentioned above.

The HEADER section is followed by a TABLES section. I2DXF makes use of the data read from the parameter file and outputs relevant data into the LTYPE, STYLE and LAYER tables. These tables define the DXF linetype, text style and layer characteristics.

All line types referenced in the parameter file are output to the LTYPE table. Each one is compared to the default line types (CONTINUOUS, DASHED etc.) defined in the 'DATA PREPARATION' section, and the relevant fields filled. Any not found in this default setup will be output as if for 'CONTINUOUS'.

All text styles referenced in the parameter file are output to the STYLE table. Each one is compared to the default text styles (STANDARD, ROMANC or ROMANT) defined in the 'DATA PREPARATION' section, and the relevant fields filled. Any not found in this default setup will be output as if for 'STANDARD'.

All layer entries in the parameter file will create a LAYER table entry. The layer colour (62 group) will be set to the original IFF layer number.

The next section is the BLOCKS section which defines the symbol definition. Each INSERT entry in the parameter file references a BLOCK entry, with the parameter file colour being used for the 62 group entry. The entities constituting the BLOCK are omitted, and only the symbol name, colour and default line type CONTINUOUS are used.

The last section in a DXF file is the ENTITIES section which contains the graphical data of the drawing. The graphical entities supported by I2DXF are LINE, POINT, CIRCLE, ARC, TRACE, SOLID, TEXT, INSERT and POLYLINE. The program accesses the IFF file, sequentially extracting each feature code (FC) and its resident layer name, and from its FRT entry individual feature characteristics can be extracted and output to the DXF file entity referenced in the parameter file for the FC.

The parameter file layer entries are used to map the IFF layer number to DXF layer name. If a particular IFF layer is missing from the parameter file, the program searches for the next EO entry, omitting all features in this layer. In this way undesired layers may be skipped from the translation. An error message will result if the end of layer EO entry can not be found.

Text heights (TH entries) are by default assumed to be in units of mm./100. However, they can be treated as point sizes if the /POINTSIZ qualifier is given. Both value types are converted to mm. in the DXF file. If the TH entry is absent, or its value zero, the FRT value (mm.) is used, if present. If not, the text height will be omitted.

The z coordinates in any ZS (3-d) entry is output to the group 30 field complementing the groups 10 and 20 for x and y. Any height in type 2 or 3 AC entry is output to a group 38 elevation field for the particular entity.

Users are strongly advised to refer to the 'DATA PREPARATION' section about the mapping restrictions from IFF features to DXF entities.

Informational messages about the conversion will be displayed at various stages of the process provided the /LOG qualifier is included in the user command line.

---

## EXAMPLES

\$ I2DXF CONTOURS CONTOURS2/FRT=DXF/PARAMETER=HERE:TEST

ELAPSED: 0 00:00:52.68 CPU: 0:00:27.82 BUFIO: 45 DIRIO: 472 FAULTS: 285  
\$

This example demonstrates the use of I2DXF without the /LOG qualifier.  
The input IFF file LSL\$IF:CONTOURS.IFF has been successfully processed  
and a DXF format disk file LSL\$DXF:CONTOURS2.DXF has been produced.

\$ I2DXF

\_Input IFF file : **ROADS/LOG/FRT=HERE:TRANSPORT**

\_Output DXF file: **ROADS2/PARAMETER=HERE:TEST.DXF2I\_PAR**

%I2DXF-I-OPENFRT, FRT file HERE:TRANSPORT.FRT opened for read

%LSLLIB-I-IFFOPENED, LSL\$DATAROOT:[LSL.IFF]ROADS.IFF;1 opened for read

%I2DXF-I-OPENDXF, DXF file LSL\$DXF:ROADS2.DXF;0 opened for write

%DXFPAR-I-OPNPAR, PARAMETER file: HERE:TEST.DXF2I\_PAR opened for read

%DXFPAR-I-LAYERFND, LAYERs found: 4

%DXFPAR-I-ENTITYFND, ENTITYs found: 12

%DXFPAR-I-BLOCKFND, BLOCK(symbol)s found: 1

%DXFPAR-I-LTYPEFND, LTYPEs found: 3

%DXFPAR-I-STYLEFND, STYLEs found: 2

%I2DXF-I-MAPEXTENT, Map extents: X 0.0 500.0, Y 0.0 500.0

%I2DXF-I-NEWOLFND, LAYER 0 found

%I2DXF-I-NEWOLFND, LAYER 1 found

%I2DXF-I-NEWOLFND, LAYER 2 found

%I2DXF-I-NEWOLFND, LAYER 3 found

%I2DXF-I-FEAFND, number of FEATURES found: 34

%I2DXF-I-PROCSUCC, number of features SUCCESSFULLY processed: 34

ELAPSED: 0 00:01:16.84 CPU: 0:00:28.06 BUFIO: 73 DIRIO: 495 FAULTS: 276  
\$

This example demonstrates the use of I2DXF with /LOG qualifier to  
enable more information about the conversion to be displayed. It also  
demonstrates the use of a pre-defined logical name 'HERE:' to  
re-direct input/output. Users may note that we are using the same  
PARAMETER file for both direction of conversion.

-----  
**MESSAGES (SUCCESS)**

These messages are used to indicate that the program has succeeded in performing some action, and do not require any user action.

NORMAL, Successful completion

**Explanation:** The program has completed successfully.

**User action:** None.

---

**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.

ATFSNFC, at FSN 'number', FC 'number'

**Explanation:** This message is used as a complement to warning and error messages. It informs the user of the current FSN, and FC positions.

**User action:** None.

ATPOSITION, at ('real', 'real')

**Explanation:** This message is used as a complement to warning and error messages. It informs the user of the current position.

**User action:** None.

FEAFND, Number of FEATURES found: 'number'

**Explanation:** This message informs the user of the total number of features found in the input IFF file. This message will only be displayed when the /LOG qualifier is included.

**User action:** None.

MAPEXTENT, Map extents: X 'min' 'max', Y 'min' 'max'.

**Explanation:** This message informs the user of the map extents which will be stored in the HEADER section of the DXF file. The data is read from the 'RA' entry of the IFF file. This message will only be displayed when the /LOG option is included.

**User action:** None.

NEWLAYFND, LAYER 'number' found

**Explanation:** This message informs the user that the given layer in the IFF file has been found. This message will only be displayed when the /LOG option is included.

**User action:** None.

OPENDXF, DXF file 'name' opened for write

**Explanation:** This message informs the user the DXF file has been opened for write.

**User action:** None.

OPENFRT, FRT file 'name' opened for read

**Explanation:** This message informs the user the FRT file has been opened for read.

**User action:** None.

PROCESUCC, Number of features SUCCESSFULLY processed: 'number'

**Explanation:** This message informs the user of the total number of features that have been successfully processed by the program. This message will only be displayed when the /LOG option is included.

**User action:** None.

RDHDRACS, Reading HEADER AC's

**Explanation:** The area feature in layer 0 is being read for the header variables in AC's.

**User action:** None.

-----  
**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.

ANGCOIN, Angles calculated for ARC coincident.

**Explanation:** The angles calculated from the data given in the IFF file are coincident with each other. This is obviously an error from the input data. If the angles are really equal, the arc should be stored as a CIRCUMCIRCLE rather than an ARC. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** Check the validity of the input data in the input IFF file.

BADCROSREF, Invalid cross-reference between IFF and DXF entry.

**Explanation:** Invalid cross-reference between current IFF feature read from the given IFF file and the DXF ENTITY given in the parameter file. Users should note that the mapping between IFF features and DXF ENTITY are restricted by their graphical nature. Users should refer to the 'DATA PREPARATION' section of the CONVERT user's guide for further details and the mapping restrictions. The program will continue ignoring the current IFF feature. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** Cross-check the IFF feature code entry with its DXF ENTITY type.

BADPTSLIN, Invalid number of points for DXF linear feature.

**Explanation:** Number of points for a feature read from the IFF file is invalid for the definition of its DXF counterparts. Users should note that the number of points allowed for the definition of a feature is defined by its type. For example, a DXF LINE entity can only accommodate two data points. The program will continue ignoring the current IFF feature. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** Cross-check the validity of data in both the IFF and PARAMETER files.

DEFTEXHGT, Default text height 'real' mm. used for text entity

**Explanation:** Default DXF text height is used for the current entity because there was no valid text height entry found for the IFF file feature, the parameter file or the FRT file. This message may be turned off by use of the /QUIET qualifier if there are too many occurrences of this situation.

**User action:** None.



LINEUSED, DXF LINE entity type used instead of POLYLINE.

**Explanation:** DXF LINE entity type is used for the current feature instead of POLYLINE. POLYLINE is the entity type currently specified by the user in the parameter file. LINE entity type is used here because the number of data points found within the current feature is two. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** None.

MDDEFAULT, Default values used for the origin offset

**Explanation:** This message informs the user that the program was only able to find a 'MD' type 1 map descriptor entry in the input IFF file when the /ABSOLUTE qualifier was given, rather than a type2 containing an origin offset. A default origin offset of (0,0) will be used.

**User action:** This may be due to the corruption of data in the IFF file, so check the IFF file.

POLYUSED, DXF POLYLINE entity type used instead of LINE.

**Explanation:** DXF POLYLINE entity type is used for the current feature instead of LINE. LINE is the entity type currently specified by the user in the parameter file. POLYLINE entity type is used here because the number of data points found within the current feature is more than two. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** None.

PROCFAIL, Number of features FAILED to process: 'number'

**Explanation:** This message informs the user of the total number of features that are failed to process by the program. Any errors encountered during the processing would have been indicated in preceding messages.

**User action:** According to previous messages.

SINGPNTDEL, Single point deleted

**Explanation:** A single point has been deleted from a POLYLINE feature.

**User action:** Check the geometry. A common reason is the potential creation of single point line features for the closing invisible moves for nested polygons.

SINGPTSTR, Single point IFF line feature not translated.

**Explanation:** Single point IFF line feature in the given IFF file cannot be translated into its DXF LINE counterparts. Users should note that DXF LINE entity requires exactly two data points to be defined. The program will continue ignoring the current IFF feature. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** Ensure that the IFF file does not contain one-point lines.

UNAANGINS, Unable to calculate angle for INSERT entity.

**Explanation:** Unable to calculate the rotation angle for DXF INSERT entity from IFF SYMBOL feature. This might be caused by two coincident data point entries in the input IFF file. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** Check the validity of the input data in the input IFF file.

UNAARCRAD, Unable to calculate radius for ARC entity.

**Explanation:** The three points given in the input data set cannot form an circular arc. This is obviously an error from the input data. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** Check the validity of the input data in the input IFF file.

UNABCALAR, Unable to calculate angles for ARC entity.

**Explanation:** This message warns the user that the angles for the definition of an ARC in DXF format could not be calculated. This may be caused by invalid input data points. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** Check the validity of the input data in the input IFF file.

UNABCALCE, Unable to calculate centre for CIRCLE entity.

**Explanation:** This message indicates that the program cannot calculate the radius and centre of a CIRCLE entry from the input IFF data. This is obviously an error from the input data.

**User action:** Check the validity of the input data in the input IFF file.

UNABMATFC, Unable to match FC 'number' (FSN 'number') from parameter file.

**Explanation:** Unable to match the feature code read in the IFF file with those presented in the parameter file. This may be caused by the incorrect cross-referencing of parameter file or the corruption of the data itself.

**User action:** Cross-check the IFF file with its corresponding parameter file.

UNDEFLAY, Undefined IFF layer 'number' found

**Explanation:** Undefined IFF layer number found in input IFF file. User must note that all IFF layers within an input IFF file must be defined and mapped in the parameter file. The program will not be able to process this layer, and any features in it will be ignored.

**User action:** Re-define the LAYER table entries in the parameter file.

UNDEFPOINTSIZE, Undefined point size 'integer' in parameter file.

**Explanation:** The DXF text size did not have a usable point size match in the TEXT SIZE section of the parameter file.

**User action:** Check the parameter file.

-----  
**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.

BADCMD, Invalid entry found in User Command Line

**Explanation:** Invalid entry or entries are found in the User Command Line.

**User action:** Re-enter the command line.

BADCOORD, Bad IFF coordinates.

**Explanation:** This message informs the user that an invalid coordinate pair has been read in from the input IFF file. It will be raised when there is an insufficiency or redundancy of the coordinate pairs in a specified input IFF feature. This could be caused by the data corruption of the input file. This message will be followed by a 'ATFSNFC' message specifying its FSN location.

**User action:** Check the input IFF file against data corruption.

BADDXF, Unable to extract DXF filename

**Explanation:** This message informs the user that the program is unable to extract the DXF filename from the input command line.

**User action:** Re-enter the command line.

BADENDOFLAY, Unable to find end of layer 'number'

**Explanation:** There was an error searching for the next EO entry signifying the end of the current layer. There was no entry for the layer in the parameter file, and the layer should have been skipped over. useraction

**User action:**

BADFRT, Unable to extract FRT filename

**Explanation:** This message informs the user that the program is unable to extract the FRT filename from the input command line.

**User action:** Re-enter the command line.

BADHDRFTR, Unable to find header feature

**Explanation:** The /HEADER qualifier was given to read the header variables from AC type 6 entries, but the area feature (FSN 1, FC 0 in layer 0) containing the AC's could not be found. It should be the first feature in the IFF file, as any features preceding it will be ignored in the search if /HEADER is given.

**User action:** Ensure that this feature is present, or omit the /HEADER qualifier.

BADIFF, Unable to extract IFF filename

**Explanation:** This message informs the user that the program is unable to extract the IFF filename from the input command line.

**User action:** Re-enter the command line.

BADNUMDP, Unacceptable number of decimal places: %N

**Explanation:** This message informs the user that there was an unacceptable number of decimal places for real coordinate output values given with the /DECIMALPLACES qualifier. The supplied number should be between 1 and 9.

**User action:** Re-enter the command line.

BADPAR, Unable to extract PARAMETER filename

**Explanation:** This message informs the user that the program is unable to extract the parameter filename from the input command line.

**User action:** Re-enter the command line.

BADRA, Unable to read 'RA' entry from IFF file

**Explanation:** This message informs the user that the program was unable to read the 'RA' range entry from the input IFF file. This may be due to the corruption of data in the IFF file.

**User action:** Check the input IFF file against data corruption.

MDABSENT, Unable to find 'MD' entry in IFF file

**Explanation:** This message informs the user that the program was unable to find the 'MD' map descriptor entry from the input IFF file when the /ABSOLUTE qualifier was given. This may be due to the corruption of data in the IFF file.

**User action:** Check the input IFF file against data corruption.

UNABOPDXF, Unable to open DXF file for output

**Explanation:** This message informs the user that the program is unable to open the DXF file with the supplied filename for output.

**User action:** Check if the system resource is available.

---

**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

### DXF2I UTILITY

---

## UTILITY DXF2I

---

### FUNCTION

DXF2I is an Auto-CAD DXF ASCII format to IFF converter. It produces a standard IFF file with output directed to disc. For a particular DXF file, it may instead produce a basic parameter file to be used in future runs of DXF2I on this and similar DXF files.

---

### FORMAT

\$ DXF2I input-file-spec [output-file-spec]

Command qualifiers	Defaults
/[NO]ABSOLUTE	/NOABSOLUTE
/FRT=file-spec	No defaults
/[NO]HEADER	/NOHEADER
/[NO]LOG	/NOLOG
/PARAMETER=file-spec	No defaults
/[NO]POINTSIZE	/NOPOINTSIZE
/[NO]QUIET	/NOQUIET
/[NO]SCAN=file-spec	No defaults

---

### PROMPTS

Input DXF filename : input-file-spec  
Output IFF filename : output-file-spec/qualifiers

---

### PARAMETERS

input-file-spec

- specifies the DXF file which is to be converted into IFF format. Any part of the file name which is not specified will be taken from the default specification 'LSL\$DXF:DXF.DXF'.

output-file-spec

- specifies the IFF file which is to be created. Any part of the file name which is not explicitly given will be taken from the default specification 'LSL\$IF:IFF.IFF'.

If the /SCAN qualifier is used to produce a default parameter file, with no IFF output, the output IFF filespec parameter is not required. (In this case the /PARAMETER qualifier is also forbidden).



Note that both logical names, 'LSL\$IF:' and 'LSL\$DXF:' must be defined, or overridden.

---

## COMMAND QUALIFIERS

### /[NO]ABSOLUTE

- enables the output of DXF coordinates as local values in the IFF file with respect to the origin contained in the \$EXTMIN header variable. The default action is to output the DXF coordinates as local IFF coordinates.

### /FRT=file-spec

- specifies an FRT file containing definitions of feature codes to be read in. These definitions will then be used to identify the features read in the IFF file. The default is LSL\$FRT:DXF.FRT. This qualifier is mandatory.

### /[NO]HEADER

- enables the creation of an area feature in layer 0 from the DXF file header variables \$EXTMIN/\$EXTMAX (also defining the IFF range) to which certain DXF header variables are output to text fields of AC entries.

### /[NO]LOG

- enables a range of informational messages concerning the progress of the data translation to be displayed at the terminal.

### /PARAMETER=file-spec

- specifies the parameter file which defines the transformation of entity types, line type/text style/block name and colour in DXF to feature codes in the IFF file. It also defines the transformation of layer names in DXF to layer numbers in IFF. The parameter file-spec is parsed from the default LSL\$LOOKUP:DXF.PAR. Any part of the file name which is not specified by the /PARAMETER qualifier will be taken from the above default specification. This qualifier is forbidden with the /SCAN qualifier.

### /[NO]POINTSIZ

- specifies that the DXF text heights (in mm.) are converted to point sizes for the TH (text height) entries, instead of the usual mm./100 for the default of /NOPPOINTSIZ.

/[NO]QUIET

- enables the suppression of a range of warning messages output during the data translation, eg. RANGEUPDATE warning that a coordinate exceeded the header range, MISENTCOL warning about the absence of a colour entry in an entity, COLNOTMAT warning that the colour given in the parameter file does not match that of the specified FC in the FRT file, and DEFLINTYP and DEFTEXSTY indicating that a default line type (from \$CELTYPE or 'CONTINUOUS'), text style (from \$TEXTSTYLE or 'STANDARD') or layer name (from \$CLAYER) is being used, and should be present in the parameter file, or a default text height (from \$TEXTSIZE or 5 mm.) will be used.

/SCAN=file-spec

- specifies an output parameter file to be created from the DXF file, with default values for the IFF settings. This scan file can then be used as the parameter file for further runs of DXF2I with the specified FRT file. The scan file-spec is parsed from the default LSL\$LOOKUP:DXF.SCAN. Any part of the file name which is not specified by the /SCAN qualifier will be taken from the above default specification. This qualifier is forbidden with the /PARAMETER qualifier, or with an output IFF file parameter on the command line.

---

## DESCRIPTION

After the command line has been decoded, relevant files will be opened. The mode of operation of DXF2I depends mainly on the presence of the /SCAN qualifier, ie. whether to create an output IFF file using an existing parameter file, or to create a default parameter (or scan) file from the contents of the DXF file. The /SCAN mode of operation is described in detail later. The following description relates to the normal operation of creating an IFF file.

The program first scans through the parameter file to extract the definitions of the lookup mapping between the two formats. Lookup mapping of IFF/DXF layers and that of IFF features/DXF entities will then be defined and stored. If a /LOG qualifier is included, informational messages will be displayed to inform the user of the number of layers, features/entities, symbols/blocks, line types and text styles found in the parameter file. Any illegal command entry will cause a warning message to be issued. Other errors detected at this stage will cause the program to abort.

After the mapping between the two formats is successfully defined, the program then accesses the Feature Representation Table (FRT) file to check for the validity of Feature Code (FC) and Colour entries in the ENTITY lookup table. Again, the program will warn if any of the feature codes specified in the parameter file cannot be found in the FRT file. Otherwise, the Graphical Type (GT) of the feature code will be stored for later matching purposes. The program will also cross check the colours specified in the parameter file and those found in the FRT file. In case of any difference, a warning message will be displayed. Users should note that colours explicitly specified in the parameter file will have a higher priority over those in the FRT file

for a given feature code.

The next stage of the conversion is the scanning of the HEADER section in the DXF file. The entries \$EXTMIN and \$EXTMAX are used to set up the 'RA' range and 'CP' entries of the IFF file. The range values will be updated by the coordinate values from the ENTITIES or BLOCKS sections as they are encountered in the DXF file (with a warning message), and the 'RA' entry updated at the end of the program run. The \$CECOLOR entry (if there is one) will be read for any colour by layer flag, and is stored for later use (see below), and the @CELTYPE entry for any linetype by layer flag. Any other default settings are stored.

If the DXF file for any reason does not contain a HEADER section, default header IFF entries will be written and a suitable warning message given. This means that any parameter file will be meaningless for a DXF file that was supposed to be colour-by-layer, for example, as the layers and flags will be missing.

By default, a type 2 map descriptor entry (MD2) is created with all fields unset, with a map scale MD2SCL field set to 1.0 and the MD2LOC local origin set to (0,0).

If the /HEADER qualifier was given, a bounding area feature is created in layer 0 with FC 0, FSN 1 from the DXF header extents, and certain header variables are output to AC entries for this feature. The header variables transferred to AC's (type 6) at present are:

\$CECOLOR	Entity colour number: 0 = BYBLOCK, 256 = BYLAYER
\$CELTYPE	Entity linetype name or BYBLOCK or BYLAYER
\$CLAYER	Current layer name
\$LUNITS	Units format for coordinates and distances
\$LUPREC	Units precision for coordinates and distances
\$PDMODE	Point display mode
\$PDSIZE	Point display size
\$TDCREATE	Date/time of drawing creation
\$TDINDWG	Cumulative editing time for this drawing
\$TDUPDATE	Date/time of last drawing update
\$TDUSRTIMER	User elapsed timer
\$TEXTSIZE	Default text height
\$TEXTSTYLE	Current text style name
\$UCSNAME	Name of current UCS
\$UCSORG	Origin of current UCS (in WCS)
\$UCSXDIR	Direction of current UCS X axis (in World coords)
\$UCSYDIR	Direction of current UCS Y axis (in World coords)

The text fields of the AC entries will be in the following format:

```
'$CECOLOR 256'  
'$TEXTSIZE 3.0'  
'$TEXTSTYLE STANDARD'  
'$UCSXDIR 1.0 0.0 0.0'
```

If it is desired to enter different HEADER values into DXF with I2DXF, this format must be observed, with spaces between the variable name and its value or values.

The next section to be scanned is the TABLES section. At present, only three tables, LTYPE, LAYER and STYLE, are used. Data from the parameter file is used to check if there are any undefined LTYPE, LAYER or STYLE entries, with warnings given if there were any absences. If the entities are coloured or line-typed by layer, the colour index (62 group) and line type (6 group) of the LAYER table will be stored. Any error encountered at this stage will cause the program to abort.

The next section to be scanned is the BLOCKS section. Each BLOCK has a unique name and is referenced by DXF INSERT entities. These BLOCK names should be referenced to symbol FC's in the parameter file. The program will cross check these BLOCK names found against those entries in the parameter file, with warnings given if there were any absences.

The BLOCKS may be composed of entities, building up a complex symbol, referenced to their local coordinates, and would be output at the defining location and scale of the INSERT entity later. These BLOCKS may also be composed of other BLOCKS in a hierarchy of symbol definitions. Because there is at present no mechanism in DXF2I of storing these complex arrangements of entities in a symbol, the BLOCKS entities may be output to the IFF file as they are read here as if they were normal entities. This is achieved by specifying an EXPAND entry in the parameter file for the BLOCK (symbol) name, together with x,y,z offsets and scalings used to convert the internal symbol coordinates to world coordinates. These would be the INSERTion point coordinates for a single level of BLOCK nesting. For a multiply nested hierarchy of blocks, these offsets and scalings should represent the combination of the individual INSERT positions.

If the offsets or scalings in the DXF file are not known, put an EXPAND entry in the parameter file with default offsets of 0 and default scalings of 1, together with an INSERT entry for all known BLOCKS in the file, and wait for the warning messages DIFFEXPOFF, DIFFEXPZOFF, DIFFEXPSCALE or DIFFEXPZSCALE which will give out any offsets and scalings that are different from the defaults.

The whole symbol may still be represented by an IFF symbol with an INSERT symbol\_name in the parameter file to derive the FC in the normal way.

For complex hierarchies of BLOCKS, several symbols may be output to the IFF file, representing different levels of the hierarchy at each INSERT point, and may it may be better to leave out the higher levels from the parameter file which will only generate the message UNDEFBLK, without processing the symbol, relying on the lower level BLOCK entities being output as features through the EXPAND mechanism. This is illustrated in the example parameter file below.

It is possible for entities to be coloured BYBLOCK. This is not supported by DXF2I, and is overridden by any colouring by layer. It is also possible to set the line type BYBLOCK, and in this case the

line type 'BYBLOCK' must be present in the parameter file for a FC match to be found. Here again the layer colour (if any) will override the colour entry of '0' (equivalent to BYBLOCK).

The ENTITIES section contains the graphical data of the drawing. The graphical entities supported by DXF2I are LINE, POINT, CIRCLE, ARC, TRACE, SOLID, TEXT, INSERT and POLYLINE. The program then reads the DXF input file sequentially to extract each entity's type, name, colour and, according to the entity types, their other individual characteristics. If the entities are coloured by LAYER, the colour index value of the entity's resident layer will override its individual colour index, and the layer colour must be present in the parameter file for the particular entity type/line type combination for a IFF feature code to be found.

If the group 62 entity colour field is absent without colouring by layer, the feature code will be derived from the parameter file using the first occurrence of entity type and line style, so that the coordinate data may still be transferred.

Text heights in mm. in the DXF file are converted to the default units of mm./100 for TH entries, unless the /POINTSIZE qualifier is given, when they will be converted into point sizes.

If present, the group 30 z coordinate fields will be converted into ZS entries. Any group 38 elevation fields will be converted to type 3 AC entries.

Users are strongly advised to refer to 'DATA PREPARATION' section about the mapping restrictions from DXF entities to IFF features.

Informational messages about the conversion will be displayed at various stages of the process provided the /LOG qualifier is included in the user command line. Some warning messages may be turned off with the /QUIET qualifier.

In the /SCAN mode of operation of DXF2I, a default parameter file is to be created from the DXF file, and no output IFF file created. /SCAN is intended to be of use in cases where a DXF file has been received without a description of the entities or layers within it. It allows the operator to produce a useable template parameter file.

The messages output are essentially the same as for a normal run of DXF2I, mainly to inform on the validity of the input DXF file, without the output of any concerning the mismatch of entities to parameter file. Apart from the /SCAN = scan-file, the command line is the same as for the normal run of DXF2I except that the output IFF file is not required, and the /PARAMETER file qualifier is forbidden. The FRT file must still be provided for the scan to find a suitable feature code to be output to the SCAN file.

After conversion using this template, the resultant IFF file can be viewed using LITES2. The template file can then be edited to choose more appropriate feature codes, and the DXF data converted again to obtain the required IFF file. When /SCAN has been used on an unknown DXF file, it is important to consider the scale of the data, as this

affects the representation. Program IINFO in the IMP package will report the range of coordinates encountered, and programs IPATCH or ITRANS/DESCRIPTOR in the IMP package can be used to set the scale in the IFF MD entry, which will be used by display programs such as LITES2. Command SHOW SCALE in LITES2 is useful to check that the scale is set sensibly (eg. that "Extent of working area" is a valid paper size).

First the DXF layer names are read from the TABLES section LAYERs or BLOCKs and output to the SCAN file, with the IFF layer number increased by one for each output. Then the ENTITIES section is read. For each combination of entity type, line type/text style/block name and colour, a new line is output to the SCAN file with a suitable feature code obtained from the FRT file (the first feature code in the file for the relevant graphical type contained in the FRT file). If there is no suitable FC available, a value of -1 will be used to flag an unset FC value, which must be edited before the scan file can be used as the parameter file for subsequent normal runs of DXF2I on this DXF file.

Each entity line and layer line output to the SCAN file is stored for later comparison to ensure that they are all unique combinations. A message TOOMNYENTS or TOOMNYLAYS will be output if either array limit is about to be exceeded, after which no more comparisons are possible and the entity lines are output regardless of their uniqueness.

One important message to be noted is ERRSCANLAY, where a layer is referenced in the entities section that was not present in the LAYER TABLEs or BLOCKs sections. As the program is now writing the entities to the scan file, it is too late to output the missing layers, and the user must edit the scan file to replace these missing layers before using it as the parameter file for the DXF file.

---

**EXAMPLE PARAMETER FILE**

This example parameter illustrates the use of INSERT entities for a complex hierarchy of BLOCKS, avoiding the output of a symbol at every INSERT, relying only on the output of the lower level BLOCK entities to define the features.

```

!
!
!
!Parameter lookup file to transform a filespec.dxf to a filespec.iff
!

! NOTE that COLOUR-BY-LAYER overrides COLOUR-BY-BLOCK, and the layer colours
! as defined in the LAYER TABLES must be used for the ENTITIES to FC section
! of this parameter file
!
! Some entities use the BYBLOCK linetype which must also be present with the
! respective layer colour
!
!
!
!
!      DXFLAYERNAME          IFFLAYERID
!
LAYER      0                      0
LAYER      8                      8
LAYER      FRAME                  1
LAYER      GRID                   2
LAYER      SPOT-LEVEL-MARKERS     3
LAYER      GROUND-MODEL           4
LAYER      SPOT-LEVELS            5
LAYER      PROMINENTCONTOUR-LEVELS 6
LAYER      CONTOUR-LEVELS         7
!
!ENTITYLOOKUPTABLE
!
!      ENTITYTYPE      NAME          FRT      COLOUR  IFFFC
!
ENTITY     POLYLINE     CONTINUOUS    1         1
ENTITY     LINE          CONTINUOUS    6         2
ENTITY     POINT         2           5
ENTITY     TEXT          STANDARD     4        11
!
ENTITY     LINE          CONTINUOUS    1         1
!
!
! for CONTOURS.DXF
!
! prominent contour levels
ENTITY     TEXT          STANDARD     7        11
!
! contour levels/grid
ENTITY     TEXT          STANDARD     8        40
!

```

```

! contours/frame
ENTITY LINE BYBLOCK 7 1
ENTITY LINE CONTINUOUS 7 1
ENTITY POLYLINE BYBLOCK 7 1
ENTITY POLYLINE CONTINUOUS 7 1
!
! grid
ENTITY POLYLINE CONTINUOUS 8 2
ENTITY LINE CONTINUOUS 8 2
!
! spot level points
ENTITY POINT 7 5
!
! spot levels
ENTITY TEXT STANDARD 1 50
!
! spot level markers - Use symbol, do not EXPAND
ENTITY INSERT CENTRED 9 8
!ENTITY EXPAND CENTRED 0 0 0 1 1 1
!
!
! do not expand CONTOURS ( = PROMINENT,MAJOR and MINOR-CONTOURS INSERTS)
!ENTITY INSERT CONTOURS 255 20
!ENTITY EXPAND CONTOURS -6004 -1011 0 20 20 1
!
! expand SPOT-LEVEL-POINTS to POINT entities
!ENTITY INSERT SPOT-LEVEL-POINTS 255 31
ENTITY EXPAND SPOT-LEVEL-POINTS -6004 -1011 0 20 20 1
!
!
! give the BLOCK names to be referenced by later INSERT entities
! the INSERT entities are translated by DXF2I as symbols, and
! translating all these would leave a lot of symbols at (0,0)
!
! EXPAND all contours
!
!ENTITY INSERT 25M-CONTOURS 7 33
ENTITY EXPAND 25M-CONTOURS -6004 -1011 0 20 20 1

!ENTITY INSERT 50M-CONTOURS 7 5
ENTITY EXPAND 50M-CONTOURS -6004 -1011 0 20 20 1

!ENTITY INSERT 200M-CONTOURS 7 5
ENTITY EXPAND 200M-CONTOURS -6004 -1011 0 20 20 1

!ENTITY INSERT 225M-CONTOURS 7 5
ENTITY EXPAND 225M-CONTOURS -6004 -1011 0 20 20 1

!ENTITY INSERT PROMINENT-CONTOURS 7 34
!ENTITY EXPAND PROMINENT-CONTOURS -6004 -1011 0 20 20 1
!
!
!ENTITY INSERT 30M-CONTOURS 7 33
ENTITY EXPAND 30M-CONTOURS -6004 -1011 0 20 20 1

!ENTITY INSERT 40M-CONTOURS 7 5

```



ENTITY	EXPAND	40M-CONTOURS	-6004	-1011	0	20	20	1
!ENTITY	INSERT	220M-CONTOURS	7	5				
ENTITY	EXPAND	220M-CONTOURS	-6004	-1011	0	20	20	1
!ENTITY	INSERT	230M-CONTOURS	7	5				
ENTITY	EXPAND	230M-CONTOURS	-6004	-1011	0	20	20	1
!ENTITY	INSERT	MAJOR-CONTOURS	7	34				
!ENTITY	EXPAND	MAJOR-CONTOURS	-6004	-1011	0	20	20	1
!								
!								
!ENTITY	INSERT	35M-CONTOURS	7	33				
ENTITY	EXPAND	35M-CONTOURS	-6004	-1011	0	20	20	1
!ENTITY	INSERT	45M-CONTOURS	7	5				
ENTITY	EXPAND	45M-CONTOURS	-6004	-1011	0	20	20	1
!ENTITY	INSERT	215M-CONTOURS	7	5				
ENTITY	EXPAND	215M-CONTOURS	-6004	-1011	0	20	20	1
!ENTITY	INSERT	235M-CONTOURS	7	5				
ENTITY	EXPAND	235M-CONTOURS	-6004	-1011	0	20	20	1
!ENTITY	INSERT	MINOR-CONTOURS	7	34				
!ENTITY	EXPAND	MINOR-CONTOURS	-6004	-1011	0	20	20	1

!  
!  
! The BLOCK structures in CONTOURS.DXF  
!  
! Final ENTITIES contain INSERTS for 'CONTOURS' BLOCK  
! and 'SPOT-LEVEL-POINTS' BLOCK  
! and frame/grid ENTITIES  
!  
!  
!

```

!
!           CONTOURS
!           |
!           |
!  MINOR-CONTOURS  MAJOR-CONTOURS  PROMININENT-CONTOURS
!           |           |           |
!           |           |           |
!    35M-CONTOURS    30M-CONTOURS    25M-CONTOURS
!
!
!

```

! Contours consist of TEXT in their respective 'CONTOUR-LEVEL' layer  
! and POLYLINES in layer '0'

! CENTRED05 = symbol at INSERT

!  
!  
!

---

## EXAMPLES

```
$ DXF2I CONTOURS CONTOURS2/FRT=TERRAIN/PARAMETER=TEST.DXF2IPAR
  ELAPSED: 00:01:32.06  CPU: 0:00:53.65  BUFIO: 29  DIRIO: 126  FAULTS: 108
$
```

This example demonstrates the use of DXF2I. DXF file LSL\$DXF:CONTOURS.DXF has been successfully processed and an IFF format disk file LSL\$IF:CONTOURS2.IFF has been produced.

```
$ DXF2I
 _Input DXF file : HERE:CULTURE/LOG/FRT=HERE:CITY
 _Output IFF file: HERE:CULTURE2/PARAMETER=HERE:TRANSFORM.DXF_PAR
%DXF2I-I-FRTOPNREAD, FRT file: HERE:CITY.FRT opened for read
%DXF2I-I-IFFOPNWRIT, IFF file: HERE:CULTURE2.IFF opened for write
%DXF2I-I-DXFOPNREAD, DXF file: HERE:CULTURE.DXF opened for read
%DXFPAR-I-OPNPAR, PARAMETER file: HERE:TRANSFORM.DXF_PAR opened for read
%DXFPAR-I-LAYERFND, LAYERS found: 3
%DXFPAR-I-ENTITYFND, ENTITYs found: 13
%DXFPAR-I-BLOCKFND, BLOCK(symbol)s found: 2
%DXFPAR-I-LTYPEFND, LTYPEs found: 4
%DXFPAR-I-STYLEFND, STYLEs found: 3
%DXF2I-I-RDHDSC, Reading HEADER of DXF file
%DXF2I-I-MAPEXTENT, Map extents: X 100.0 2000.0, Y 10.0 3000.0
%DXF2I-I-RDTBSC, Reading TABLES of DXF file
%DXF2I-I-RDBKSC, Reading BLOCKS of DXF file
%DXF2I-I-RDENSC, Reading ENTITIES of DXF file
%DXF2I-I-FEAFND, number of ENTITIES found: 123
%DXF2I-I-PROCSUCC, number of entities SUCCESSFULLY processed: 123
ELAPSED: 00:01:32.06  CPU: 0:00:53.65  BUFIO: 29  DIRIO: 126  FAULTS: 108
$
```

This example demonstrates the use of DXF2I with /LOG qualifier to enable more information about the conversion to be displayed. It also demonstrates the use of a pre-defined logical name 'HERE:' to re-direct input/output.

-----  
**MESSAGES (SUCCESS)**

These messages are used to indicate that the program has succeeded in performing some action, and do not require any user action.

NORMAL, Successful completion

**Explanation:** The program has completed successfully.

**User action:** None.

---

**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.

COLBYLAY, DXF entities coloured by LAYER.

**Explanation:** This message indicates that the DXF entities are coloured by LAYER. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

DEFIFFHDR, Default IFF header written in absence of DXF header.

**Explanation:** This message indicates that the program is writing default IFF file header entries as the DXF file did not have a header section.  
useraction

**User action:**

DXFOPNREAD, DXF file: 'filename' opened for read.

**Explanation:** This message indicates that the DXF file is opened for read only. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

ENTFND, number of ENTITIES found: 'number'

**Explanation:** This message informs the user of the total number of DXF entities found in the input DXF file. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

FRTOPNREAD, FRT file: 'filename' opened for read.

**Explanation:** This message indicates that the FRT file is opened for read only. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

IFFOPNWRITE, IFF file: 'filename' created for write.

**Explanation:** This message indicates that the IFF file is opened for write. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

LTYPEBYLAY, DXF entity line types by LAYER.

**Explanation:** This message indicates that the DXF entity line types are set by LAYER. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

MAPEXTENT, Map extents: X 'min' 'max', Y 'min' 'max'

**Explanation:** This message indicates the map extents in the input DXF file. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

PROCESUCC, number of entities SUCCESSFULLY processed: 'number'

**Explanation:** This message informs the user of the total number of entities that have been successfully processed by the program. This message will only be displayed if the /LOG option is specified.

**User action:** None.

RDBKSC, Reading BLOCKS section of DXF file.

**Explanation:** This message indicates that the program is reading DXF file BLOCKS section. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

RDENSC, Reading ENTITIES section of DXF file.

**Explanation:** This message indicates that the program is reading DXF file ENTITIES section. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

RDHDSC, Reading HEADER section of DXF file.

**Explanation:** This message indicates that the program is reading DXF file HEADER section. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

RDTBSC, Reading TABLES section of DXF file.

**Explanation:** This message indicates that the program is reading DXF file TABLES section. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

SCANOPNWRITE, SCAN file: 'filename' opened for write.

**Explanation:** This message indicates that the SCAN file is opened for write only. This message will only be displayed if the /LOG qualifier is specified.

**User action:** None.

WRTHDRFTR, Writing HEADER feature to IFF file.

**Explanation:** This message indicates that the program is writing the feature (FSN 1, FC 0 in layer 0) to contain the HEADER variables in the text fields of AC's type 6. This message will only be displayed if the /LOG qualifier is specified with the /HEADER qualifier.

**User action:** None.

-----  
**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.

DEFLAYNAM, Default layer name '*string*' used for entity at line '*number*'

**Explanation:** Default DXF layer name (from \$CLAYER variable in DXF header) is used for the current entity because there was no text style '*6*' entry found for the entity in the DXF file. This default layer name must be present in any parameter file for DXF2I to find a IFF layer number. This message may be turned off by use of the /QUIET qualifier if there are too many occurrences of this situation.

**User action:** None.

DEFLINTYP, Default line type '*CONTINUOUS*' used at line '*number*'

**Explanation:** Default DXF line type '*CONTINUOUS*' is used for the current entity. This is because there is no line type entry found for the entity in the DXF file, and this default line type must be present in any parameter file for DXF2I to find a linetype/colour combination to FC pairing. This message may be turned off by use of the /QUIET qualifier if there are too many occurrences of this situation.

**User action:** None.

DEFTEXHGT, Default text height '*real*' mm. used for entity at line '*number*'

**Explanation:** Default DXF text height (either from \$TEXTSIZE variable in DXF header, or 5) is used for the current entity because there was no valid text height '*40*' entry found for the entity in the DXF file. This message may be turned off by use of the /QUIET qualifier if there are too many occurrences of this situation.

**User action:** None.

DEFTEXSTY, Default text style '*string*' used for entity at line '*number*'

**Explanation:** Default DXF text style (either from \$TEXTSTYLE variable in DXF header, or '*STANDARD*') is used for the current entity because there was no text style '*7*' entry found for the entity in the DXF file. This default text style must be present in any parameter file for DXF2I to find a textstyle/colour combination to FC pairing. This message may be turned off by use of the /QUIET qualifier if there are too many occurrences of this situation.

**User action:** None.

DIFFEXPOFF, Inconsistent offsets ('x' 'y') for symbol 'name' at line 'number'.

**Explanation:** The x,y offsets supplied with the symbol EXPAND entry in the parameter file, (already used to output the symbol from the BLOCKS section) were different from the INSERT entity values (displayed here and used for the symbol).

**User action:** Check the values in the parameter file EXPAND entry, or DXF INSERT entity.

DIFFEXPSCALE, Inconsistent scales ('xscale' 'yscale') for symbol 'name' at line 'number'.

**Explanation:** The x,y scales supplied with the symbol EXPAND entry in the parameter file, (already used to output the symbol from the BLOCKS section) were different from the INSERT entity values (displayed here and used for the symbol).

**User action:** Check the values in the parameter file EXPAND entry, or DXF INSERT entity.

DIFFEXPZOFF, Inconsistent Z offset ('z') for symbol 'name' at line 'number'.

**Explanation:** The z offset supplied with the symbol EXPAND entry in the parameter file, (already used to output the symbol from the BLOCKS section) was different from the INSERT entity value (displayed here and used for the symbol).

**User action:** Check the value in the parameter file EXPAND entry, or DXF INSERT entity.

DIFFEXPZSCALE, Inconsistent Z scale ('zscale') for symbol 'name' at line 'number'.

**Explanation:** The z scale supplied with the symbol EXPAND entry in the parameter file, (already used to output the symbol from the BLOCKS section) was different from the INSERT entity value (displayed here and used for the symbol).

**User action:** Check the value in the parameter file EXPAND entry, or DXF INSERT entity.

ERRLAYCOL, Error reading colour index entry for DXF layer 'name' at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the colour index entry for the given DXF layer at the indicated location. Users should note that if the data in the DXF file is to be coloured by LAYER, every DXF layer should have a colour index attached to it.

**User action:** Ensure that there is a valid colour index entry attached to each DXF layer if the BYLAYER flag is set in \$CECOLOUR (= 256).



ERRLAYLTYPE, Error reading line type entry for DXF layer 'name' at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the line type entry for the given DXF layer at the indicated location. Users should note that if the data in the DXF file is to be organised with line type by LAYER, every DXF layer should have a line type attached to it.

**User action:** Ensure that there is a valid line type entry attached to each DXF layer if the BYLAYER flag is set in \$CELTYPE.

ERRSCANLAY, DXF LAYER 'name' not found in DXF file, at line 'number'.

**Explanation:** The given DXF layer name encountered in the DXF file could not be found in the DXF TABLE section file, and will not have been output to the SCAN file.

**User action:** Ensure that all layer names in the DXF file are defined in the SCAN file LAYER table before using it as the PARAMETER file for the DXF FILE.

MISARCANG, Missing DXF arc angle(s) at line 'number'

**Explanation:** Missing DXF starting and/or ending angle(s) entry for the definition of an arc entity at the indicated location. The program will continue ignoring the current entity.

**User action:** Check the validity of DXF arc angle entry at the specified location.

MISCOOR, Missing x or y coordinate at line 'number'

**Explanation:** Missing coordinate in the current DXF entity entry at the indicated location. The program will continue ignoring the current DXF entity.

**User action:** Check the validity of data entries at the indicated location in the input DXF file.

MISENTBLK, Missing DXF block name at line 'number'

**Explanation:** Missing DXF block name entry for the definition of an insert entity at the indicated location. The program will continue ignoring the current entity.

**User action:** Check the validity of DXF block name entry at the specified location.

MISENTCOL, Missing colour number entry at line 'number'

**Explanation:** The expected colour number entry associated with each DXF feature was not found. Users should note that all the DXF entities must have a colour number attached to them unless they are organised by layer. The colour will be ignored in the derivation of the feature code, and only the first combination of entity type and style will be found so that the conversion may proceed. This message may be turned off by use of the /QUIET qualifier if there are too many occurrences of this situation.

**User action:** Check the validity of colour number entry in the input DXF file.

MISENTLAY, Missing layer entry at line 'number'

**Explanation:** The expected layer entry associated with each DXF feature was not found. Users should note that all the DXF entities must be referenced to a layer.

**User action:** Check the validity of LAYER entry in the input DXF file.

MISENTRAD, Missing DXF circle radius at line 'number'

**Explanation:** Missing DXF circle radius entry at the indicated location. The program will continue ignoring the current entity.

**User action:** Check the validity of DXF radius entry in the input DXF file.

MISENTTXT, Missing text value at line 'number'

**Explanation:** Missing text value in the current DXF entity entry at the indicated location. The program will continue ignoring the current DXF entity.

**User action:** Check the validity of text entry at the indicated location.

MISLINCOOR, Only one coordinate in linear feature at line 'number'

**Explanation:** In the current linear feature there is only one coordinate. There must be at least two to constitute a line.

**User action:** Check the validity of data at the specified location.

MISPOLCOOR, Only one coordinate in polyline feature at line 'number'

**Explanation:** In the current polyline feature there is only one coordinate. There must be at least two to constitute a polyline.

**User action:** Check the validity of data at the specified location.

PROCEFAIL, number of entities FAILED to process: 'number'

**Explanation:** This message informs the user of the total number of entities that are failed to process by the program. Any errors encountered during the processing would have been indicated in preceding messages.

**User action:** According to previous messages.

RANGEUPDATE, header range needed updating near line 'number'.

**Explanation:** The range extents in the DXF header were exceeded by a coordinate at the indicated line, and the IFF range modified. This message may be turned off by use of the /QUIET qualifier if there are too many occurrences of this situation.

**User action:** None.

UNDEFBLK, Undefined DXF BLOCK 'name' referenced at line 'number'.

**Explanation:** An undefined DXF block (symbol) name is referenced at the indicated location in the input DXF file. Users should note that the mapping between DXF blocks and IFF feature codes must be defined in the parameter file ENTITY INSERT table.

**User action:** Check the mapping between DXF blocks and IFF feature codes.

UNDEFLAY, Undefined DXF layer 'name' referenced at line 'number'.

**Explanation:** An undefined DXF layer name is referenced at the indicated location in the input DXF file. Users should note that the mapping between DXF and IFF layers must be defined in the parameter file LAYER table.

**User action:** Check the mapping between DXF and IFF layers.

UNDEFLIN, Undefined DXF LTYPE 'name' referenced at line 'number'.

**Explanation:** An undefined DXF line type name is referenced at the indicated location in the input DXF file. Users should note that the mapping between DXF line types and IFF feature codes must be defined in the parameter file ENTITY table.

**User action:** Check the mapping between DXF line types and IFF feature codes.

UNDEFPOINTSIZE, Undefined point size for 'real' mm. in parameter file.

**Explanation:** The DXF text size did not have a usable point size match in the TEXT SIZE section of the parameter file, and the last specified point size is used as a default.

**User action:** Check the parameter file.

UNDEFTEX, Undefined DXF STYLE 'name' referenced at line 'number'.

**Explanation:** An undefined DXF text style name is referenced at the indicated location in the input DXF file. Users should note that the mapping between DXF text styles and IFF feature codes must be defined in the parameter file ENTITY TEXT table.

**User action:** Check the mapping between DXF text styles and IFF feature codes.

UNEQNUMCO, Uneven number of x and y coords near line 'number'.

**Explanation:** The feature did not have complete pairs of coordinates therefore it is likely the whole feature may be corrupt

**User action:** Check the validity of data at the specified location.

-----  
**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.

ERRCREIFF, Error creating IFF file: 'filename'

**Explanation:** Unable to create, or error status encountered during the attempt to create the IFF file for output. The program will abort if this error occurs.

**User action:** Check if the system resource is available.

ERRDXFNAM, Error reading DXF filename from command line.

**Explanation:** Unable to extract, or error status encountered during the attempt to read the DXF filename from command line. The program will abort if this error occurs.

**User action:** Re-enter the DXF filename on command line.

ERRENTACVAL, Error reading AC height entry for DXF entity at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the AC height entry for a DXF entity at the indicated location. The program will continue, ignoring the current entity.

**User action:** Ensure that there is a valid entry at the indicated location.

ERRENTANG, Error reading angle entry for DXF entity at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the angle entry for a DXF circle entity at the indicated location. The program will continue ignoring the current entity.

**User action:** Ensure that there is a valid angle entry at the indicated location.

ERRENTCOL, Error reading colour index entry for DXF entity at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the colour index entry for a DXF entity at the indicated location. The program will continue, ignoring the current entity.

**User action:** Ensure that there is a valid colour index entry at the indicated location.

ERRENTCOOR, Error reading coordinate value for DXF entity at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the coordinate value entry for a DXF entity at the indicated location. The program will continue, ignoring the current entity.

**User action:** Ensure that there is a valid coordinate value entry at the indicated location.

ERRENTRAD, Error reading radius entry for DXF entity at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the radius entry for a DXF circle entity at the indicated location. The program will continue ignoring the current entity.

**User action:** Ensure that there is a valid radius entry at the indicated location.

ERRFRTNAM, Error reading FRT filename from command line.

**Explanation:** Unable to extract, or error status encountered during the attempt to read the FRT filename from command line. The program will abort if this error occurs.

**User action:** Re-enter the FRT filename on command line.

ERRIFFNAM, Error reading IFF filename from command line.

**Explanation:** Unable to extract, or error status encountered during the attempt to read the IFF filename from command line. The program will abort if this error occurs.

**User action:** Re-enter the IFF filename on command line.

ERRLAYNAM, Error reading DXF layer name at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the DXF layer name at the indicated location. This error may be caused by an empty entry in the relevant field.

**User action:** Ensure that there is a valid layer name entry at the indicated location.

ERRLINTYP, Error reading DXF line type name at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the DXF line type name at the indicated location. This error may be caused by an empty entry in the relevant field.

**User action:** Ensure that there is a valid line type name entry at the indicated location.

ERRMATBLK, DXF block 'name' not found in parameter file, at line 'number'.

**Explanation:** The given DXF block name encountered in the DXF file could not be found in the parameter file. Users should note that the mapping between DXF entities and IFF features must be defined in the parameter file ENTITY table. The program will continue ignoring the current entity.

**User action:** Ensure that all block names in the DXF file are defined in the parameter file ENTITY table.

ERRMATFC, Error matching DXF entities and IFF FC at line 'number'.

**Explanation:** Unable to find, or error status encountered during the attempt to match the DXF entity's characteristics, e.g. line type and colour, with IFF feature code (FC) entries in the PARAMETER file. Users should note that the mapping between DXF entities and IFF features must be defined in the parameter file ENTITY table. The program will continue ignoring the current entity.

**User action:** Check the validity of DXF entries in the DXF file at the indicated location and the DXF entities/IFF features mapping in the parameter file.

ERRMATLAY, DXF LAYER 'name' not found in parameter file, at line 'number'.

**Explanation:** The given DXF layer name encountered in the DXF file could not be found in the parameter file. Users should note that the the mapping between DXF and IFF layers must be defined in the parameter file LAYER table. The program will continue ignoring the current entity.

**User action:** Ensure that all layer names in the DXF file are defined in the parameter file LAYER table.

ERRMATLIN, DXF line type 'name' not found in parameter file, at line 'number'.

**Explanation:** The given DXF line type name encountered in the DXF file could not be found in the parameter file. Users should note that the mapping between DXF entities and IFF features must be defined in the parameter file. The program will continue ignoring the current entity.

**User action:** Ensure that all line type names in the DXF file are defined in the parameter file ENTITY table.

ERRMATSTY, DXF text style 'name' not found in parameter file, at line 'number'.

**Explanation:** The given DXF text style name encountered in the DXF file could not be found in the parameter file. Users should note that the mapping between DXF entities and IFF features must be defined in the parameter file. The program will continue ignoring the current entity.

**User action:** Ensure that all text style names in the DXF file are defined in the parameter file ENTITY table.

ERROPNDXF, Error opening DXF file: 'filename'

**Explanation:** Unable to open, or error status encountered during the attempt to open the DXF file for input. The program will abort if this error occurs.

**User action:** Check the existence of the DXF file.

ERROPNFRT, Error opening FRT file.

**Explanation:** Unable to open, or error status encountered during the attempt to open the FRT file for input. The program will abort if this error occurs.

**User action:** Check the existence of the FRT file.

ERROPNSCAN, Error opening SCAN file: 'filename'

**Explanation:** Unable to open, or error status encountered during the attempt to open the SCAN file for write. The program will abort if this error occurs.

**User action:** Check the validity of the scan file name or directory.

ERRPARNAM, Error reading PARAMETER filename from command line.

**Explanation:** Unable to extract, or error status encountered during the attempt to read the PARAMETER filename from command line. The program will abort if this error occurs.

**User action:** Re-enter the PARAMETER filename on command line.

ERRPOLFLG, Error reading DXF polyline close flag at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the DXF polyline close flag entry at the indicated location. The close flag entry is used to indicate whether the polyline is closed or not. The program will continue ignoring the current entity.

**User action:** Ensure that there is a valid polyline close flag entry at the indicated location.

ERRSCANAM, Error reading SCAN filename from command line.

**Explanation:** Unable to extract, or error status encountered during the attempt to read the SCAN filename from command line.

**User action:** Re-enter the SCAN filename on command line.



ERRSELDXF, Error selecting DXF file: 'filename'

**Explanation:** Unable to select, or error status encountered during the attempt to select the DXF file for input. The program will abort if this error occurs.

**User action:** Check the existence of the DXF file.

ERRSYMNAM, Error reading DXF block name at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the DXF block name at the indicated location. This error may be caused by a missing entry in the relevant field.

**User action:** Ensure that there is a valid block name entry at the indicated location.

ERRSYMSCA, Error reading DXF symbol scale at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the DXF symbol scale at the indicated location. The DXF symbol scale is used to define an insert entity. The program will continue ignoring the current entity.

**User action:** Ensure that there is a valid symbol scale entry at the indicated location.

ERRTEXHGT, Error reading DXF text height at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the DXF text height, or height less than or equal to 0, at the indicated location. A default text height (either from \$TEXTSIZE variable in DXF header, or 5) will be used.

**User action:** Ensure that there is a valid text height entry at the indicated location.

ERRTEXROT, Error reading DXF text rotation at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the DXF text rotation entry at the indicated location. Text rotation entry is used to define the rotation of the text. The program will continue ignoring the current entity.

**User action:** Ensure that there is a valid text rotation entry at the indicated location.

ERRTEXSTR, Error reading DXF text string at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the DXF text string at the indicated location. Text string entry is used to define the location of the text. The program will continue ignoring the current entity.

**User action:** Ensure that there is a valid text string entry at the indicated location.

ERRTEXSTY, Error reading DXF text style name at line 'number'.

**Explanation:** Unable to read, or error status encountered during the attempt to read the DXF text style name at the indicated location. This error may be caused by an empty entry in the relevant field. A default text style (either from \$TEXTSTYLE variable in DXF header, or 'STANDARD') will be used.

**User action:** Ensure that there is a valid text style entry at the indicated location.

NOPARSCAN, No /PARAMETER or /SCAN on command line.

**Explanation:** Neither the /PARAMETER nor /SCAN qualifier was given on the command file.

**User action:** Re-enter the command line with one of these qualifiers.

RDLINERR, Error reading line in DXF file at line 'number'

**Explanation:** This message indicates that the program cannot be continued due to the detection of an error when reading the DXF file.

**User action:** According to the preceding error message.

TOOMNYENTS, Too many unique entities for /SCAN - maximum allowed is 'number'

**Explanation:** This message indicates that there too many different entity entries for output with the /SCAN option. The program can no longer check for the uniqueness of any subsequent entries, and will just output them regardless.

**User action:** Break the DXF file up into smaller sections and rerun DXF2I, or consult Laser-scan for a modification to the program.

TOOMNYLAYS, Too many unique layers for /SCAN - maximum allowed is 'number'

**Explanation:** This message indicates that there too many different layer entries for output with the /SCAN option. The program can no longer check for the uniqueness of any subsequent entries, and will just output them regardless.

**User action:** Break the DXF file up into smaller sections and rerun DXF2I, or consult Laser-scan for a modification to the program.

TOOMNYPNTS, Too many points in feature at line 'number' - maximum allowed is 'number'

**Explanation:** This message indicates that the reading of a POLYLINE feature cannot be continued due to it containing too many points at the specified line.

**User action:** Break the feature up into smaller sections and rerun DXF2I.

UNACONT, Error status encountered reading DXF file, unable to continue.

**Explanation:** This message indicates that a feature or section cannot be continued due to the detection of a severe error when reading the DXF file.

**User action:** According to the preceding error message.

---

**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.