Laser-Scan Ltd.

CONVERT PACKAGE

IFFGDB Reference

Issue 1.0 - 17-June-1991

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

Document "IFFGDB REFERENCE", Category "REFERENCE"
Document Issue 1.0   S Townrow (modified 17-Jun-1991)

CONTENTS

IFFGDB - Change Record i

PREFACE 1
Intended audience 1
Structure of this document 1
Conventions used in this document 2

CHAPTER 1 GDB FORMAT DESCRIPTION

FORMAT DESCRIPTION 1-1
Introduction 1-1
The SICAD/GDB Format 1-1
The SICAD/GDB feature element structure 1-2

CHAPTER 2 GDB DATA PREPARATION

DATA PREPARATION 2-1
Comparison of SICAD/GDB and IFF 2-1
Feature Translation 2-1
Area Features 2-2

CHAPTER 3 I2GDB UTILITY

UTILITY I2GDB 3-1
FUNCTION 3-1
FORMAT 3-1
PROMPT 3-1
PARAMETERS 3-1
COMMAND QUALIFIERS 3-2
RESTRICTIONS 3-2
DESCRIPTION 3-3
Command line 3-3
PARAMETERS FILE 3-3
LAYER 3-3
SYMBOL 3-3
FC 3-4
PROGRAM ACTION 3-5
EXAMPLES 3-8
MESSAGES (SUCCESS) 3-9
MESSAGES (ERROR) 3-10
MESSAGES (OTHER) 3-12

IFFGDB - Change Record

Version 1.0 S Townrow 18-April-1991

Module IFFGDB - New section, **Area Features**, added to DATA PREPARATION section to describe how the IFF file should be prepared if common area lines are not to be duplicated when translating from IFF to GDB.

Version 1.1 S Townrow 17-June-1991

Module IFFGDB - Reorganised package documentation.

PREFACE

Intended audience

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

Structure of this document

This document is composed of 2 major sections.

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

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

MODULE - the name of the CONVERT module.

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

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

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

UTILITY - the name of the utility.

FUNCTION - a synopsis of what the utility does.

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

PROMPT - how it prompts the user.

PARAMETERS - description of expected command parameters.

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

RESTRICTIONS - a summary of restrictions on the use of

qualifiers

DESCRIPTION - the definitive description of the utility action.

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

EXAMPLES - annotated examples of utility useage.

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

Conventions used in this document

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

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

CHAPTER 1

GDB FORMAT DESCRIPTION

FORMAT DESCRIPTION

Introduction

The program I2GDB provides a translation from Laser-Scan's Internal Feature Format (IFF) to the SICAD/GDB (Geographical Data Base) ASCII disc file format. For a brief comparison of the way IFF and GDB regard data, see the section on 'Data Preparation' below.

The SICAD/GDB Format

The IFF data is output to a SICAD/GDB ASCII disc file with the format defined in this summary.

The basic unit of a GDB file is the feature element. Each element consists of a header line which must contain the element type (ETYP=xx) and element number (ENUM=n), and optionally the hierarchy level (STU=n), the overlay number (EB=n), the line style (SM=n) and line thickness or colour (ST=n):

```
ETYP=xx STU=n ENUM=n EB=n ST=n SM=n
```

This header line is followed by a number of parameter and descriptor lines, the number and type depending on the element type, and which consist of the parameter name and its value.

It is the element type which describes what kind of object it is (and roughly translates to the IFF graphical type), while the element number is a unique number assigned to each different feature component, and is the number by which multiple occurrences of the same feature element are referenced. This means that any element need only occur once in full detail within a GDB file, and any other features to which it belongs, and which belong to it, need only refer to this element number in a single line.

The hierarchy level is required for the feature element types which are themselves composed of other element types. For example, a line segment in the top hierarchy (STU=1) consists of two point elements in a lower hierarchy (STU=2).

Within any feature element parameter which requires coordinate information, a HEX representation should be used if possible in preference to the actual real values to preserve accuracy on transfer to other formats.

SICAD/GDB format comprises many feature element types, of which only a subset are used by I2GDB because of the relatively small number of distinct feature types (graphical type) used in IFF files. These are described below, with their required relevant parameter fields.

Any angles in the parameter fields are measured in degrees anti-clockwise from the horizontal.

Most measurements in the parameter fields are in absolute ground metres.

Any comment lines begin with a '*'.

The SICAD/GDB feature element structure

The SICAD/GDB feature elements that are used in the I2GDB transfer routine are listed below, together with their various required parameter fields. The use of these feature elements for each of the IFF graphical types is described in the preparation section of this manual, and in the I2GDB transfer description section.

PG Graphical Point element

The basic graphical point element, and used to define any feature vertices. This element may or may not define a symbol at the vertex depending on the relevant parameter field entry.

```
ETYP=PG STU=n ENUM=n EB=n
X ZC35FF50800000000    -1535.3145
Y Z4412048A00000000     4612.5413
PKZ Z
HGEL n
PNR 0
```

X,Y are the position of the point in either HEX or real value format, or both.

PKZ defines the type of symbol to use at the point/vertex if any; PKZ Z signifies that no symbol is to be used, and is therefore used in any PG elements describing the lower levels of IFF linear features except for the symbol string. Other common symbols are

S	small dot
T	circle, r=1.15mm, dot in centre
U	circle, r=0.4mm
V	thick dot
W	circle, r=0.6mm
X	cross, 5x5mm
Y	triangle, sides=2mm.

but the user may define other values using up to 4 characters.

PNR is the point number.

HGEL is used to indicate a point height (z-coord).

LI Graphical Line element

The basic graphical line element, and used to define some linear feature components. This element is delimited by two PG point elements in a lower hierarchy. Some higher order feature elements are broken into constituent LI components eg. area features (FL).

```
ETYP=LI STU=n ENUM=n EB=n
```

An HGEL height parameter may be applied to the line.

LY Graphical Poly-line element

The polygon or poly-line element used to define some linear features as a string of lines. This feature requires two PG point elements, in a lower hierarchy, to define the first and last vertices, as well as the intermediate coordinates in the parameter fields.

```
ETYP=LY STU=n ENUM=n EB=n
NAM
FLD ZC35FF50800000000    -1535.3145
    Z4412048A00000000     4612.5413
        ..
        ..
    ZC35F66EA00000000    -1526.4321
    Z44121BC300000000     4635.7629
```

NAM is the feature element name.

FLD describes the feature vertex coordinates. If the first and last points are different, then they are defined in two PG point elements in a lower hierarchy and excluded from the FLD block.

An HGEL height parameter may be applied to the line.

SN Interpolated line or Spline element

The spline or interpolated linear feature element, and used to define some linear features. This feature requires two PG point elements to define the first and last vertices, in a lower hierarchy, as well as the intermediate coordinates in the parameter fields.

```
ETYP=SN STU=n ENUM=n EB=n
ZSP FREI
XMI Z44EF70E9     61296.9102
YMI Z44339031     13200.1914
XMA Z44F2F0B2     62192.6953
YMA Z4434CD7A     13517.4766
LEN
FRE
```

```
WAN
WEN
FLD ZC35FF50800000000    -1535.3145
    Z4412048A00000000     4612.5413
        ..
        ..
    ZC35F66EA00000000    -1526.4321
    Z44121BC300000000     4635.7629
```

ZSP is the spline type.
(i) FREI (free spline, zero curvature at both ends)
(ii) SCHL (closed spline)
(iii) STAN (tangent fixed at start, free at end)
(iv) STAE (tangent fixed at both ends)

XMI,YMI,XMA,YMA describe the bounding limits of the interpolated curve in absolute ground metres.

LEN is the length of the interpolated curve.

WAN,WEN are the angles at the beginning and end of the interpolated curve.

FLD describes the feature vertex coordinates, except for the first and last ones, which are defined in two PG point elements in a lower hierarchy in all but closed splines.

FL Area feature element

The area fill feature element. This feature is defined by a series of LI or LY line elements at a hierarchy one lower, which are themselves defined by two PG point elements at a hierarchy two lower.

```
ETYP=FL STU=n ENUM=n EB=n
NAM
X Z44EF70E9     61296.9102
Y Z44339031     13200.1914
W
SA
FLA
```

NAM is the feature element name.

X,Y are the area feature reference coordinates.

W is the angle of the hatching used for shading.

SA is the spacing between the hatching lines used for shading.

FLA is the area of the feature.

FL areas containing curve sections will not be output by the converter.

BO Curve/arc element

The arc or curve feature element. This feature is defined by a pair of PG point elements at a lower hierarchy.

```
ETYP=BO STU=n ENUM=n EB=n
X Z44EF70E9     61296.9102
Y Z44339031     13200.1914
R
W
```

X,Y are the coordinates of the centre of the arc.

R is the radius of the arc.

W is the angle subtended by the arc.

KR Circle feature element

The circle feature element.

```
ETYP=KR STU=n ENUM=n EB=n
X Z44EF70E9     61296.9102
Y Z44339031     13200.1914
R
```

X,Y are the coordinates of the centre of the circle.

R is the circle radius.

SY Symbol feature element

The symbol feature element.

```
ETYP=SY STU=n ENUM=n EB=n
NAM
X Z44EF70E9     61296.9102
Y Z44339031     13200.1914
W
F
S
X1 Z44EF70E9     61296.9102
Y1 Z44339031     13200.1914
X2 Z44F2F0B2     62192.6953
Y2 Z4434CD7A     13517.4766
```

NAM is the feature element name.

X,Y are the coordinates of the symbol.

W is the symbol rotation angle.

F is the scale factor.

S is the reflection flag: 0 for no reflection, 1 for reflection about the x-axis and 2 for reflection about the y-axis.

X1,Y1,X2,Y2 are the coordinates of the bounding box. These are in ground metres relative to X,Y.

TX Text feature element

The text feature element.

```
ETYP=TX STU=n ENUM=n EB=n
X Z44EF70E9     61296.9102
Y Z44339031     13200.1914
N
H
W
A
M
TXT
```

X,Y are the coordinates of the text reference point.

N is the number of characters in the text string.

H is the text height.

W is the text rotation angle.

A is the grid justification that the text reference coordinates represent: L for the left-hand bottom point, R for the right-hand bottom point, M for the middle bottom point or Z for the central position.

M is the ratio of character width to height (default 0.81).

TXT is the actual text string.

FR Cadastral feature element

The cadastral label element.

```
ETYP=FR STU=n ENUM=n EB=n
X Z44EF70E9     61296.9102
Y Z44339031     13200.1914
NR1 123
```

```
NR2 4
H
W
NDX 1000.0000
NDY 1000.0000
```

X,Y are the coordinates of the cadastral reference point.

NR1,NR2 are the cadastral numbers of the parcel point which are output on the map as:

```
   123
   ---
    4
```

H is the text height.

W is the text rotation angle.

NDX,NDY if present represent the offset from X,Y at which the label should be output. If not present then the label is output at the parcel point, X,Y.

CHAPTER 2

GDB DATA PREPARATION

DATA PREPARATION

Comparison of SICAD/GDB and IFF

IFF is a feature orientated 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.

SICAD/GDB format to some extent shares this concept of a 'feature', and a GDB file produced by I2GDB is essentially an ASCII representation of an IFF file on a feature by feature basis.

However, a fully structured SICAD/GDB file has imposed upon its constituent elements some form of feature hierarchy, referred to by the STU=n fields in the feature element header lines. As an example, an area feature consists of an area (FL) element, consisting of line segments (LI), and each of these is defined by two point (PG) elements. In this way the line segment belonging to several higher level features need only appear once in full detail in the GDB file, and any further references to it only need refer to it by its unique element number (ENUM=n) in the element header line. As IFF files have no such structure imposed upon them, each feature is decomposed appropriately and the components assigned a unique element number in all but the simplest cases of sharing.

The decomposition of IFF features is dependent on the graphical type (GT) of the feature held in the FRT file, referenced by feature code (FC), so it is essential that an appropriate FRT file is supplied to the program via the /FRT = file-spec qualifier for each run of I2GDB. Additional control is provided by the use of a parameters file.

Feature Translation

The I2GDB Program Action section of this document gives the GDB features output when particular IFF graphical types are encountered. The user can control these by use of a parameters file which defines translations where there are alternatives. Where a data field in required in the GDB and there is no direct equivalent in the IFF file, the user may attach the value required by means of a feature attribute (ancillary code) or a per-point attribute. For this purpose a special section is set up in the FRT file. The following ACD entries provide the basic set of parameters and descriptors that the user may wish to use. In principle the ACD table may be extended to include any other GDB descriptors.

```
ACD TABLE 1
!             Code    Name    Min     Max
ACD   C       1       PKZ                     ! point symbol
ACD   I       2       PNR     1               ! point number
ACD   R       3       WAN     -360.0 360.0    ! spline start tangent
ACD   R       4       WEN     -360.0 360.0    ! spline end tangent
ACD   I       5       NDX                     ! cadastral offset x
ACD   I       6       NDY                     ! cadastral offset y
ACD   R       7       W       -90.0   90.0    ! area hatching angle
```

When the converter receives a feature or point with an attribute from this table it outputs the name and value of the attribute, for example "WAN 45.0". The value is output as an integer, real or 4-character string according to the type of the ACD entry. For attributes attached to a feature, a special extension exists for character type attributes. If the value "USTX" or "ustx" is received, then the text entry of the AC is used instead of its value. (See IFF-FEATURE reference section for description of text entries of ACs).

Point heights, HGEL, will be output to a GDB file if a feature in an IFF file has an AC 2 or AC 3 attribute. These attributes both specify contour heights. Point heights will also be output if ZS coordinates or CB coordinates with a 'Z' point attribute are used in the IFF file.

A cadastral symbol should be represented in an IFF file as a text element with a text string of the form n1/n2. For example, in textual form:

```
NF 101
FS 1
ST
100 700
RO 0
TX 123/4
EF
```

Area Features

The I2GDB utility can prevent lines which are common to two areas being written twice to the output file. This, however, requires the IFF file to be prepared in such a way that the program will know if area lines are unique or are shared with other areas in the same layer.

For this to work correctly, the following points should be noted:

- It will only apply to areas (ETYP=FL) intended to be written as lines (ETYP=LI) as opposed to polylines (ETYP=LY). This must be defined in the parameter file.

- Each point of an area should be given a point attribute. This attribute flags the line (which the point begins - in the direction the area was digitised) as being common to another area. This 'shared-code' attribute should be a unique integer or zero. If it is non-zero, it denotes the

line as being common to another area (whose corresponding line must have the same unique shared-code attribute on the starting point of the line) and zero denotes that the line does not belong to another area.

- As areas may be digitised in any order, it is vital that the points have the correct attributes associated with them and if this is not the case, unpredictable results may occur.

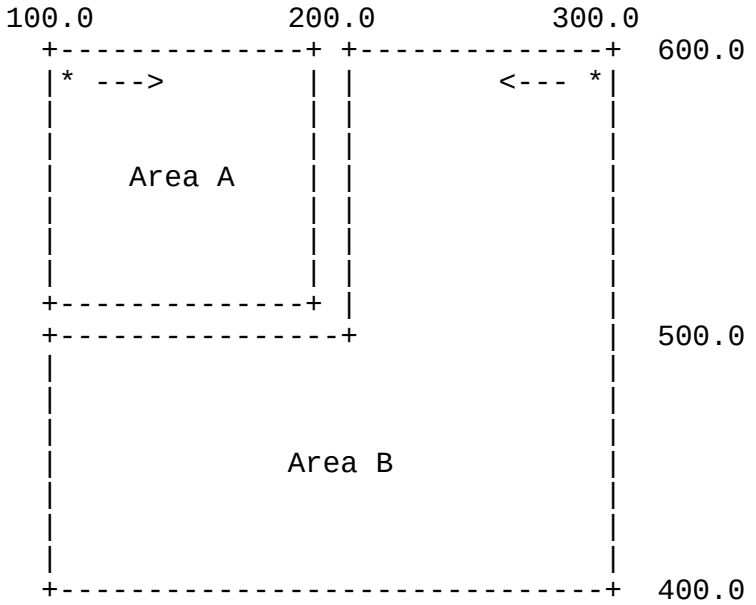
- Duplicate points are not eliminated by this process as only the ENUM of each common **line** is held.

- Only areas with common lines that are in the same layer will be dealt with.

- The shared-code attribute should be included in the ACD part of the FRT file and should be of type integer and called 'SHARED' as shown here:

```
ACD TABLE 1
!       Type    Code    Name    Min     Max
ACD     C       1       PKZ
ACD     I       2       PNR     1
ACD     R       3       WAN
ACD     R       4       WEN
!
! etc
!
ACD     I       10      SHARED
```

Consider the example below which consists of two areas, A and B, which have two lines in common. The areas have been shown apart for clarity but do actually coincide. The first point in each area is denoted by '*' and the arrow gives the direction in which it was digitised.



The point attributes should be attached to the points as follows:

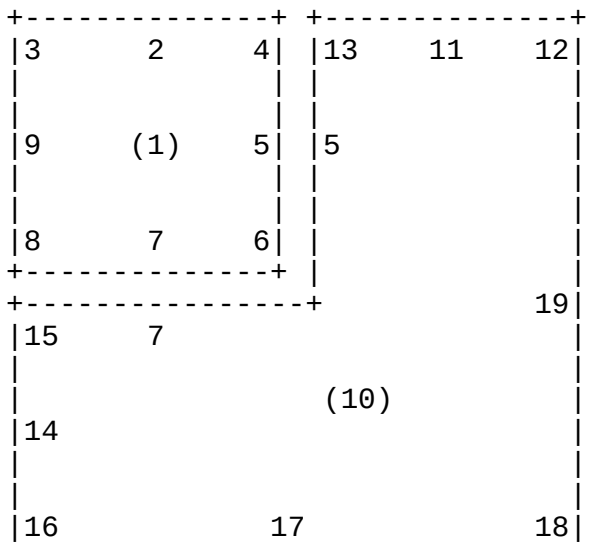
For area A, the Coordinate Block entry should look like:

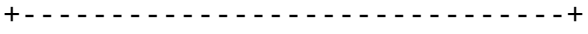
```
CB FL 0
   GT 12
   NR 5
   NC 3
   NA 0
   HC 91 92 1010              ! 'SHARED' attribute has code 1010 in
ACD
   DA 100.0 600.0     0       ! line is not common
      200.0 600.0     1       ! given unique value of 1
      200.0 500.0     2       ! given unique value of 2
      100.0 500.0     0       ! line is not common
      100.0 600.0     0       ! line is not common
```

For area B, the Coordinate Block entry should look like:

```
CB FL 0
   GT 12
   NR 7
   NC 3
   NA 0
   HC 91 92 1010              ! 'SHARED' attribute has code 1010 in
ACD
   DA 300.0 600.0     0       ! line is not common
      200.0 600.0     1       ! corresponds to value in area A
      200.0 500.0     2       ! corresponds to value in area A
      100.0 500.0     0       ! line is not common
      100.0 400.0     0       ! line is not common
      300.0 400.0     0       ! line is not common
      300.0 600.0     0       ! line is not common
```

When we execute I2GDB, with all the above correctly defined, the element numbers (ENUMs) will be associated with their elements as shown:





Where the ENUMs in brackets are those of the two areas (FL). Those adjacent to lines are of the lines (LI) and those in corners are of points (PG).

As can be seen, lines with ENUMs 5 and 7 will not be duplicated but the end points of the whole common area will be.

CHAPTER 3

I2GDB UTILITY

UTILITY I2GDB

FUNCTION

I2GDB reads an Internal Feature Format (IFF) file, and produces a file on disc in SICAD/GDB (Geographical Data Base) format.

FORMAT

$ I2GDB input-IFF-file-spec output-GDB-file-spec

Command qualifiers	**Defaults**
/[NO]DEBUG	/NODEBUG
/FRT=file-spec	/FRT=LSL$FRT:OS.FRT
/[NO]LOG	/NOLOG
/PARAMETERS=file-spec	/PARAMETERS= LSL$FRT:GDBPARAMS.PAR

PROMPT

_Input IFF file: input-IFF-file-spec

_Output GDB file: output-GDB-file-spec

PARAMETERS

input-IFF-file-spec

- This parameter specifies the name of an IFF file, and is compulsory. The data written to the output file are read from this file. Only one filename may be specified for each run of the program.
  The default device and extension LSL$IF:filename.IFF are applied to the input file specification when it is parsed.

output-GDB-file-spec

- This parameter specifies the SICAD/GDB data file spec, and is compulsory. The data read from the input IFF file is written to this file. Only one filename may be specified for each run of the program. The default filename and extension GDB.GDB are applied to the output file specification when it is parsed.

COMMAND QUALIFIERS

/DEBUG
/NODEBUG (default)

- When /DEBUG is present, the I2GDB utility will output many more diagnostic messages as it processes the input file than it would under normal running, or with the /LOG qualifier given. Each IFF entry is signalled, with the FSN and ISN given for the NF entries, and the number of coordinate points and their visibility given for ST/ZS/CB entries, as an aid to tracing any IFF conversion problems.

/FRT = file-spec

- The /FRT command qualifier specifies a Feature Representation Table (FRT) file which the program will read to determine the graphical types (GT) of the IFF features. The FRT qualifier with file-spec is compulsory.
  The default file specification is LSL$FRT:OS.FRT, against which the file specification given with /FRT qualifier is parsed.

/PARAMETERS = file-spec

- The /PARAMETERS command qualifier specifies a Parameters file which the program will read to determine how particular IFF constructs map onto GDB constructs. The PARAMETERS qualifier with file-spec is not compulsory.
  The default file specification is LSL$FRT:GDBPARAMS.PAR, against which the file specification given with /PARAMETERS qualifiers is parsed.

/LOG
/NOLOG (default)

- When /LOG is present, the I2GDB utility will output diagnostic messages about any file creation or deletion as it processes the input IFF file so that any disc errors may be traced.

RESTRICTIONS

Only one IFF file is allowed in each run of I2GDB, to produce a single GDB file.

DESCRIPTION

Command line

The symbol I2GDB is normally set up as:

```
I2GDB == "$lsl$exe:i2gdb"
```

and the program may then be used as if it were a normal VMS utility.

PARAMETERS FILE

A parameters file uses three sorts of commands to map IFF constructs onto GDB constructs.

LAYER

The LAYER command is followed by an IFF layer number, and a corresponding GDB layer number to use as output for the EB parameter.

```
LAYER 0 31 !  IFF layer 0 -> GDB EB 31
```

If no LAYER command has been given for a particular layer encountered, then I2GDB uses the IFF layer number, modulo 32, as the GDB layer number.

All LAYER commands in the parameters file should appear in ascending order of IFF layer number.

SYMBOL

The SYMBOL command takes an FRT symbol number (SC), a corresponding 8-character name to use in the GDB NAM field and an optional scale factor.

```
SYMBOL 21 XCROSS       ! IFF SC 21 -> GDB NAM "XCROSS"
SYMBOL 22 XCROSS 3.0   ! IFF SC 22 -> GDB NAM "XCROSS", scale
factor 3.0
```

If no SYMBOL command has been given for a particular SC encountered, then I2GDB creates a name from the "SC" followed by the IFF SC number, e.g., "SC21".

The scale factor is used to map the IFF unit symbol square onto a symbol square in SICAD. The IFF unit symbol square extends from (-1,-1) to (+1,+1), the SICAD symbol square is of arbitrary size, set by the creator of the symbol library.

The value of scale factor to be entered in the parameters file should be

(Length of side of GDB symbol square) / (Length of side of IFF symbol square).

For example, if the GDB symbols used are defined on (-400,-400) to (+400,+400) then the scale factor should be 400. A default of 1 is assumed if no value is given.

There is a special use of the SYMBOL command shown below:

```
SYMBOL 23 PG          ! IFF SC 23 -> PG output
```

In this case the symbol will be output as if it was a single point in STU level 1. This would be done as a way of entering a geodetic fix point or similar.

All SYMBOL commands in the parameters file should appear in ascending order of IFF SC number. See note below on corresponding FC entries.

FC

The FC command takes an FRT feature code (FC) and provides ST and SM integer values. It also provides flags to say whether a line is to be output using LI,LY constructs, what kind of spline to use for translating curve elements, whether to output area using FL constructs, and whether to output a text using TX or FR constructs.

```
FC 123 ST 3 SM 4 LI   ! IFF FC 123 -> GDB LI with ST and SMs.
FC 124 ST 3 LY FL     ! IFF FC 124 -> GDB FL area using LY.
FC 125 TX M 0.9        ! IFF FC 125 -> text, aspect ratio(M) =
0.9
FC 126 FR             ! IFF FC 125 -> GDB FR cadastral label.
FC 127 SN ZSP FREI    ! IFF FC 127 -> GDB FREI (free) spline.
```

The allowable spline types are:
(i) FREI (free spline, zero curvature at both ends)
(ii) SCHL (closed spline)
(iii) STAN (tangent fixed at start, free at end)
(iv) STAE (tangent fixed at both ends)

If no FC command has been given for a particular FC encountered, then the default action depends on the graphical type (GT) of the feature as found from the FRT. For linear features, LIs are used in preference to LYs. For text features TX is used with no FR output, aspect ratio 0.81. For area features, the area is output as line elements. For curves, the curve is output as a free spline. For No ST or SM entries are made.

Note carefully that the SYMBOL entry specifies the translation only of the IFF secondary codes. Therefore it is still necessary to give FC entries for the features which use these symbols.

Note that if insufficient info is available to the converter, for example through missing attribute data, it may use default values anyway.

Note that a FL element is output using LI line segments unless an explicit LY flag has been given in the relevant FC command.

If defaults are used, a set of messages of the form:

"x features with FC y have been output using default translations"

will be output at the end of the program.

PROGRAM ACTION

I2GDB is a utility to transfer an Internal Feature Format (IFF) disc file to a SICAD/GDB format ASCII disc file.

Note that only a single dataset may be transferred during a single run of the program.

For a full description of the SICAD/GDB output file structure and format, see the Format section of this module's documentation.

After decoding the command line for input and output file names and the qualifiers, the FRT file is read and the values stored in internal arrays indexed by feature code (FC).

The supplied IFF file is opened with input revision level of one so that any ST or ZS entries are translated into CB entries, from which x and y coordinates are read and possibly z coordinates and PKZ symbol definitions. Z coordinates are used to produce HGEL parameters on points.

After writing the range values to a header comment line, I2GDB translates the IFF file entry by entry into the relevant GDB feature element hierarchy or STU level.

Any overlay (NO) entry is recorded for writing to the EB level entry of each feature element.

Contour line heights as attribute codes (ACs) 2 and 3 are detected and used to produce HGEL parameters on lines.

Any information in TX, TH or RO entries is stored for later use depending on the graphical type of the feature.

What I2GDB does with coordinate information from any CBs (and converted ST/ZS entries) depends on the graphical type (GT) of the feature derived from the feature code (FC) in the FS entry and on entries in the parameters file.

Linear feature (GT 1) coordinates are output at the top level (STU=1) either as line segment (LI) feature elements or as poly-line (LY) feature elements according to the relevant FC command in the parameters file. The default is LI if there is no parameters file or no relevant FC command. Each LI segment is

defined by a pair of point (PG) elements at a lower level (STU=2). A LY feature has the first and last vertices output to point (PG) elements at a lower level (STU=2) unless the polyline is closed. In this case the start point is repeated in the FLD field if the LY element. The default is to output "zero" point symbols parameters (PKZ Z), but if a "PKZ" attribute is attached to a point in a coordinate block then a non-zero PKZ parameter will be output.

Line features with only two points will be output as LIs even if the parameter file specifies that feature is to be output as an LY.

Polyline (LY) and spline (SN) features are split after 48 points (including the two end points). In the case of a 49 point feature, it will be split earlier into two LYs or SNs.

Interpolated curve features (GT6) are output to spline (SN) feature elements at the top level (STU=1), with the first and last vertices output to point (PG) elements in a lower level (STU=2) unless the spline is closed. The type of spline output depends on the translation set up in the parameters file. For STAN and STAE splines the user must have attached WAN and WEN angles as appropriate as feature level attributes. If these are missing, then a FREI spline will be output. Similarly, if a SCHL spline is not in fact closed, then a FREI spline will be output.

Symbol string features (GT11) are output as a series of symbols at the top level (STU=1), one at each vertex of the string.

Area features (GT12) are output to area (FL) elements at the top level (STU=1) provided an entry of the form:

```
FC 124 ST 3 LY FL
```

exists in the parameters file. The segments of the bounding polygon are decomposed into either a series of LI segments or a LY feature according to the parameters file FC command. The default is to use LI segments. Any hatching information present from the FRT file is used if possible for the parameter fields. If no relevant FC command exists in the parameters file then the feature will be output as simple LI segments with no areal parameters. If the SC entry of the FRT is 101->106, ST is set to 0, invisible perimeter. Note that if the user attaches an attribute named 'W' to the area which holds real values, then the converter outputs this value instead of the angle derived from the FRT.

Part arc features (GT2,3,4) are output to curve (BO) feature elements at STU=1, with the end vertices used for lower level (STU=2) point (PG) elements, with no symbols. The radius and angle subtended by the arc are calculated from the three edge coordinates or edge-centre-edge coordinates and used in the relevant parameter fields. Note that all BO feature elements have anticlockwise sense, and so clockwise arcs from IFF have their sense reversed.

Full circle features (GT5) are output to circle (KR) elements, with the radius calculated from the three edge coordinates and used in the relevant parameter field.

All symbol features (GT7,8,9) are output to symbol (SY) elements at STU=1. The values from the IFF RO and TH entries, and from the FRT file, are used to calculate the various parameter field entries. The symbol scale from the parameter file is used to ensure the GDB symbols have the correct scale factor.

Text features (GT10) are output to text (TX) elements at STU=1 unless an FC command in the parameters file for the FC encountered specifies an FR cadastral label. For example:

```
FC 125 FR
```

In the FR case, the text string should be of the form n1/n2 and is used to construct the two cadastral number parameters of the features.

In both TX and FR cases, the values from the IFF RO and TH entries and from the FRT file are used to calculate the various parameter field entries. For TX output, text features that are components of a text status (TS) string are processed as separate features. For FR output, the feature attributes NDX and NDY are processed to give cadastral offsets.

If a linear feature contains invisible line breaks, the sections are output to GDB as separate features.

EXAMPLES

$ **I2GDB/FRT=OS TEST_GDB GDB <CR>**

```
  Now processing the IFF file LSL$IF:TEST_GDB.IFF to GDB.GDB

%I2GDB-S-NORMAL, successful completion
 ELAPSED:    0 00:00:08.15  CPU: 0:00:01.82  BUFIO: 18  DIRIO: 61  FAULTS: 210
```

This example shows a normal run of I2GDB successfully converting the IFF file LSL$IF:TEST_GDB.IFF into the GDB file GDB.GDB.

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 terminated successfully.

User action: None.

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.

CBREADERR, error reading coordinate block

Explanation: Could not find x or y coordinates in the coordinate block.

User action: Check integrity of IFF file.

CMDLNERR, error reading the command line

Explanation: There was an error reading the supplied command line.

User action: Resupply the command line with the correct parameters and/or qualifiers. In particular ensure that a valid FRT file-spec is supplied.

FRTOPNERR, error opening FRT file

Explanation: The FRT file given in the command line could not be opened.

User action: Check that the FRT file exists, or is not corrupted.

GDBLEVELERR, level number 'integer' too large for GDB

Explanation: The IFF overlay number is too large for the GDB EB=level parameter, which must lie in the range 0 to 31.

User action: Supply the IFF overlay numbers within the correct range and rerun the program.

GDBOPNERR, error opening GDB file

Explanation: The GDB file given in the command line could not be opened.

User action: Check that there is sufficient privilege for file creation, or that there is no disc error.

IFFOPNERR, error opening IFF file

Explanation: The IFF file given in the command line could not be opened.

User action: Check that the IFF file exists, or is not corrupted.

INVALFRT, invalid FRT file

Explanation: The FRT file contains erroneous data

User action: Check the entries in the FRT file.

NOFRTENTRY, feature with FC 'integer' encountered with no FRT entry

Explanation: A feature was encountered in the IFF file with a feature code that was not listed in the supplied FRT file.

User action: Supply a valid FRT file and rerun the program.

PAROPNERR, error opening PARAMETERS file

Explanation: The PARAMETERS file given in the command line could not be opened.

User action: Check that the PARAMETERS file exists or is not corrupted.

PARSTXERR, syntax error in parameter file

Explanation: The PARAMETERS file contains a line with incorrect syntax.

User action: Check syntax, and check LAYERs, SYMBOLs and FCs in ascending order.

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.