Laser-Scan Ltd.

CONVERT PACKAGE

IFFMAPDATA Reference

Issue 1.1 - 2-November-1994

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

Document "IFFMAPDATA REFERENCE", Category "REFERENCE"
Document Issue 1.1   S Townrow (modified 02-Nov-1994)

CONTENTS

IFFMAPDATA - Change Record

Version 1.0   S Townrow 12-September-1994

Module IFFMAPDATA - Documentation created for new MAPDATA2I and I2MAPDATA utilities.

Version 1.0   S Townrow 12-September-1994

Module IFFMAPDATA - The I2MAPDATA message section has been updated to include the message UNSETSCALE and the second example run of the program has had the filenames put in the correct order.

Module IFFMAPDATA - The MAPDATA2I reference manual has been modified to state the the IFF range values will be written to the Corner Points (CP) entry.

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

MAPDATA FORMAT DESCRIPTION

FORMAT DESCRIPTION

The MAPDATA Format

This section describes parts of the MapData format created by the PC based MapData system which are read by MAPDATA2I and also created by the I2MAPDATA utility.

The format is a very simple one which holds only line and point features with one attribute. Text is not supported.

A MapData file will begin with three header lines consisting of two blank lines followed by a line of the form:

```
     190.000     260.000  0.01000000   * * * * 1.000000
```

The meaning of these fields is not well-defined but the first two are suspected to be the X and Y extents of the file.

Each feature is started with a line which consists of the keyword `SEGMENT', the feature serial number (FSN) and space for a single text attribute, eg:

```
SEGMENT 1  **  A14 (A45)  **
```

The attribute is located after the first two asterisks and first two spaces and is followed by two spaces and two asterisks.

The feature code of the feature is held on the next line and is given after the `CODE' keyword as shown in this example:

```
CODE 25.61
```

MapData feature codes lie in the range 1.1 to 127.127 and are treated as a major code before the point and a minor code after the point rather than as a real number. This is because 56.20 would be the same feature code as 56.2 if treated as a real number.

After the `CODE' line defining the feature code are listed all of the coordinate points in the feature together with flags which determine their representation. For example:

```
495.940 145.820 0.000 C H
495.720 145.790 0.000 C H
495.520 145.800 0.000 H
492.470 145.980 0.000 C L
495.400 146.090 0.000 C L
493.340 146.170 0.000 C L
495.310 146.280 0.000 L
495.220 146.270 0.000
487.290 144.470 0.000
```

The first three real numbers are the X, Y and Z ordinates of the point respectively and are held in kilometre values.

Each point can optionally have one of three flags after the Z value. A `C' flag means that this point has curved status when drawn and the absence of this flag means that the section is straight. The curve flags in a feature will result in a curve being fitted through those points which have the `C' flag starting from the point BEFORE the first flag, eg.

```
495.520 145.800 0.000
495.470 145.980 0.000 C
495.400 146.090 0.000 C
495.340 146.170 0.000 C
495.310 146.280 0.000
495.220 146.270 0.000
```

This will result in a curve being fitted through the first 4 points. At least three points are required to generate a curve which means that the second and third points should have curved status. A single curved point on its own (with no adjacent curved points) will not be treated as a curve and drawn straight.

The `H' and `L' flags on a point determine the priority of the section from that point to the next when drawing. The `H' means high priority and the `L' means low priority and specifies that this road section should be drawn as an over section or an under section respectively. Absence of either of these flags denotes the road section as being of normal priority.

There is nothing to terminate a MapData file, the last line will simply contain the last coordinate of the last feature.

Example MapData File

Start of file:

```
     400.000     300.000  0.01000000   * * * * 1.000000
SEGMENT 1  **  Attribute string  **
CODE 20.1
495.940 145.820 0.000 C H
495.720 145.790 0.000 C H
495.520 145.800 0.000 H
495.470 145.980 0.000 C L
495.400 146.090 0.000 C L
495.340 146.170 0.000 C H
495.310 146.280 0.000
495.220 146.270 0.000 C H
495.140 146.220 0.000 C H
495.090 146.130 0.000 C H
495.070 146.050 0.000 H
494.760 146.030 0.000 H
494.740 145.920 0.000 H
495.140 145.860 0.000
495.190 145.370 0.000 L
495.040 145.280 0.000 L
495.240 145.120 0.000
SEGMENT 2  **    **
CODE 20.4
487.440 114.820 0.000
488.720 125.790 0.000
489.520 124.800 0.000
480.470 129.980 0.000
481.800 110.090 0.000
482.130 117.170 0.000
```

CHAPTER 2

MAPDATA DATA PREPARATION

DATA PREPARATION

Comparison of MapData 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.

The MapData format shares this concept of a 'feature' but only supports simple lines and symbols. A feature consists of a feature serial number, a feature code, one optional text attribute and the coordinate(s). The main difference between the MapData and the IFF Data Model is that MapData allows different parts of the same feature to have different representations. This is held in the MapData file by assigning to each point of the coordinates a flag which can denote the point is to be drawn as part of a curve and/or at high or low priority (over or under other features).

Since the IFF Data Model cannot directly mirror this representation scheme at the per-point level the data can be altered by various options described below.

MapData has its own feature coding scheme whereby feature codes lie in the range 1.1 to 127.127 which are treated as a major and minor codes not as a real number. There must exist a mapping between these and IFF feature codes (which are integers that lie in the range 1 to 32767) and this is described in the parameter file mechanism below.

Feature Translation

When converting from IFF to MapData it is only possible to translate simple IFF features. These are lines (which can be interpolated curves) and symbols only. Text features are ignored since the MapData format does not support them.

All arcs and circle features (GTs 2, 3, 4, 5) in IFF are defined by three coordinates; edge-centre-edge or three edge points. These must be interpolated sufficiently to enable these features to be translated as simple linear features.

Interpolated curves (GT 6) are considered to be a simple linear feature.

Symbol features (GTs 7, 8, 9) are all converted into MapData features and GTs 8 and 9 will generate point features with two points (the second denoting its orientation).

Symbol string features (GT 11) are considered to be simple linear features, and the symbols defining the vertices are lost.

Fill area features (GT 12) are considered to be a closed linear feature, and any pattern fill information is lost.

Preparation for data to be transferred between MapData and IFF

It is quite likely that IFF will be used as the digitising format and the data subsequently transferred to a MapData system. On the other hand, data may be transferred from a MapData system to IFF for complex editing operations, or for other spatial processing functions such as structuring or polygon formation.

Whatever the purpose for transferring data, the relationship between IFF feature codes and MapData feature codes, if these are to be preserved, should be defined.

The implementation of this requires

- o a parameter file defining feature code correlation, and

- o an FRT file (together with SRI and TRI files) containing an ACD section

The main purpose of the parameter file is to define how the MapData feature codes map onto IFF feature codes but may also contain definitions for various tolerances and meta-data (such as local origin and scale-factor).

MAPDATA2I and I2MAPDATA will read the same parameter file and this is encouraged to ensure accurate and consistent translation. This is particularly useful when batch converting multiple files from the same volume of data as the various meta-data will be the same.

The MAPDATA2I parameter file

This file contains two types of entry :-

a) the environment settings, and
b) the feature code lookup table

The environment setting keywords

These keywords are the same as many of the DCL command line qualifiers and can be used to save the need to specify an option each time on the command line which is particularly useful when batch processing many files.

Example keywords follow :-

```
!
!
CURVE_AC      CODE    2001
PRIORITY_AC   NAME    Priority
TEXT_AC       NAME    Text
!
```

```
CORNER_POINT_SEPARATION      40.0
!
CURVE_SEPARATION             40.0
CURVE_CHORD                  10.0
CURVE_DEVIATION              0.5
!
SCALE                        200000
OFFSET                       400000 100000
MULTIPLY                     1000
!
DEFAULT_LINE_FC              1
DEFAULT_SYMBOL_FC            25
!
!
```

For an explanation of these keywords, see the corresponding command line qualifiers documented in the MAPDATA2I Reference Manual.

Any I2MAPDATA keywords which are present in the parameter file will be read and ignored by MAPDATA2I, but any undefined words or syntax will be reported.

The feature code lookup table

Each line of the 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. Any text starting with the "!" character is regarded as a comment, and is ignored. Comments can appear on the same lines as commands, or on separate lines. Note that commands may be in upper or lower case.

The feature code lookup table describes the translations required to produce IFF feature codes from MapData feature codes. Each line is prefixed with the CODE command which tells the program how to translate a particular feature code.
A line has the following form:-

CODE IFF-code MD-fc Interp Curve-code

where the fields are :-

CODE - the command CODE states that this line is an entry in the feature code lookup table.

IFF-code - the IFF feature code that will be used for the IFF feature which mirrors the MapData feature with the feature code defined by **MD-fc**.

MD-fc - the MapData feature code which maps onto the above IFF feature code. The feature code should lie in the range 1.1 to 127.127.

Interp - the interpretation of the MapData feature when creating a corresponding IFF feature. This can be blank in which case the IFF feature is an exact copy of the MapData feature or it can be one of the following three keywords:

STRAIGHT - this should be used when the feature is predominantly straight and any curved sections will be interpolated (smoothed) by having additional points stored in the feature. The output graphical type for the feature should be linear (GT 1) and the degree of smoothing is controlled by the tolerances of qualifiers /CURVE_SEPARATION, /CURVE_CHORD and /CURVE_DEVIATION or their corresponding parameter file keywords.

CURVE - this should be used when the feature is predominantly curved with a few straight sections within which are maintained by inserting new points along the section to hold the shape. These features should have a curve graphical type (GT 6) and the insertion of controlling point is done at a distance specified by /CORNER_POINT_SEPARATION from the non-curved point.

BREAK - this option can be used to break the feature when there is a change between the curve and straight sections. This will require that two feature codes are present in the parameter file for the two types of section. The parameter file should contain the IFF straight FC, the MapData code, the keyword `BREAK' and the IFF curve FC.

Curve-code - the IFF feature code for the curve section of broken features. This should only be present in the parameter file when the BREAK option is used. The feature code supplied should be of graphical type 6.

Symbol features should not have an **Interp** option as these apply to linear features only.

Rotatable symbols in MapData are held with two points, the first being the locating point and the second giving its orientation (from the first). These can be converted to IFF by ensuring that the destination IFF feature code for the feature is of graphical type 8 (a rotated symbol). In this case the IFF feature will have the single locating point and the corresponding rotation written to IFF RO entry of the feature.

An example follows :-

```
!
! Feature code lookup section of parameter file
!
!CODE  IFF-fc   MD-fc   interp   Curve-fc
!
CODE    1201    12.1    CURVE                   !Open (route)
CODE    1202    12.2    CURVE                   !Under construction (route)
CODE    1203    12.3    CURVE                   !In tunnel
CODE    1212    12.12                           !Junc open restricted access
CODE    1213    12.13                           !Junction open-full access
CODE    1214    12.14                           !Junction under construction
CODE    1224    12.24                           !Location
CODE    1225    12.25                           !Symbol placement
CODE    1232    12.32   CURVE                   !Link road
CODE    1401    14.1    STRAIGHT                !Open (route)
CODE    1402    14.2    STRAIGHT                !Under construction (route)
CODE    1403    14.3    STRAIGHT                !In tunnel
CODE    1412    14.12                           !Junc open restricted access
CODE    1413    14.13                           !Junction open-full access
CODE    1432    14.32   BREAK      999          !Link road
CODE    1501    15.1    BREAK      999          !Open (route)
CODE    1502    15.2    BREAK      999          !Under construction (route)
CODE    1503    15.3    BREAK      999          !In tunnel
CODE    1512    15.12                           !Junc open restricted access
CODE    1513    15.13                           !Junction open-full access
CODE    1524    15.24                           !Location
CODE    1525    15.25                           !Symbol placement
CODE    1532    15.32   CURVE                   !Link road
CODE    1601    16.1    CURVE                   !Dual carriageway
CODE    1602    16.2    CURVE                   !Under construction (route)
CODE    1603    16.3    CURVE                   !In tunnel
!
```

The I2MAPDATA parameter file

This file is almost identical to the MapData2I parameter file but only contains an attribute description table.

This file contains two types of entry :-

a) the environment settings, and
b) the feature code lookup table

The environment setting keywords

These keywords are the same as many of the DCL command line qualifiers and can be used to save the need to specify an option each time on the command line which is particularly useful when batch processing many files.

Example keywords follow :-

```
!
CURVE_AC       CODE    2001
PRIORITY_AC    NAME    Priority
TEXT_AC        NAME    Text
!
ADD_OFFSET
DIVIDE                 1000
!
```

For an explanation of these keywords, see the corresponding command line qualifiers documented in the I2MAPDATA Reference Manual.

Any MAPDATA2I keywords which are present in the parameter file will be read and ignored by I2MAPDATA, but any undefined words or syntax will be reported.

The feature code lookup table

Each line of the 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. Any text starting with the "!" character is regarded as a comment, and is ignored. Comments can appear on the same lines as commands, or on separate lines. Note that commands may be in upper or lower case.

The feature code lookup table describes the translations required to produce MapData feature codes from IFF feature codes. Each line is prefixed with the CODE command which tells the program how to translate a particular feature code.
A line has the following form:-

CODE   IFF-code   MD-fc   Interp   Curve-code

where the fields are :-

CODE - the command CODE states that this line is an entry in the feature code lookup table.

IFF-code - the IFF feature code which maps onto the output MapData feature code described by **MD-fc**.

MD-fc - the MapData feature code which is used for the output feature when converting a feature with feature code **IFF-code**. The feature code should lie in the range 1.1 to 127.127.

Interp - the interpretation of the MapData used by MAPDATA2I only. I2MAPDATA will ignore this field but it was included here for completeness as the same parameter file can be read by I2MAPDATA and MAPDATA2I.

Curve-code - the IFF feature code for the curve section which is only used by MAPDATA2I. I2MAPDATA will ignore this field but it was included here for completeness as the same parameter file can be read by I2MAPDATA and MAPDATA2I.

Rotatable symbols in MapData are held with two points, the first being the locating point and the second giving its orientation (from the first). These can be converted from IFF symbol features (denoted by the appropriate bits set in the FS entry) if they have two points and are of graphical type 8. The second MapData point will be derived by converting the angle in the IFF RO entry into a second point.

Here is an example of a MAPDATA parameter file for use with both MAPDATA2I and I2MAPDATA.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!                            MapData Parameter File
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! Environment setting for MAPDATA2I
!
CURVE_AC        CODE    2001
PRIORITY_AC     NAME    Priority
TEXT_AC         NAME    Text
!
CORNER_POINT_SEPARATION 40.0
!
CURVE_SEPARATION        40.0
CURVE_CHORD             10.0
CURVE_DEVIATION         0.5
!
SCALE                   200000
OFFSET                  400000 100000
MULTIPLY                1000
!
DEFAULT_LINE_FC         1
DEFAULT_SYMBOL_FC       25
!
!
! Environment settings for I2MAPDATA
!
ADD_OFFSET
DIVIDE                  1000
!
!
! Feature code lookup table for MAPDATA2I and I2MAPDATA
!
!
!CODE  IFF-fc   MD-fc   interp   Curve-fc
!
CODE    1201    12.1    CURVE                   !Open (route)
CODE    1202    12.2    CURVE                   !Under construction (route)
CODE    1203    12.3    CURVE                   !In tunnel
CODE    1212    12.12                           !Junc open restricted access
CODE    1213    12.13                           !Junction open-full access
CODE    1214    12.14                           !Junction under construction
CODE    1224    12.24                           !Location
CODE    1225    12.25                           !Symbol placement
CODE    1232    12.32   CURVE                   !Link road
CODE    1401    14.1    STRAIGHT                !Open (route)
CODE    1402    14.2    STRAIGHT                !Under construction (route)
CODE    1403    14.3    STRAIGHT                !In tunnel
CODE    1412    14.12                           !Junc open restricted access
CODE    1413    14.13                           !Junction open-full access
CODE    1432    14.32   BREAK      999          !Link road
CODE    1501    15.1    BREAK      999          !Open (route)
CODE    1502    15.2    BREAK      999          !Under construction (route)
CODE    1503    15.3    BREAK      999          !In tunnel
CODE    1512    15.12                           !Junc open restricted access
CODE    1513    15.13                           !Junction open-full access
CODE    1524    15.24                           !Location
CODE    1525    15.25                           !Symbol placement
CODE    1532    15.32   CURVE                   !Link road
CODE    1601    16.1    CURVE                   !Dual carriageway
CODE    1602    16.2    CURVE                   !Under construction (route)
CODE    1603    16.3    CURVE                   !In tunnel
!
```

CHAPTER 3

I2MAPDATA UTILITY

UTILITY I2MAPDATA

FUNCTION

I2MAPDATA reads a Laser-Scan IFF file and produces a disk based MapData file. A comprehensive parameter file and FRT are required to provide a lookup mechanism between IFF and MapData feature codes.

FORMAT

```
$ I2MAPDATA input-IFF-file-spec output-MAPDATA-file-spec
```

Command qualifiers	**Defaults**
/[NO]ADD_OFFSET	Add origin
/[NO]CURVE_AC = keyword	No curve AC defined
/DIVIDE = real	Division factor = 1.0
/[NO]FRT_FILE = file-spec	No FRT
/[NO]LOG	No logging
/[NO]PARAMETER_FILE = file-spec	No parameter file
/[NO]PRIORITY_AC = keyword	No priority AC defined
/[NO]TEXT_AC = keyword	No text AC defined

PROMPT

```
_Input-IFF-file:          input-IFF-file-spec
_Output-MAPDATA-file:     output-MAPDATA-file-spec
```

PARAMETERS

input-IFF-file-spec

- This parameter specifies the name of the input IFF file, and is compulsory. The data written to the output file is read from this file. Only one filename may be specified each time the program is executed.

  The default device and extension `LSL$IF:IFF.IFF' is applied to the input file specification when it is parsed.

output-MAPDATA-file-spec

- This compulsory parameter specifies the name of the output MapData file which contains the data converted from the input IFF file. If any parts of parameter are not given, the missing components are taken from the template `SYS$DISK:MAPDATA.MAP'

COMMAND QUALIFIERS

/[NO]ADD_OFFSET

- specifies that the local origin of the data (held in the IFF Map Descriptor) is to be added to the coordinates. This addition of the offset is performed before the scaling is applied from the /DIVIDE qualifier. This qualifier will enable the user to recreate the MapData coordinates that were fed into MAPDATA2I as /ADD_OFFSET and /DIVIDE mirror the /OFFSET and /MULTIPLY qualifiers in MAPDATA2I. The default action is to add the local origin to the coordinates.

/[NO]CURVE_AC = (CODE:integer)

/[NO]CURVE_AC = (NAME:string)

- specifies the number or name of the attribute code which holds the per-point attribute for the curve flags. This AC can be specified using the code or name as defined in the ACD section of the FRT. If not specified, no curve flags will be present in the output file.

/DIVIDE = real

- this specifies the scaling factor which is applied to all IFF coordinates in order to get them into the correct units, typically in kilometres for MapData. It is the inverse function of /MULTIPLY in MAPDATA2I and is applied after the origin offset is added if specified by /ADD_OFFSET. The default is to apply a divide factor of 1.0 (ie nothing).

/[NO]FRT_FILE = file-spec

- specifies a Feature Representation Table (FRT) file which the program will use to interpret the AC entries specified. In order for the program to check the AC entries, there must be an ACD section in the FRT. The FRT file-spec is parsed against the default `LSL$FRT:FRT.FRT'.

/[NO]LOG

- When /LOG is present, the I2MAPDATA utility will output diagnostic messages about any file creation and what is happening as it processes the input IFF file so that any errors may be traced.

/[NO]PARAMETER=file-spec

- specifies the parameter file, which defines the mapping of IFF feature codes into MAPDATA feature codes and any tolerances or attribute codes. For a description of the parameter file, see the DATA

PREPARATION section. NOTE: This qualifier is compulsory.

/[NO]PRIORITY_AC = (CODE:integer)

/[NO]PRIORITY_AC = (NAME:string)

- specifies the number or name of the attribute code which holds the per-point attribute for the priority flags. This AC can be specified using the code or name as defined in the ACD section of the FRT. If not specified, no priority flags will be present in the output file.

/[NO]TEXT_AC = (CODE:integer)

/[NO]TEXT_AC = (NAME:string)

- specifies the number or name of the IFF attribute code whose optional text string holds an attribute which may be stored with the MapData feature. This AC can be specified using the code or name as defined in the ACD section of the FRT. If not specified, no text attribute will be present in the output file.

RESTRICTIONS

Command line qualifiers take precedence over their corresponding keywords in the parameter file.

DESCRIPTION

Command line

The symbol I2MAPDATA is normally set up as:

```
I2MAPDATA == "$LSL$EXE:I2MAPDATA"
```

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

Parameter File

This file contains the tolerances, commands and the feature code lookup table which describes the translations from IFF feature code to MapData feature code. The format of the table is described in the DATA PREPARATION section under the heading `The I2MAPDATA parameter file', and it will probably have been set up by the LAMPS system manager and remain relatively static at most installations.

Program Action

After decoding and checking the command line qualifiers and parameters, the program reads and validates the parameter file. The program is designed to read the same parameter file as MAPDATA2I uses but will only use the IFF to MapData feature code mapping.

The IFF file is opened and the Map Descriptor (MD) entry is read to determine the local origin if required. The first overlay (NO entry) is found and if this is layer 0, all features within it are ignored.

Each feature in turn is read and converted into a corresponding MapData with the appropriate feature code. The feature code is assigned to the output MapData feature based on the IFF code if defined in the parameter file or using the default.

If there is an attribute held in an IFF AC entry (in the optional text string), then this can be stored in the MapData feature, providing the appropriate /TEXT_AC code or name was used.

If the /CURVE_AC or /PRIORITY_AC are used, then any curve or priority flags present on individual coordinate points will be written to the corresponding points in the output string. Only points with curve flags (`C ') are maintained and interpolated (or inserted) points which are stored as `I ' are discarded. High and low priority flags (stored as `H ' and `L ') will also be maintained.

Any oriented IFF symbols (denoted by the appropriate bits in the FS entry) are turned into two point Mapdata features where the second point is generated 4 sheet mms from first. This second point is calculated using the angle held in the IFF rotation (RO) entry and if this was created by MAPDATA2I, then no loss of data will occur. As already stated, the length of the vector will be 4 sheet mms in length. In order to calculate this length, the IFF scale is required from the Map Descriptor. If this is unset (ie set to zero) then the program will assume the length to be 10 IFF units.

EXAMPLES

```
$ I2MAPDATA/FRT=HERE:X/PARAMETER=HERE:X.PAR/LOG/DIVIDE=1000 X.IFF X.MAP

%I2MAPDATA-I-READFRT, reading FRT file HERE:X.FRT
%I2MAPDATA-I-READPAR, reading parameter file HERE:X.PAR
%I2MAPDATA-I-CODES, 659 feature codes found in parameter file
%LSLLIB-I-IFFOPENED, LSL$DATA_ROOT:[MAPDATA]X.IFF;35 opened for read
%I2MAPDATA-I-MAPOPENED, X.MAP opened for write
%I2MAPDATA-I-IGNLAY, layer 0 found and ignored
%I2MAPDATA-I-LAYER, layer 1 found
%I2MAPDATA-W-UNKCURFLG, Unknown CURVE flag on point 2 of feature 1
%I2MAPDATA-W-UNKPRIOFLG, Unknown PRIORITY flag on point 4 of feature 1
%I2MAPDATA-W-UNKCURFLG, Unknown CURVE flag on point 11 of feature 1
%I2MAPDATA-W-UNKCURFLG, Unknown CURVE flag on point 13 of feature 1
%I2MAPDATA-W-UNKPRIOFLG, Unknown PRIORITY flag on point 13 of feature 1
%I2MAPDATA-I-MAPCLOSED, MAPDATA file closed
```

```
%I2MAPDATA-I-IFFCLOSED, IFF file closed
 ELAPSED:    0 00:00:05.36  CPU: 0:00:02.64  BUFIO: 27  DIRIO: 40  FAULTS: 464
```

This example shows a conversion successfully taking place but with a few warnings to alert the user that various things may be wrong with the feature with FSN 1. Use of the /LOG file show useful information such as the fact that layer 0 has been ignored. The reported problems (the warnings) were that the curve and priority flags on various points in the feature did not have the correct values. These points will be transferred with no flags and the messages are just to alert the user. If the user is not concerned about the reported problems then the warnings can be ignored since a valid IFF file was produced.

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.

CODES, 'integer' feature code'(s)' found in parameter file

> **Explanation:** This message appears if /LOG was specified on the command line, and informs the user that the parameter file has been successfully read, giving the number of feature codes found in the CODE entries.
>
> **User action:** None.

IFFCLOSED, IFF file closed

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

IGNLAY, layer 0 found and ignored

> **Explanation:** This message appears if /LOG was specified on the command line, and informs the user that layer zero (usually reserved for registration marks etc.) has been found and all feature within it will be ignored.
>
> **User action:** None.

LAYER, layer 'number' found

> **Explanation:** This message appears if /LOG was specified on the command line, and informs the user that the given layer in the IFF file has been found.
>
> **User action:** None.

MAPCLOSED, MAPDATA file closed

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

MAPOPENED, 'file-spec' opened for write

> **Explanation:** This message appears if /LOG was specified on the command line, and informs the user that the MAPDATA output file has been successfully opened.
>
> **User action:** None.

NOFRT, no FRT file has been specified

Explanation: This message will appear in the absence of /FRT on the command line, or if /NOFRT was specified.

User action: None.

READFRT, reading FRT file 'file-spec'

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

User action: None.

READPAR, reading parameter file 'file-spec'

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

User action: None.

UNSETSCALE, IFF scale unset - symbol lines will be 'number' long

Explanation: MapData holds rotated symbols as a vector line whose angle defines the symbol orientation. I2MAPDATA generates the second point of this vector from the first point and the rotation held in the IFF RO entry. The length of this vector is calculated as being 4 sheet millimeters which can only be derived if the IFF scale (in the Map Descriptor) is set and the units are in metres. This message will appear if the scale is unset and the vector length will default to 10 IFF units.

User action: If the reported vector length is acceptable, then no further action is required, otherwise the IFF scale should be set and the program run again.

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.

BADATTCODE, attribute code 'code' out of range ('min' to 'max')

Explanation: An invalid IFF attribute code (outwith the given range) has been read either from the command line or from the parameter file. If the attribute code has been read from the parameter file, the LINEINPAR message indicating the offending line number will follow, and the program will continue to read the file, ignoring this line. Otherwise program execution will terminate.

User action: For a parameter file error, amend the attribute code on the relevant line in the file, and for a command line error, respecify the command using an appropriate attribute code. The program can then be run again.

BADFC, feature code 'fc' out of range ('min' to 'max')

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

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

BADMDCODE, invalid MAPDATA feature code 'string'

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

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

GOTFC, FC 'fc' has already been defined

Explanation: The given feature code has already been defined in the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

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

LINEINPAR, line 'number' of parameter file

Explanation: This message always appears after an error has occurred while decoding a line in the parameter file, and it informs the user of the offending line number. The previous message will have been generated either by LSLLIB or by the program itself, and will be reduced in status to a warning. The offending line in the parameter file will be ignored.

User action: If the error message was generated by LSLLIB, it most likely indicates an error in the parameter file which should be amended. Otherwise, see the relevant error message explanation.

MAXCODES, number of CODE entries exceeds maximum ('max')

Explanation: There are more than the allowed maximum number of CODE entries in the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Edit the parameter file to reduce the number of CODE entries, and run the program again. If there is an ongoing requirement for an increased maximum, please contact Laser-Scan.

NOSTRING, cannot read character string

Explanation: A character string cannot be read where expected from the parameter file. A character string in this context may be an attribute name, or a MAPDATA feature code. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Edit the parameter file to include the attribute name or MAPDATA feature code where expected, and if necessary run the program again.

TOOMANYPTS, symbol has 'number' points, FSN 'fsn' ignored

Explanation: Symbol features can have no more than 2 points. Such a feature has been found with more than the maximum, so the feature is ignored and processing continues. This error may indicate either that the symbol bits in the FS entry have been incorrectly set, or that the feature has the wrong feature code, also in the FS entry.

User action: Use the IMP utility IPATCH to correct the FS entry. Look at the HI entry to check the history of the file in order to ascertain where in the flowline the IFF file became corrupted.

UNEXPENDIFF, unexpected end of IFF file

Explanation: The end of the IFF file was reached unexpectedly, before an end of job (EJ) entry had been found. The map data will have been correctly processed, but this message may indicate that the IFF file had been improperly closed by a previous process, and some data may be missing.

User action: Run IMEND on the IFF file to insert the missing entries at the end. Examine the file using LITES2 to determine the integrity of the data. If there is some missing data, run the IMP utility IINFO to look at the file history, which may indicate at what stage things went wrong.

UNKCURFLG, Unknown CURVE flag on point 'number' of feature 'fsn'

Explanation: The reported point had a per-point attribute which was specified as being a curve flag and yet it contained an invalid entry. The valid entries for a curve flag are 'C   ', '   ' and 'I   ' which define the point as being curved, undefined or interpolated respectively. If the IFF file was created by MAPDATA2I and all attributes correctly set up then I2MAPDATA will convert the data with no problems. If the data was edited or manipulated in some way, it is possible that the flag could have been corrupted.

User action: The point will be given an undefined flag and will still be present in the output MapData file. If this is acceptable then this message can be ignored, otherwise the offending point should be investigated and corrected.

UNKFC, FC 'fc' has not been defined, FSN 'fsn' ignored

Explanation: The given feature code in the IFF file was not recognised because it has not been defined in the parameter file. The feature is ignored and processing continues.

User action: If the feature code is wanted, it should be defined in the parameter file and the program run again.

UNKPRIOFLG, Unknown PRIORITY flag on point 'number' of feature 'fsn'

Explanation: The reported point had a per-point attribute which was specified as being a priority flag and yet it contained an invalid entry. The valid entries for a priority flag are 'H   ', 'L   ' and '   ' which define the point as being of high, low or normal priority respectively. If the IFF file was created by MAPDATA2I and all attributes correctly set up then I2MAPDATA will convert the data with no problems. If the data was edited or manipulated in some way, it is possible that the flag could have been corrupted.

User action: The point will be given an undefined flag and will still be present in the output MapData file. If this is acceptable then this message can be ignored, otherwise the offending point should be investigated and corrected.

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.

ERRCLOMAP, error closing MAPDATA output file 'file-spec'

Explanation: The MAPDATA output file could not be closed. An additional LSLLIB message will follow, giving the reason for the failure. If this is due to a system error, the system error message will also appear. The program attempts to close the IFF input file, and then exits.

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed. Once the problem has been solved, you can run the program again.

ERRCLOPAR, error closing parameter file 'file-spec'

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

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed. Once the problem has been solved, you can run the program again.

ERROPNMAP, error opening MAPDATA output file 'file-spec'

Explanation: The MAPDATA output file could not be opened. An additional LSLLIB message will follow, giving the reason for the failure. If this is due to a system error, the system error message will also appear. The program attempts to close the IFF input file, and then exits.

User action: Ascertain the cause of the failure from the messages given. There may be a simple remedy, e.g. a change in the directory specification. Alternatively, this error could be due to a more serious problem such as a system failure, in which case the System Manager should be informed. Once the problem has been solved, you can run the program again.

ERROPNPAR, error opening parameter file 'file-spec'

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

User action: Ascertain the cause of the failure from the messages given. There may be a simple remedy, e.g. a change in the directory specification. Alternatively, this error could be due to a more serious problem such as a system failure, in which case the System Manager should be informed. Once the problem has been solved, you can run the program again.

ERRRDFRT, error reading FRT file 'file-spec'

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

User action: Dependent upon the associated error messages.

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

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

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed. Once the problem has been solved, you can run the program again.

IFFCLOERR, cannot close IFF file

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

User action: Look up the IFF error code in the IFFLIB Reference Manual (part of the MAPPING package documentation) and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed. Once the problem has been solved, you can run the program again.

IFFREADERR, error reading from IFF file 'name'

Explanation: An error has occurred while reading the IFF file. This may indicate a more serious system error. The message will be preceded by an IFF library message which will give more details, including a two-character code.

User action: Look up the IFF error code in the IFFLIB Reference Manual (part of the MAPPING package documentation) and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed. Once the problem has been solved, you can run the program again.

NOCODES, no CODE entries in parameter file

Explanation: There are no CODE entries in the parameter file. Without these, the program cannot translate the IFF feature data.

User action: Create a parameter file containing the appropriate CODE entries, and run the program again.

NOMD, cannot find MD entry in IFF file

Explanation: If the origin offset is to be applied to IFF coordinates, I2MAPDATA looks for the Map Descriptor entry immediately after opening the IFF file. If the descriptor is type 2, it is read in to obtain the origin offset. If a map descriptor is not found, the program exits and this message appears, indicating that the IFF file is invalid and therefore cannot be used.

User action: Look at the IFF file using the IMP utility IPATCH, in order to ascertain the level of corruption. The HIstory entry may indicate how the IFF file came to have no MD entry.

UNKATTCODE, cannot find attribute code 'code'

Explanation: An IFF attribute code specified either on the command line or in the parameter file cannot be found in the ACD tables. The message immediately preceding this one will give further information as to why the code was not recognised. Program execution will terminate.

User action: If the attribute code was specified on the command line, respecify the command line using the correct code. If the attribute code was incorrectly specified in the parameter file, amend the attribute code in the file. Run the program again.

UNKATTNAME, cannot find attribute name 'name'

Explanation: An IFF attribute name specified either on the command line or in the parameter file cannot be found in the ACD tables. The message immediately preceding this one will give further information as to why the name was not recognised, and the NOFRT message may follow. Program execution will terminate.

User action: If the NOFRT message indicates that an FRT file is required, respecify the command line with the /FRT qualifier. If the /FRT qualifier is used, ensure that the appropriate ACD definition appears in the FRT file. If the attribute name was specified on the command line, respecify the command line using the correct name. If the attribute name was incorrectly specified in the parameter file, amend the attribute name in the file. Run the program again.

UNSETMD, IFF MD entry not type 2

Explanation: If the origin offset is to be applied to IFF coordinates, I2MAPDATA looks for the Map Descriptor entry immediately after opening the IFF file. If the descriptor is type 2, it is read in to obtain the origin offset. If a map descriptor is not found, the program exits and this message appears, indicating that the IFF file is invalid and therefore cannot be used.

User action: Look at the IFF file using the IMP utility IPATCH, in order to ascertain the level of corruption. The HIstory entry may indicate how the IFF file came to have no MD entry.

MESSAGES (FATAL)

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

UNKCMD, unknown command "'string'"

> **Explanation:** The given command in the parameter file was successfully decoded, but was not recognised during interpretation.
>
> **User action:** This message is most likely due to an error in the program and should be reported to Laser-Scan.

MESSAGES (OTHER)

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

CHAPTER 4

MAPDATA2I UTILITY

UTILITY MAPDATA2I

FUNCTION

MAPDATA2I reads a MAPDATA disk file and produces a Laser-Scan IFF file. A comprehensive parameter file and FRT are required to provide a lookup mechanism between MAPDATA and IFF feature codes.

FORMAT

```
$ MAPDATA2I input-MAPDATA-file-spec output-IFF-file-spec
```

Command qualifiers	**Defaults**
/CORNER_POINT_SEPARATION = real	Corner point separation = 1.0
/[NO]CURVE_AC = keyword	No curve AC defined
/CURVE_SEPARATION = real	Curve point separation = 0.0
/CURVE_CHORD = real	Curve arc to chord distance = 0.0
/CURVE_DEVIATION = real	Curve angular deviation = 0.0
/DEFAULT_LINE_FC = integer	Default line feature code = 0
/DEFAULT_SYMBOL_FC = integer	Default symbol feature code = 0
/[NO]FRT = file-spec	No FRT
/[NO]LOG	No logging
/MULTIPLY = real	Multiplication factor = 1.0
/OFFSET = (real,real)	Origin offset = (0.0,0.0)
/[NO]PARAMETER = file-spec	No parameter file
/[NO]PRIORITY_AC = keyword	No priority AC defined
/SCALE = real	Scale = 1.0
/[NO]TEXT_AC = keyword	No text AC defined

PROMPT

```
_Input-MAPDATA-file:        input-MAPDATA-file-spec
_Output-IFF-file:           output-IFF-file-spec
```

PARAMETERS

input-MAPDATA-file-spec

- This parameter specifies the name of the input MAPDATA file, and is compulsory. The data written to the output file is read from this file. Only one filename may be specified each time the program is executed.

  The default device and extension `SYS$DISK:MAPDATA.MAP' is applied to the input file specification when it is parsed.

output-IFF-file-spec

- This compulsory parameter specifies the name of the output IFF file which contains the data converted from the input MAPDATA file. If any parts of parameter are not given, the missing components are taken from the template `LSL$IF:IFF.IFF'.

COMMAND QUALIFIERS

/CORNER_POINT_SEPARATION = real

- This qualifier specifies the distance (in IFF units) from the end of a straight line section that a shape point will be inserted to control the shape of the feature. This only applies when a feature is being treated as predominantly curved (graphical type of 6) and is to have new points inserted to constrain straight sections. The distance is applied after the coordinates have been scaled with the factor specified by /MULTIPLY. If the distance specified exceeds the length of a straight line segment the shape point will be inserted at a distance equal to a quarter of the segment length from the end.

/[NO]CURVE_AC = (CODE:integer)

/[NO]CURVE_AC = (NAME:string)

- specifies the number or name of the attribute code used to hold the per-point attribute for the curve flags. This AC can be specified using the code or name as defined in the ACD section of the FRT. If not specified, no curve flags will be present in the output file.

/CURVE_SEPARATION = real

- specifies the minimum distance between output data points when Akima cubic interpolation is applied. This distance should be specified in IFF units as it is used after the /MULTIPLY and /OFFSET values have been applied.

/CURVE_CHORD = real

- specifies the arc to chord distance between output data points when Akima cubic interpolation is applied. This distance should be specified in IFF units as it is used after the /MULTIPLY and /OFFSET values have been applied.

/CURVE_DEVIATION = real

- specifies the angular deviation between output data points when Akima cubic interpolation is applied. This angle must be specified in radians.

/DEFAULT_LINE_FC = integer

- this qualifier specifies the feature code to be assigned to any line features which have no mapping present in the parameter file. When a MapData line feature is found, its feature code is looked up in the parameter file to determine the IFF feature code. If the MapData feature code is not listed in the parameter file, then the default is used. This default feature code should be present in the FRT you are using and have the correct graphical type to prevent conflicts.

/DEFAULT_SYMBOL_FC = integer

- this qualifier specifies the feature code to be assigned to any symbol features which have no mapping present in the parameter file. When a MapData symbol feature is found, its feature code is looked up in the parameter file to determine the IFF feature code. If the MapData feature code is not listed in the parameter file, then the default is used. This default feature code should be present in the FRT you are using and have the correct graphical type to prevent conflicts.

/[NO]FRT = file-spec

- specifies a Feature Representation Table (FRT) file which the program will use to interpret the AC entries specified in the parameter file and identify if features have been assigned a feature code with the correct graphical type. In order for the program to check the AC entries, there must be an ACD section in the FRT. The FRT file-spec is parsed against the default LSL$FRT:file-spec.FRT. This qualifier must be used when the parameter file is given.

/[NO]LOG

- When /LOG is present, the MAPDATA2I utility will output diagnostic messages about any file creation and what is happening as it processes the input MAPDATA file so that any errors may be traced.

/MULTIPLY = real

- this specifies a scaling factor which is applied to all coordinates in order to get them into the correct units, typically in metres on the ground. It can be used in conjunction with /OFFSET to get the file in UK National Grid with the coordinates in metres relative to a local origin held in the Map Descriptor.

/OFFSET = (real,real)

- specifies the local origin of the data which is both written to the IFF Map Descriptor and is subtracted from the coordinates. This subtraction of the offset is performed after the scaling is applied from the /MULTIPLY qualifier. Therefore, if the coordinates are in kilometres and /MULTIPLY=1000 is used to get this into metres, then the offset should be applied in metres eg /OFFSET=(400000.0,100000.0).

/[NO]PARAMETER=file-spec

- specifies the parameter file, which defines the mapping of MAPDATA feature codes into IFF feature codes and is used in conjunction with the FRT file. For a description of the parameter file, see the DATA PREPARATION section.

/[NO]PRIORITY_AC = (CODE:integer)

/[NO]PRIORITY_AC = (NAME:string)

- specifies the number or name of the attribute code used to hold the per-point attribute for the priority flags. This AC can be specified using the code or name as defined in the ACD section of the FRT. If not specified, no priority flags will be present in the output file.

/SCALE = real

- specifies the scale of the output IFF file which is set in the IFF Map Descriptor (MD) entry.

/[NO]TEXT_AC = (CODE:integer)

/[NO]TEXT_AC = (NAME:string)

- specifies the number or name of the attribute code used to hold the pre-feature attribute which may be present in the MapData feature. This AC can be specified using the code or name as defined in the ACD section of the FRT. If not specified, no text attribute will be present in the output file.

RESTRICTIONS

Command line qualifiers take precedence over their corresponding keywords in the parameter file.

DESCRIPTION

Command line

The symbol MAPDATA2I is normally set up as:

```
MAPDATA2I == "$LSL$EXE:MAPDATA2I"
```

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

Parameter File

This file contains the feature code lookup table which describes the translations from MapData feature code to IFF feature code and how each feature should be interpreted. It is important that this file and the FRT are consistent and accurate. The format of the table is described in the DATA PREPARATION section under the heading `The MAPDATA2I parameter file', and it will probably have been set up by the LAMPS system manager and remain relatively static at most installations.

Program Action

After decoding and checking the command line qualifiers and parameters, the program reads and validates the parameter file.

The IFF file is now created containing the IFF Range (RA), History (HI), Map Header (MH), Map Descriptor (MD), New Section (NS), Cubic Coefficients (CC) and Corner Point (CP) entries.

The MapData file is opened and each feature in turn is read and converted into a corresponding IFF feature with the interpretation as defined in the parameter file. The feature code is assigned to the output IFF feature based on the MapData code if defined in the parameter file or by using the default.

If the /TEXT_AC qualifier was used correctly, any MapData text attribute present on the feature will be written to the optional text string of the IFF AC defined.

Similarly, if /CURVE_AC or /PRIORITY_AC is used, then any curve or priority flags present on individual coordinate points will be written to the corresponding points in the IFF coordinate string (the CB entry). Any curve flags will be held as the string `C ' in a CB attribute. High and low priority flags will be stored as `H ' and `L ' respectively and interpolated (or inserted) points will be stored as `I '. It is therefore best that the ACs used to hold these attributes are defined in the ACD section as being of type character.

The interpretation of the feature can be one of four possible types:

- o Any curve flags on the MapData feature are ignored and the IFF feature uses the graphical type appropriate to the IFF feature code. This action would be taken when there is no keyword in the parameter file.

- o The feature is predominantly straight and any curved sections will be interpolated (smoothed) by having additional points stored in the feature. The output graphical type for the feature should be linear (GT 1) and the parameter file keyword should be `STRAIGHT'. The degree of smoothing is controlled by the tolerances of qualifiers /CURVE_SEPARATION, /CURVE_CHORD and /CURVE_DEVIATION or their corresponding parameter file keywords.

- o The feature is predominantly curved with a few straight sections within which are maintained by inserting new points along the section to hold the shape. These features should have a curve graphical type (GT 6) and the parameter file keyword should be `CURVE'.

- o The MapData feature will be broken into two types of feature when there is a change between the curve and straight sections. This will require that two feature codes are present in the parameter file for the two types of section. The parameter file should contain the IFF straight FC, the MapData code, the keyword `BREAK' and the IFF curve FC.

These options are specified in the parameter file as shown in the DATA PREPARATION section.

Rotatable symbols in MapData are held with two points, the first being the locating point and the second giving its orientation (from the first). If a MapData feature code maps onto an IFF feature code with a graphical type of 8 (GT 8 is a rotated symbol), then the IFF feature will have the single locating point and the corresponding rotation written to IFF RO entry of the feature.

Finally, when all features have been processed, the range (RA) entry information in the IFF file is updated to reflect all points within the file which is then closed. The same range values are written to the IFF Corner Point (CP) entry on the basis that some information there is better the none.

EXAMPLES

```
$ MAPDATA2I/FRT=BILBO/PARAMETER=GANDALF.PAR/SCALE=200000/MULTIPLY=1000/LOG     -
/OFFSET=(400000,100000) TESTMAP.MAP TESTMAP.IFF

%MAPDATA2I-I-OPNFRT, FRT file BILBO.FRT opened for read
%MAPDATA2I-I-OPNPAR, PARAMETER file GANDALF.PAR opened for read
%MAPDATA2I-W-FEATNOTCURV, Curved feature with FC 20.11 not graphical type 6
%MAPDATA2I-W-FEATNOTCURV, Curved feature with FC 20.30 not graphical type 6
%MAPDATA2I-W-FEATNOTCURV, Curved feature with FC 20.51 not graphical type 6
%MAPDATA2I-W-FEATNOTCURV, Curved feature with FC 20.80 not graphical type 6
%MAPDATA2I-W-FEATNOTCURV, Curved feature with FC 24.11 not graphical type 6
%MAPDATA2I-W-FEATNOTCURV, Curved feature with FC 30.104 not graphical type 6
%MAPDATA2I-I-FCFND, 32 feature codes found in parameter file
%MAPDATA2I-I-IFFOPN, Created output IFF file LSL$IF:TESTMAP.IFF
%MAPDATA2I-W-FCDEFAULT, default FC used for feature code 20.1 at line 5
%MAPDATA2I-W-FCDEFAULT, default FC used for feature code 20.1 at line 14
%MAPDATA2I-W-FCDEFAULT, default FC used for feature code 20.1 at line 23
%MAPDATA2I-W-FCDEFAULT, default FC used for feature code 20.1 at line 32
%MAPDATA2I-W-FCDEFAULT, default FC used for feature code 20.1 at line 41
 ELAPSED:    0 00:00:01.89  CPU: 0:00:00.66  BUFIO: 29  DIRIO: 27  FAULTS: 352
```

This example shows a conversion successfully taking place but with a few warnings to alert the user that various things may not have been set up correctly. The FEATNOTCURV warning occurred because the reported feature codes were present in the parameter file as graphical type 1 rather than 6 when the the interpretation of these features was intended to be as curves. The other warning FCDEFAULT reports each occurrence of a feature with MapData feature code 20.1 has been given the default feature code because it was not present in the parameter file. If the user is not concerned about the reported problems then the warnings can be ignored since a valid IFF file was produced.

```
$ MAPDATA2I/FRT=BILBO/PARAMETER=GOLLUM.PAR/SCALE=200000/MULTIPLY=1000/LOG      -
/OFFSET=(400000,100000)/TEXT_AC=(NAME:Text) TESTMAP.MAP TESTMAP.IFF

%MAPDATA2I-I-OPNFRT, FRT file BILBO.FRT opened for read
%MAPDATA2I-I-OPNPAR, PARAMETER file GOLLUM.PAR opened for read
%MAPDATA2I-I-FCFND, 660 feature codes found in parameter file
%MAPDATA2I-I-IFFOPN, Created output IFF file LSL$IF:TESTMAP.IFF
 ELAPSED:    0 00:00:06.44  CPU: 0:00:03.25  BUFIO: 22  DIRIO: 65  FAULTS: 398
```

This example shows a simple conversion successfully taking place with no problems occurring. The /TEXT_AC qualifier was used to specify (by name) that the AC to hold the feature attribute is AC 6 (which is named as `Text' in the default Laser-Scan ACD table 0). Many other qualifiers have been left off as they were present in the parameter file as keywords. This is a particularly useful mechanism when batch processing many files which all have the same values for the various tolerances.

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.

FCFND, 'integer' feature code'(s)' found in parameter file

> **Explanation:** This message informs the user that the parameter file has been successfully read, and feature codes found.
>
> **User action:** None.

IFFOPN, Created output IFF file 'file-spec'

> **Explanation:** This message informs the user that the IFF file has been successfully created and opened for write.
>
> **User action:** None.

LINEINPAR, line 'number' of parameter file

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

NOFRT, no FRT file has been specified

> **Explanation:** This message will appear in the absence of /FRT on the command line, or if /NOFRT was specified.
>
> **User action:** This will result in a IFF file containing line and symbol features that only have default feature codes. If a comprehensive mapping from MapData feature code to IFF feature code is required, the user should supply an accurate parameter file and FRT.

OPNFRT, FRT file 'filename' opened for read

> **Explanation:** This message informs the user that the FRT file has been successfully opened for read. This message will be displayed when the /LOG qualifier is selected.
>
> **User action:** None.

OPNPAR, PARAMETER file 'filename' 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.

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.

BADATTCODE, attribute code 'code' out of range ('min' to 'max')

Explanation: An invalid IFF attribute code (outside the given range) has been read either from the command line or from the parameter file. If the attribute code has been read from the parameter file, the LINEINPAR message indicating the offending line number will follow, and the program will continue to read the file, ignoring this line. Otherwise program execution will terminate.

User action: For a parameter file error, amend the attribute code on the relevant line in the file, and for a command line error, respecify the command using an appropriate attribute code. The program can then be run again.

BADFC, feature code 'fc' out of range ('min' to 'max')

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

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

BADMDCODE, invalid MAPDATA feature code 'string'

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

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

ERRFNDFC, could not find FC 'number' in FRT file

Explanation: The IFF feature code, (read from the parameter file) could not be found in the FRT file. This error may be caused either by entering an incorrect FC entry in the parameter file, or by specifying an incorrect FRT file. The program will use the relevant default feature code.

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

FCDEFAULT, default FC used for feature code 'code' at line 'number'

Explanation: The MAPDATA feature at the given line has a default feature code assigned as the MAPDATA code has not been given in the supplied parameter file.

User action: Amend the parameter file if required.

FEATNOTCURV, Curved feature with FC 'code' not graphical type 6

Explanation: The CODE entry in the parameter file defined the reported feature code to be interpreted as a predominantly curved feature and to insert control points to constraint the shape. However, the feature code given was not defined as being of graphical type 6 (interpolated curve) in the FRT.

User action: This warning is to make the user aware of possible conflicts. If this is what was intended then this message can be ignored.

FEATNOTSTR, Straight feature with FC 'code' not graphical type 1 or 12

Explanation: The CODE entry in the parameter file defined the reported feature code to be interpreted as a predominantly straight feature and to interpolate the curve points into the feature. However, the feature code given was not defined as being of graphical type 1 (linear) or 12 (area) in the FRT.

User action: This warning is to make the user aware of possible conflicts. If this is what was intended then this message can be ignored.

NOSTRING, cannot read character string

Explanation: A character string cannot be read where expected from the parameter file. A character string in this context may be an attribute name, or a MAPDATA feature code. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Edit the parameter file to include the attribute name or MAPDATA feature code where expected, and if necessary run the program again.

NOTLINEFTR, FC 'fc' is not defined as a linear feature at line 'number'

Explanation: The feature found on the reported line has more than one point and yet was not assigned a linear feature code (with Graphical Type 1 or 6). The only exception to this is when a two point feature is assigned to be a scalable symbol and this is done by mapping the MapData feature code onto and IFF feature code which has a Graphical Type of 8.

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

NOTSYMBFTR, FC 'fc' is not defined as a symbol feature at line 'number'

Explanation: The feature found on the reported line has only one point and yet was not assigned mapping onto a symbol feature (with Graphical Type 7).

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

TOOMNYDEFS, Too many parameter file definitions. Maximum is 'number'

Explanation: Each `CODE' line entry which gives a MapData feature code to IFF feature code mapping is stored in an array and this limit has been exceeded. The maximum number permitted is as shown and any definitions beyond this number will be ignored. The program will continue as normal.

User action: Please contact Laser-Scan and ask if this limit can be increased.

UNKCMD, unknown command "'command'"

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

User action: Check the reported command in the parameter file and ensure that it is defined and used correctly.

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.

BADTOLS, All smoothing tolerances are 0.0 - specify at least one

Explanation: The Akima smoothing tolerances CURVE_SEPARATION, CURVE_CHORD and CURVE_DEVIATION were all zero. At least one must be specified for the smoothing to work.

User action: Run the program again with correct values.

BADVALUE, Bad value given to qualifier or keyword command 'name'

Explanation: The value given to the reported qualifier or parameter file keyword is unreasonable.

User action: Run the program again with a sensible value.

ERRCLOPAR, error closing parameter file 'file-spec'

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

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed. Once the problem has been solved, you can run the program again.

ERROPNFRT, error reading FRT file 'filename'

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

User action: Dependent upon the associated error messages.

ERROPNPAR, error opening parameter file 'filename'

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

User action: Ascertain the cause of the failure from the messages given. There may be a simple remedy, e.g. a change in the directory specification. Alternatively, this error could be due to a more serious problem such as a system failure, in which case the System Manager should be informed. Once the problem has been solved, you can run the program again.

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

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

User action: Ascertain the cause of the failure from the messages given, and take the appropriate remedial action. If the error is due to a serious problem such as a system failure, the System Manager should be informed. Once the problem has been solved, you can run the program again.

NOCOORD, no valid coordinate entries at line 'number'

Explanation: There were no valid coordinate entries in the given MAPDATA feature.

User action: Check the MAPDATA file and correct the errors.

NOPARFC, no valid entries in parameter file

Explanation: There were no valid entity or FC entries in the given parameter file.

User action: Check the parameter file and correct the errors.

OUTBUFFULL, Coordinate buffer full - limit of 'number' reached

Explanation: The output buffer which holds the coordinates has filled up and the program will stop immediately. The current size of this buffer is as reported.

User action: Contact Laser-Scan and request if this limit can be increased.

TWOVALUES, Error in qualifier /OFFSET - two values required

Explanation: The /OFFSET qualifier requires two values, one for the x and one for the y coordinate of the origin offset. The format for this qualifier is /OFFSET=(x,y) where x and y are real numbers.

User action: When using the /OFFSET qualifier, ensure that two values are always given.

UNACREIFF, Unable to create specified IFF file 'file-spec'

Explanation: It was not possible to open the IFF file as specified. Further information should be obtainable from the adjoining system message.

User action: Repeat the specification after taking notice of the IFFLIB error messages.

UNAOPN, Unable to open MAPDATA file 'file-spec'

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

User action: Ascertain the cause of the failure from the messages given. There may be a simple remedy, e.g. a change in the directory specification. Alternatively, this error could be due to a more serious problem such as a system failure, in which case the System Manager should be informed. Once the problem has been solved, you can run the program again.

UNKATTCODE, cannot find attribute code 'code'

Explanation: An IFF attribute code specified either on the command line or in the parameter file cannot be found in the ACD tables. The message immediately preceding this one will give further information as to why the code was not recognised. Program execution will terminate.

User action: If the attribute code was specified on the command line, respecify the command line using the correct code. If the attribute code was incorrectly specified in the parameter file, amend the attribute code in the file. Run the program again.

UNKATTNAME, cannot find attribute name 'name'

Explanation: An IFF attribute name specified either on the command line or in the parameter file cannot be found in the ACD tables. The message immediately preceding this one will give further information as to why the name was not recognised, and the NOFRT message may follow. Program execution will terminate.

User action: If the NOFRT message indicates that an FRT file is required, respecify the command line with the /FRT qualifier. If the /FRT qualifier is used, ensure that the appropriate ACD definition appears in the FRT file. If the attribute name was specified on the command line, respecify the command line using the correct name. If the attribute name was incorrectly specified in the parameter file, amend the attribute name in the file. Run the program again.

MESSAGES (OTHER)

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