Laser-Scan Ltd.

CONVERT PACKAGE

IFFMSDS Reference

Issue 1.0 - 17-June-1991

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

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

CONTENTS

IFFMSDS - Change Record i

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

CHAPTER 1 MSDS FORMAT DESCRIPTION

FORMAT DESCRIPTION 1-1
Introduction 1-1
MSDS data 1-1
Comparison of MSDS and IFF 1-2

CHAPTER 2 MSDS2I UTILITY

UTILITY MSDS2I 2-1
FUNCTION 2-1
FORMAT 2-1
PROMPTS 2-1
PARAMETERS 2-1
COMMAND QUALIFIERS 2-2
DESCRIPTION 2-3
MSDS2I and logical names 2-4
EXAMPLE 2-4
MESSAGES (INFORMATIONAL) 2-6
MESSAGES (WARNING) 2-7
MESSAGES (ERROR) 2-8
MESSAGES (FATAL) 2-11
MESSAGES (OTHER) 2-12

IFFMSDS - Change Record

Version 1.0 S Townrow 17-June-1991

Module IFFMSDS - 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

MSDS FORMAT DESCRIPTION

FORMAT DESCRIPTION

Introduction

The Military Survey Data Structure (MSDS) exchange format is a UK interim format for the exchange of topologically structured vector digital data. It will be replaced when the NATO Digital Geographic Information Working Group (DGIWG) standards are complete.

The MSDS format expected is an ASCII text format, as described to Laser-Scan in the document "MIL SVY DATA STRUCTURE (MSDS) EXCHANGE FORMAT".

The IFF files produced are intended for processing directly to produce graphic output. For more information on the IFF file structure see the **IFF User's Guide** of the Laser-Scan MAPPING package documentation.

MSDS data

MSDS data comes in an ASCII file. This will usually be a ANSI file on magnetic tape. The record length is variable, with a maximum length of 4096 bytes. Each input MSDS file will be output to one IFF file.

MSDS consists of topologically structured data gathered into "THEMES" or layers. The data within a THEME is divided into two levels, the feature level and the spatial level. The feature level combines the resulting spatial entities with attribute data to represent a real world geographic entity.

There are three spatial entities, which each relate directly to a feature type. These are:

1. Node - a zero-dimensional entity, representing a single significant coordinate. Nodes are related directly to Point features.

2. EDGE - a one-dimensional entity representing a set of coordinates. Line features consist of one or more connected edges. Edges also bound the third spatial entity:

3. FACE - a two-dimensional entity representing the largest connected area, enclosed by oriented edges. Area features consist of one or more faces that share a common boundary.

The simple features (points, lines and areas) can be combined, with further attribute data to form complex features.

The coordinates in MSDS are stored as integer coordinates, usually as centimetres on the projection. X,Y and Z values are stored for each point.

Comparison of MSDS and IFF

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

MSDS format shares this concept of a 'feature', at the simple feature level. Each simple feature has a name that can be related to an IFF feature code. This translation is effected by a simple lookup table that relates the MSDS feature names with IFF feature codes. Complex features can be defined in MSDS in terms of simple features, but are ignored in the conversion to IFF.

MSDS has several concepts which have no direct equivalence in unstructured feature-based IFF files. The MSDS2I program is intended for producing IFF files for graphic display and hardcopy output, and hence does not attempt to preserve such MSDS properties as shared geometry or complex features. It would be possible to convert the concept of THEMEs into LAYERs in IFF, but at the moment all features are simply written to layer 1 in the IFF file.

CHAPTER 2

MSDS2I UTILITY

UTILITY MSDS2I

FUNCTION

MSDS2I is a MSDS (**M**ilitary **S**urvey **D**ata **E**xchange) format to IFF converter. It produces a standard IFF file intended for graphics display or hardcopy plotting.

FORMAT

```
$ MSDS2I input-file-spec output-file-spec
```

Command qualifiers	**Defaults**
/DIVIDE_BY = integer	/DIVIDE_BY = 100
/[NO]LOG	/NOLOG
/[NO]ORIGIN = (xval,yval)	/NOORIGIN
/SCALE = integer	/SCALE = 50000
/TRANSLATION = file-spec	/TRANSLATION=LSL$LOOKUP:MSDS.TRAN

PROMPTS

```
Input MSDS filename :   input-file-spec
Output IFF filename :   output-file-spec
```

PARAMETERS

input-file-spec

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

output-file-spec

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

COMMAND QUALIFIERS

/DIVIDE_BY
/DIVIDE_BY = 100 (default)

- defines the ratio of the size of the units in the MSDS file and the IFF file. By default the coordinates in the MSDS file represent centimetres on the projection and the coordinates in the IFF file represent metres on the projection.

/LOG
/NOLOG (default)

- if specified, progress of the conversion is output to the screen. This qualifier is recommended if the program is being run interactively.

/ORIGIN = (xval,yval)
/NOORIGIN (default)

- specify the origin of the output coordinate system **in MSDS coordinates.**
  This qualifier allows the user to specify the origin of the coordinate system to be used in the IFF file. By default the south west corner of the coordinates in the MSDS geometrical features (edges and nodes) will be used. See IFFLIB documentation for more information about origin offsets in IFF files.

/SCALE
/SCALE = 50000 (default)

- selects the scale of the output file to be stored in the map descriptor. See IFFLIB documentation for more information about the scale factor in the map descriptor of IFF files.

/TRANSLATION
/TRANSLATION = file-spec

- specifies the file to be used for the translation between MSDS feature names and IFF feature codes.

  This is an ASCII file. Blank lines, and lines starting with "!" are ignored. Other lines should contain the MSDS feature name (maximum of 12 characters) followed by an integer representing the IFF feature code. A short example file follows:

```
!
! file to associate MSDS names with IFF feature codes
!
! Name                code            comment ( if required)
!=====================================

AAAAAAAA      100
BBBBBBBB      101
CCCCCCCC      102
DDDDDDDD      103

! the end
```

DESCRIPTION

MSDS2I converts MSDS files into IFF format. The input is a standard ASCII file, which may be a disk file or an ANSI file on magnetic tape.

NOTE

> The MSDS format allows files with records up to the block size of 4096 bytes. While the program MSDS2I is happy to deal with these records some difficulty has been experienced in transferring files with records longer than 2048 bytes from tape to disk. The VMS utilities COPY and CONVERT will not deal with such files. Indeed such files cannot be "TYPED".

In converting MSDS data to IFF, the hierarchical nature of the data is lost, and the THEME and COMPLEX FEATURE levels of abstraction are ignored.

The SIMPLE features (areas, lines and points) of the MSDS data are written into one IFF layer with a feature code based on the feature name. The relationship between the MSDS feature name and the IFF feature code is contained in a translation file. The default name for this file is LSL$LOOKUP:MSDS.TRAN, but this default may be overridden by the /TRANSLATION switch. An example file is supplied in LSL$PUBLIC_ROOT:[CONVERT.LOOKUP] along with corresponding FRT and SRI files (MSDS.FRT and MSDS.SRI). The TRI file is immaterial as MSDS does not support text features.

If the program fails to find the MSDS names of the simple features in the translation file, then these features are not transferred to the IFF file. The number of features so ignored are listed as a warning message. This mechanism may be used as a method of selecting which MSDS features to convert.

Attributes associated with the MSDS features, and their values, are stored in the text part of type 6 ACs in IFF features. This is done with no interpretation of the attribute codes or their values.

MSDS2I and logical names

MSDS2I requires that the following logical names are set up:

1. LSL$MSDS: - this should point to the directory (or device) that contains the MSDS file to be converted. In particular this may, conveniently point to the magnetic tape drive. (e.g. MUA0:)

2. LSL$LOOKUP: - this should point to the directory containing the MSDS translation file which is to be used for converting MSDS feature names to IFF feature codes.

3. LSL$IF: - this should be set up to point to a directory to which the IFF file is to be written.

Any of these three logical names may be overridden by explicit specification of the directory concerned.

EXAMPLE

```
$ msds2i here:msds msds/log
%MSDS2I-I-TRANSOPEN, MSDS translation file "LSL$LOOKUP:MSDS.TRAN" opened for
read
%MSDS2I-I-MSDSOPEN, MSDS input file "HERE:MSDS.NFF" opened for read
%LSLLIB-I-IFFOPENED, LSL$SOURCEROOT:[CONVERT.MSDS2I]MSDS.IFF;95 opened for write

Scanning msds file "HERE:MSDS.NFF"...
Dataset contains 635 features, 3240 topological entities
------------------------------------------------------------------------
Entity ID =     1, type = complex feature
Entity ID =   137, type = simple area feature
Entity ID =   314, type = simple line feature
Entity ID =   580, type = simple point feature
Entity ID =   636, type = face
Entity ID =  1264, type = edge
Entity ID =  2855, type = node
%MSDS2I-I-MSDSREWOUND, MSDS input file "HERE:MSDS.NFF" rewound

Reading msds file "HERE:MSDS.NFF"....
Dataset contains 635 features, 3240 topological entities
------------------------------------------------------------------------
Entity ID =     1, type = complex feature
Entity ID =   137, type = simple area feature
Entity ID =   314, type = simple line feature
Entity ID =   580, type = simple point feature
Entity ID =   636, type = face
Entity ID =  1264, type = edge
Entity ID =  2855, type = node
------------------------------------------------------------------------
Outputting data to IFF
Origin is 55695698,577254556,0
Scale is 1:50000
Coordinates will be divided by 100
------------------------------------------------------------------------
```

```
Writing area features
------------------------------------------------------------------------
Writing line features
------------------------------------------------------------------------
Writing point features
------------------------------------------------------------------------
 ELAPSED:    0 00:05:28.37  CPU: 0:03:01.63  BUFIO: 66  DIRIO: 1399  FAULTS:
3029
$
```

This example demonstrates the use of MSDS2I. MSDS file HERE:MSDS.NFF (A disk file) has been processed, using the default translation file LSL$LOOKUP:MSDS.TRAN, to produce an IFF file LSL$IF:MSDS.IFF. As the /LOG qualifier has been given, the various stages the program goes through are recorded.

```
$ msds2i here:msds msds/trans=msdsshort

%MSDS2I-W-IGNAMES, 49 features with names not found in translation file ignored
..   27 feature(s) with name "GROUNDSURA"
..   22 feature(s) with name "PYLONP"
 ELAPSED:    0 00:05:06.76  CPU: 0:02:49.21  BUFIO: 29  DIRIO: 1439  FAULTS:
2333
```

This example demonstrates the use of MSDS2I with a translation table that does not contain all the feature names that exist in the input data. The MSDS file has been successfully converted, with a warning of the features that were not processed.

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.

MSDSOPEN, MSDS input file "'filename'" opened for read

Explanation: If the user specifies the /LOG qualifier, then the program will confirm that it has opened the input file with this message.

User action: None.

MSDSREWOUND, MSDS input file "'filename'" rewound

Explanation: If the user specifies the /LOG qualifier, then the program will confirm that it has rewound the input file with this message.

User action: None.

TRANSOPEN, MSDS translation file "'filename'" opened for read

Explanation: If the user specifies the /LOG qualifier, then the program will confirm that it has opened the translation file with this message.

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.

ATTLEN, attribute value length 'integer' is greater than stated maximum 'integer'

Explanation: The program has read an attribute value which is longer than the stated maximum attribute value length for that field (this maximum length follows the value in the dataset). The full length of the attribute value will be used. The program will also output the offending string to the terminal.

User action: Report the problem to the supplier of the data.

CLOSERR, error closing file "'filename'"

Explanation: The program has found some problem in closing the specified file. The accompanying LSLLIB messages should explain the nature of the error.

User action: Dependent upon the associated error messages.

IGNAMES, 'integer' features with names not found in translation file ignored

Explanation: No translation was possible between the name of these features and a feature code in the translation file. This may be because the operator did not required these features output to IFF.

User action: If these features were required, edit the translation file to include a translation for these names and re-run the program

STRTOOLNG, string too long in record 'integer' - more than 'integer' characters

Explanation: The program has attempted to read a field as a string, but the field is longer than the space allocated for it by the program. This error message is produced, followed by the truncated string, and the truncated string is used by the program.

User action: If the fields is actually not too long, consult Laser-Scan to have the program altered. Otherwise, report the problem to the supplier of the data.

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.

COMMA, failed to find expected comma separator in record 'integer'

Explanation: Fields in MSDS are separated by commas. This error is given if the program has tried to read a separator, and instead found some other character.

User action: Inspect the data, and report the error to the data supplier.

FTABLE, error reading feature code translation table "'filename'"

Explanation: An error has occurred while reading the translation table. The associated error messages should clarify the problem.

User action: Dependent upon the associated error messages.

INFILE, error opening input MSDS file "'filename'"

Explanation: The program has failed to open the requested input file. The associated error messages should clarify the problem.

User action: Dependent upon the associated error messages.

ISOLATE, node isolation error - isolation flag='integer', pointer count='integer'

Explanation: In an MSDS node record, if the isolation flag is zero then the pointer count must be one or more, whereas if it is non-zero, then the pointer count must be zero or greater. This error is given if those conditions are not met.

User action: Report the problem to the data supplier.

MISSEDGE, cannot find oriented edge 'integer' in assembling current line

Explanation: When outputting a line feature, the program looks up all of the oriented edges composing each line in the list of edges for the dataset. This message means that the current line feature claims to contain an oriented edge that does not exist.

User action: Report the problem to the supplier of the data.

MISSNODE, cannot find node 'integer' in assembling current point

Explanation: When outputting a point feature, the program looks up all of the nodes composing that point feature. This message means that the current point feature claims to contain a node that does not exist.

User action: Report the problem to the supplier of the data.

MSDSREWIND, failed to rewind MSDS input file "'filename'"

Explanation: The program scans the input file to find the amount of space required to store the data in it. It then rewinds the input file and reads the data. This message means that this rewind has failed. The associated messages should clarify the reason for the error.

User action: Dependent upon the associated error messages.

NOTHEME, record 'integer' contains relation name "'string'" instead of "theme_comp_to_own"

Explanation: The second relation in a relation list should be "theme_comp_to_own", which lists the theme(s) containing this feature. In this record, that is not so.

User action: Report the problem to the supplier of the data.

ORIGIN, /ORIGIN must have 2 numbers (x,y) as argument

Explanation: The user has specified more or fewer than two coordinates to the /ORIGIN command.

User action: Repeat the command, specifying all three coordinates, or omit the origin command.

READREC, error reading record 'integer'

Explanation: The program is unable to read the specified record. The associated error messages should help to explain why.

User action: Dependent upon the associated error messages.

RELCNT, relation count is 'integer' (should be 3) in record 'integer'

Explanation: The relation count field of a feature record is defined to have the value 3. However, this is not so in this record.

User action: Report the problem to the supplier of the data.

UNEXPEOL, unexpected end-of-line in record 'integer', reading 'string'

Explanation: The program encountered the end of an MSDS record whilst trying to read the specified field or separator. The program will exit.

User action: Inspect the MSDS data, and report the error to the data supplier.

UNEXPREC, entity 'integer' is of unknown type 'integer', in record 'integer'

Explanation: The entity with the specified ID, in the specified record, has a recod type that is not in the range 0-6. The program cannot read this record.

User action: Report this problem to the data supplier.

UPDENT, error udpating 'string' entry - entry found "'string'", ielen 'integer'

Explanation: An error occurred trying to reposition back to the specified entry to update it before closing the IFF file. The file is corrupt.

User action: Investigate what is wrong with the file. This problem may reflect a corruption of the program, in which case report it to Laser-Scan with as many details as possible.

MESSAGES (FATAL)

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

ABORT, aborting

Explanation: Used within the program to communicate a severe error.

User action: None.

INTERNAL, internal error whilst 'processing'

Explanation: An unexpected condition has ocurred while processing the data.

User action: Save the input data and contact Laser-Scan

NOROOM, no room to store 'string' - more than 'integer' instances found

Explanation: Within the program, various arrays are used to store the data relating to the MSDS dataset. This message is output if one of those arrays has become full.

User action: Report this error message to Laser-Scan, noting what it is that was being stored, and request that the array be increased in size.

NOVIRTMEM, failed to allocate dynamic memory

Explanation: Within the program, memory is allocated dynamically, depending on the size of MSDS dataset. This message is output when this allocation fails.

User action: Report this error message to your system manager, along with the associated system error. He may be able to increase your virtual page count to allow this data set to be converted.

STRNGERR, error in dynamic string handling

Explanation: Within the program, strings are handled dynamically. This message is output when this handling fails.

User action: Report this error message to your system manager, along with the associated system error. He may be able to increase your virtual page count to allow this data set to be converted.

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.