

Laser-Scan Ltd.

CONVERT PACKAGE

IFFARCINFO Reference

Issue 1.4 - 23-February-1993

Copyright (C) 1993 Laser-Scan Ltd
Science Park, Milton Road, Cambridge, England CB4 4FY tel: (01223) 420414

Document "IFFARCINFO REFERENCE", Category "REFERENCE"
Document Issue 1.4 S Townrow (modified 23-Feb-1993)

CONTENTS

| | | |
|-----------|--|------|
| | IFFARCINFO - Change Record | i |
| | PREFACE | 1 |
| | Intended audience | 1 |
| | Structure of this document | 1 |
| | Conventions used in this document | 2 |
| CHAPTER 1 | ARC/INFO FORMAT DESCRIPTION | |
| | FORMAT DESCRIPTION | 1-1 |
| | The ARC/INFO EXPORT Format | 1-1 |
| | Example EXPORT File | 1-10 |
| CHAPTER 2 | ARC/INFO DATA PREPARATION | |
| | DATA PREPARATION | 2-1 |
| | Comparison of ARC/INFO and IFF | 2-1 |
| | Feature Translation | 2-1 |
| | Preparation for data to be transferred between EXPORT and IFF | 2-2 |
| | The ARC2I parameter file | 2-3 |
| | The default feature codes | 2-3 |
| | The attribute description table | 2-3 |
| | The annotation feature codes | 2-5 |
| | The I2ARC parameter file | 2-6 |
| | The attribute description table | 2-6 |
| CHAPTER 3 | I2ARC UTILITY | |
| | UTILITY I2ARC | 3-1 |
| | FUNCTION | 3-1 |
| | FORMAT | 3-1 |
| | PROMPT | 3-1 |
| | PARAMETERS | 3-1 |
| | COMMAND QUALIFIERS | 3-2 |
| | RESTRICTIONS | 3-3 |
| | DESCRIPTION | 3-3 |
| | Command line | 3-3 |
| | PROGRAM ACTION | 3-4 |
| | EXAMPLES | 3-7 |
| | MESSAGES (SUCCESS) | 3-9 |
| | MESSAGES (INFORMATIONAL) | 3-10 |
| | MESSAGES (WARNING) | 3-11 |
| | MESSAGES (ERROR) | 3-13 |
| | MESSAGES (OTHER) | 3-15 |
| CHAPTER 4 | ARC2I UTILITY | |
| | UTILITY ARC2I | 4-1 |
| | FUNCTION | 4-1 |

| | |
|------------------------------------|------|
| FORMAT | 4-1 |
| PROMPT | 4-1 |
| PARAMETERS | 4-1 |
| COMMAND QUALIFIERS | 4-2 |
| RESTRICTIONS | 4-5 |
| DESCRIPTION | 4-7 |
| Command line | 4-7 |
| Parameter File | 4-7 |
| Program Action | 4-7 |
| EXAMPLES | 4-8 |
| MESSAGES (INFORMATIONAL) | 4-11 |
| MESSAGES (WARNING) | 4-12 |
| MESSAGES (ERROR) | 4-17 |
| MESSAGES (OTHER) | 4-22 |

IFFARCINFO - Change Record

Version 1.0 S Townrow 04-April-1991

- Module IFFARCINFO - Addition of a /VERBOSE qualifier to ARC2I to provide messages describing the success/failure of assigning attributes to their respective arcs or points.
- New messages NOARCATT and NOPNTATT added to ARC2I which report the failure to assign any attributes to a given point or arc.

Version 1.1 S Townrow 31-May-1991

- Module IFFARCINFO - New ARC2I qualifiers /BLOCKSIZE, /ID_TO_FC and /REWIND added which are reflected in the documentation.

Version 1.2 S Townrow 17-June-1991

- Module IFFARCINFO - Reorganised package documentation.

Version 1.3 S Townrow 23-July-1991

- Module IFFARCINFO - New message ARC2I__MAXFSN added to ARC2I messages section.

Version 1.4 S Townrow 23-Feb-1993

- Module IFFARCINFO - The FORMAT DESCRIPTION section has been modified to include a description of ARC/INFO version 6.0 text annotation.
- The DATA PREPARATION section had been expanded to include a description of how to set up a new annotation feature code section in the parameter file.
 - The ARC2I reference manual now documents the level of support for ARC/INFO version 6.0 annotation and to include the new qualifiers /ARC_ID and /POINT_ID.

- Three new messages, BADFCODE, DEFAULTFC and MISSCMD have been added to the ARC2I message section.

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 assignment 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

ARC/INFO FORMAT DESCRIPTION

FORMAT DESCRIPTION

The ARC/INFO EXPORT Format

This section describes parts of the EXPORT files created by the ARC/INFO system which are read by ARC2I and also some file parts created by the I2ARC utility.

The EXPORT files being discussed are in uncompressed ASCII form which is not the standard representation. By default, they are produced by the ARC/INFO system in compressed form which is unreadable. This can be overridden using the ARC/INFO TAPEWRITE facility by specifying the compression argument as 'NONE' which should be the last thing on the line. This will produce an ASCII file with the extension '.E00'. Continuation files (.E01, .E02 etc) can be produced using TAPEWRITE, but these are not supported and no attempt to process them should be made.

The EXPORT format maintains three basic feature structures, namely linear features, symbol features and text features, each having their own section in the EXPORT file called ARC, LAB and TXT (or TX6) respectively. In addition to these, there is another information section (IFO) containing information relevant to the whole EXPORT file. This includes tables holding attribute information associated with features from the ARC and LAB sections.

All features within the EXPORT file are referenced by an internal ID number unique within the dataset, and which are updated sequentially on reading the IFF file.

An example EXPORT file can be found at the end of this section which contains examples of all the relevant sections.

An EXPORT file will begin with a dataset 'header' of the form:

EXP 0

to be followed by feature coordinate information grouped into the three basic feature type sections, each section having its own one line header of ARC 2, LAB 2, TXT 2, or TX6 2 designating the relevant section. These headers, with following value of 2, denote that the real coordinates within the file are of single precision and have a field size of 14. ARC2I will also process files containing double precision reals which will have header lines ARC 3, LAB 3, TXT 3 and TX6 3 and a field size of 24.

All linear feature coordinates follow in the ARC section, with the format:

```
ARC 2
      I1          0          0          0          0          0          N1
x          y          x          y
x          y          x          y
x          y          x          y
x          y
```

for each feature in turn, consisting of an ID line and coordinates,

```
      In          0          0          0          0          0          Nn
x          y          x          y
-1          0          0          0          0          0
```

where I1, In etc. are the internal ID numbers of the feature to which the subsequent coordinates belong, and N1, Nn etc. are the numbers of coordinate x, y pairs to follow. Only the x and y values are stored in the EXPORT file, and no three dimensional z information is present.

The maximum number of points for a feature is 500 and any more would have been carried over and a new feature created.

The x and y coordinates are output in exponential format: 1.000000E+02. The end of all sections in the EXPORT file are signified by the -1 ... line, which must be present even in the absence of any x, y coordinates:

```
ARC 2
      -1          0          0          0          0          0          0
```

The next section of an EXPORT file contains all symbol feature coordinate information in the LAB section, in the format:

```
LAB 2
      U1          I1 x          y
x          y          x          y
```

for each feature in turn, consisting of an ID line and coordinates,

```
      Un          In x          y
x          y          x          y
-1          0 0.0000000E+00 0.0000000E+00
```

where U1, Un are the User-ID numbers and I1, In are the internal ID numbers of the feature to which these coordinates belong. After the LAB header, the coordinates for each symbol are duplicated twice on the next line.

The x and y coordinates are output in exponential format: 1.000000E+02. In the absence of any symbol features, the EXPORT file must still contain:

```
LAB 2
      -1          0 0.0000000E+00 0.0000000E+00
```

The third and last major coordinate section of an EXPORT file contains all text feature information in the TXT section in one of two forms. Below version 6.0 of ARC/INFO the text section has the format:

```
TXT 2
      N          2          0          S          len
      x1          x2          x3          x4          y1
      y2          y3          y4          0.0000000E+00 0.0000000E+00
      0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 sz
      -1.0000000E+02
      text
```

for each feature in turn, consisting of a header line, coordinates and text,

```

      N          2          0          S          len
      x1          x2          x3          x4          y1
      y2          y3          y4          0.0000000E+00 0.0000000E+00
      0.0000000E+00 0.0000000E+00 0.0000000E+00 0.0000000E+00 sz
      -1.0000000E+02
      text
      -1          0          0          0          0          0          0
```

where N is the level number to which the feature belonged, S is the EXPORT symbol number of the feature, and len is the number of characters in the text string. Values x1, y1, x2, y2, x3, y3, x4 and y4 may not all be present although x1 and y1 must. They define the point, line, arc, or spline along which the text feature lies depending on the number of points present. Value sz is the text height and the others can be ignored. The text string is output in the 'text' line.

The x and y coordinates are output in exponential format: 1.000000E+02. In the absence of any text features, the EXPORT file must still contain:

```
TXT 2
      -1          0          0          0          0          0          0
```

ARC/INFO version 6.0 and above writes EXPORTS files containing the text in a more flexible format. This includes banked (composite) texts and splines containing many more points. Text attributes are can now be created which were not available before this version. However, these attributes are not currently supported and text attributes in the TAT table will be ignored.

The format of the version 6.0 text section is as follows:

```
TX6  2
<sub-class>
      0          N      npts      0          S      ncomp      len
      0          0          0      0          0          0          0
      0          0          0      0          0          0          0
      0          0          0      0          0          0          0
      c1          c2      .      .      c(ncomp)      0          0
      0          0          0      0          0          0          0
      0          0          0      0          0          0          0
0.0000000E+00
ht 0.0000000E+00 0.0000000E+00
x1          y1
x2          y2
..          ..
x(npts-1)   y(npts-1)
x(npts)     y(npts)
text
      -1          0          0          0          0          0          0
```

where N is the level number to which the feature belonged, npts is the number of points in the spline around which the text is bent, ncomp is the number of text components, S is the EXPORT symbol number of the feature, and len is the number of characters in the text string.

The range of values c1 to c(ncomp) denote the starting point in the list of coordinates which follow for each text component and the values should not exceed npts. The value ht is the text and the text string is output in the 'text' line.

In the absence of any text features, the EXPORT file must still contain:

```
TX6  2
      -1          0          0          0          0          0          0
```

After all this feature coordinate information must follow information on the dataset as a whole, in the IFO section, together with some formatting instructions.

After the IFO section header line there will be several tables, possibly in any order but usually alphabetical. However, the simplest will be discussed first. The range information which denotes the maximum and minimum values of all data points within the EXPORT file is held in the .BND table. This is as follows:

```

IFO 2
filename.BND                XX    4    4    16                1
XMIN                      4-1    14-1  12 3 60-1  -1  -1-1                1
YMIN                      4-1    54-1  12 3 60-1  -1  -1-1                2
XMAX                      4-1    94-1  12 3 60-1  -1  -1-1                3
YMAX                      4-1   134-1  12 3 60-1  -1  -1-1                4
  xmin                ymin                xmax                ymax
  
```

where xmin, xmax, ymin, ymax are the range values output in exponential format.

Next, the corner point values are held in the .TIC table and may contain more than four points. In this case, a polygon has been described which surrounds all data points within the EXPORT file. The format of the .TIC table is:

```

filename.TIC                XX    3    3    12                N
IDTIC                      4-1    14-1  5-1 50-1  -1  -1-1                1
XTIC                      4-1    54-1  12 3 60-1  -1  -1-1                2
YTIC                      4-1    94-1  12 3 60-1  -1  -1-1                3
    1 x                    y
    2 x                    y
    3 x                    y
    4 x                    y
      .                    .
      .                    .
    N x                    y
  
```

where x, y are the point values output in exponential format and N is the number of points in the polygon.

The next table, the .FSN, is not a recognised ARC/INFO table and is only produced from I2ARC. The .FSN table cross references the internal EXPORT ID feature numbers to the IFF FSN, FC and overlay numbers, in the format:

```

filename.FSN                XX    4    4    16                M
filename-ID                4-1    14-1  5-1 50-1  -1  -1-1                1
FSN                      4-1    54-1  5-1 50-1  -1  -1-1                2
FC                      4-1    94-1  5-1 50-1  -1  -1-1                3
LAYER                    4-1   134-1  5-1 50-1  -1  -1-1                4
    id                fsn                fc                layer
    id                fsn                fc                layer
  
```

for each feature in the EXPORT dataset.

where M is the maximum ID number in the dataset, ie. the total number of features. Note that some IFF features will have been broken up

into several EXPORT features with distinct ID numbers (eg. TS text component strings).

The .LUT table again is not a recognised ARC/INFO table and is produced by I2ARC. This IFF feature code to EXPORT symbol number conversion table looks like:

| | | | | | | | | | |
|--------------|-----|--------|-----|------|----|------|--|---|---|
| filename.LUT | | | | 2 | 2 | 8 | | N | |
| FC | 4-1 | 14-1 | 5-1 | 50-1 | -1 | -1-1 | | | 1 |
| SYMBOL | 4-1 | 54-1 | 5-1 | 50-1 | -1 | -1-1 | | | 2 |
| | fc | symbol | | | | | | | |
| | fc | symbol | | | | | | | |

for each FC used in the IFF file.

where N is the total number of FCs used in the IFF file.

The next table to be described is the Arc Attribute Table (.AAT) which always contains at least 7 basic entries to describe the arcs in the ARC section. This is a standard ARC/INFO table and can be read by ARC2I using the parameter file mechanism described in the next section. The 7 lines following the .AAT line should always be present. Any additional attributes are appended to the end of the table.

| | | | | | | | | | |
|--------------|-----|-------|----|----|------|-----|------|--|------|
| filename.AAT | | | | XX | N+7 | N+7 | 30 | | E |
| FNODE | 4-1 | 14-1 | 5 | 0 | i0-1 | -1 | -1-1 | | 1- |
| TNODE | 4-1 | 54-1 | 5 | 0 | i0-1 | -1 | -1-1 | | 2- |
| LPOLY | 4-1 | 94-1 | 5 | 0 | i0-1 | -1 | -1-1 | | 3- |
| RPOLY | 4-1 | 134-1 | 5 | 0 | i0-1 | -1 | -1-1 | | 4- |
| LENGTH | 4-1 | 174-1 | 12 | 3 | i0-1 | -1 | -1-1 | | 5- |
| filename | 4-1 | 214-1 | 5 | 0 | i0-1 | -1 | -1-1 | | 6- |
| filename-ID | 4-1 | 254-1 | 5 | 0 | i0-1 | -1 | -1-1 | | 7- |
| attr1 | f-1 | 294-1 | 2 | 0 | i0-1 | -1 | -1-1 | | 8- |
| . | | | | | | | | | |
| . | | | | | | | | | |
| . | | | | | | | | | |
| attrN | f-1 | 294-1 | 2 | 0 | i0-1 | -1 | -1-1 | | N+7- |

where N is the number of additional attributes in the table itself and E is the number of attribute entries following the table. 'i' is the item datatype value and lies in the range 1 to 6. Associated with this is the item field width, 'f', which is only used if 'i' is 2 or 3.

Here is a summary of the EXPORT item datatypes:

| Datatype | Width | Description |
|----------|-------|---|
| 1 | 8 | date held in 8 characters |
| 2 | f | character type. Width 'f' taken from table |
| 3 | f | integer. Width 'f' taken from table |
| 4 | 14 | floating point number stored in width of 14 |
| 5 | 11 | binary integer stored in width of 11 |
| 6 | 14 | floating point number stored in width of 14 |

Following the .AAT table are the actual attribute data entries and are as many as there are arcs in the ARC section. The entries consist of the 7 basic attributes and any additional ones are appended one after the other in field widths described above. An example of this is given at the end of the section.

The next table to be described is a standard ARC/INFO table and is the Point/Polygon Attribute Table (.PAT) which always contains at least 4 basic attribute entries to describe the points in the LAB section. If the area and perimeter in the table are set to zero, then the .PAT is actually a point table, otherwise it is a polygon table. However, ARC2I does not maintain any link/node structure and so any attributes in the .PAT are assigned to the LAB points irrespective of whether the table is really a point or polygon one.

These 4 lines following the .PAT line should always be present. Any additional attributes are appended to the end of the table.

| | | | | | | | | | |
|--------------|-----|-------|------|------|-----|------|--|---|------|
| filename.PAT | | | XX | N+4 | N+4 | 30 | | E | |
| AREA | 4-1 | 14-1 | 12 3 | i0-1 | -1 | -1-1 | | | 1- |
| PERIMETER | 4-1 | 54-1 | 12 3 | i0-1 | -1 | -1-1 | | | 2- |
| filename | 4-1 | 94-1 | 5 0 | i0-1 | -1 | -1-1 | | | 3- |
| filename-ID | 4-1 | 134-1 | 5 0 | i0-1 | -1 | -1-1 | | | 4- |
| attr1 | f-1 | 294-1 | 2 0 | i0-1 | -1 | -1-1 | | | 5- |
| . | | | | | | | | | |
| . | | | | | | | | | |
| . | | | | | | | | | |
| attrN | f-1 | 294-1 | 2 0 | i0-1 | -1 | -1-1 | | | N+4- |

where N is the number of additional attributes in the table itself and E is the number of attribute entries following the table. 'i' is the item datatype value and lies in the range 1 to 6. Associated with this is the item field width, 'f' which is only used if 'i' is 2 or 3.

This table is very similar to the .AAT table and the datatypes and field widths are the same. However, the first entry following the .PAT table, typically contains a negative area and has an internal ID number of zero. This entry refers to the world seed point for the whole map and can be ignored.

It is important that the internal ID number held in the .PAT is the same as the internal ID number of the associated point in the LAB section in order for the attribute to be transferred.

The example at the end of this section shows typical .AAT and .PAT tables and their following attribute entries.

I2ARC uses a different mechanism to the .AAT and .PAT tables to hold attribute information this takes the form of 2 INFO tables, .IAC and .RAC, which are described below. ARC2I can read the .AAT and .PAT tables but ignores the .FSN, .LUT, .IAC and .RAC tables that are produced by I2ARC.

The .IAC holds the integer information extracted from any ACs in the IFF file, and takes the form:

| | | | | | | | | | |
|--------------|------|-------|------|-------|--------|------|--|---|---|
| filename.IAC | | | | 4 | 4 | 52 | | N | |
| filename-ID | 4-1 | 14-1 | 5-1 | 50-1 | -1 | -1-1 | | | 1 |
| TYPE | 4-1 | 54-1 | 5-1 | 50-1 | -1 | -1-1 | | | 2 |
| VALUE | 4-1 | 94-1 | 5-1 | 50-1 | -1 | -1-1 | | | 3 |
| TEXT | 40-1 | 134-1 | 40-1 | 20-1 | -1 | -1-1 | | | 4 |
| | id | type | | value | "text" | | | | |
| | id | type | | value | "text" | | | | |

for each integer AC in the dataset.

where N is the total number of integer ACs in the IFF file.

For each integer AC is given the feature ID followed by the AC type, its value (which may represent a date, time or character as well as a normal integer) and any optional text given with the AC.

In the absence of any integer ACs the EXPORT file must still contain an entry with no values:

| | | | | | | | | | |
|--------------|------|-------|------|------|----|------|--|---|---|
| filename.IAC | | | | 4 | 4 | 52 | | 0 | |
| filename-ID | 4-1 | 14-1 | 5-1 | 50-1 | -1 | -1-1 | | | 1 |
| TYPE | 4-1 | 54-1 | 5-1 | 50-1 | -1 | -1-1 | | | 2 |
| VALUE | 4-1 | 94-1 | 5-1 | 50-1 | -1 | -1-1 | | | 3 |
| TEXT | 40-1 | 134-1 | 40-1 | 20-1 | -1 | -1-1 | | | 4 |

Now the real ACs are held in the .RAC as follows:

| | | | | | | | | | |
|--------------|------|-------|------|-------|--------|------|--|---|---|
| filename.RAC | | | | 4 | 4 | 52 | | N | |
| filename-ID | 4-1 | 14-1 | 5-1 | 50-1 | -1 | -1-1 | | | 1 |
| TYPE | 4-1 | 54-1 | 5-1 | 50-1 | -1 | -1-1 | | | 2 |
| VALUE | 4-1 | 94-1 | 5-1 | 50-1 | -1 | -1-1 | | | 3 |
| TEXT | 40-1 | 134-1 | 40-1 | 20-1 | -1 | -1-1 | | | 4 |
| | id | type | | value | "text" | | | | |
| | id | type | | value | "text" | | | | |

for each real AC in the dataset.

where N is the total number of real ACs in the IFF file.

For each real AC is given the feature ID followed by the AC type, its value and any optional text given with the AC.

In the absence of any real ACs the EXPORT file must still contain an entry with no values:

| | | | | | | | | |
|--------------|------|-------|------|------|----|------|---|---|
| filename.RAC | | | | 4 | 4 | 52 | 0 | |
| filename-ID | 4-1 | 14-1 | 5-1 | 50-1 | -1 | -1-1 | | 1 |
| TYPE | 4-1 | 54-1 | 5-1 | 50-1 | -1 | -1-1 | | 2 |
| VALUE | 4-1 | 94-1 | 5-1 | 50-1 | -1 | -1-1 | | 3 |
| TEXT | 40-1 | 134-1 | 40-1 | 20-1 | -1 | -1-1 | | 4 |

Finally, the EXPORT file should be terminated by the lines

EOI
EOS

Example EXPORT File

EXP 0 ORCA\$DUA0:[ARC50]TEST.E00

ARC 2

| | | | | | | |
|---------------|---------------|---------------|---------------|---|---|---|
| 1 | 0 | 2 | 2 | 2 | 3 | 4 |
| 4.5175273E+04 | 7.3694828E+04 | 4.5175273E+04 | 7.4143547E+04 | | | |
| 4.5285164E+04 | 7.3694828E+04 | 4.5175273E+04 | 7.3694828E+04 | | | |
| 2 | 0 | 3 | 2 | 2 | 2 | 2 |
| 4.5175273E+04 | 7.3319359E+04 | 4.5175273E+04 | 7.3694828E+04 | | | |
| 3 | 0 | 4 | 4 | 4 | 2 | 5 |
| 5.3737664E+04 | 5.7833430E+04 | 5.3871344E+04 | 5.7833430E+04 | | | |
| 5.3871344E+04 | 5.7948016E+04 | 5.3737664E+04 | 5.7948016E+04 | | | |
| 5.3737664E+04 | 5.7833430E+04 | | | | | |
| -1 | 0 | 0 | 0 | 0 | 0 | 0 |

LAB 2

| | | | | | |
|---------------|---------------|---------------|---------------|--|--|
| 1 | 2 | 5.0058543E+04 | 6.5691727E+04 | | |
| 5.0058543E+04 | 6.5691727E+04 | 5.0058543E+04 | 6.5691727E+04 | | |
| 2 | 3 | 5.0058664E+04 | 6.5691570E+04 | | |
| 5.0058664E+04 | 6.5691570E+04 | 5.0058664E+04 | 6.5691570E+04 | | |
| 3 | 4 | 4.5211902E+04 | 7.3844398E+04 | | |
| 4.5211902E+04 | 7.3844398E+04 | 4.5211902E+04 | 7.3844398E+04 | | |
| -1 | 0 | 0.0000000E+00 | 0.0000000E+00 | | |

TXT 2

| | | | | | |
|----------------|---------------|---------------|---------------|---------------|--|
| 1 | 2 | 0 | 5 | 13 | |
| 4.5431688E+04 | 4.5678945E+04 | 0.0000000E+00 | 0.0000000E+00 | 6.4180004E+04 | |
| 6.4637887E+04 | 0.0000000E+00 | 0.0000000E+00 | 0.0000000E+00 | 0.0000000E+00 | |
| 0.0000000E+00 | 0.0000000E+00 | 0.0000000E+00 | 0.0000000E+00 | 8.0000000E+01 | |
| -1.0000000E+02 | | | | | |

Walker County

| | | | | | |
|----------------|---------------|---------------|---------------|---------------|--|
| 1 | 2 | 0 | 5 | 11 | |
| 4.5341785E+04 | 4.5422746E+04 | 0.0000000E+00 | 0.0000000E+00 | 6.4298734E+04 | |
| 6.4450539E+04 | 0.0000000E+00 | 0.0000000E+00 | 0.0000000E+00 | 0.0000000E+00 | |
| 0.0000000E+00 | 0.0000000E+00 | 0.0000000E+00 | 0.0000000E+00 | 8.0000000E+01 | |
| -1.0000000E+02 | | | | | |

Dade County

| | | | | | | |
|----|---|---|---|---|---|---|
| -1 | 0 | 0 | 0 | 0 | 0 | 0 |
|----|---|---|---|---|---|---|

SIN 2

EOX

LOG 2

198911 12136 1 14 1131IMPORT COVER FILE1 TEST

~

EOL

IFO 2

TEST.AAT

| | | | | | | | | |
|---------|-----|-------|------|------|----|------|---|----|
| | | | XX | 8 | 8 | 30 | 3 | |
| FNODE# | 4-1 | 14-1 | 5 0 | 50-1 | -1 | -1-1 | | 1- |
| TNODE# | 4-1 | 54-1 | 5 0 | 50-1 | -1 | -1-1 | | 2- |
| LPOLY# | 4-1 | 94-1 | 5 0 | 50-1 | -1 | -1-1 | | 3- |
| RPOLY# | 4-1 | 134-1 | 5 0 | 50-1 | -1 | -1-1 | | 4- |
| LENGTH | 4-1 | 174-1 | 12 3 | 60-1 | -1 | -1-1 | | 5- |
| TEST# | 4-1 | 214-1 | 5 0 | 50-1 | -1 | -1-1 | | 6- |
| TEST-ID | 4-1 | 254-1 | 5 0 | 50-1 | -1 | -1-1 | | 7- |
| TYPE | 2-1 | 294-1 | 2 0 | 30-1 | -1 | -1-1 | | 8- |

| | | | | | | |
|---|---|---|---|---------------|---|---|
| 2 | 2 | 2 | 3 | 1.0205883E+03 | 1 | 1 |
| 1 | | | | | | |
| 3 | 2 | 2 | 2 | 3.7546875E+02 | 2 | 2 |

```

1
4          4          4          2 4.9653125E+02          3          3
4
TEST.BND
XMIN          4-1    14-1    12 3 60-1    -1    -1-1          1-
YMIN          4-1    54-1    12 3 60-1    -1    -1-1          2-
XMAX          4-1    94-1    12 3 60-1    -1    -1-1          3-
YMAX          4-1   134-1    12 3 60-1    -1    -1-1          4-
  4.0351262E+04 5.2367867E+04 5.9762277E+04 7.9025125E+04
TEST.PAT
AREA          4-1    14-1    12 3 60-1    -1    -1-1          1-
PERIMETER     4-1    54-1    12 3 60-1    -1    -1-1          2-
TEST#         4-1    94-1     5 0 50-1    -1    -1-1          3-
TEST-ID       4-1   134-1     5 0 50-1    -1    -1-1          4-
TYPE          3-1   174-1     3 0 20-1    -1    -1-1          5-
-5.1723462E+08 9.2115969E+04          1          0XXX
  5.1719466E+08 9.3633086E+04          2          1BCK
  2.4654992E+04 1.0205883E+03          3          2ARR
  1.5317813E+04 4.9653125E+02          4          3BOX
TEST.TIC
IDTIC         4-1    14-1     5 0 50-1    -1    -1-1          1-
XTIC          4-1    54-1    12 3 60-1    -1    -1-1          2-
YTIC          4-1    94-1    12 3 60-1    -1    -1-1          3-
  1 4.8294227E+04 7.4051867E+04
  2 4.8068285E+04 8.7915570E+04
  3 5.9459316E+04 8.8108656E+04
  4 5.9702652E+04 7.4244648E+04
EOI
EOS

```

CHAPTER 2

ARC/INFO DATA PREPARATION

DATA PREPARATION

Comparison of ARC/INFO 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 ARC/INFO format shares this concept of a 'feature' but does not hold all information relevant to a single feature in the same place within the dataset, but rather separates the information into different sections so that, for example, the coordinates of the points defining the feature are stored away from any attribute information that the IFF feature might have. The contents of these different sections are indexed by the internal EXPORT ID number, which is cross-referenced to the IFF feature serial numbers and feature codes in later tables.

ARC/INFO has its own feature coding scheme whereby the FRT file is read and any features unique in graphical type (GT), colour, width, size and secondary code are assigned a sequential symbol number unique within the three major ARC/INFO feature types of linear (ARC), symbol (LAB) and text (TXT). These are cross-referenced back to IFF feature codes in later tables.

It is essential that a feature representation table (FRT) file be supplied to I2ARC with the IFF file, as the graphical type of any feature is of prime importance in the correct placement of entries through the ARC/INFO file.

Feature Translation

When converting from IFF to ARC/INFO, it is not possible to translate fully some IFF graphical types, (the only features to be translated exactly are the linear feature with GT 1 consisting simply of a string of coordinates, and the text feature with GT 10 consisting of a character string, size, orientation and position). These are listed below, together with their approximations.

All arc (true arcs, not to be mistaken with the ARC/INFO definition of an arc) 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) in IFF may have size or orientation information contained in the FRT file or in the IFF RO or TH entries, or a second defining point. This information is lost in the translation to ARC/INFO which can only consider a symbol as a simple entity.

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 EXPORT and IFF

It is quite likely that IFF will be used as the digitising format and the data subsequently transferred to an EXPORT database. On the other hand, data may be transferred from an EXPORT database 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 attributes and EXPORT attributes, if these are to be preserved, should be defined.

The implementation of this requires

- o a parameter file defining attribute correlation, and
- o an FRT file (together with SRI and TRI files) containing an ACD section

The parameter file defines how the EXPORT attributes will be represented in IFF AC entries. It is cross referenced with the ACD section in the FRT by name or code to get its datatype. A mechanism also exists to specify whether the attribute is to go into the AC text or value field. If it is to go into the text field, the datatype in the ACD is ignored. If the datatype in the ACD conflicts with the datatype of the attribute in the EXPORT file, a warning will be given and the attribute ignored.

In converting from ARC2I, the attributes in any coverage tables other than the .AAT or .PAT will be ignored. Attributes in the .AAT and .PAT table will be ignored unless the appropriate definitions are present in the parameter file. When transferring attributes, the .AAT and .PAT attribute information should be in the same order as with the ARC and LAB sections respectively. The ID numbers in the .PAT entries should match the ID number and order of the corresponding points in the LAB section. If one attribute is wrong this may throw the ordering out and cause the following points to be given no attributes.

When transferring data from EXPORT, ARC2I will only produce ST entries unless the output revision level has been set to 1 within the system. Text features will contain TH entries holding the text height. All features will have the text/symbol bit in the FS entry set to the appropriate value to indicate the feature type. All data will be

output to layer 1 in the IFF file.

The ARC2I parameter file

This file contains three parts :-

- a) the default line, symbol and text feature codes, and
- b) the attribute description table, and
- c) the annotation feature codes

The default feature codes

The default feature codes are optional. If they are missing, the default feature codes are 1, 2 and 3 for lines, symbols, and text respectively. These defaults can be overridden using the commands DEF_LINE, DEF_SYMBOL and DEF_TEXT in the parameter file to allow users to give these feature types the feature codes of their choice.

An example follows :-

```
!  
! Type          <fc>  
!  
DEF_LINE        10  
DEF_SYMBOL      20  
DEF_TEXT        30  
!
```

The attribute description 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 attribute description table describes the translations required to produce IFF attributes from EXPORT attributes. Each line is prefixed with the ADT command which tells the program how to translate a particular attribute.

A line has one of the following two forms :-

| ADT CODE | code | entry | field | Ename | Etable |
|----------|------|-------|-------|-------|--------|
| ADT NAME | name | entry | field | Ename | Etable |

where the fields are :-

| | | |
|------|---|---|
| ADT | - | the command ADT which states that this line is an entry in the attribute description table. |
| CODE | - | the sub-command CODE which states that the attribute will be described by its IFF attribute code. |
| NAME | - | the sub-command NAME which states that the attribute |

- will be described by its IFF attribute name.
- code** - the IFF attribute code, an integer in the range 0 to 32767, as derived from the appropriate ACD entry in an FRT file.
 - name** - the IFF attribute name, as given by the appropriate ACD entry in an FRT file.
 - entry** - a command specifying which IFF entry the attribute is to be transferred to - the only valid command is
AC which specifies that an AC entry is to be used
 - field** - a sub-command to AC, specifying which field the attribute is to be transferred to - valid commands are either
VALUE which specifies that the AC value field is to be used, or
TEXT which specifies that the AC text field is to be used
 - Ename** - the EXPORT attribute name, a string which can be up to 16 characters in length
 - Etable** - a sub-command which denotes which EXPORT table the entry is referring to - the .AAT table or the .PAT table. Logically, the valid commands are either
AAT which denotes this entry as referring to the .AAT table, or
PAT which denotes this entry as referring to the .PAT table

The purpose of the 'Etable' column is because attributes of the same name can be used for lines as for symbols. They may even have different datatypes. In this case, two entries should be given, one with this field having command AAT and the other with command PAT. They should have different unique names or codes to reference the datatype in the ACD section of the FRT.

If a code is specified, it should reflect the ACD table which the attribute refers to. ACD table 1 entries range from 1000-1999, ACD table 2 entries range from 2000-2999 and so on.

An example follows :-

| | NAME/CODE | Group | AC VALUE/TEXT | EXPORT-att | EXPORT-table |
|-----|-----------|----------|---------------|------------|--------------|
| ADT | NAME | MajorTwo | AC VALUE | MAJOR2 | AAT |
| ADT | NAME | MinorTwo | AC VALUE | MINOR2 | AAT |
| ADT | CODE | 1050 | AC TEXT | LINE | AAT |
| ADT | NAME | Spot | AC VALUE | SPOT | AAT |
| ADT | NAME | MinorTwo | AC VALUE | MINOR2 | PAT |
| ADT | CODE | 1120 | AC TEXT | TYPE | PAT |
| ADT | NAME | XCoord | AC VALUE | X-COORD | PAT |
| ADT | NAME | YCoord | AC VALUE | Y-COORD | PAT |
| ADT | NAME | SiteId | AC TEXT | SITE-ID | PAT |
| ADT | NAME | SiteType | AC TEXT | SITE-TYPE | PAT |
| ADT | CODE | 1230 | AC VALUE | BEG-A | PAT |

The annotation feature codes

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 annotation feature code table describes the lookup between text LEVEL and SYMBOL numbers into IFF feature codes. Each line is prefixed with the ANNO command which tells the program that a line is part of this table. A feature code can be assigned to a text feature on the basis of its LEVEL code, SYMBOL code or both.

A line has one of the following three forms :-

```

ANNO  FC  fcode  LEVEL  lcode  SYMBOL  scode
ANNO  FC  fcode  LEVEL  lcode
ANNO  FC  fcode  SYMBOL  scode

```

where the fields are :-

- ANNO** - the command ANNO which states that this line is an entry in the annotation feature code table.
- FC** - the sub-command FC which will be followed by the feature code to be derived from the LEVEL code, SYMBOL code or both.
- fcode** - the IFF feature code, an integer in the range 0 to 32767, to be assigned to a feature with the corresponding LEVEL or SYMBOL codes.
- LEVEL** - the LEVEL sub-command denotes that the feature code will be assigned to text features with the following LEVEL code. If the SYMBOL command is also given on this line, then the feature code can only be assigned if both the LEVEL and SYMBOL codes match. If neither match in the whole table, the default feature code is used.
- lcode** - the value of the LEVEL code used to determine if this feature code is to be assigned to a feature.
- SYMBOL** - the SYMBOL sub-command denotes that the feature code will be assigned to text features with the following SYMBOL code. If the LEVEL command was also given on this line, then the feature code can only be assigned if both the LEVEL and SYMBOL codes match. If neither match in the whole table, the default feature code is used.
- scode** - the value of the SYMBOL code used to determine if this feature code is to be assigned to a feature.

NOTE

The order of entries in the ANNO table is important. The program performs a simple linear search through the table to assign feature codes on the basis of a first match between LEVEL and SYMBOL codes. So if, for example, you required an IFF feature code of 1001 to be assigned to features which have an ARC/INFO LEVEL code of 2 and SYMBOL code of 5, but a feature code of 1002 to be assigned to features with a LEVEL code of 2 irrespective of the SYMBOL code, then the feature code 1001 must appear in the list first. If the feature code 1002 was put in the table first, this would match everything with LEVEL code 2 and no features would be assigned feature code 1001.

An example follows :-

```
!
!      FC      fcode   LEVEL  lcode    SYMBOL   scode
!
ANNO    FC      1000    LEVEL    1
ANNO    FC      1001    LEVEL    2          SYMBOL   5
ANNO    FC      1002    LEVEL    2
ANNO    FC      1003    SYMBOL    3
ANNO    FC      1003    SYMBOL    6
ANNO    FC      1003    SYMBOL    7
!
```

The example also shows that the same feature code may be assigned on the basis of different LEVEL or SYMBOL codes.

The I2ARC parameter file

This file is almost identical to the ARC2I parameter file but only contains an attribute description table.

The attribute description 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 attribute description table describes the translations required to produce EXPORT attributes from IFF attribute. Each line is prefixed with the ADT command which tells the program how to translate a particular attribute.

A line has one of the following two forms :-

ADT CODE code entry field Ename

ADT NAME name entry field Ename

where the fields are :-

- ADT** - the command ADT which states that this line is an entry in the attribute description table.
- CODE** - the sub-command CODE which states that the attribute will be described by its IFF attribute code.
- NAME** - the sub-command NAME which states that the attribute will be described by its IFF attribute name.
- code** - the IFF attribute code, an integer in the range 0 to 32767, as derived from the appropriate ACD entry in an FRT file.
- name** - the IFF attribute name, as given by the appropriate ACD entry in an FRT file.
- entry** - a command specifying which IFF entry the attribute is to be transferred to - the only valid command is
 AC which specifies that an AC entry is to be used
- field** - a sub-command to AC, specifying which field the attribute is to be transferred to - valid commands are either
 VALUE which specifies that the AC value field is to be used, or
 TEXT which specifies that the AC text field is to be used
- Ename** - the EXPORT attribute name, a string which can be up to 16 characters in length

The difference between this and the ARC2I parameter file is the absence of the 'Etable' column because attributes of IFF line features automatically go into the .AAT table and attributes of IFF symbol features go in the .PAT table.

The AC entry and its sub-commands TEXT and VALUE were included for consistency with the ARC2I parameter file but are ignored. This way I2ARC can use an ARC2I parameter file with minimal changes. This entry is not used because I2ARC reads the ACD datatype from the FRT to determine which field to read the value from. For example, if the ACD datatype is 'C' then the text is read from the optional text string and the integer and real fields are ignored.

If a code is specified, it should reflect the ACD table which the attribute refers to. ACD table 1 entries range from 1000-1999, ACD table 2 entries range from 2000-2999 and so on.

An example follows :-

```
!  
!      NAME/CODE  Group          AC  VALUE/TEXT    EXPORT-att  
!  
ADT      NAME      MajorTwo      AC  VALUE      MAJOR2  
ADT      NAME      MinorTwo      AC  VALUE      MINOR2  
ADT      CODE      1050          AC  TEXT  
          LINE  
ADT      NAME      Spot          AC  VALUE      SPOT  
ADT      NAME      MinorTwo      AC  VALUE      MINOR2  
ADT      CODE      1120          AC  TEXT      TYPE  
ADT      NAME      XCoord        AC  VALUE      X-COORD  
ADT      NAME      YCoord        AC  VALUE      Y-COORD  
ADT      NAME      SiteId        AC  TEXT      SITE-ID  
ADT      NAME      SiteType      AC  TEXT      SITE-TYPE  
ADT      CODE      1230          AC  VALUE      BEG-A  
!
```

An example ARC2I parameter file is given below, followed by the ACD section of the FRT with which it is associated.

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! ARC/INFO Parameter file for attribute conversion in ARC2I
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! Default feature codes
!
!           <fc>
DEF_LINE      10
DEF_SYMBOL    20
DEF_TEXT      30
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! Attribute Definitions
!
!      NAME/CODE  Group          AC VALUE/TEXT    EXPORT-att    EXPORT-table
!
ADT      NAME      MajorOne      AC VALUE      MAJOR1        AAT
ADT      NAME      MinorOne      AC VALUE      MINOR1        AAT
ADT      NAME      MajorTwo      AC VALUE      MAJOR2        AAT
ADT      NAME      MinorTwo      AC VALUE      MINOR2        AAT
ADT      NAME      Line          AC TEXT       LINE          AAT
ADT      NAME      Spot          AC VALUE      SPOT          AAT
ADT      NAME      Type_aat      AC VALUE      TYPE          AAT
!
ADT      NAME      MajorOne      AC VALUE      MAJOR1        PAT
ADT      NAME      MinorOne      AC VALUE      MINOR1        PAT
ADT      NAME      MajorTwo      AC VALUE      MAJOR2        PAT
ADT      NAME      MinorTwo      AC VALUE      MINOR2        PAT
ADT      NAME      Type_pat      AC TEXT       TYPE          PAT
ADT      NAME      XCoord        AC VALUE      X-COORD       PAT
ADT      NAME      YCoord        AC VALUE      Y-COORD       PAT
ADT      NAME      SiteId        AC TEXT       SITE-ID       PAT
ADT      NAME      SiteType      AC TEXT       SITE-TYPE     PAT
ADT      NAME      BeginDate     AC VALUE      BEG-A         PAT
ADT      NAME      EndDate       AC VALUE      END-A         PAT
ADT      NAME      PhotoDates    AC TEXT       PHOTO-DATES   PAT
ADT      NAME      System        AC TEXT       SYSTEM        PAT
ADT      NAME      SubSystem     AC TEXT       SUBSYSTEM     PAT
ADT      NAME      Class         AC TEXT       CLASS         PAT
ADT      NAME      SubClass      AC TEXT       SUBCLASS      PAT
ADT      NAME      Modifier      AC TEXT       MODIFIER      PAT
ADT      NAME      LCCode        AC TEXT       LCCODE        PAT
ADT      NAME      LUCode        AC VALUE      LUCODE        PAT
ADT      CODE      1968          AC TEXT       F-1968        PAT
ADT      CODE      1973          AC TEXT       F-1973        PAT
ADT      CODE      1976          AC TEXT       F-1976        PAT
ADT      CODE      1980          AC TEXT       F-1980        PAT
ADT      CODE      1983          AC TEXT       F-1983        PAT
ADT      CODE      1984          AC TEXT       F-1984        PAT
ADT      CODE      1985          AC TEXT       F-1985        PAT

```

| | | | | | | |
|------|------|-------|--------|---------|--------|-------|
| ADT | CODE | 1986 | | AC TEXT | F-1986 | PAT |
| ADT | CODE | 1987 | | AC TEXT | F-1987 | PAT |
| ADT | CODE | 1988 | | AC TEXT | F-1988 | PAT |
| ADT | CODE | 1989 | | AC TEXT | F-1989 | PAT |
| ! | | | | | | |
| ! | FC | fcode | LEVEL | lcode | SYMBOL | scode |
| ! | | | | | | |
| ANNO | FC | 1000 | LEVEL | 1 | | |
| ANNO | FC | 1001 | LEVEL | 2 | SYMBOL | 5 |
| ANNO | FC | 1002 | LEVEL | 2 | | |
| ANNO | FC | 1003 | SYMBOL | 3 | | |
| ANNO | FC | 1003 | SYMBOL | 6 | | |
| ANNO | FC | 1003 | SYMBOL | 7 | | |
| ! | | | | | | |

Note that in this example, the CODE for the last few entries reflect the date associated with the attributes. This is just for clarity because integers are not permitted in the ACD group names which were called 'TextA' etc. instead of 'F-1968' etc.

The ACD part of the FRT:

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! ACD part of FRT for ARC/INFO EXPORT attribute translation with ARC2I
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ACD      TABLE      1
!
!      datatype      code      group
!
ACD      I           10      MajorOne
ACD      I           20      MinorOne
ACD      I           30      MajorTwo
ACD      I           40      MinorTwo
ACD      I           50      Type_aat
ACD      C           60      Line
ACD      I           70      Spot
ACD      C           80      Type_pat
ACD      R           90      XCoord
ACD      R          100      YCoord
ACD      C          110      SiteId
ACD      C          120      SiteType
ACD      D          130      BeginDate
ACD      D          140      EndDate
ACD      C          150      PhotoDates
ACD      C          160      System
ACD      C          170      SubSystem
ACD      C          180      Class
ACD      C          190      SubClass
ACD      C          200      Modifier
ACD      C          210      LCCode
ACD      I          220      LUCode
ACD      C          968      TextA      ! F-1968
ACD      C          973      TextB      ! F-1973
ACD      C          976      TextC      ! F-1976

```

| | | | | |
|-----|---|-----|-------|----------|
| ACD | C | 980 | TextD | ! F-1980 |
| ACD | C | 983 | TextE | ! F-1983 |
| ACD | C | 984 | TextF | ! F-1984 |
| ACD | C | 985 | TextG | ! F-1985 |
| ACD | C | 986 | TextH | ! F-1986 |
| ACD | C | 987 | TextI | ! F-1987 |
| ACD | C | 988 | TextJ | ! F-1988 |
| ACD | C | 989 | TextK | ! F-1989 |

CHAPTER 3

I2ARC UTILITY

UTILITY I2ARC

FUNCTION

I2ARC reads an Internal Feature Format (IFF) file, and produces a file on disc in ARC/INFO export format.

FORMAT

\$ I2ARC input-IFF-file-spec output-ARC-name

| Command qualifiers | Defaults |
|----------------------|----------------------|
| /[NO]DEBUG | /NODEBUG |
| /FRT=file-spec | /FRT=LSL\$FRT:OS.FRT |
| /[NO]INFO_TABLES | /NOINFO_TABLES |
| /[NO]LOG | /NOLOG |
| /PARAMETER=file-spec | no parameter file |

PROMPT

| | |
|-------------------|---------------------|
| _Input IFF file: | input-IFF-file-spec |
| _Output ARC name: | output-ARC-name |

PARAMETERS

input-IFF-file-spec

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

output-ARC-name

- This parameter specifies the name of an ARC/INFO data file name, and is compulsory. Only the name of the output file must be specified, without any directory or extension attributes, and the output ARC/INFO file-spec will be constructed with the extension .E00 and written to the current directory. The data read from the input IFF file are written to this file. Only one filename may be specified for each run of the program. The maximum length for the output filename is 12 characters and if more are given, it will be truncated. This is because the name appears in the tables in the IFO section and a name with more than 12 characters will cause the field columns to be misaligned.

COMMAND QUALIFIERS

/DEBUG

/NODEBUG (default)

- When /DEBUG is present, the I2ARC utility will output many more diagnostic messages as it processes the input file than it would under normal running, or with the /LOG qualifier given. Each IFF entry is signalled, with the FSN and ISN given for the NF entries, and the number of coordinate points and their visibility given for ST/ZS/CB entries, as an aid to tracing any IFF conversion problems. If the program run should fail for some reason, all temporary files are left for further examination to assess the cause of failure.

/FRT=file-spec

- The /FRT command qualifier specifies a Feature Representation Table (FRT) file which the program will read to determine the graphical types (GT) of the IFF features, and for information on any AC entries. The FRT qualifier with filespec is compulsory. The default file specification is LSL\$FRT:OS.FRT. If a file specification is given with /FRT qualifier, the file-spec is parsed against this default.

/INFO_TABLES

/NOINFO_TABLES (default)

- When /INFO_TABLES is present, the I2ARC utility will put any AC attribute information into ARC/INFO .AAT and .PAT tables rather than the .IAC and .RAC ones. The .AAT and .PAT tables are the standard way of holding attribute information within ARC/INFO and are described in the FORMAT DESCRIPTION section at the beginning of the chapter. If the qualifier is present, then /PARAMETER must also be used to specify a file containing the attribute names as they will appear in the EXPORT tables. If the qualifier is not present (or negated), the parameter file is not required and any attributes are written in the form of the .IAC and .RAC tables. This is the default action.

/LOG

/NOLOG (default)

- When /LOG is present, the I2ARC utility will output diagnostic messages about any file creation or deletion as it processes the input IFF file so that any disc errors may be traced.

/PARAMETER=file-spec

- The /PARAMETER command qualifier specifies a parameter file which holds information about the translation of attributes into the output EXPORT file. It is used in conjunction with /INFO_TABLES as it only describes the attributes that go into the .AAT and .PAT tables. A description and an example of the parameter file can be found in the DATA PREPARATION section of this chapter.

RESTRICTIONS

There are certain types of IFF feature that I2ARC does not translate fully. These include:

Arc and Circle features (GTs 2, 3, 4, 5) are output as linear features, with interpolated points between the start and end points used to define the feature.

Interpolated curves (GT 6) and Symbol strings (GT 11) are output as simple linear features. This means that the symbol information for a Symbol string is lost.

Oriented or scaled symbols (GT 8, 9), which I2ARC treats as simple symbol features (GT 7), lose any size or rotation information supplied either in the FRT file or in the IFF file in TH or RO entries, or a second size and orientation defining point.

Area fill features (GT 12) are output as a closed linear feature.

When using /INFO_TABLES, I2ARC can only produce EXPORT item datatypes 1, 2, 5 and 6 in the .AAT and .PAT tables. This is because an integer ('I') datatype in the ACD part of the FRT could map onto EXPORT datatypes 3 or 5. Similarly, a real ('R') datatype in the ACD could map onto EXPORT types 4 or 6.

I2ARC maintains no structure in the data and so the FNODE, TNODE, LPOLY, RPOLY, AREA, LENGTH and PERIMETER values in the .AAT and .PAT tables are set to zero.

DESCRIPTION

Command line

The symbol I2ARC is normally set up as:

```
I2ARC == "$ls1$exe:i2arc"
```

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

PROGRAM ACTION

I2ARC is a utility to transfer an Internal Feature Format (IFF) disc file to an ARC/INFO export format ASCII disc file.

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

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

After decoding the command line for input and output file names and the qualifiers, the FRT file is read. Each feature code that is unique in graphical type, colour, width, size and secondary code is assigned a unique internal Symbol code sequential within the three basic feature types: Symbol (GT 7, 8, 9), Text (GT 10) or Linear (the remainder), for the three ARC/INFO sections of ARC, LAB and TXT respectively.

If it was specified on the command line, the parameter file is then read to get the EXPORT attribute names associated with AC types.

The supplied IFF file is opened with input revision level of zero so that any CB entries are translated into ST or ZS entries, from which only the x and y coordinates are read and used for the ARC/INFO data file.

Because the ARC/INFO format stores the various information types separately in different sections of the export file, temporary data files are created to store the IFF data as it is read. These are: \$\$LAB.TMP for Symbol features, \$\$TXT.TMP for Text features, \$\$FSN.TMP for the FSN, FC, Overlay and internal ID number cross-reference table, \$\$IAC.TMP for integer AC information and \$\$RAC.TMP for real AC information. If /INFO_TABLES was specified, any AC information will be written to data files, \$\$AAT.TMP and \$\$PAT.TMP. All these files are copied across to the main EXPORT file after the IFF file has been read, and are then deleted.

After writing the ARC/INFO file header of EXP 0 to the ARC/INFO file, and the three coordinate section header lines of ARC 2 to the ARC/INFO file, LAB 2 to \$\$LAB.TMP and TXT 2 to \$\$TXT.TMP, I2ARC translates the IFF file entry by entry into the relevant ARC/INFO file or temporary file.

The IFF header entries of relevance to ARC/INFO are the range (RA) and corner points (CP) are stored for later use in the IFO information section's filename.BND table and filename.TIC tables respectively.

Any overlay (NO) entry is recorded for use in the filename.FSN cross-reference table of the IFO section, and for any text features encountered.

On encountering a new feature (NF), the serial number (FSN) is recorded for the internal ARC/INFO ID to IFF FSN cross-reference table in the IFO section, as also is the feature code (FC). This

information is output to \$\$FSN.TMP at the end of feature (EF) entry, for later copying to the ARC/INFO file, as ID, FSN, FC and Layer in the filename.FSN section in IFO.

If /NOINFO_TABLES is given or missing, any information from attribute code (AC) entries encountered is written to the \$\$IAC.TMP or \$\$RAC.TMP files depending on whether the AC value is integer (or character or date or time) or real, for later copying to the ARC/INFO file filename.IAC or filename.RAC section of IFO.

If /INFO_TABLES is given, an initial pass of the IFF file is made to build a list of all the ACs in the file in order to create the tables. Due to the nature of the .AAT and .PAT tables, which hold an entry for each attribute of each line or symbol, there may be some excess information. The program has to produce a file containing the maximum number of attributes per feature for all features to cover the worst case.

Any information in TX, TH or RO entries is stored for later use depending on the graphical type of the feature: for symbol features it is discarded, whereas for text features it may be used to calculate start and end positions of the text string when the coordinate entry has been encountered.

What I2ARC does with coordinate information from any ST or ZS entries (and CBs converted to ST/ZS) depends on the graphical type (GT) of the feature in its FC. Linear feature coordinates can be output to the ARC/INFO file immediately as they are the first of the three feature types (ARC, LAB, TXT) in the ARC/INFO format. The IFF graphical types which are considered to be linear and for which the coordinate information is used without any change or interpolation are line (GT 1), interpolated curve (GT 6), symbol string (GT 11) where any symbol information is lost, and fill area (GT 12) considered to be a closed linear feature. All arc or circle features (GTs 2, 3, 4, 5) have their three-point coordinate information interpolated into a suitable number of points to suitably define the feature, and then output to the ARC/INFO file ARC section.

All IFF coordinate blocks belonging to the same feature with no invisible line breaks are output to ARC/INFO in the one block, up to a maximum number of x, y pairs of 500, when a new block must be started having the same ID number. If the feature contains invisible line breaks, the sections are output to ARC/INFO in separate coordinate blocks with the same ID number.

If the feature is a symbol (GTs 7, 8, 9) the coordinate information is output to \$\$LAB.TMP for later copying to the ARC/INFO file LAB section in the appropriate format. ARC/INFO cannot handle any symbol size or orientation information that may have been held in the FRT file or TH or RO entries, or a second defining point, and this is discarded.

All text feature (GT 10) coordinate information is written to \$\$TXT.TMP in a suitable format for later copying to the ARC/INFO file TXT section. The text status components (TS) of a multi-text string are copied separately as distinct features with distinct ID numbers. The text size in the FRT file, or size and orientation information

held in the IFF file TH or RO entries is used to calculate start and end positions of the text string.

Any other IFF entries are reflected to the user and ignored.

On completion of the IFF reading, the end of section lines of -1 ... are written to the three coordinate sections in ARC/INFO (ARC), \$\$LAB.TMP (LAB) and \$\$TXT.TMP (TXT) and then \$\$LAB.TMP and \$\$TXT.TMP are copied to the ARC/INFO file and deleted.

The IFO information sections are next completed. First the header IFO 2 followed by the range and corner point data into filename.BND and filename.TIC in suitable formats. \$\$FSN.TMP is then copied across containing the ID, FSN, FC and Layer reference table, followed by \$\$IAC.TMP for integer AC data and \$\$RAC.TMP for real AC data. Finally the filename.LUT table is written out for the IFF FC to ARC/INFO symbol number reference, and the ARC/INFO file finished with the EOI and EOS terminators.

EXAMPLES

\$ I2ARC/FRT=OS TW_ANG1 TEST_ARC <CR>

HI
MH
MD
CC

ELAPSED: 0 00:00:08.15 CPU: 0:00:01.82 BUFIO: 18 DIRIO: 61 FAULTS: 210

This example shows a normal run of I2ARC successfully converting the IFF file LSL\$IF:TW_ANG1.IFF into the ARC/INFO file TEST_ARC.E00.

\$ I2ARC/LOG/INFO_TABLES/FRT=THERE:ST/PAR=THERE:ARCINFO.DAT BTEST BTEST <CR>

Opening IFF file LSL\$IF:BTEST.IFF
Read FRT file THERE:ST.FRT
Opening parameter file THERE:ARCINFO.DAT
Opening ARC/INFO file BTEST.e00
Opening work files \$\$lab.tmp
 \$\$txt.tmp
 \$\$fsn.tmp
 \$\$aat.tmp
 \$\$pat.tmp

Now processing the IFF file to ARC/INFO

HI
MH
MD
NS
CC

Closing IFF file LSL\$IF:BTEST.IFF
Removing work file \$\$lab.tmp
Removing work file \$\$txt.tmp
Removing work file \$\$fsn.tmp
Removing work file \$\$aat.tmp
Removing work file \$\$pat.tmp
Closing ARC/INFO file BTEST.e00

ELAPSED: 0 00:00:08.15 CPU: 0:00:01.82 BUFIO: 18 DIRIO: 61 FAULTS: 210

This example show the use of the /LOG, /INFO_TABLES and /PARAMETER qualifiers to process a file and produce .AAT and .PAT tables.

MESSAGES (SUCCESS)

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

NORMAL, successful completion

Explanation: The program has terminated successfully.

User action: None.

MESSAGES (INFORMATIONAL)

These messages give information only, and require no immediate action by the user. They are used to provide information on the current state of the program, or to supply explanatory information in support of a warning or error message.

LINEINPAR, line 'number' of parameter file

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

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

MESSAGES (WARNING)

These messages are output when an error has occurred that can be corrected immediately by the user or that the program will attempt to overcome.

BADATTCODE, attribute code 'code' out of range

Explanation: An invalid IFF attribute code (outwith the range 0 to 32767) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the attribute code in the parameter file, and if necessary run the program again.

BADATTNAME, invalid attribute name 'name'

Explanation: An invalid IFF or EXPORT attribute name has been read from the parameter file. This message may be followed by an LSLLIB message indicating an error in reading the name. Alternatively, this message may be preceded by a message indicating why the name was not recognised. The LINEINPAR message indicating the parameter file line number will appear last, and the program will continue to read the file, ignoring this line.

User action: Amend the attribute name in the parameter file, and if necessary run the program again.

DEFAULTACD, cannot find attribute code 'code', using default values

Explanation: The given IFF attribute code (as read from the parameter file) could not be found amongst the current definitions. Default values will be used instead.

User action: Ensure that the appropriate ACD definition appears in the FRT file if you are using one. Otherwise, amend the attribute code in the parameter file if it is in error. If necessary run the program again.

MISSATTDEF, ACD type 'number' ('name') is not defined in parameter file

Explanation: The ACD type specified in the FRT does not have a corresponding entry in the parameter file which is necessary to obtain the EXPORT attribute name to go in the .AAT or .PAT table. The attribute will be ignored.

User action: If this attribute is required in the translation, there must be an line in the parameter file defining the EXPORT attribute name and various other information associated with it. Refer to the DATA PREPARATION section in the CONVERT User Guide for details of the I2ARC paramter file.

NOATTNAME, cannot read attribute name

Explanation: An IFF or EXPORT attribute name cannot be read where expected from the parameter file. This message will be followed by an LSLLIB message indicating why the name could not be read. The LINEINPAR message indicating the parameter file line number will also follow, and the program will continue to read the file, ignoring this line.

User action: Edit the parameter file to include the attribute name in the parameter file where expected, and if necessary run the program again. In the unlikely event that no errors are found in the parameter file, please report the problem to Laser-Scan.

NOTEXT, AC 'number' has ACD type C, but no text field present

Explanation: The AC type was declared in the ACD part of the FRT as being a character datatype and when the IFF file was scanned, no text fields were found. This attribute will not be included in the translation. The presence of the text field is essential when creating .AAT and .PAT tables in order to determine the maximum field size for the attribute. Hence this error only occurs when creating .AAT and .PAT attribute tables.

User action: If the attribute in question is required in the translation, the IFF AC entry must contain the text field to determine the maximum size for this character field.

OUTFIL, Reducing filename to 'name'

Explanation: The output filename was reduced to 12 character in length so that the columns in the EXPORT attribute tables are not affected.

User action: None.

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: This message is most likely due to an error in the program and should be reported to Laser-Scan.

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.

ARCNAM, only the name of the ARCINFO file can be specified

Explanation: The ARCINFO output file name specified on the command line must only be a file name - that is, it must not include a node, device, directory, extension or version number.

User action: Run the program again, only specifying the ARCINFO file name.

ARCOPNERR, error opening ARC/INFO file

Explanation: The ARC/INFO file derived from the name given in the command line could not be opened.

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

CMDLNERR, error reading the command line

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

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

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

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

User action: Depends upon the associated LSLLIB message.

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

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

User action: Depends upon the associated LSLLIB messages.

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.

FRTOPNERR, error opening FRT file

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

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

IFFOPNERR, error opening IFF file

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

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

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

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

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

PAROPNERR, error opening parameter file

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

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

TMPFILERR, error opening temporary files

Explanation: There was an error opening the temporary work files \$\$lab.tmp, \$\$txt.tmp, \$\$fsn.tmp, \$\$iac.tmp, \$\$rac.tmp, \$\$aat.tmp or \$\$pat.tmp.

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

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

ARC2I UTILITY

UTILITY ARC2I

FUNCTION

ARC2I reads an uncompressed ARC/INFO EXPORT tape or disk file and produces a Laser-Scan IFF file. If EXPORT attributes are desired in the translation, a parameter file must be specified to reflect how they will be represented in the IFF file. This parameter file is used in conjunction with the ACD part of an FRT file.

FORMAT

\$ ARC2I input-EXPORT-file-spec [output-IFF-file-spec]

Command qualifiers

Defaults

| | |
|-------------------------|---------------------|
| /ARC_ID=keyword | /ARC_ID = USER_ID |
| /BLOCKSIZE=integer | Blocksize = 8000 |
| /FRT=file-spec | No FRT |
| /[NO]ID_TO_FC | Use default FCs |
| /[NO]LOG | /NOLOG |
| /PARAMETER=file-spec | No parameter file |
| /POINT_ID=keyword | /POINT_ID = USER_ID |
| /[NO]REWIND | /NOREWIND |
| /SCALE=integer | Scale = 0 |
| /SELECT=(integer[,...]) | All tape files |
| /[NO]VERBOSE | /NOVERBOSE |

PROMPT

_Input EXPORT-file: input-EXPORT-file-spec

PARAMETERS

input-EXPORT-file-spec

- This parameter specifies the name of the input EXPORT 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:filename.E00' is applied to the input file specification when it is parsed.

When processing files on tape, the input file name should be the magnetic tape device which has been mounted with /FOREIGN. The EXAMPLES section shows tape files being processed.

output-IFF-file-spec

- This optional parameter specifies the name of the output IFF file which contains the data converted from the input EXPORT file. If the parameter is not given, the output file will have the same name as the input file but with the default device and extension 'LSL\$IF:filename.IFF' applied where 'filename' is the input file name.

When processing tape files, this parameter is only permitted if the /SELECT qualifier was also given with one argument. This file-spec is then given to the single tape file. If /SELECT was missing or given with more than one argument, the output file-spec is ignored as each output file will have the coverage name held in each EXPORT file.

COMMAND QUALIFIERS

/ARC_ID=keyword
/ARC_ID=USER_ID (default)

- This qualifier specifies which fields to use when matching arc attributes to their features. The two fields available are the User-ID and the Internal-ID fields and so the keywords are USER_ID and INTERNAL_ID.

The first field of each line feature in the ARC section is the Internal-ID and the second is the User-ID. Note that this is the opposite way around to the LAB section.

The first seven default attributes in the Arc Attribute Table (AAT) contain two ID attributes, at least one of which should have values which reflect the User-ID or Internal-ID of the ARC section. The attribute called the coverage name followed by a '#' is the Internal-ID (eg COVERAGE#) and the attribute called the coverage name followed by '-ID' is the User-ID (eg COVERAGE-ID).

The field that is used to match features must be unique and sorted so that each entry in the AAT table matches exactly the entry in the corresponding ARC section.

If this qualifier is not given, the User-ID field will be used to match any attributes to their features.

/BLOCKSIZE=integer

- This qualifier is used to specify the magnetic tape blocksize. The blocksize does not have to be exactly divisible by the EXPORT format record size (80 characters). The program will process files whose records span tape blocks. The default block size is 8000 and the maximum is 65535.

/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. In order for the program to do this, 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.

/ID_TO_FC

/NOID_TO_FC (default)

- This qualifier is specifically for use with data supplied by Bartholomews. It is used to extract Feature Codes (FC) of features from their User-ID fields in the EXPORT file. Since IFF will only hold a maximum FC number of 32767, the ID will be subject to some adjustments before being used as a feature code. These adjustments area as follows:

- User-IDs of zero are removed.
- User-IDs which are less than 32768 are left unchanged.
- User-IDs between 32768 and 139000 (inclusive) have their first 3 digits removed.
- User-IDs which are greater than 139000 will have their first 2 digits removed.

These adjustments will only produce unique feature codes for IDs greater than 32768 if the EXPORT file contains a certain set of IDs that will not conflict with others when they have their digits removed.

Use of this qualifier overrides the default FCs and the ones held in the parameter file. It only applies to linear and point features. Text feature will have the default feature code or that defined in the parameter file.

Users of other ARC/INFO data may safely use this qualifier to extract the FC from the User-ID field providing the values are less than 32768.

/LOG

/NOLOG (default)

- When /LOG is present, the ARC2I utility will output diagnostic messages about any file creation and what is happening as it processes the input EXPORT file so that any errors may be traced.

/PARAMETER=file-spec

- specifies the parameter file, which defines the transformation of EXPORT attributes into IFF AC entries. It is used in conjunction with the ACD section of the FRT file. For a description of the parameter

file, see the DATA PREPARATION section.

/POINT_ID=keyword

/POINT_ID=USER_ID (default)

- This qualifier specifies which fields to use when matching point attributes to their features. The two fields available are the User-ID and the Internal-ID fields and so the keywords are USER_ID and INTERNAL_ID.

The first field of each point in the LAB section is the User-ID and the second is the Internal-ID number. Note that this is the opposite way around to the ARC section.

The first four default attributes in the Point Attribute Table (PAT) contain two ID attributes, at least one of which should have values which reflect the User-ID or Internal-ID of the LAB section. The attribute called the coverage name followed by a '#' is the Internal-ID (eg COVERAGE#) and the attribute called the coverage name followed by '-ID' is the User-ID (eg COVERAGE-ID).

Often the LAB section will have unique User-ID values for polygon seeds but zeros in the second field (the Internal-ID) and so the /POINT_ID need not be given as this is the default.

The field that is used to match features must be unique and sorted so that each entry in the PAT table matches exactly the entry in the corresponding LAB section.

If this qualifier is not given, the User-ID field will be used to match any attributes to their features.

/REWIND (default)

/NOREWIND

- This qualifier defines whether the magnetic tape device is to be rewound **BEFORE** processing begins. When initialising the tape, the default is to ensure that the tape is at the beginning. The /NOREWIND is useful when processing very large volumes of tape data that needs to be removed in sections so that the /SELECT values do not start from the beginning of the tape.

/SCALE=integer

- This is to specify the scale of the output IFF file which is used to update the scale in the Map Descriptor (MD) entry and to convert text heights into hundredths of a millimetre on the sheet in the IFF (TH) entry. If no /SCALE qualifier is given, the Map Descriptor is left blank and the text entries will have heights taken directly from the EXPORT file.

/SELECT=(integer[,...])

- used to select the files to be processed by their position on tape. The maximum number should not exceed the number of files on the tape. The /SELECT qualifier will accept single, multiple or ranges of integer arguments. Ranges may be specified with starting and ending integers that are separated by a colon, for example /SELECT=2:4. When specifying more than one range they should be separated with a comma, for example /SELECT=(2:4,6,8:10). The EXAMPLES section shows the use of this qualifier.

The arguments to /SELECT are relative the position of the tape head which will depend on the use of the /REWIND qualifier.

For example, to convert files from tape in groups of ten:

If using /NOREWIND, the program should be invoked using /SELECT=(1:10) repeatedly because the tape head will remain after each group of ten files.

If using /REWIND, the program should be invoked using /SELECT=(1:10) and then /SELECT=(11:20) and so on, because the tape head will return to the start of the tape each time. This will slow the process down as the program has to skip files until it reaches the desired position.

If, however, it is required to start reading the tape from say the 51st file from the beginning, to reach this point use /SELECT=(51:60) and thereafter (with /NOREWIND) use /SELECT=(1:10).

/VERBOSE

/NOVERBOSE (default)

- When /VERBOSE is present, the ARC2I utility will output detailed information about the ordering of the Internal-IDs of points and arcs. It is intended for use when attributes cannot be associated with their respective arcs or points due to the Internal-IDs being in a different order. It should be noted that the program only makes a limited attempt to correct the attribute ordering in the AAT and PAT tables and files should ideally contain unique, ordered Internal-IDs.

RESTRICTIONS

ARC2I can only convert EXPORT files that are created with no compression. This implies that you can list the file on your terminal and read the strings of numbers which make up the arc and points etc. The ARC/INFO TAPEWRITE facility will produce compressed files by default and this should be overridden by specifying 'NONE' as the last command on the line to disable the compression.

ARC2I produces unstructured data with no polygonal information. This can be achieved in IFF using IMP utilities ILINK and IPOLYGON.

All attributes associated with line features must be held in the .AAT coverage table in the IFO section and attributes associated with point features must be held in the .PAT table. Attributes held in any other table are ignored.

ARC2I does not handle unloaded text. Any text within the EXPORT file is taken from the TXT (or TX6) section of the file and not from any updated text section in the form of a .TXT table under the IFO section. It is therefore essential that the updated text is loaded back into the TXT section of the EXPORT file.

If there are TXT entries in the EXPORT file with more than 2 points defining the arc or spline along which a text feature lies, they are ignored. ARC2I will only produce text features with no orientation or with an orientation equal to the angle between the origin and the line defined by the first two points in the feature.

The program partially supports ARC/INFO version 6.0 annotation text features as held in the TX6 section. The extent of the support is as follows:

- o BANKED TEXT - IFF composite text features will be created if the ARC feature has the relevant information defining it as composite.
- o ANGLED TEXT - Text with one point will have an orientation (IFF RO entry) of zero. Texts with two points will have an orientation equal to the angle between the two points.
- o BENT TEXT - Text digitised to follow an arc or curved feature is not supported. If the bent text consists of one component, then its orientation will be defined as the angle between the first and last points of the curve. If the text contains multiple components along one curve, each component will start on a point of that curve. Each component will therefore be oriented at the angle defined by the first and last point for each component's segment of the curve. The last point of a component's curve segment is the point before the one which starts the next component.
- o STRETCHED TEXT - Text features will not be stretched in anyway to fit between points, be it by inserting additional spaces or by breaking the text string up into individual component characters.
- o TEXT ATTRIBUTES - ARC/INFO Version 6.0 text attributes are not currently supported. Any text attribute tables (TAT) will be ignored.

The EXPORT sections ARC, LAB and TXT (or TX6) are assumed to appear in this order in the file. (ie alphabetical).

Only single .E00 files can be processed as ARC2I does not handle continuation files (.E01, .E02 etc). Therefore the 'max_lines' argument to the ARC/INFO EXPORT facility should not be invoked.

The program only makes a limited attempt to correct any discrepancies with the attribute ordering in the AAT and PAT tables and files should ideally contain unique, ordered Internal-IDs.

DESCRIPTION

Command line

The symbol ARC2I is normally set up as:

```
ARC2I == "$ls1$exe:arc2i"
```

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

Parameter File

This file contains the attribute description table which describes the translations required to produce IFF AC entries from EXPORT attribute data held in the .AAT and .PAT tables. The format of the table is described in the DATA PREPARATION section under the heading 'The ARC2I 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 the command line, the program creates a temporary file which consists of a copy of the IFO section of the EXPORT file. This is used to process the ARC and LAB sections in parallel with IFO section in case attributes from tables such as the .AAT or .PAT need to be given to features. This temporary file will have the name 'filename.ARC2I_TMP' where 'filename' is the output filename.

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 CP information being those values in the .TIC table of the IFO section. If this table contains more than four points, then a maximum/minimum box is calculated for the area.

The EXPORT ARC section is now processed to extract the arcs within the file and these are written to the IFF file with Graphical Type (GT) 1. The default feature code for these line features is FC 1 unless the parameter file was specified. If so, the feature code will be the value associated with the DEF_LINE command. If the .AAT table is present and contains attributes that are correctly defined in the parameter file, then the linear feature will be given the associated attribute in the form of an IFF AC entry.

Label points are now created using the information held in the LAB section of the EXPORT file. These symbols have GT 8 and the feature code will be the default of FC 2 or the DEF_SYMBOL in the parameter file if it was given. Any associated attributes, correctly defined in the parameter file, will be taken from the .PAT and assigned to the symbol feature as an AC entry. This will only be done if the label

point ID number taken from the LAB section matches the label point ID number in the .PAT entry. It is therefore important that the order of the LAB entries matches that of the .PAT entries if the attributes are to be maintained in the translation.

Any text features will now be extracted from the TXT section if the file was created from an ARC/INFO system before version 6.0. The orientation of the text depends on the number of points specifying the line along which it lies. One point will result in an orientation of zero. Two points will result in an orientation between the origin and the line created by these first two points. Points in excess of two are ignored as ARC2I does not create text features that follow an arc or spline.

Files created by ARC/INFO version 6.0 or later will have annotation extracted from the TX6 section of the file and appropriate text features created. See the RESTRICTIONS section for detail of the level of support for ARC/INFO version 6.0 text.

Text features will be of GT 10 and can be assigned a feature code in one of three ways. If the LEVEL and SYMBOL codes of the text feature in ARC matches an entry in the parameter file, it is given that features. If this is not the case and the DEF_TEXT value was defined in the parameter file, it will be assigned this feature code. Failing both these condition the default feature code of 3 is used.

Finally the range (RA) entry information in the IFF file is updated to reflect all points within the file which is then closed. The temporary file is closed and deleted and providing no error occurred, the user should never see this file.

EXAMPLES

\$ ARC2I/SCALE=50000/PAR=HERE:ARCINFO.DAT/FRT=HERE:ACD_TABLE MUB1:

Initializing tape MUB1: for read - Please wait.

IFF file LSL\$IF:ANNO.IFF created.
IFF file LSL\$IF:ANNOX.IFF created.
IFF file LSL\$IF:BDY.IFF created.
IFF file LSL\$IF:CHATT_SIST.IFF created.
IFF file LSL\$IF:CHATT_SITE.IFF created.
IFF file LSL\$IF:CHATT_SITEP.IFF created.
IFF file LSL\$IF:DURHAM_SIST.IFF created.
IFF file LSL\$IF:DURHAM_SITE.IFF created.
IFF file LSL\$IF:FLOOD.IFF created.
IFF file LSL\$IF:FORTO_SIST.IFF created.
IFF file LSL\$IF:H02180HY.IFF created.
IFF file LSL\$IF:H02180LC.IFF created.
IFF file LSL\$IF:H02180LU.IFF created.
IFF file LSL\$IF:H02180RD.IFF created.
End of Tape.

ELAPSED: 0 00:15:44.85 CPU: 0:03:35.43 BUFIO: 787 DIRIO: 2265 FAULTS:

This examples shows that all the files on a magnetic tape have been successfully converted to IFF. Since the /SELECT qualifier was not specified, all files on the tape have been processed until the tape mark was found. The name of each output IFF file was that of the coverage name held in each EXPORT file as it was processed. The absence of the /LOG qualifier reduced the amount of information fed back to the user.

```
$ ARC2I/LOG/SCALE=50000/PAR=HERE:ARCINFO.DAT/FRT=HERE:ACD_TABLE -  
_$ /SELECT=(8,12:14) MUB1:
```

```
Parameter file : LSL$SOURCE_ROOT:[CONVERT.ARC2I]ARCINFO.DAT;62  
FRT file       : LSL$SOURCE_ROOT:[CONVERT.ARC2I]ACD_TABLE.FRT;31  
Initializing tape MUB1: for read - Please wait.  
Looking for start of next EXPORT file.
```

```
PROCESSING file 8 on tape  
Searching file for coverage name...  
IFF file LSL$IF:DURHAM_SITE.IFF created.  
Reading arcs from ARC section of EXPORT file.  
Reading label points from LAB section of EXPORT file.  
Reading text from TXT section of EXPORT file.  
21 feature(s) written to IFF file.  
Files closed.  
Looking for start of next EXPORT file.
```

```
PROCESSING file 12 on tape  
Searching file for coverage name...  
IFF file LSL$IF:H02180LC.IFF created.  
Reading arcs from ARC section of EXPORT file.  
Reading label points from LAB section of EXPORT file.  
87 feature(s) written to IFF file.  
Files closed.
```

```
PROCESSING file 13 on tape  
Searching file for coverage name...  
IFF file LSL$IF:H02180LU.IFF created.  
Reading arcs from ARC section of EXPORT file.  
Reading label points from LAB section of EXPORT file.  
Reading text from TXT section of EXPORT file.  
794 feature(s) written to IFF file.  
Files closed.
```

```
PROCESSING file 14 on tape  
Searching file for coverage name...  
IFF file LSL$IF:H02180RD.IFF created.  
Reading arcs from ARC section of EXPORT file.  
Reading label points from LAB section of EXPORT file.  
Reading text from TXT section of EXPORT file.  
286 feature(s) written to IFF file.  
Files closed.
```

```
ELAPSED:      0 00:09:11.35  CPU: 0:01:58.68  BUFIO: 428  DIRIO: 1388  FAULTS:  
206  
$
```

This example illustrates the use of the /SELECT qualifier to process only certain files held on a tape. This may be necessary when, for example, different parameter or FRT files are needed to process certain files. It also shows the more verbose output messages given to the user with the use of the /LOG qualifier.

```
$ ARC2I/LOG/SCALE=50000/PAR=HERE:ARCINFO.DAT/FRT=HERE:ACD_TABLE TESTFILE1 TEST
```

```
Parameter file : LSL$DATA_ROOT:[ARCINFO.DATA]ARCINFO.DAT;56
FRT file       : LSL$DATA_ROOT:[ARCINFO.DATA]ACD_TABLE.FRT;27
EXPORT file SYS$DISK:TESTFILE1.E00;0 opened for read
IFF file LSL$IF:TEST.IFF created.
Reading arcs from ARC section of EXPORT file.
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
%ARC2I-W-ATTRTYPE, attribute TYPE not declared as real in FRT - ignored
Reading text from TXT section of EXPORT file.
14 feature(s) written to IFF file.
Files closed.
```

```
ELAPSED:      0 00:00:11.45  CPU: 0:00:02.56  BUFIO: 23  DIRIO: 49  FAULTS: 292
$
```

This example shows the result of processing a disk file with an output filename specified to prevent the the IFF file having the same name as the input file. It also shows a warning message being generated due to a problem with an attribute called 'TYPE'. This message means that the value of TYPE was held in the EXPORT file as a floating point number but had a conflicting datatype specified in the FRT. Usually, if there is a problem with an attribute it will not be written to the IFF file rather than writing a value that is incorrect. The message can be ignored unless the attribute 'TYPE' is desired in the output IFF file in which case the relevant action should be taken and the program run again.

MESSAGES (INFORMATIONAL)

These messages give information only, and require no immediate action by the user. They are used to provide information on the current state of the program, or to supply explanatory information in support of a warning or error message.

LINEINPAR, line 'number' of parameter file

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

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

MAXFSN, Maximum Feature Serial Number exceeded. Restarting FSNs from 1.

Explanation: This message occurs when more than 32767 features have been written to the IFF file. The Feature Serial Number of features will start from 1 again. This will result in an IFF file with duplicate FSNs.

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.

ATTFLD, attribute 'name' not defined as AC text field - ignored

Explanation: The character attribute was not defined in the parameter file as going into the AC text field. This requires that the field column of the attribute contains the command TEXT.

User action: Edit the parameter file to ensure that the attribute in question will go into the text field of the AC entry, and run the program again if required.

ATTRTYPE, attribute 'name' not declared as 'type' in FRT - ignored

Explanation: The datatype of the attribute specified in the FRT does not reflect the type stored in the EXPORT file. See the DATA PREPARATION section describing the relation between the EXPORT item type and the FRT datatypes.

User action: Amend the FRT file to contain the correct datatype for the attribute in question, and if necessary run the program again.

BADATTCODE, attribute code 'code' out of range

Explanation: An invalid IFF attribute code (outside the range 0 to 32767) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

User action: Amend the attribute code in the parameter file, and if necessary run the program again.

BADATTNAME, invalid attribute name 'name'

Explanation: An invalid IFF or EXPORT attribute name has been read from the parameter file. This message may be followed by an LSLLIB message indicating an error in reading the name. Alternatively, this message may be preceded by a message indicating why the name was not recognised. The LINEINPAR message indicating the parameter file line number will appear last, and the program will continue to read the file, ignoring this line.

User action: Amend the attribute name in the parameter file, and if necessary run the program again.

BADFCODE, feature code 'code' out of range

Explanation: An invalid IFF feature code (outside the range 0 to 32767) has been read from the parameter file. The LINEINPAR message indicating the parameter file line number will follow, and the program will continue to read the file, ignoring this line.

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

BLKSIZ, Magtape block size defaulted to 'number' bytes.

Explanation: The /BLOCKSIZE was missing or too large when used with a magtape device.

User action: If the defaulted blocksize is that at which the tape was written then no user action is necessary. If it is not, abort the program and try again using the correct block size.

DEFAULTACD, cannot find attribute code 'code', using default values

Explanation: The given IFF attribute code (as read from the parameter file) could not be found amongst the current definitions. Default values will be used instead.

User action: Ensure that the appropriate ACD definition appears in the FRT file if you are using one. Otherwise, amend the attribute code in the parameter file if it is in error. If necessary run the program again.

DEFAULTFC, No text FC for LEVEL 'integer' and SYMBOL 'integer' in parameter file

Explanation: A feature code for the text feature with the reported LEVEL and SYMBOL numbers was not found in the parameter file and so the default text feature code was used. This default for text features is defined in the parameter file by the DEF_TEXT command or 3 if not defined. This message will only occur for ARC/INFO v6.0 annotation and if the /LOG qualifier was used.

User action: If the default feature code given to the text feature is satisfactory then this message can be ignored. If not, amend the parameter file to include an entry for the reported LEVEL and SYMBOL number and if necessary run the program again.

ERRDATE, error in attribute 'name' date value 'string' - using default

Explanation: One or more of the date components in the given string could not be read. The attribute is given the minimum default date of 17-NOV-1858. This error will occur if the date string contains components that cannot be converted into integers; a typical example being spaces. The attribute name given may enable the offending date to be located.

User action: The date string may be corrected and the program run again if required.

MISSCMD, missing command after "'command'"

Explanation: The message is given when the ANNO command in the parameter file was expected to be followed by another which it was not. 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: This message is most likely due to an error in the parameter file which should be corrected and the program run again if necessary.

NOARCATT, no attribute(s) given to arc with ARC ID 'number'

Explanation: No attributes were assigned to the arc with the given internal ID number. This is because the internal ID number in the ARC section does not match that in the AAT table.

User action: Ensure that the EXPORT file being processed contains unique, ordered internal IDs. This should be possible to do in the ARC/INFO system before writing the file to tape.

NOATTNAME, cannot read attribute name

Explanation: An IFF or EXPORT attribute name cannot be read where expected from the parameter file. This message will be followed by an LSLLIB message indicating why the name could not be read. The LINEINPAR message indicating the parameter file line number will also follow, and the program will continue to read the file, ignoring this line.

User action: Edit the parameter file to include the attribute name in the parameter file where expected, and if necessary run the program again. In the unlikely event that no errors are found in the parameter file, please report the problem to Laser-Scan.

NOCP, insufficient or missing corner point information in EXPORT file

Explanation: This warning message occurs when the input EXPORT file contains less than 4 corner points in the .TIC table the IFO section, or when the .TIC table is missing. The values in the IFF CP entry file will default to zero.

User action: If necessary, ensure that the .TIC table is present in the IFO section and that it contains at least 4 corner points.

NOPAR, no parameter file specified - any attributes ignored

Explanation: The parameter file was not specified on the command line and any attributes found in the EXPORT file will not be included in the translation but will be ignored.

User action: If the EXPORT attributes are desired in the translation, a parameter file must be created and specified on the command line. See the relevant documentation on the parameter file.

NOPNTATT, no attribute(s) given to point with LAB ID 'number'

Explanation: No attributes were assigned to the point with the given internal ID number. This is because the internal ID number in the LAB section does not match that in the PAT table.

User action: Ensure that the EXPORT file being processed contains unique, ordered internal IDs. This should be possible to do in the ARC/INFO system before writing the file to tape.

OUTFILIGNORE, output filename ignored when used with more than one tape file

Explanation: An output file specification was given on the command line for an input file on tape. This is permitted when /SELECT has one argument but any more and it is ignored and the name for the output file will be the EXPORT coverage name held in each file on the tape.

User action: None.

QUALIGNORE, /SELECT qualifier ignored. Only for use with tape

Explanation: As /SELECT does not apply to disk files it will be ignored.

User action: None.

TXTPTS, text points ignored - first two used only

Explanation: A text feature held in the EXPORT file contained more than 2 points which define the curve along which it lies. At present ARC2I only uses the first one or two points to give the text feature no orientation or an orientation equal to the angle between the origin and the first two points.

User action: None.

UNDEFATT, attribute 'name' undefined for 'export' table - ignored

Explanation: An attribute was found which has not been associated with the .PAT or .AAT table in which it appears in the EXPORT file. This is done by having an entry in the parameter file for the attribute such that the table columns contains 'PAT' or 'AAT' depending on the table in which it appears. If the attribute appears in both the .AAT and the .PAT table then two entries must be given. They may be of different datatypes and this is specified in the FRT and is reference through the parameter file by attribute name or code.

User action: Edit the parameter file to ensure that the table column of the offending attribute reflects the table in which it occurs, and if necessary run the program again.

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: This message is most likely due to an error in the program and should be reported to Laser-Scan.

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.

ERRBCKSPC, error backspacing through tape file

Explanation: An error occurred while rewinding the file specified. The associated LSLLIB and system error messages may clarify the problem.

User action: The user should not normally encounter this error, however the associated LSLLIB or system error message may indicate the problem. If the problem is not apparent from these messages please contact Laser-Scan.

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

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

User action: Depends upon the associated LSLLIB message.

ERRIFF, error writing an entry to the IFF output file

Explanation: An internal error was detected when writing an IFF entry to the output IFF file.

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

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

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

User action: Depends upon the associated LSLLIB messages.

ERRRDFRT, error reading FRT file 'file-spec'

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

User action: Dependent upon the associated error messages.

ERRRDNUM, error reading number from 'location'

Explanation: Some form of error occurred when reading an integer or real from the specified part of the EXPORT file.

User action: Use a text editor to examine the input EXPORT file and ensure that the suspect section within the EXPORT file is not corrupted.

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.

ERRRDSTR, error reading string from 'location'

Explanation: Some form of error occurred when reading an string from the specified part of the EXPORT file.

User action: Use a text editor to examine the input EXPORT file and ensure that the suspect section within the EXPORT file is not corrupted.

ERRRWD, error rewinding to file "'file-spec'"

Explanation: An error occurred while rewinding the file specified. The associated LSLLIB and system error messages may clarify the problem.

User action: The user should not normally encounter this error, however the associated LSLLIB or system error message may indicate the problem. If the problem is not apparent from these messages please contact Laser-Scan.

ERRWRT, error writing to file "'file-spec'"

Explanation: An error occurred while writing to the file specified. The associated LSLLIB and system error messages may clarify the problem.

User action: The user should not normally encounter this error, however the associated LSLLIB or system error message may indicate the problem. If the problem is not apparent from these messages please contact Laser-Scan.

INVFILNAM, invalid filename "'file-spec'"

Explanation: The EXPORT input file specification is not valid. The specified filename will not parse against the given skeleton: SYS\$DISK:.E00;0.

User action: Specify a correct filename including device name if necessary.

MAGTAPE, error initialising magtape device "'device-name'"

Explanation: An error occurred in initialising the magnetic tape drive for reading. The associated LSLLIB and system error messages should clarify the problem.

User action: Dependent upon the associated messages. One possible cause of problems is that the tape has not been mounted.

MISSINGINFO, Insufficient basic information in 'export' table

Explanation: The EXPORT .AAT table contained less than its 7 basic components or the .PAT table contained less than its 4 basic components.

User action: Ensure that both the .AAT and .PAT tables in the IFO section contain their basic 7 or 4 components respectively.

NOCOVERNAM, no coverage name in EXPORT file

Explanation: The EXPORT input file does not contain a coverage name which must be used to name the output files when processing several files on magnetic tape. This is the name used to denote each table within the IFO section. For example, the .AAT table begins with 'coverage.AAT' and the .PAT table begins with 'coverage.PAT' etc.

User action: Ensure that the EXPORT file contains a coverage name. This means in effect that the IFO section must contain at least one table. A .AAT table should be present if there are points in the ARC section and a .PAT table should be present if there are points in the LAB or CNT sections.

NODEVICE, unknown device name "'device'" given

Explanation: The device name given on the command line was not recognised.

User action: Specify a legal device name on the command line.

NOEXP, No EXP entry on first line of EXPORT file

Explanation: The EXPORT file did not contain an EXP entry on the first line. This is essential as it denotes the start of a new file. A possible cause may be that the EXPORT file is a continuation file (E01 or E02 etc) which is not permitted.

User action: Ensure that this line exists as the first in the EXPORT file and that it is not a continuation file.

NOFILES, no more files exist on the tape

Explanation: A file selected using /SELECT has a position number greater than the number of files on the tape. Any existing files converted and closed will not be affected by this error.

User action: If all files were converted as expected then this error can be ignored. Otherwise ensure that the /SELECT list accurately reflects the files on tape to be converted.

NOFILNAM, no filename in EXPORT file specification "'file-spec'"

Explanation: The specified EXPORT input file-spec is a valid device name but the filename has not been specified.

User action: Rerun ARC2I with the correct EXPORT filename and device specification.

NOFRT, specified FRT file does not exist

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

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

NOIFO, no IFO section in EXPORT file "'file-spec'"

Explanation: The specified EXPORT input file has no IFO section.

User action: Ensure that ARC2I is only performed on EXPORT files that contain an IFO section.

NOPARFILE, specified parameter file does not exist

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

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

NORA, no RA entry could be found in IFF file

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

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

NOTMNTFOR, magtape device "'device'" must be mounted foreign

Explanation: The specified EXPORT input file-spec has been parsed as a magtape device but the tape is not mounted foreign.

User action: Mount the tape with the MOUNT/FOREIGN command and rerun ARC2I.

RDLNEXP, error reading a line from input EXPORT file

Explanation: An error occurred while reading a line from the EXPORT file. This will be followed by an LSLLIB and possibly a system error message.

User action: The user should not normally encounter this error, however the associated LSLLIB or system error message may indicate the problem. If the problem is not apparent from these messages please contact Laser-Scan.

RDLNTMP, error reading a line from temporary file

Explanation: An error occurred while reading a line from the temporary file. This will be followed by an LSLLIB and possibly a system error message.

User action: The user should not normally encounter this error, however the associated LSLLIB or system error message may indicate the problem. If the problem is not apparent from these messages please contact Laser-Scan.

UNEXPEOF, unexpected end-of-file while reading EXPORT file

Explanation: The end-of-file marker was read from the EXPORT file when more line were expected.

User action: Ensure that the input EXPORT file has complete information in both the ARC and the INFO sections.

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.