

Penmap/Map500 UNIVERSAL EXPORT FORMAT

Information

Issue date: 8th April 2005

Detailing file versions up to 1.1

Please note that the information in this document is subject to change as this format is still under development, however any changes should be of a minor nature.

To receive the most recent copy of this document please email a request to:

ErikSchuetz@penmap.com

stating the issue date on your current copy.

INDEX

- 1 INTRODUCTION
 - 1.1 BASIC DATA FORMATS
- 2 HEADER SECTION
3. NODE SECTION
 - 3.1 INITIAL NODE DATA LINE
 - 3.2 SPECIAL NODE DATA LINES
 - 3.2.1 NODE OBSERVATION DATA
 - 3.3 ADDITIONAL NODE DATA LINES
 - 3.4 FREE NODE
 - 3.5 BILATERATION NODE
 - 3.6 TRILATERATION NODE
 - 3.7 CHAIN AND OFFSET NODE
 - 3.8 EXTEND NODE
 - 3.9 INTERSECTION NODE
 - 3.10 NORMAL NODE
 - 3.11 PARALLEL NODE
 - 3.12 TOTAL STATION NODE
 - 3.13 TOTAL STATION VERIFY NODE
 - 3.14 RESECTION NODE
 - 3.15 BEARING AND DISTANCE NODE
 - 3.16 TRAVERSE NODE
 - 3.17 GPS NODE
 - 3.18 GPS VERIFY NODE
 - 3.19 ROADWHEEL NODE
 - 3.20 CONSTRUCTION NODE
 - 3.21 REFERENCE OBSERVATIONS
- 4 VERIFY SECTION
5. GRAPHICS SECTION
 - 5.1 COMMON GRAPHICS FIELDS
 - 5.2 GRAPHIC POINT
 - 5.3 GRAPHIC PERMANENT STATION
 - 5.4 SINGLE POINT CIRCLE (Centre point and radius)
 - 5.5 TWO POINT CIRCLE (Centre and circumference points)
 - 5.6 THREE POINT CIRCLE (Three circumference points)
 - 5.7 SINGLE BEZIER CURVE
 - 5.8 COMPOUND BEZIER CURVE
 - 5.9 GRAPHIC ARC
 - 5.10 TANGENTIAL ARC
 - 5.11 GRAPHIC LINE
 - 5.12 SINGLE POINT SYMBOL
 - 5.13 TWO POINT SYMBOL
 - 5.14 SINGLE POINT TEXT
 - 5.15 TWO POINT TEXT
 - 5.16 THREE POINT TEXT
 - 5.17 AXIALLY ALIGNED RECTANGLE
 - 5.18 THREE POINT ALIGNED RECTANGLE
 - 5.19 TWO POINT RECTANGLE WITH HEIGHT
6. GISDEFINITIONS
 - 6.1 FIELD DEFINITIONS
 - 6.2 DATALIST DEFINITION
 - 6.3 OPTION TREE DEFINITION
7. GISRECORD SECTION
8. LINSTYLE SECTION
9. LAYER SECTION
10. POINTSTYLE SECTION

- 11. DIMENSION SECTION
 - 11.1 LINEAR DIMENSION
 - 11.2 ANGLE DIMENSION
 - 11.3 BEARING DIMENSION
- 12. SYMBOLS SECTION
 - 12.1 INITIAL DATA LINE
 - 12.2 ELEMENT DATA LINES
 - 12.2.1 SYMBOL POINT ELEMENT
 - 12.2.2 SYMBOL CIRCLE ELEMENT
 - 12.2.3 SYMBOL ARC ELEMENT
 - 12.2.4 SYMBOL LINE ELEMENT
 - 12.2.5 SYMBOL POLYLINE ELEMENT
 - 12.2.6 SYMBOL TEXT ELEMENT
 - 12.2.7 SYMBOL COMPOUND BEZIER ELEMENT
- 13. APPENDIX 1 - SAVING FROM PENMAP/MAP500
- 14. APPENDIX 2 - LOADING INTO PENMAP/MAP500
- 15. APPENDIX 3 - IMPORT POSSIBILITIES
- 16. APPENDIX 4 – VERSION DIFFERENCES

1. INTRODUCTION

This format is designed for the complete import/export of a Penmap/Map500 survey.

Universal import/export files have the file extension '.unv'

It is in an ASCII comma separated format.

Each line of the file is terminated with a carriage return character followed by a line feed character.

No single line can exceed 2000 characters.

The universal export file is divided into a number of sections.

Each section has a header line and a tailer line, for example the survey points (nodes) section is preceded by a single line:

NODES

and terminated by a single line:

ENDNODES

Data lines within all sections start with an 'Index' field.

The index field starts with a single letter prefix to identify the type of object to which the data line refers, followed by a unique number for the object.

For header data the prefix letter is 'H'.

For nodes (survey points) the prefix letter is 'N'.

For graphics the prefix letter is 'G'.

For GIS recordobjects the prefix letter is 'D'.

For GIS database definitions the prefix letter is 'X'

For Layer definitions the prefix letter is 'L'.

For Linestyle definitions the prefix letter is 'S'

For Symbol definitions the prefix letter is 'M'.

For Dimensioning definitions the prefix letter is 'K'.

For Pointstyle definitions the prefix letter is 'P'.

For DTM definitions the prefix letter is 'T'.

These prefixes are used throughout the data structure. For example the NODES section of the file might contain a node with the index 'N211'. Any other data line in any other section can contain a reference to this node by simply referring to 'N211'.

1.1 BASIC DATA FORMATS

STRINGS

String values will be enclosed in double parenthesis, for example, "some text" .

FLAGS

Certain values have a 'True' or 'False' value only. Within this document such values are referred to as flags. Within the 'unv' file these values will be represented by **T** for True, and by **F** for False.

NUMBERS

Redundant leading and trailing zeros should be suppressed.

Certain values can be invalid in some circumstances. Where this is the case the data will be entered as NAN (abbreviation for 'Not A Number') and a value of zero can be assumed for such an entry.

Certain values (i.e RGB vales) are more normally expressed in hexadecimal. All hexadecimal numbers must be prefixed with '\$', for example:

\$FF0080

TIME

Time values are stored with leading zeros suppressed, 24 hour clock as HH:MM:SS.HH for example:

9:27:5.78

15:32.17.09

DATE

Date values are stored with leading zeros suppressed as DD/MM/YYYY for example:

11/3/1998

25/12/1990

2. HEADER SECTION

The HEADER section contains general information about the survey.

This section starts with the header line:

HEADER

and ends with the tailer line

ENDHEADER

All general information must be defined between this header and tailer.

Currently there are two compulsory data lines defined for the header section which will typically look like this:

H0,"Penmap",0,2

This consists of:

A index/descriptor (always 'H1' for this information),

The name of the program that created this file,

The file major version number (an integer between 0 and 999),

The file minor version number (an integer between 0 and 9),

H1,"gfx","METRES",-455,-450,545,550,20:25:11.84,2/11/1998

This consists of:

A index/descriptor (always 'H1' for this information),

The survey name,

The survey units (as a text string),

The survey area westmost extent,

The survey area southmost extent,

The survey area eastmost extent,

The survey area northmost extent,

The survey time,

The survey date.

Universal files below version 1.0 may have a DIMENSIONS section included in the header which sets dimensioning parameters. Versions 1.0 and above use the “descriptor H4” version which is defined later. The “descriptor H2” version will typically look like this:

H2,"Arial",\$FF,0,0,3,1,2,0,3,2,0,1,0,0,0,128,0,T,0,24929,562987693768710.125,T,T

This consists of:

A index/descriptor (always ‘H2’ for this information),
The font name,
The dimensioning colour,
The dimensioning line style 0 (Solid), 1 (Dash), 2 (Dot), 3 (DashDot), or 4 (DashDotDot),
The dimension display 0 (Map values only), 1 (Measured values only), or 2 (All values),
The arrowhead length,
The arrowhead width ,
The display state 0 (Hide), 1 (Lines only), or 2 (Lines + Text)
The gap between arrowhead and ref. Line in pixels,
The text height,
The text width,
Angular dimension line radius,
The text gap in pixels,
The fixed standard deviation,
The ‘per units’ standard deviation,
The angular standard deviation,
The net count,
The net tolerance,
The dimrec^.showindex [REQUIRES ELABORATION]
The dimrec^.netredunmin, [REQUIRES ELABORATION]
The dimrec^.netellipsc, [REQUIRES ELABORATION]
The dimrec^.netellipscsz, [REQUIRES ELABORATION]
The dimrec^.netellipsh), REQUIRES ELABORATION]
The dimrec^.netellipAPRIORI, [REQUIRES ELABORATION],

If the survey contains captured images, the following line may be present in the header and is used internally by Penmap/MAP500 to derive image names.

H3,1,0

This consists of:

A index/descriptor (always ‘H3’ for this information)
A parameter count (currently 1)
1st parameter – ImageNameBase number (same as value set in Penmap.ini for imagenamebase)

If the DIMENSIONS section is included in Universal files from 1.0 onwards, then an additional header line should be present which sets dimensioning parameters. It will typically look like this:

H4,"Arial",2,1.5,3,2,0,2,1,2,1,0.002,0,0.0008,20,0.001,T,0.001,9,90,T,T,\$FF0000,\$FF0000,0,\$0,\$408000,1,\$0,\$8000,1,F

This consists of:

A index/descriptor (always 'H4' for this information),

The font name,

The dimension display 0 (Map values only), 1 (Measured values only), or 2 (All values)

The arrowhead length,

The arrowhead width,

The display state 0 (Hide), 1 (Lines only), or 2 (Lines + Text)

The gap between arrowhead and ref. Line (now in survey units),

The text height,

The text width,

Angular dimension line radius,

The text gap (now in survey units),

The fixed standard deviation,

The 'per units' standard deviation,

The angular standard deviation,

The net count,

The net tolerance,

The dimrec^.showindex [REQUIRES ELABORATION]

The dimrec^.netredunmin, [REQUIRES ELABORATION]

The dimrec^.netellipsc, [REQUIRES ELABORATION]

The dimrec^.netellipsz, [REQUIRES ELABORATION]

The dimrec^.netellipsh, [REQUIRES ELABORATION]

The dimrec^.netellipAPRIORI, [REQUIRES ELABORATION],

The dimensioning text colour,

The dimensioning line colour,

The dimensioning line style 0 (Solid), 1 (Dash), 2 (Dot), 3 (DashDot), or 4 (DashDotDot),

The offset dimensioning text colour,

The offset dimensioning line colour,

The offset dimensioning line style 0 (Solid), 1 (Dash), 2 (Dot), 3 (DashDot), or 4 (DashDotDot),

The chain dimensioning text colour,

The chain dimensioning line colour,

The chain dimensioning line style 0 (Solid), 1 (Dash), 2 (Dot), 3 (DashDot), or 4 (DashDotDot),

The text "white background" flag

3. NODE SECTION

Nodes are the points contained in a survey. All node data is contained in a single section. All node data follows a uniform convention.

This section starts with the header line:

NODES

and ends with the tailer line

ENDNODES

All node information must be defined between this header and tailer.

A node may require several data lines to define it fully, but EVERY node has a common initial data line which gives basic information about the node. The initial line also indicates the form of the additional lines which are required to fully define the node.

3.1 INITIAL NODE DATA LINE

This initial line has the following format:

Index, Descriptor, Status, Type, Easting, Northing, Level

Index - This is a unique identifier for the node, it starts with 'N' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the data which follows it. In the case of the node initial line this number is always '1'. The descriptor field can have values between 0 and 65535 for other types of node information which follow the initial data line for that node.

Status - This is a number used to indicate the status of the node, this can have a value 0,1,2 or 3 which refers to OLD, NEW, CHANGED or DELETED.

Type - This is a number between 0 and 65535 which is used to identify the type of the node. This field is used to determine the additional data lines (if any) which will follow to fully define the node. The current possible values for the node type are:

- 1 = Free node
- 2 = Bilateration node
- 3 = Trilateration node
- 4 = Chain and offset node
- 5 = Extend node
- 6 = Intersection node
- 7 = Normal node
- 8 = Parallel node
- 10 = Total station node
- 11 = Resection node
- 12 = Bearing node
- 13 = Traverse node
- 14 = GPS node
- 15 = Roadwheel node
- 17 = Verify node

Easting, Northing, Level - These values are in survey units.

Example of an initial node line for an 'old' bilateration node:

N211,1,0,2,306941.7907,292422.6170,206.0975

3.2 SPECIAL NODE DATA LINES

NODE OBSERVATION DATA

A node may have additional data derived from the Penmap/Map500 node observation system. This data is application specific and is effectively a 'mini-database' attached to the node.

If node observation data is present then this will appear anywhere (usually as the second line which defines a node) after the initial node data line. Node observation data is fully defined by a single data line. An example of a node observation data is:

N241,0,"7,2,RERS,,Road edge right"

The line is constructed as follows:

The unique node index (same as for the initial data line),

The descriptor (value of 0 for node observation data),

Node observation 'export string'

An initial node data line for the node must exist for this special data to bear any significance.

Please note that the observation data is only used in some applications and will be absent for many surveys. The 'export string' is the standard import/export string for the particular observation system in use. These will be detailed in the relevant literature on specific observation systems.

3.3 ADDITIONAL NODE DATA LINES

Important: *None of the following data line descriptions contain node observation data lines.*

With the exception of the 'free node' which is fully defined within the initial node line, all nodes require additional data lines to fully define them.

All additional node data lines start off with two common fields which determine the format of the data which follows. These two common fields are:

Index, Descriptor,

Index - This is the unique identifier for the node as determined by the initial data line for that node.

Descriptor - This is a number which is used to identify the form of the data which follows the descriptor.

Where a node is fully defined by just one additional data line, the descriptor is numerically identical to the 'type' field for the initial data line for that node.

Where multiple data lines are needed to define the node, the descriptor will vary, but the least significant two digits of the descriptor will be numerically identical to the 'type' field for the initial data line for that node.

Currently defined descriptors are as follows:

- 0 = Node observation data
- 2 = Bilateration node
- 3 = Trilateration node
- 4 = Chain and offset node
- 5 = Extend node
- 6 = Intersection node
- 7 = Normal node
- 8 = Parallel node
- 10 = Total station node
- 1010 = Total station reference observation
- 11 = Resection node
- 1011 = Resection node reference observation
- 12 = Bearing node
- 13 = Traverse node
- 1013 = Traverse node reference observation
- 14 = GPS node
- 1014 = GPS node additional
- 2014 = GPS node data extra
- 3014 = GPS node mean correction data
- 15 = Roadwheel node
- 1015 = Roadwheel node list
- 17 = Total StationVerify node
- 18 = GPS Verify node

3.4 FREE NODE

A free node is fully defined by the initial data line for the node. An example of a free node is:
N211,1,0,1,306941.7907,292422.6170,206.0975

3.5 BILATERATION NODE

A bilateration node is fully defined by two data lines. An example of a bilateration node is:
N2545,1,0,2,306864.1189,292491.1416,0.0000
N2545,2,25.0000,24.0000,N1285,N1315

The first line is the initial data line for the node. The following line is constructed as follows:
The unique node index (same as for the initial data line),
The descriptor (value of 2 for a bilateration node),
Distance to first reference node in survey units,
Distance to second reference node in survey units,
Unique index of first reference node,
Unique index of second reference node.

3.6 TRILATERATION NODE

A trilateration node is fully defined by two data lines. An example of a trilateration node is:
N2651,1,0,3,306872.6305,292480.5328,0.0000
N2651,3,-25.0000,24.0000,12.0000,N1285,N1315,N2545

The first line is the initial data line for the node. The following line is constructed as follows:
The unique node index (same as for the initial data line),
The descriptor (value of 3 for a trilateration node),
Distance to first reference node in survey units,
Distance to second reference node in survey units,
Distance to third reference node in survey units,
Unique index of first reference node,
Unique index of second reference node,
Unique index of third reference node.

3.7 CHAIN AND OFFSET NODE

A chain and offset node is fully defined by two data lines. An example of a chain and offset node is:
N2792,1,0,4,306959.6281,292519.1371,0.0000
N2792,4,10.6000,5.7000,3.3000,N1555,N1585

The first line is the initial data line for the node. The following line is constructed as follows:
The unique node index (same as for the initial data line),
The descriptor (value of 4 for a chain and offset node),
Chain distance in survey units,
Offset distance in survey units,
Measured distance in survey units,
Unique index of first reference node,
Unique index of second reference node.

3.8 EXTEND NODE

An extend node is fully defined by two data lines. An example of an extend node is:

N2905,1,0,5,306899.7241,292518.0202,0.0000
N2905,5,10.4500,N1495,N1465

The first line is the initial data line for the node. The following line is constructed as follows:

The unique node index (same as for the initial data line),
The descriptor (value of 5 for an extend node),
Extend distance in survey units,
Unique index of first reference node,
Unique index of second reference node.

3.9 INTERSECTION NODE

An intersection node is fully defined by two data lines. An example of an intersection node is:

N3062,1,0,6,306861.9823,292448.7193,0.0000
N3062,6,N3032,N3002,N1405,N1435

The first line is the initial data line for the node. The following line is constructed as follows:

The unique node index (same as for the initial data line),
The descriptor (value of 6 for an intersection node),
Unique index of first reference node,
Unique index of second reference node,
Unique index of third reference node,
Unique index of fourth reference node.

3.10 NORMAL NODE

A normal node is fully defined by two data lines. An example of a normal node is:

N3412,1,0,7,306951.8389,292487.3813,0.0000
N3412,7,N1525,N1555,N1615

The first line is the initial data line for the node. The following line is constructed as follows:

The unique node index (same as for the initial data line),
The descriptor (value of 7 for a normal node),
Unique index of first reference node,
Unique index of second reference node,
Unique index of third reference node.

3.11 PARALLEL NODE

A parallel node is fully defined by two data lines. An example of a parallel node is:

N3681,1,0,8,307020.0000,292395.0000,0.0000
N3681,8,0.0000,N1915,N1945,N3529

The first line is the initial data line for the node. The following line is constructed as follows:

The unique node index (same as for the initial data line),
The descriptor (value of 8 for a parallel node),
Distance in survey units from third reference node,
Unique index of first reference node,
Unique index of second reference node,
Unique index of third reference node.

3.12 TOTAL STATION NODE

A total station node node is fully defined by a variable number of data lines. Three data lines are required for the basic information defining the node, additional data lines define the reference observations for the node. An example of a total station node is:

```
N30491,1,0,10,415581.5026,433035.0899,-0.2319
N30491,10,N25981,1,1,0,1.6,0,0,303.79694,90.49417,38.659,F,55.1924,4,10,47,54,89,1998,5,28,4,0
N30491,2010,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
N30491,1010,1,-1,F,F,14.84778,90.54694,33.909,1.5,N23371,0,0,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
N30491,1010,2,-1,F,F,335.21083,90.41111,31.779,1.5,N23281,0,0,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
N30491,1010,3,-1,F,F,317.00722,90.47639,37.949,1.5,N23221,0,0,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
N30491,1010,4,-1,F,F,266.81944,89.80278,17.132,1.5,N22831,0,0,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
```

The first line is the initial data line for the node.

The second line is constructed as follows:

```
N30491,10,N25981,1,1,0,1.6,0,0,303.79694,90.49417,38.659,F,55.1924,4,10:47:54.89,28/5/1998,0
```

The unique node index (same as for the initial data line),
The descriptor (value of 10 for a total station node),
The node identifying the total station position,
The stationtype which can have the values 0 (Fixed), 1 (Free), or 2 (Check),
The scaling factor,
The theodolite height,
The staff height,
The left/right offset used when prism could not be put exactly on point,
The forward/backward offset used when prism could not be put exactly on point,
The horizontal angle,
The vertical angle,
The distance from the total station to the prism,
The 'has forced correction' flag,
The correction value,
Count of reference observations,
The time of creation of the node,
The date of creation of the node,
Instrument position check status 0 (None), 1 (OK), or 2 (FAIL).

SEE NEXT PAGE FOR DESCRIPTION OF THIRD DATA LINE

The third line is constructed as follows:

N30491,2010,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0

The unique node index (same as for the initial data line),
The descriptor (value of 2010 for a total station node),
The prism offset,
The secondary scaling factor,
The standard deviation constant for distance measurement error,
The standard deviation rate (per m or per ft) for distance measurement error,
The standard deviation constant for angle measurement functions,
The standard deviation constant for the instrument centering error,
The 'Use nachbarschaftstreue Anpassung values' flag,
The dxnachbar value,
The dynachbar value,
The nachbar exponent,
The temperature (in degrees Celcius),
The pressure (in Kpa),
The mean sea level height in metres from sea level,
The Earth radius in metres,
The 'Use curvature effect on distance measurement' flag,
The weather model 0 (No model), 1 (Old model), 2 (New model),
The level weighting constant,
Instrument position check status 0 (None), 1 (OK), or 2 (FAIL).

FOR A DESCRIPTION OF ADDITIONAL REFERENCE OBSERVATION LINES SEE THE
SUBSECTION ENTITLED 'REFERENCE OBSERVATIONS'

3.13 TOTAL STATION VERIFY NODE

This node type is almost identical to the Total Station node. It differs from a Total Station node in the following respects only:

1. In the initial data line, the Type field (4th field) will be 17 rather than 10
2. In the second line, the Descriptor field (2nd field) will be 17 rather than 10

All other data lines will follow the same form as a Total Station Node

An example of a Total Station Verify node would be:

```
N30491,1,0,17,415581.5026,433035.0899,-0.2319
N30491,17,N25981,1,1,0,1.6,0,0,303.79694,90.49417,38.659,F,55.1924,4,10,47,54,89,1998,5,28,4,0
N30491,2010,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
N30491,1010,1,-1,F,F,14.84778,90.54694,33.909,1.5,N23371,0,0,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
N30491,1010,2,-1,F,F,335.21083,90.41111,31.779,1.5,N23281,0,0,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
N30491,1010,3,-1,F,F,317.00722,90.47639,37.949,1.5,N23221,0,0,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
N30491,1010,4,-1,F,F,266.81944,89.80278,17.132,1.5,N22831,0,0,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0
```

3.14 RESECTION NODE

A resection node is fully defined by a variable number of data lines. Two data lines are required for the basic information defining the node, additional data lines define the reference observations for the node. An example of a resection node is:

```
N25981,1,0,11,415582.1844,432996.4383,1.7015  
N25981,11,T,0,0.7614,0.4343,0.1554,415582.1844,432996.4383,1.7015,"Tienstra",1,55.1924,0,0.1025,28/5/1998,F,0
```

The first line is the initial data line for the node.

The second line is constructed as follows:

The unique node index (same as for the initial data line),

The descriptor (value of 11 for a resection node),

The 'Resection calculation valid' flag,

The total station height,

The easting,northing,level of standard deviation error,

The easting,northing,level of the resection coordinate. Invalid if 'Resection calculation valid' flag is

False,

The Algorithm name,

The scaling value,

The orientation of the instrument,

The standard deviation of the scaling factor,

The standard deviation of the orientation,

The date of creation of the node,

The 'Scale is calculated' flag,

The count of reference observations which follow.

FOR A DESCRIPTION OF ADDITIONAL REFERENCE OBSERVATION LINES SEE THE
SUBSECTION ENTITLED 'REFERENCE OBSERVATIONS'

3.15 BEARING AND DISTANCE NODE

A bearing and distance node is fully defined by two data lines. An example of a bearing and distance node is:

N301,1,0,12,51.5895,84.5925,0
N301,12,15,25.46,N211

The first line is the initial data line for the node. The following line is constructed as follows:
The unique node index (same as for the initial data line),
The descriptor (value of 12 for a bearing and distance node),
The bearing angle,
Distance in survey units from the reference node,
Unique index of reference node.

3.16 TRAVERSE NODE

AT THIS POINT IN TIME THE TRAVERSE NODE EXPORT IS UNDER DISCUSSION. EXPORT FILE VERSION 0.2 WILL EXPORT ALL TRAVERSE NODES AS FREE NODES.

[REQUIRES ELABORATION - DISCUSSION NEEDED]

1013 = Traverse node reference observation

3.17 GPS NODE

*INTRODUCED IN TEST FORM IN EXPORT FILE VERSION 0.3,
CORRECTLY IMPLEMENTED IN EXPORT FILE VERSION 0.4.
EARLIER VERSIONS EXPORT ALL GPS NODES AS FREE NODES.*

A GPS node is fully defined by two, three, four, or five data lines. An example of a GPS node is:

```
N2141,1,0,14,306891.4307,292425.7207,0.7837
N2141,14,"UK Ordnance Survey","Ord Sur Brit '36-
Mean",F,306891.4307,292425.7207,0.7837,0.9166710198,-0.058860077,0.7837459855,2,N211
N2141,1014,2,2,0.19,0.269,-1,-1,-1,-1,3.3,2.83,4.3473,1,6,0,1,51000
N2141,5014,4,0.7165,0.3946,0.4569,0.0861,0.0078,0.0092,0.0228,0.0341,0
The first line is the initial data line for the node.
```

The second line is constructed as follows:

The unique node index (same as for the initial data line),
The descriptor (value of 14 for a GPS node),
The Projection Name string,
The Spheroid Name string,
The 'IsCalibrated flag',
Point Coordinate Easting,
Point Coordinate Northing,
Point Coordinate Level,
Point Latitude in radians,
Point Longitude in radians,
Point Altitude,

Collection Mode which can be:

- 0 = STATIC MEAN
- 1 = STATIC MEAN CORRECTED
- 2 = STATIC LAST
- 3 = AT DISTANCE INTERVALS
- 4 = AT TIME INTERVALS
- 5 = MOVE SYMBOL
- 6 = MOVE ROTATE SYMBOL
- 7 = OFFSET
- 8 = RELATIVE TO PERMANENT STATION

BaseStation Node Index

The third (optional) line is defined as follows:

The unique node index (same as for the initial data line),

The descriptor (value of 1014 for a GPS node),

The format type (Currently always 1, or 2), if this value is 2 then the fourth node data line should be present.

The RMS type which can be

0 : None - there are no RMS error values

1 : ENA - RMS entries for Easting, Northing, and Altitude error

2 : HV - RMS entries for Horizontal and Vertical error

3 : ENAC - RMS entries for Easting, Northing, Altitude, and Covariance error

An array of 6 RMS values (currently a maximum of four of these are used, set to -1 if unused),

The vertical dilution of precision value,

The horizontal dilution of precision value,

The positional dilution of precision value,

The RTK Age - the delaytime when the measurement was taken,

The svsgps - number of GPS satellites,

The svsglo - number of Glonass satellites,

The timetype which can be:

0 : None,

1 : UTC (Followed by a UTCTIME value)

2 : Week/Sec (Followed by a week number, and a Seconds value)

The fourth (optional, see “format type” in third data line) line is defined as follows:

The unique node index (same as for the initial data line),

The descriptor (value of 5014 for a GPS node),

The position mode which can be

0 : Unknown

1 : Autonomous Mode

2 : DGPS mode

3 : RTK (code mode)

4 : DGPS (float mode)

5 : RTK (fixed mode)

then 9 covariance values:

dx

dy

dz

xx

yy

zz

xy

xz

yz

If the GPS receiver is not providing covariance data, due for example to lack of RTK then the nine covariance values will all be zero.

The fifth (optional) data line line is defined as follows:

The unique node index (same as for the initial data line),

The descriptor (value of 6014 for a GPS node),

Count of flags in this data line,

Flag_1,

Flag_2,

...Flag_n

Defined flags so far are..

Flag_1 – Has been “Post Process Corrected” flag.

Flag_2 – Differential correction state which can be

0 : Unknown

1 : Not differentially corrected

2 : Is differentially corrected

The sixth (optional) data line is defined as follows:

The unique node index (same as for the initial data line),
The descriptor (value of 8014 for a GPS node),
The PDOP for this node,
The 2D RMS for this node,
The PDOP limit set when this node was placed,
The 2D RMS limit set when this node was placed,
The elevation mask in force when this node was placed.

3.18 GPS VERIFY NODE

This node type is almost identical to the GPS node. It differs from a GPS node in the following respects only:

1. In the initial data line, the Type field (4th field) will be 18 rather than 14.
2. In the second line, the Descriptor field (2nd field) will be 18 rather than 14.
3. In the third line, the Descriptor field (2nd field) will be 1018 rather than 1014.
4. In the fourth line, the Descriptor field (2nd field) will be 5018 rather than 5014.
5. In the fifth line, the Descriptor field (2nd field) will be 6018 rather than 6014.
6. In the sixth line, the Descriptor field (2nd field) will be 8018 rather than 6018.

All other data lines will follow the same form as a GPS Node

3.19 ROADWHEEL NODE

A roadwheel node is fully defined by a variable number of data lines. Two data lines are required for the basic information defining the node, additional data lines contain the survey nodes which define the mid-section of the roadwheel path. An example of a roadwheel node is:

```
N1215,1,0,15,196.412,253.6317,0  
N1215,15,7,N1081,2870.91,250,32,0,  
N1215,1015,1,7,N211,N241,N271,N301,N331,N361,N391
```

The first line is the initial data line for the node.
The second line is constructed as follows:
The unique node index (same as for the initial data line),
The descriptor (value of 15 for a roadwheel node),
The count of nodes which make up the roadwheel path,
The index of the final node for the path,
The map path length,
The chain value,
The offset value,
The offset bias.

The additional data lines are constructed as follows:

```
N1215,1015,1,7,N211,N241,N271,N301,N331,N361,N391
```

The unique node index (same as for the initial data line),
The descriptor (value of 1015 for a Roadwheel node list),
The list number (should increment from 1 upwards for each list for any given node),
The count of nodes which make up the roadwheel path,
The path node indices.

3.20 CONSTRUCTION NODE

A construction node is fully defined by two data lines. An example of a bearing and distance node is:

N807,1,0,21,-422.5932,248.1483,0
N807,21,1,102,N511,N541,N481,0,N361,N391,0

The first line is the initial data line for the node. The following line is constructed as follows:

The unique node index (same as for the initial data line),

The descriptor (value of 21 for a construction node),

The 'intindex' value,

The first construction 'senserecord',

The second construction 'senserecord'.

The 'senserecords' vary in length according to their type. They consist of an ID, to identify the type, followed by a number of fields determined by their type

The types of 'senserecord' are:

parallel_construct - ID = 0, + 3 fields

normal_construct - ID = 1, + 4 fields

oblique_construct - ID = 2, + 5 fields

bearing_construct - ID = 3, + 2 fields

circle1_construct - ID = 100, + 2 fields

circle2_construct - ID = 101, + 4 fields

circle3_construct - ID = 102, + 3 fields

point_construct - ID = 1000, + 1 field.

3.21 REFERENCE OBSERVATIONS

Total station and resection nodes can have reference observation information. This information will follow the two data lines necessary to define the total station or resection node.

Typically each reference observation data line will be of the form:

N30491,1010,1,-1,F,F,14.84778,90.54694,33.909,1.5,N23371,0,0,1,0,1,0,0,0,0,F,0,0,1,15,1013,0,6370000,F,0,30,0,0,0,0,1

These reference observation lines are constructed as follows:

- The unique node index (same as for the initial data line of the node),
- The descriptor (value of 1010 for a total station node reference observations, 1011 for resection node reference observations),
- The reference observation number (should increment from 1 upwards for each observation for any given node),
- The measurement type which can have values -1 (None), 0 (Angle), or 1 (Distance),
- The 'Horizontal frozen' flag,
- The 'Level frozen' flag,
- The horizontal angle,
- The vertical angle,
- The distance,
- The staff height,
- The node index for the reference node,
- The left/right offset used when prism could not be put exactly on point,
- The forward/backward offset used when prism could not be put exactly on point,
- The prism offset,
- The secondary scaling factor,
- The standard deviation constant for distance measurement error,
- The standard deviation rate (per m or per ft) for distance measurement error,
- The standard deviation constant for angle measurement functions,
- The standard deviation constant for the instrument centering error,
- The 'Use nachbarhaftstreue Anpassung values' flag,
- The dxnachbar value,
- The dynachbar value,
- The nachbar exponent,
- The temperature (in degrees Celcius),
- The pressure (in Kpa),
- The mean sea level height in metres from sea level,
- The Earth radius in metres,
- The 'Use curvature effect on distance measurement' flag,
- The weather model 0 (No model), 1 (Old model), 2 (New model),
- The level weighting constant,
- Instrument position check status 0 (None), 1 (OK), or 2 (FAIL),
- Instrument collimation error - usually zero,
- Index correction of EDM - usually zero,
- CheckVelOrder [REQUIRES ELABORATION]
- EDM scale - default = 1.

4. VERIFY SECTION

Survey nodes may have references to verify nodes which are used to update or correct their position. All verification references are stored in the VERIFY section, however this data contains indices into the NODES section which must be present for the verification section to be meaningful.

This section starts with the header line:

VERIFY

and ends with the trailer line

ENDVERIFY

All verification data lines adhere to the following format:

Index, Descriptor, NodeIndex, IncludeFlag, VerifyCount, (VerifyData)

Index - This is the unique identifier for the verification line, it starts with 'V' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the verification data which follows, currently this is always '0'

NodeIndex - This is the unique identifier for the node within the NODES section.

IncludeFlag - This indicates whether to include the original node coordinate in determination of the node position.

VerifyCount - The count of the verification nodes for this node.

VerifyData - This consists of one or more entries following this format:

FieldCount - Number of fields for this verification node. This will be 2 if Flagtype field is omitted - Normal type assumed, or 3 if Flagtype field is included for all other types. *Although a Normal verification node can be stored in three field with a '0' flagtype, the default method of storing the Normal type is with two fields only, the NodeIndex and the Include Flag.*

NodeIndex - Index of this node in the NODES section

IncludeFlag - Indicates whether to include this node in the determination of the node position.

For any type verification node other than the 'Normal' type, this flag should always be 'False'.

Flagtype - Type of verification node, this can be:

- 0: Normal
- 1: Type A
- 2: Type B
- 3: Setout
- 129: Type A - Frozen
- 130: Type B - Frozen
- 131: Setout - Frozen

For example, a verification section for two nodes would look like this:

VERIFY

N1609,T,1,2,N1829,T

N1829,F,1,3,N2049,T,0

ENDVERIFY

5. GRAPHICS SECTION

Graphics are the lines, arcs, curves etc. which give meaning to the survey. This does NOT include such things as symbol definitions which are held in a separate section. All survey graphics data is contained in a single section. All graphics data follows a uniform convention.

This section starts with the header line:

GRAPHICS

and ends with the tailer line

ENDGRAPHICS

All graphics information must be defined between this header and tailer.

A complex graphic may require several data lines to define it fully, but most graphics are fully defined with a single data line. Generally graphics data lines do not contain coordinate information, instead they contain one or more node indexes which provide the coordinates. For example a simple line definition would not contain coordinates for the start and end of the line, but would contain references to two nodes which define the positions of the start and end of the line. It is therefore important that node indexes within graphics data lines should exist as node definitions in the NODES section of the file. The NODES section must be present for the GRAPHICS section to be meaningful

5.1 COMMON GRAPHICS FIELDS

ALL graphics data lines start with seven common fields, these are:

Index, Descriptor, Status, Layer, Pattern remaining flag, Pattern remaining percentage, Primary Node Index

Index - This is a unique identifier for the graphic, it starts with 'G' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the graphics data which follows it. The descriptor field can have values between 0 and 65535. Currently these are:

10 = A point

11 = A permanent station

20 = A circle defined by centre point and a point on the circumference

21 = A circle defined by a centre point, and the radius

22 = A circle defined by three points on the circumference

31 = A single bezier curve = 31;

32 = A compound bezier curve

1032 = A compound bezier cross-reference list

40 = An arc defined by a start, mid, and an end point

42 = A tangential arc

60 = A single line

70 = A symbol referenced to a single point

71 = A symbol referenced to two points

80 = Text referenced to a single point

81 = Text referenced to two points

82 = Text referenced to three points

90 = An axially aligned rectangle defined by two points on opposite corners

91 = A rectangle defined by three points

92 = A rectangle defined by two points and a height

Status - This is a number used to indicate the status of the node, this can have a value 0,1,2 or 3 which refers to OLD, NEW, CHANGED or DELETED.

Layer - This is a reference into the LAYERS section identifying the graphic layer

Pattern remaining flag - Use the 'Pattern remaining percentage' value if True.

Pattern remaining percent age - Used when a line style is continued along a path.

Primary Node Index - All graphics objects are attached to at least one node. This field holds the node index for the first (or only) node for any graphics object.

5.2 GRAPHIC POINT

A graphic point is fully defined by a single data line. An example of a graphic point is:

G211,10,0,L7,T,0,N211,P3,1.2

The fields following the common fields are:

The point style index,

The point size.

To specify a point using the default point style, the point style index should be set to P-1, for example:

G211,10,0,L7,T,0,N211,P-1,1.2

A point thus specified will display within Penmap according to the Misc>Graphic Point setting.

5.3 GRAPHIC PERMANENT STATION

A graphic permanent station is fully defined by a single data line. An example of a graphic permanent station is:

G1557,11,0, L0,T,0,N1141, "A permanent station",0.2,0.1,0,2,24

The fields following the common fields are:

The permanent station text,

The text height,

The text width,

The text rotation,

The text x_align value (Windows based) 0 (align left), 2 (align right), or 6 (align centre),

The text y_align value (Windows based) 0 (align top), 8 (align bottom), or 24 (align baseline).

5.4 SINGLE POINT CIRCLE (Centre point and radius)

This type of circle is fully defined by a single data line. An example of a single point circle is:

G619,21,0,L0,T,0,N691,50

The node index in the common fields denotes the circle centre. The field following the common fields is:

The radius of the circle.

5.5 TWO POINT CIRCLE (Centre and circumference points)

This type of circle is fully defined by a single data line. An example of a two point circle is:

G531,20,0,L0,T,0,N541,N571

The node index in the common fields denotes the circle centre. The field following the common fields is:

The node index of the point on the circumference.

5.6 THREE POINT CIRCLE (Three circumference points)

This type of circle is fully defined by a single data line. An example of a three point circle is:

G573,22,0,L0,T,0,N601,N631,N661

The node index in the common fields denotes a point on the circle circumference. The fields following the common fields are:

The node index of the second point on the circumference,

The node index of the third point on the circumference.

5.7 SINGLE BEZIER CURVE

This type of curve is fully defined by a single data line, however this object never exists on its own.

For a single bezier curve to have relevance within Penmap/Map500 it must be referenced by a compound bezier curve (described after this object). An example of a single bezier curve is:

G849,31,0,L0,T,0,N931,N961,-81.6667,-160,0,-46.3713,-184.3279,0

The fields following the common fields are:

The node index for the curve end point,

The first control point easting coordinate,

The first control point northing coordinate,

The first control point level,

The second control point easting coordinate,

The second control point northing coordinate,

The second control point level.

5.8 COMPOUND BEZIER CURVE

This type of curve is fully defined by two or more data lines. Compound bezier curves consist of references to single bezier curves. An example of a compound bezier curve is:

G803,32,0,L0,T,0,N931,2
G803,1032,1,2,G849,G939

The field following the common fields is:

The number of single bezier curves which make up this curve

Additional data lines contain the single bezier curve references and are constructed as follows:

The unique graphic index (same as for the first data line of the curve),

The data descriptor, always 1032 indicating a bezier curve cross-reference list,

The list number (should increment from 1 upwards for each list for any given node),

The single curve count,

A list of 'count' single bezier curve graphic indices.

(The above example has a single curve count of 2 and therefore contains 2 single curve indices)

5.9 GRAPHIC ARC

An arc is fully defined by a single data line. An example of a graphic arc is:

G427,40,0,L0,T,0,N421,N451,N481

The fields following the common fields are:

The node index for the arc mid point,

The node index for the arc end point.

5.10 TANGENTIAL ARC

A tangential arc is fully defined by a single data line. An example of a tangential arc is:

G473,42,0,L0,T,0,N481,N511,-0.8682,0.4961

The fields following the common fields are:

The node index for the arc end point, TGraphArcTan.startsin, TGraphArcTan.startcos.

The startsin and startcos values refer to the sine and cosine values for the start angle of the arc. This is referenced to the horizontal axis, left to right, such that:

An arc starting horizontally left to right would be considered to have a starting angle of 0 degrees, and therefore a sine of 0 and a cosine of 1

An arc starting vertically upwards would be considered to have a starting angle of 90 degrees, and therefore a sine of 1 and a cosine of 0

An arc starting horizontally right to left would be considered to have a starting angle of 180 degrees, and therefore a sine of 0 and a cosine of -1

An arc starting vertically downwards would be considered to have a starting angle of 270 degrees, and therefore a sine of -1 and a cosine of 0

5.11 GRAPHIC LINE

A graphic line is fully defined by a single data line. An example of a graphic point is:

G259,60,0,L0,T,0,N241,N271

The field following the common fields is:

The node index for the line end point.

5.12 SINGLE POINT SYMBOL

A single point symbol is fully defined by a single data line. An example of a single point symbol is:

G1440,70,0,L0,T,0,N1111,M1,50.8643,46.6177,0

The fields following the common fields are:

The symbol index (SYMBOL section must exist),

The symbol height,

The symbol width,

The symbol rotation.

5.13 TWO POINT SYMBOL

A two point symbol is fully defined by a single data line. An example of a two point symbol is:

G694,71,0,L3,T,0,N917,N947,0.1815,-0,0,0.9818,5.9048,0,M14, 34.2388,34.2388,-46.25367

The fields following the common fields are:

The second node index,
The first internal anchor x coordinate,
The first internal anchor y coordinate,
The first internal anchor z coordinate,
The second internal anchor x coordinate,
The second internal anchor y coordinate,
The second internal anchor z coordinate,
The symbol index (SYMBOL section must exist),
The symbol height,
The symbol width,
The symbol rotation.

5.14 SINGLE POINT TEXT

Single point text is fully defined by a single data line. The single node reference sets the start point for the text. An example of single point text is:

G1176,80,0,L7,T,0,N1051,"Example text",31.5092,8.3073,0,0,24

The fields following the common fields are:

The text,
The text height,
The text width,
The text rotation,
The text x_align value (Windows based) 0 (align left), 2 (align right), or 6 (align centre),
The text y_align value (Windows based) 0 (align top), 8 (align bottom), or 24 (align baseline).

5.15 TWO POINT TEXT

Two point text is fully defined by a single data line. The two node references define the baseline of the text. An example of two point text is:

G1176,81,0,L7,T,0,N1051,N1081,"Example text",31.5092,8.3073,0,0,24

The fields following the common fields are:

The second node index,
The text,
The text height,
The text width,
The text rotation,
The text x_align value (Windows based) 0 (align left), 2 (align right), or 6 (align centre),
The text y_align value (Windows based) 0 (align top), 8 (align bottom), or 24 (align baseline).

5.16 THREE POINT TEXT

Three point text is fully defined by a single data line. The first two node references define the baseline of the text, the third defines the height. An example of three point text is:

G1176,82,0,L7,T,0,N1051,N1081,N1111,"Example text",31.5092,8.3073,0,0,24

The fields following the common fields are:

The second node index (for text baseline),

The third node index (for text height),

The text,

The text height,

The text width,

The text rotation,

The text x_align value (Windows based) 0 (align left), 2 (align right), or 6 (align centre),

The text y_align value (Windows based) 0 (align top), 8 (align bottom), or 24 (align baseline).

5.17 AXIALLY ALIGNED RECTANGLE

An axially aligned rectangle is fully defined by a single data line. The two node references define opposite corners of the rectangle. An example of an axially aligned rectangle is:

G665,90,0,L9,T,0,N721,N751

The field following the common fields is:

The second node index.

5.18 THREE POINT ALIGNED RECTANGLE

A three point rectangle is fully defined by a single data line. The first two node references define the rectangle 'baseline', the third node determines the rectangle height. An example of a three point rectangle is:

G707,91,0,L9,T,0,N781,N811,N841

The fields following the common fields are:

The second node index (baseline),

The third node index (height).

5.19 TWO POINT RECTANGLE WITH HEIGHT

A two point rectangle is fully defined by a single data line. The two node references define the rectangle 'baseline', the rectangle height is provided directly. An example of a two point rectangle with height is:

G753,92,0,L9,T,0,N871,N901,50

The fields following the common fields are:

The second node index (baseline),

The rectangle height.

6. GISDEFINITIONS

GIS definitions are the database templates which define the form of each record. This section serves to define the databases, but not to hold database records. All GIS definitions follow a uniform convention.

This section starts with the header line:

GISDEFINITIONS

and ends with the tailer line

ENDGISDEFINITIONS

All GIS definition information must be defined between this header and tailer.

A complex definition may require many data lines to define it fully, but all definitions require at least 3 lines to define a basic database. Additional lines will be required to define the fields of each database.

Database definitions consist of:

Initial database line

Field definitions...

Final database line

The **initial database line** defines a database, it consists of seven fields, these are:

Index, Descriptor, Name, Key, Database type, Symbol name, Field count

For example:

X1,0,"Building","BD",2,"BD",9

Index - This is a unique identifier for the GIS database definition, it starts with 'X' followed by a positive number with a value between 0 and 16383.

Descriptor - This is a number which is used to identify the form of the GIS data which follows it. For the initial GIS line this is always '0'

Name - This is the name of the database.

Key - This is the unique identifier (four characters max.) for the database.

Database type - This identifies the type of database. Currently the following values are permitted:

0 = Attribute database

1 = Symbol database

2 = Area database

3 = Non-graphic database

4 = Linear database

Symbol name - This is a [REQUIRES ELABORATION].

Field count - This is the count of the total number of fields for the database. Fields are numbered from zero, so a database with five fields will have fields numbered from 0 to 4.

The **final database data line** ends the database definition, it consists of eight fields, these are:

Index, Descriptor, Frozen flag, Flag field index, Graphics field index, Index field index, Rows, ID Counter

For example:

X2,2,F,0,1,3,2,0

Index - This is the unique identifier for the GIS database definition.

Descriptor - This is a number which is used to identify the form of the GIS data which follows it. For the final GIS line this is always '2'

Frozen flag - This indicates frozen state of database.

Flag field index - The field index for the Flag field for the database.

Graphics field index - The field index for the Graphics field for the database.

Index field index - The field index for the Index field for the database.

Rows - [Requires elaboration]

ID Counter - [Requires elaboration]

6.1 FIELD DEFINITIONS

A GIS database field is usually defined by a single data line, however where a field provides a pre-determined selection of choices in the form of a list, or an option tree, additional lines are required to define the data. The common field data line is as follows:

Index, Descriptor, Field number, Type, Name, Default value, Minimum value, Maximum value, Affiliation, Compulsory flag, Protected flag, Hidden flag, Display flag, Display type, Bar negcol, Bar poscol, Bar cutcol, Bar minval, Bar maxval, Bar cutval

For example:

X0,1,24,7,"Immediate Destruc. Pending","False"," ","",-1,F,F,F,T,0,1,1,1,0,0,0
or
X0,1,0,100,"Flags"," "," ","",-1,F,T,T,F,0,2,3,4,-2,5,3

Index - This is the unique identifier for the GIS database definition.

Descriptor - Always '1' for a field definition.

Field number - Zero based field number.

Type - Current values for the field type are:

0 = Integer (16 bit signed).
1 = Large integer (32 bit signed).
2 = Positive floating point number.
3 = Floating point number.
4 = Character string
5 = Word (16 bit unsigned)
6 = Option tree (Requires additional data lines)
7 = Boolean (True or False)
8 = Date
9 = Time
10 = Sketch
11 = Photo
12 = Photo-sketch
13 = Attach
14 = Data list (Requires additional data lines)
100 = Flags
101 = Graphics
102 = Global index
103 = Local index

Name - The name of field.

Default value - The default field value

Minimum value- Minimum field value

Maximum value - Maximum field value

Affiliation - [Requires elaboration]

Compulsory flag - Field must have an entry

Protected flag - Field is protected

Hidden flag - Field is hidden

Display flag - Field is displayable

DisplayType - Determines display mode 0 (Text), or 1 (Bargraph)

Bar negcol - Bargraph colour for negative values

Bar poscol - - Bargraph colour for positive values

Bar cutcol - - Bargraph colour for cutoff values

Bar minval - - Bargraph minimum display value

Bar maxval - - Bargraph maximum display value

Bar cutval - Bargraph cutoff value

6.2 DATALIST DEFINITION

The 'datalist' type of field requires additional data lines to define the predetermined selection list for the field. There is an initial data list line followed by a variable number of additional lines. A simple example:

```
X10,3,6,7  
X10,1003,6,0,3,"Banana","Apple","Mushroom","Cabbage"  
X10,1003,6,4,6,"Greengage","Pumpkin","Peach"
```

The initial datalist line consist of a fixed number of fields as follows:

Index, Descriptor, Field number, Listcount

Index - This is the unique identifier for the GIS database definition.

Descriptor - Always '3' for an initial datalist line.

Field number - Zero based field number.

ListCount - Count of data items for field

The second and subsequent lines are constructed as follows:

Index, Descriptor, Field number, First, Last, Data....

Index - This is the unique identifier for the GIS database definition.

Descriptor - Always '1003' for a data line.

Field number - Zero based field number.

First - Zero based number of first data item in this line

Last - Zero based number of last data item in this line

Data - Strings representing the predetermined field entries

6.3 OPTION TREE DEFINITION

The 'option tree' type of field requires additional data lines to define the predetermined selection tree for the field. There is an initial data list line followed by a variable number of additional lines. A simple example:

```
X9,4,6,79,14
X9,1004,6,"Cobalt",3,">1>1>1"
X9,1004,6,"Iron",3,">1>1>2"
X9,1004,6,"Magnetic",2,">1>1"
X9,1004,6,"Aluminium",3,">1>2>1"
X9,1004,6,"Tin",3,">1>2>2"
X9,1004,6,"Copper",3,">1>2>3"
X9,1004,6,"Non-magnetic",2,">1>2"
X9,1004,6,"Metal",1,">1"
X9,1004,6,"Polystyrene",2,">2>1"
X9,1004,6,"Polyvinyl",2,">2>2"
X9,1004,6,"Polyester",2,">2>3"
X9,1004,6,"Polycarbonate",2,">2>4"
X9,1004,6,"Plastic",1,">2"
X9,1004,6,"",0,""
```

The initial option tree line consist of a fixed number of fields as follows:

Index, Descriptor, Field number, MaxChars, Listcount

Index - This is the unique identifier for the GIS database definition.

Descriptor - Always '4' for an initial option tree line.

Field number - Zero based field number.

MaxChars - Maximum field size

ListCount - Count of data items (equal to number of additional lines) for field

The second and subsequent lines are constructed as follows, one option per data line:

Index, Descriptor, Field number, Data, Level, Tree position

Index - This is the unique identifier for the GIS database definition.

Descriptor - Always '1004' for an option definition line.

Field number - Zero based field number.

Data - Option data, as a string

Level - The distance from the root.

Tree position - Describes the option position on the option tree e.g. ">2>3" for third option on second branch..

7. GISRECORD SECTION

GIS records are the database records contained in a survey. All GIS record data is contained in a single section. All GIS record data follows a uniform convention. Versions of the Universal format up to and including version 0.5 have a problem with GIS records under certain circumstances. If the GIS record is an Area or Linear database with a large number of graphic elements then a problem occurs on export causing an unimportable file to be created. This problem is dealt with from version 0.6 onwards by the use of the alternative GIS record. The original style of GIS record remains valid for GIS records with a small number of graphic elements.

This section starts with the header line:

GISRECORDS

and ends with the tailer line

ENDGISRECORDS

All GIS record information must be defined between this header and tailer.

All GIS records require at least two lines to define them.

Original style GIS records have a common first line which follows this format:

Index, Descriptor, Database index, Record Status, Field Count, Graphics Count, Graphics Indices....

For example:

D0,0,X5,1,8,4,G1131,G1331,G1531,G1731

Index - This is a unique identifier for the GIS record definition, it starts with 'D' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the GIS record which follows it. For the first GIS record line this is always '0'

Database index - The index into the GISDEFINITIONS section for details on this database.

Record Status - The current status of the record, this can be:

OLD	0
NEW	1
CHANGED	2
DELETED	4

Field Count - The total number of record fields for this record.

Graphics Count - The total number of graphic indices for this record.

Graphics Indices - Comma separated graphics cross references.

Alternative style GIS records have a common first line which follows this format:

Index, Descriptor, Database index, Record Status, Field Count, Graphics Count

For example:

D0,1,X5,1,8,4

Index - This is a unique identifier for the GIS record definition, it starts with 'D' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the GIS record which follows it. For the first GIS record line this is always '1'

Database index - The index into the GISDEFINITIONS section for details on this database.

Record Status - The current status of the record, this can be:

OLD	0
NEW	1
CHANGED	2
DELETED	4

Field Count - The total number of record fields for this record.

Graphics Count - The total number of graphic indices for this record.

The alternative GIS first line is followed by one or more lines used to define the graphics indices for the record.

For example:

D0,1001,0,3,G1131,G1331,G1531,G1731

Index - This is a unique identifier for the GIS record definition, it starts with 'D' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the line as graphics indices line, it is always 1001

First - The zero based number of the first graphic index in this line.

Last - The zero based number of the last graphic index in this line.

Graphics Indices - Comma separated graphics cross references.

Subsequent data lines for both the original and alternative (for the same example record) follow this format:

Index, Descriptor, First, Last, Data...

For example (providing field data for the preceding example):
D0,1000,0,3,"1","[-157.271 , 192.394]","Global0000005","BH0002"
D0,1000,4,7,"17 / Nov / 1998","18:21:27","Bill","Monica"

Index - This is a unique identifier for the GIS record definition.

Descriptor - This is always '1000' for additional GIS record lines.

First - The zero based number of the first record field in this line.

Last - The zero based number of the last record field in this line.

Data - Comma separated field data strings.

Datalists (Introduced with version 0.8, datalists can be imported and exported.)

```
D0,0,X8,1,8,3,G211,G253,G295
D0,1000,0,2,"1","[44.270 , 154.160]","Global0000001"
D0,1000,3,5,"fred0001","30 / Aug / 2001","19:50:35"
D0,1000,6,7,"",""
D0,1002,2,6,7
D0,1003,6,0,3,4,0,4,"0.000000","0.000000","1.000000","0.000000","1.000000"
D0,1003,6,1,3,4,0,4,"2.000000","19.000000","15.000000","",""
D0,1003,6,2,3,4,0,4,"66.000000","77.000000","88.000000","99.000000","12.000000"
D0,1003,6,3,3,4,0,4,"11.000000","16.000000","32.000000","48.000000","64.000000"
D0,1003,7,0,1,2,0,2,"13.000000","43.000000","45.000000"
D0,1003,7,1,1,2,0,2,"6.000000","7.000000","8.000000"
```

This example shows a record containing two datalists. The datalist data has an initial line which follows the main GIS record and is constructed as follows:

Index, Descriptor, Field count, Field number, Field number.....

For example (providing datalist field data for the preceding example):

```
D0,1002,2,6,7
```

Index - This is a unique identifier for the GIS record definition.

Descriptor - This is always '1002' for first datalist line.

Field count - The count of the number of fields which contain a datalist.

Field number - The zero based field identifier of a datalist field. There should be an identifier for every field of datalist type. In the above example there are two datalists in the record, fields 6 and 7.

The datalist initial line is followed by a number of lines, one for each row of a datalist record. The format of these lines is as follows:

Index, Descriptor, Field number, Row number, Maximum row number, Maximum column number, First column number, Last column number, Data....

For example (providing datalist field data for the preceding example):

```
D0,1003,6,0,3,4,0,4,"0.000000","0.000000","1.000000","0.000000","1.000000"
```

Index - This is a unique identifier for the GIS record definition.

Descriptor - This is always '1003' for each datalist row-data line.

Field number - The zero based field identifier of the datalist field.

Row number - The zero based row identifier of this datalist row.

Maximum row number - The maximum row number. This value will be the same for each row dataline for a given field.

Maximum column number - The maximum column number for this row. This value can vary for each row dataline for a given field.

First column number - The zero based column identifier of a datalist field, for the first item of data on this line.

Last column number - The zero based column identifier of a datalist field, for the last item of data on this line.

Data - The data for the line, numeric data stored as strings (i.e. in quotes).

8. LINSTYLE SECTION

Linestyles are applied to graphics according to the layer they occupy. Linestyles are defined in a basic fashion, their use can be modified by fields in individual layers. The LINSTYLE section starts with the header line:

LINESTYLES

and ends with the tailer line

ENDLINESTYLES

All linestyle information must be defined between this header and tailer.

***IMPORTANT:** Penmap defines a default line style of a solid line. This must ALWAYS be included in the LINSTYLE section using a single line EXACTLY as follows:*

S0,0,"Solid",0

A Linestyle is generally defined completely using two data lines, although complex line patterns may increase this.

The initial linestyle data line is as follows:

Index, Descriptor, Name, Vector count

For example (continuing the above example):

S9,0,"Dash-box",7

Index - This is a unique identifier for the linestyle definition, it starts with 'S' followed by a positive number with a value between 0 and 16383.

Descriptor - This is a number which is used to identify the form of the linestyle information which follows it. For the initial linestyle definition line this is always '0'.

Name - The name of the linestyle.

Vector count - The number of vectors (Moves, Draws etc.) which make up the line.

Subsequent data lines for the same linestyle definition the line path and follow this format:

Index, Descriptor, First, Last, Vector data...

For example (continuing the above example):

S9,1000,0,4,1,0,0,2,40,0,1,60,-10,2,80,-10,2,80,10

S9,1000,5,6,2,60,10,2,60,-10

Index - This is the unique identifier for the linestyle definition.

Descriptor - This is always '1000' for linestyle vectors.

First - The zero based number for the first vector in this data line.

Last - The zero based number for the last vector in this data line.

Vector data - The vectors, stored in the following form:

1,x,y - for a move to x,y.

2,x,y - for a line to x,y.

3,x,y - for a filled rectangle from the current position to x,y.

9. LAYER SECTION

This section starts with the header line:

LAYERS

and ends with the tailer line

ENDLAYERS

All layer information must be defined between this header and tailer.

A layer definition is fully defined by two data lines as follows:

For Universal Format versions up to and including 1.0:

```
L1,0,"Trees",F,"FFFFFFFF",S0,1,100,0,$0,$0,T,0,F,F
```

```
L1,1000,0,1,1,F,0,24,F
```

For Universal Format versions 1.1 and later:

```
L1,0,"New",F,"FFFFFFFF",S0,1,100,0,$FF00,$AEFFAE,T,0,F,F,$FF,$FF0000,$8080FF,$FFAAAA
```

```
L1,1000,0,1,1,F,0,24,F
```

The first data line is as follows:

For Universal Format versions up to and including 1.0:

Index, Descriptor, Name, Frozen flag, Contour state, Linestyle, length, width, thickness, Line RGB, Text RGB, Bestfit flag, Text weight, Italic flag, Underline flag

For Universal Format versions 1.1 and later:

Index, Descriptor, Name, Frozen flag, Contour state, Linestyle, length, width, thickness, New Line RGB, New Text RGB, Bestfit flag, Text weight, Italic flag, Underline flag, Old Line RGB, Changed Line RGB, Old Text RGB, Changed Text RGB

For example:

```
L1,0,"New",F,"FFFFFFFF",S0,1,100,0,$FF00,$AEFFAE,T,0,F,F,$FF,$FF0000,$8080FF,$FFAAAA
```

Index - This is a unique identifier for the layer definition, it starts with 'L' followed by a positive number with a value between 0 and 16383.

Descriptor - This is a number which is used to identify the form of the layer information, this is always '0'.

Name - The name of the layer.

Frozen flag - Indicates if layer is frozen

Contour state - A packed boolean array in quotes. This represents; first five booleans are 'show surfaces' 1 to 5, last five booleans are 'show breaklines' for surfaces 1 to 5.

Linestyle - The linestyle index into the LINSTYLES section.

Length - The length property to be applied to the selected linestyle.

Width - The width property to be applied to the selected linestyle.

Thickness - The thickness property to be applied to the selected linestyle.

Line RGB - The windows RGB value for the line colour as a hexadecimal. (New Line RGB for 1.1 onwards)

Text RGB - The windows RGB value for the text colour as a hexadecimal. (New Text RGB for 1.1 onwards)

Bestfit flag - Use 'bestfit' algorithm when rendering text.

Text. weight - Windows based 'text weighting' value.

Italic flag - Render text as italics where available.

Underline flag - Render text with underlining.

Old Line RGB - The windows RGB value for the line colour as a hexadecimal. (1.1 onwards)

Changed Line RGB - The windows RGB value for the line colour as a hexadecimal. (1.1 onwards)

Old Text RGB - The windows RGB value for the text colour as a hexadecimal. (1.1 onwards)

Changed Text RGB - The windows RGB value for the text colour as a hexadecimal. (1.1 onwards)

The second data line adds details for the layer text as follows:

Index, Descriptor, Name, Frozen flag, Contour state, Linestyle, length, width, thickness, Line RGB, Text RGB, Bestfit flag, Text weight, Italic flag, Underline flag

For example:

L1,1000,0,1,1,F,0,24,F

Index - This is a unique identifier for the layer definition, it starts with 'L' followed by a positive number with a value between 0 and 16383.

Descriptor - This is a number which is used to identify the data line as layer text information, this is always '1000'.

Data version – Currently '0' (included for possible future changes).

Text height – Text height in survey units.

Text width – Text width in survey units.

Text size flag – Use height/width (above) parameters when rendering text, rather than the global text height/width.

XAlign - Windows based text X alignment value.

YAlign - Windows based text Y alignment value.

Text alignment flag – Use alignment (above) parameters when rendering text, rather than the global text alignment parameters.

10. POINTSTYLE SECTION

Point styles are provided for different representations of the graphics 'Point'.

The POINTSTYLE section starts with the header line:

POINTSTYLES

and ends with the tailer line

ENDPOINTSTYLES

All pointstyle information must be defined between this header and tailer.

***IMPORTANT:** Penmap defines a default point style of a single point. This must ALWAYS be included in the POINTSTYLE section using a single line EXACTLY as follows:*

P0,0,"Dot",0

A Pointstyle is generally defined completely using two data lines, although complex point styles may increase this.

The initial pointstyle data line is as follows:

Index, Descriptor, Name, Vector count

For example (continuing the above example):

P3,0,"Double Circle",4

Index - This is a unique identifier for the pointstyle definition, it starts with 'P' followed by a positive number with a value between 0 and 16383.

Descriptor - This is a number which is used to identify the form of the pointstyle information which follows it. For the initial pointstyle definition line this is always '0'.

Name - The name of the pointstyle.

Vector count - The number of vectors (Moves, Draws etc.) which make up the point definition.

Subsequent data lines for the same pointstyle definition the point path and follow this format:

Index, Descriptor, First, Last, Vector data...

For example (continuing the above example):

P3,1000,0,3,1,0,0,4,80,1,0,0,4,100

Index - This is the unique identifier for the pointstyle definition.

Descriptor - This is always '1000' for pointstyle vectors.

First - The zero based number for the first vector in this data line.

Last - The zero based number for the last vector in this data line.

Vector data - The vectors, stored in the following form:

- 1,x,y** - for a move to x,y.
- 2,x,y** - for a line to x,y.
- 3,x,y** - for a filled rectangle to x,y.
- 4,r** - for a hollow circle of radius r.
- 5,r** - for a filled circle of radius r.
- 10,x,y** - for a solid triangle (normal).
- 11,x,y** - for a solid triangle (rotated).
- 12,x,y** - for a solid diamond.
- 13,r** - for a solid 4 point star.
- 14,r** - for a solid 5 point star.
- 15,r** - for a solid 6 point star.
- 16,x,y** - for a solid cross (upright).
- 17,x,y** - for a solid cross (diagonal).

11. DIMENSION SECTION

The DIMENSION section starts with the header line:

DIMENSIONS
and ends with the tailer line
ENDDIMENSIONS

All dimension information must be defined between this header and tailer.

Currently there are three types of dimension defined, these are Linear, Angle, and Bearing dimensions.

If the Dimension Section is included in a file, then the 'Dimension Attributes' line in the file header should also be present.

11.1 LINEAR DIMENSION

Index, Descriptor, Value, SD1, SD2, First node, Last node

For example:

K1,1, 36.8663,0.76,0.4,N421,N451

Index - This is a unique identifier for the dimension, it starts with 'K' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the dimension information which follows it. For Linear Dimensions this is always '1'.

Value - The value of the dimension.

SD1 - Standard deviation of distance in metres

SD2 - Standard deviation of distance in metres per metre

First node - The index of the first node.

Last node - The index of the last node.

11.2 ANGLE DIMENSION

Index, Descriptor, Value, SD, First node, Second node, Third Node, Last node

For example:

K5,2,223.4038,0.25,N421,N451,N451,N481

Index - This is a unique identifier for the dimension, it starts with 'K' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the dimension information which follows it. For Angle Dimensions this is always '2'.

Value - The value of the angle dimension.

SD - Standard deviation of angle in current units of angle.

First node - The index of the first node (First vector).

Second node - The index of the second node (First vector).

Third node - The index of the third node (Second vector).

Last node - The index of the last node (Second vector).

11.3 BEARING DIMENSION

Index, Descriptor, Value, SD, First node, Last node

For example:

K6,3,235.32526,N271,N241

Index - This is a unique identifier for the dimension, it starts with 'K' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the dimension information which follows it. For Bearing Dimensions this is always '3'.

Value - The value of the bearing dimension.

SD - Standard deviation of bearing angle in current units of angle.

First node - The index of the first node.

Last node - The index of the last node.

11.4 CHAIN DIMENSION

Index, Descriptor, Value, SD1, SD2, First node, Last node

For example:

K1,5, 36.8663,0.76,0.4,N421,N451

Index - This is a unique identifier for the dimension, it starts with 'K' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the dimension information which follows it. For Chain Dimensions this is always '5'.

Value - The value of the dimension.

SD1 - Standard deviation of distance in metres

SD2 - Standard deviation of distance in metres per metre

First node - The index of the first node.

Last node - The index of the last node.

11.5 OFFSET DIMENSION

Index, Descriptor, Value, SD1, SD2, First node, Last node

For example:

K1,6, 36.8663,0.76,0.4,N421,N451,N481

Index - This is a unique identifier for the dimension, it starts with 'K' followed by a positive number with a value between 0 and 2147483647.

Descriptor - This is a number which is used to identify the form of the dimension information which follows it. For an Offset Dimensions this is always '6'.

Value - The value of the dimension.

SD1 - Standard deviation of distance in metres

SD2 - Standard deviation of distance in metres per metre

Ref. Start node - The index of the chain start node.

Ref. End node - The index of the chain end node.

Offset node - The index of the offset node.

12. SYMBOLS SECTION

The SYMBOLS section starts with the header line:

```
SYMBOLS  
and ends with the tailer line  
ENDSYMBOLS
```

All symbol information must be defined between this header and tailer.

A Symbol is defined using several data lines, although all symbols share a common initial line.

12.1 INITIAL DATA LINE

The initial symbol data line is as follows: [REQUIRES ELABORATION - SOME DISCUSSION]

Index, Descriptor, Name, BaseX, BaseY, GPS Signature, GPS AntX, GPS AntY, GPS climit, GPS minspeed, GPS RotX, GPS RotY, GPS has1, GPS has2, GPS hasr, GPS lealedge, GPS hasc, GPS hass, GIS def, Element count

For example:

M0,0,"SYMBOL90",0,0,21845,0.0021,0.2946,10,1.5,0,0,T,F,T,0,T,T,X-1,2

Index - This is a unique identifier for the symbol definition, it starts with 'M' followed by a positive number with a value between 0 and 16383.

Descriptor - This is a number which is used to identify the form of the symbol information which follows it. For the initial symbol definition line this is always '0'.

Name - The name of the symbol.

BaseX - The base X value.

BaseY - The base Y value.

GPS signature - The GPS signature

GPS AntX - antenna location x coordinate/second insertion point x coordinate.

GPS AntY - antenna location y coordinate/second insertion point y coordinate.

GPS climit - the GPS course change angular speed limit.

GPS minspeed- the GPS minimum forward speed.

GPS RotX- symbol rotation centre x coordinate.

GPS RotY- symbol rotation centre y coordinate.

GPS has1 - antenna location is valid flag or has second insertion point.

GPS has2 - unused in current version.

GPS hasr- symbol rotation centre is valid flag.

GPS lealedge - symbol leading edge 0 (N), 1 (E), 2 (S), or 3 (W).

GPS hasc- the GPS course change angular speed limit is valid flag.

GPS hass- the GPS minimum forward speed is valid flag.

GIS def - related GIS definition index

Element count - Count of additional lines to define symbol.

12.2 ELEMENT DATA LINES

12.2.1 SYMBOL POINT ELEMENT

The symbol point element is constructed as follows:

Index, Descriptor, Element number, X coord, Y coord, Z coord

for example:

M39,1,0,-47.9854,74.167,0

M39,1,1,-35.551,79.8517,0

Index - This is the unique identifier for the symbol definition.

Descriptor - For point elements this is always '1'.

Element number - The number of this element, zero based.

X coord, Y coord, Z coord - The coordinate of the point.

12.2.2 SYMBOL CIRCLE ELEMENT

The symbol circle element is constructed as follows:

Index, Descriptor, Element number, X coord, Y coord, Z coord, Radius

for example:

M43,2,0,-230.4001,283.4751,0,36.0965

Index - This is the unique identifier for the symbol definition.

Descriptor - For circle elements this is always '2'.

Element number - The number of this element, zero based.

X coord, Y coord, Z coord - The circle centre coordinate.

Radius - The circle radius.

12.2.3 SYMBOL ARC ELEMENT

The symbol arc element is constructed as follows:

Index, Descriptor, Element number, X coord, Y coord, Z coord, Degenerate, Radius, StartAngle, EndAngle

for example:

M42,3,0,-320.5095,317.1759,0,F,11.0555,72.06296,201.35667

Index - This is the unique identifier for the symbol definition.

Descriptor - For arc elements this is always '3'.

Element number - The number of this element, zero based.

X coord, Y coord, Z coord - The arc start coordinate.

Degenerate - The degenerate flag.

Radius - The arc radius.

StartAngle - The arc start angle.

EndAngle - The arc end angle.

12.2.4 SYMBOL LINE ELEMENT

The symbol line element is constructed as follows:

Index, Descriptor, Element number, X coord, Y coord, Z coord, EndX, EndY, EndZ

for example:

M39,4,0,-47.9854,74.167,0,-40.55,74.167,0

Index - This is the unique identifier for the symbol definition.

Descriptor - For line elements this is always '4'.

Element number - The number of this element, zero based.

X coord, Y coord, Z coord - The line start coordinate.

EndX, EndY, EndZ - The line end coordinate.

12.2.5 SYMBOL POLYLINE ELEMENT

The symbol polyline element is constructed as follows:

Index, Descriptor, Element number, X coord, Y coord, Z coord, Vector count

for example:

M5,5,0,12.3548,35.1488,0,6

Index - This is the unique identifier for the symbol definition.

Descriptor - For polylines this is always '5'.

Element number - The number of this element, zero based.

X coord, Y coord, Z coord - The polyline start coordinate.

Vector count - The count of line elements, which are stored in special data lines following this line as follows:

Index, Descriptor, Element number, First vector number, Last vector number, Vector coordinates...

for example (continuing the above example):

M5,1005,0,0,1,57.0642,2.958,0,12.3548,35.1488,0
M5,1005,0,2,3,10.5664,87.0117,0,53.4874,120.9908,0
M5,1005,0,4,5,112.5038,56.6093,0,91.0433,42.3023,0

Index - This is the unique identifier for the symbol definition.

Descriptor - For polyline vectors this is always '1005'.

Element number - The number of the parent element, zero based.

First vector number - The first vector in this data line, zero based.

Last vector number - The last vector in this data line, zero based.

Vector coordinates - X,Y,Z coordinates on polyline.

12.2.6 SYMBOL TEXT ELEMENT

The symbol text element is constructed as follows:

Index, Descriptor, Element number, X coord, Y coord, Z coord, Text, Height, Width, Rotation, XAlign, YAlign

for example:

M44,8,0,-418.4657,249.453,0,"Hi there",13.5209,5.8205,-347.68502,0,24

Index - This is the unique identifier for the symbol definition.

Descriptor - For text elements this is always '8'.

Element number - The number of this element, zero based.

X coord, Y coord, Z coord - The text position coordinate.

Text - The text string.

Height - The height of the text.

Width - The width of the text.

Rotation - The angle of the text.

XAlign - Windows based text X alignment value.

YAlign - Windows based text Y alignment value.

12.2.7 SYMBOL COMPOUND BEZIER ELEMENT

The symbol compound bezier element is constructed as follows:

Index, Descriptor, Element number, X coord, Y coord, Z coord, Curve count

for example:

M45,10,7,-380.4023,195.3629,0,2

Index - This is the unique identifier for the symbol definition.

Descriptor - For compound bezier elements this is always '10'.

Element number - The number of this element, zero based.

X coord, Y coord, Z coord - The compound bezier start coordinate.

Curve count - The count of bezier elements, which are stored in special data lines following this line as follows:

Index, Descriptor, Element number, First curve number, Last curve number, Curve coordinates...

for example (continuing the above example):

M45,1010,7,0,1,-380.4023,195.3629,0,-370.8864,195.5298,0,-360.1755,190.3295,0,-351.8547,195.8637,0,-351.8547,195.8637,0,-343.8816,201.1666,0,-344.5091,212.5582,0,-340.8363,220.9054,0

Index - This is the unique identifier for the symbol definition.

Descriptor - For bezier sub-elements vectors this is always '1010'.

Element number - The number of the parent element, zero based.

First vector number - The first vector in this data line, zero based.

Last vector number - The last vector in this data line, zero based.

Curve coordinates - Start, Control1, Control2, and End X,Y,Z coordinates for each bezier.

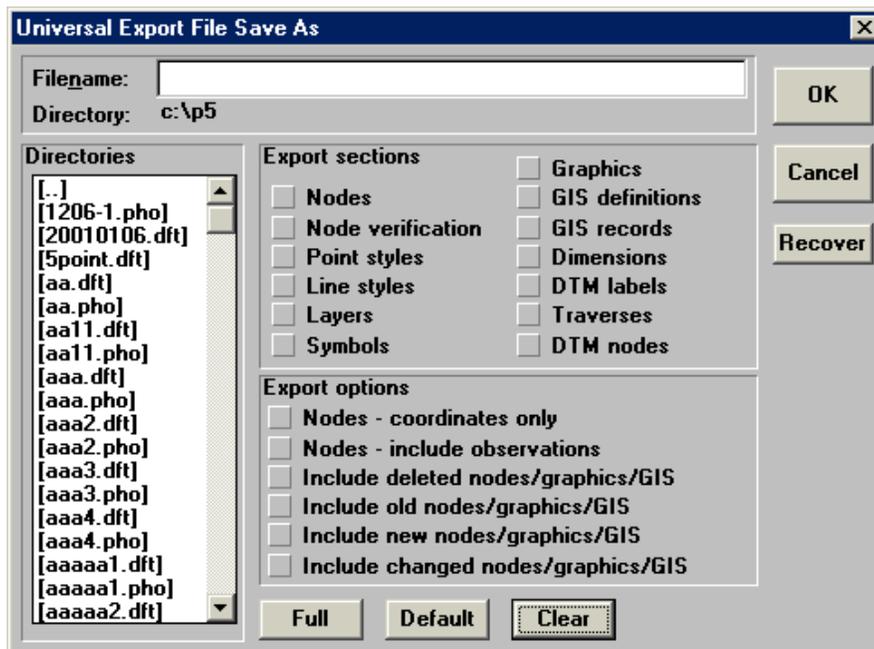
13. APPENDIX 1 - SAVING FROM PENMAP/MAP500

To create an export file, from within Penmap/Map500 use File>Export>Universal. The following dialog will appear.

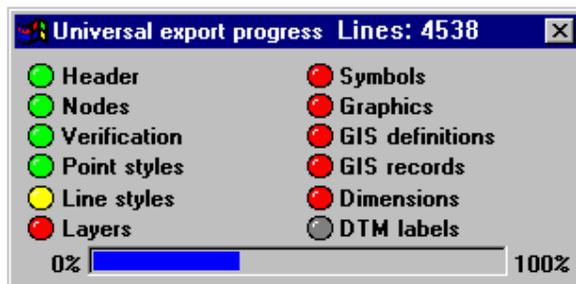
Export file names must be 8 or less characters long. It is not necessary to add the '.unv' file extension in this dialog.

Please note that if GIS records are selected for export, then the Graphics and GIS definitions sections will also be exported (as indicated by the checkboxes becoming both checked and disabled)..

Similarly if node observation export is selected as an option, then the Nodes section will also be exported.



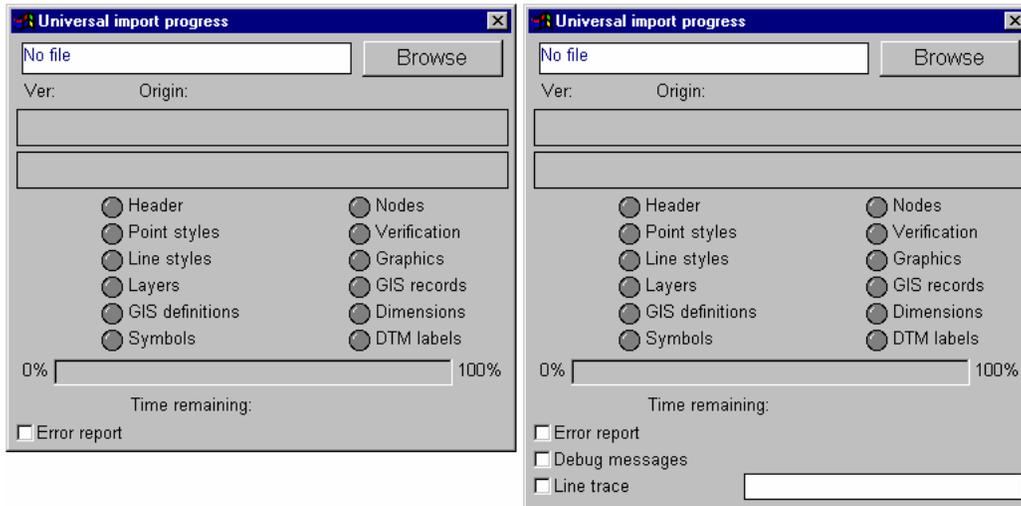
During export the following dialog will be displayed.
 A grey light indicates that the section is not to be exported.
 A red light indicates that the section is awaiting export.
 A yellow light indicates that the section is in the process of being exported.
 A green light indicates that the section has been exported.



14. APPENDIX 2 - LOADING INTO PENMAP/MAP500

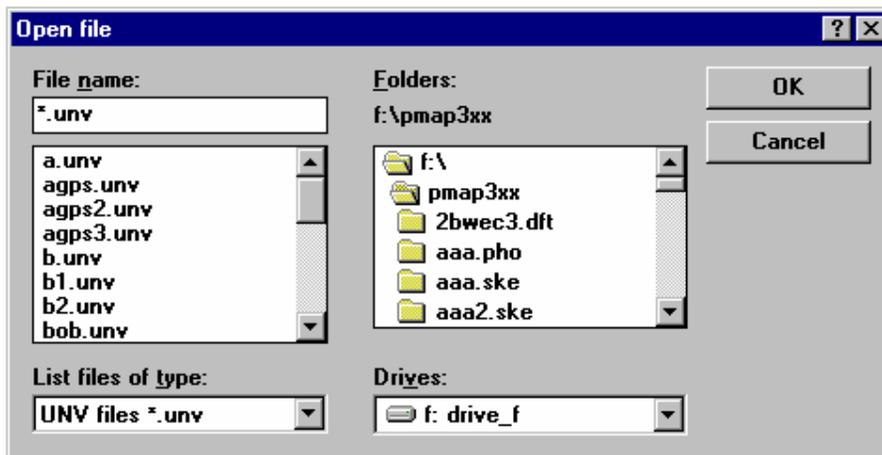
To import a Universal Format file, from within Penmap/Map500 use File>Import>Universal. The following dialog will appear.

Note: *The dialog on the right can be obtained by dragging the bottom edge of the dialog downwards. The 'Debug messages' and 'Line trace' checkboxes are intended for debugging use.*

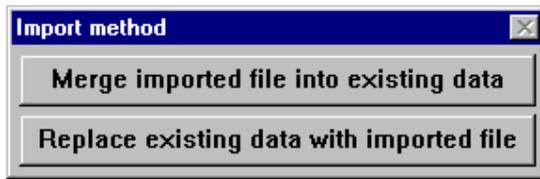


Select checkbox options first.

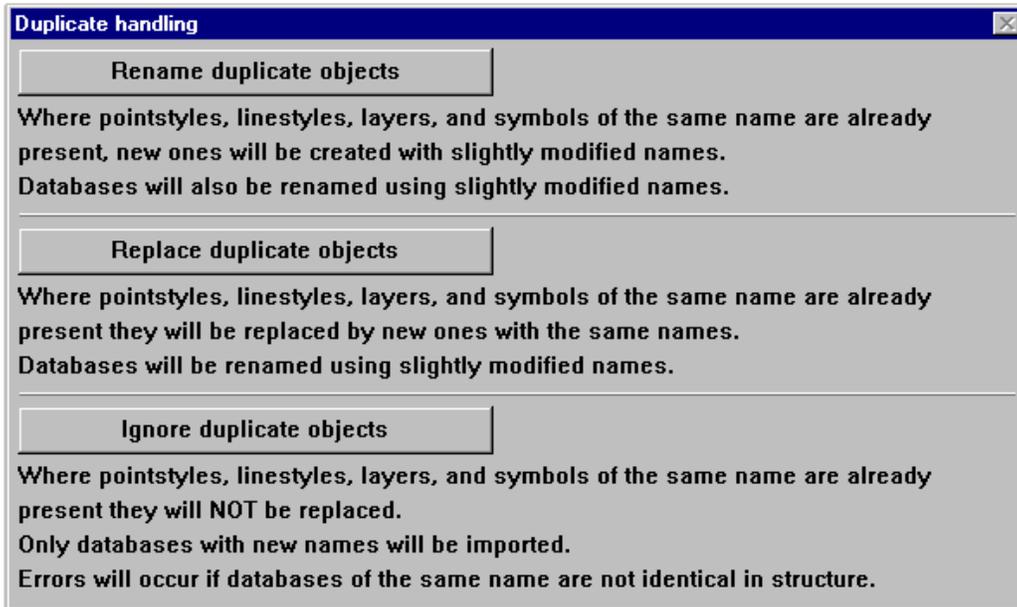
When ready hit the 'Browse' button to select a file for import, the following dialog will appear.



Once you have selected the required file for import you will be given a choice of Merge or Replace for the incoming file.

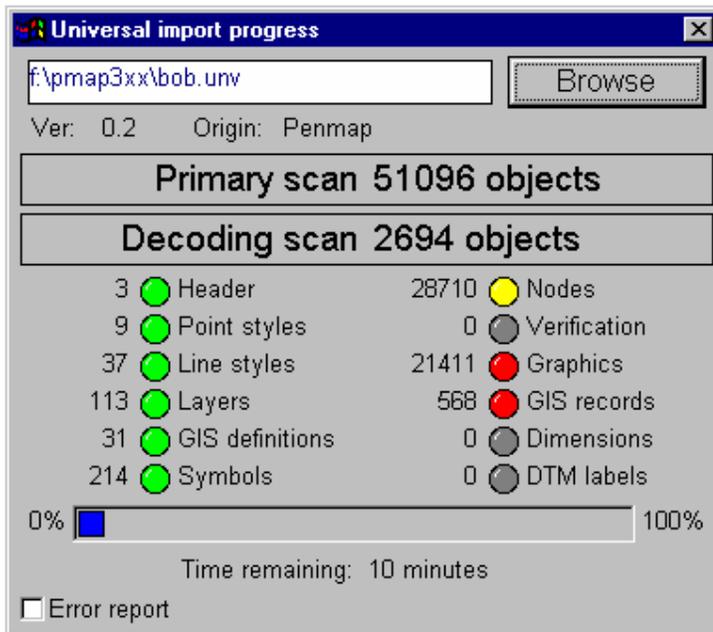


If you choose to Merge the incoming file with your existing data/setup then you will be presented with the following dialog.



Once this selection has been made the import process will start.

The Import dialog displays the progress of the import operation.



The import consists of two passes of the incoming file. The first pass enumerates and indexes all twelve import sections. The second scan decodes the file. In the dialog above there is a number and an LED indicator next to each import section. The number indicates the number of entries for that section, the LED indicates the status:

- GREY - Section absent or empty
- RED - Section awaiting decoding
- YELLOW - Section being decoded
- GREEN - Section has been decoded

15. APPENDIX 3 - IMPORT POSSIBILITIES

MERGING SURVEYS

Penmap/Map500 has suffered from a limitation in the ability to merge surveys from several surveys working on the same project. Using the Universal Format such merging is now easy provided certain precautions are taken:

1. Each surveyor should be using an identical Penmap/Map500 setup, with the same preloaded files, and therefore the same GIS definitions, Layernames, Symbols etc.
2. When merging the surveys, the first file should use 'Replace existing data with imported file' (see Appendix 2).
3. All other files should use 'Merge imported file into existing data', with the 'Ignore duplicate object' mode of 'Duplicate handling' selected.

ADDING LAYERS, SYMBOLS ETC.

Universal format files need not contain all twelve sections. It is quite easy therefore to add new layers, symbols etc. at any time by importing a Universal format file containing just the necessary sections.

PENMAP RELATED FUNCTIONS

Penmap has two Macro codes which are for specialised use with a file called 'Default.unv'.

6040 - This code will import a file called 'default.unv' from the Penmap program directory, in replace mode.

6041 - This code will import a file called 'default.unv' from the Penmap program directory, in merge mode.

These codes can be used in sequence macros (ref. Customiser), for example, to do various things like preload layers, or preload a survey.

They can also be used in the initial sequence (ref. Customiser), to preload 'Default.unv'. This is a useful way to share a setup (i.e. layers, symbols, GIS definitions, linestyles, pointstyles etc.) with other users without copying numerous files.

16. APPENDIX 4 – VERSION DIFFERENCES

This section only started with version 0.5 so information before that time is unavailable.

0.5 – Start level for this record of changes.

0.6 – Added alternative storage for GIS record. (see **Section 7. GISRECORD SECTION**)

This was necessary because the original form of the record encountered problems when GIS area or linear databases with more than a couple of hundred graphics elements were stored.

Storing such objects could overflow the 2000 character limit on any single data line in the universal format.

The solution to this problem was to provide a new alternative record for storing the information held in the first line of the original record. In the alternative form of the record, the graphics indices which were previously stored at the end of the initial line, are now stored on one or more subsequent lines immediately following the initial line.

In practice Penmap only uses this alternative form where GIS records have more than 150 graphic elements, the original form still being used for records with less graphic elements. This maintains the highest level of backwards compatibility for exported files.

This change was implemented in Patch2 for Penmap 3.70a, Map500 2.0a, the patch being issued to resolve this problem.

0.8 – Added covariance data to GPS node for supporting GPS devices, added datalist export to GIS records

Added H3 line in header for extra parameters. For Penmap3 ver 4, and MAP500 2.1 only the imagenamebase parameter is supported

0.9 – Added text size/alignment data line to Layer definition

Added further point style vector options to the pointstyle definitions.

Added fifth GPS node definition line

1.0 – Chain and Offset Dimension objects added to fit in with the Chain and Offset enhancements in Penmap 5.0 and Map500 2.42. Header DIMENSION parameters line has two alternative forms, old (descriptor H2) and new (descriptor H4) to cover altered/additional dimension parameters. All version 1.0 files should contain the H4 line in place of the H2 line.

1.1 – Layer definition expanded. From Penmap 5.1 / MAP500 2.7 it is possible to have 6 colours for each layer. Old/new/changed graphics colours, plus old/new/changed text colours. The layer definition has been expanded to include these extra colours. The GPS node definition has expanded to include a new optional line carrying PDOP, 2D RMS, and elevation mask information.