

Enigma Airports file format

This document describes the Enigma Airports file format.
The Airports data file contains detail information on airports and airfields.

This file format has been created to support the Enigma series of EFIS instruments from MGL Avionics.

MGL Avionics grants any interested party the right to use this Airports data format whether this is in support of a MGL Avionics product or not.

Any party making use of this document and described file format will do so at their own discretion, responsibility and risk.

MGL Avionics places this document and the data format it describes in the public domain in order to foster the use of a common, compact airports data file format that is particularly optimized for use with resource limited systems and/or systems that require fast access to individual items in the airports files in order to increase overall systems performance.

In order to allow recognition of the format any party adopting this format agrees to refer to this format as the "Enigma Airports data format". All data using this format should be considered to be in the public domain and the implementer will make reasonable effort to allow any interested party to use the data.

Exceptions:

MGL Avionics reserves copyright and denies use of this data format for any military or related purpose.

Comments are welcome: info@MGLAvionics.co.za

Airports data format

The Airports file has a fixed name when used with an instrument, however any file name could be used. The default filename is "AIRPORTS.EWD".

Within this file an unrestricted number of airport records are stored.

The file starts with a 32 bit pointer to the start of the first flexible length airport record followed immediately with an index to all the airport records. Each index entry contains a type identifier, a six character airport identifier (1 to 6 characters in size), a 32 bit pointer to the start of the airport record in the file and finally a geographic position of the airfield (latitude and longitude).

The six character airport identifier is also found in the Enigma Waypoint file format and this should be considered as the key to link the two databases. For example, the waypoint file format has the long name "Cape town international airport" while the short name (or identifier) is "FACT". "FACT" will then be used as identifier in the airport data file index.

Kind:	byte (See airport types further on)
Identifier:	1 byte length of string (1..6), followed by six ASCII characters
Pointer:	32 bit signed integer
Latitude:	32 bit signed integer
Longitude:	32 bit signed integer

Each index entry is exactly 20 bytes in size. The first index entry starts at byte location 4 in the file.

Please note: The index must be sorted by identifier from lowest to highest using common ASCII sort criteria. (For example the identifier "ABC" is lower in value than "DE"). The sorting is vital as binary searches are prescribed to quickly locate a particular entry.

Strings are stored starting with a length byte, followed by a field that is the size of the maximum number of characters that can be stored in the string. Unused characters in a string are "don't care" values.

All integer values are stored in "little endian" format (also known as Intel style). The LSB occupies the lowest address location in the file.

Airport types

Defined airport types are currently any of the following:

- 0 – unspecified type
- 1 – 'AIRPORT' Typical assignment for medium sized airports
- 2 – 'MAJOR AIRPORT' Typical assignment for large and international airports
- 3 – 'SEAPLANE BASE'
- 4 – 'AIRFIELD' Typical assignment for smaller municipal airfields, glider fields etc

5 – 'PRIVATE AIRFIELD'

6 – 'ULTRALIGHT FIELD'

8 – 'HELIPORT'

These type identifiers are compatible with the waypoint type identifiers used in the Enigma Waypoint file format

Please note: If you add a airport type that you believe is of general interest – please let us know so we can update this document. Contact e-mail: info@MGLAvionics.co.za

Position data format

The Enigma position format is used in all Enigma data files. Enigma format allows usage of 32 bit integers while allowing position resolution to less than 3 feet.

Position latitude and longitude are stored as signed 32 bit integers.

Degrees are multiples of the value 180000 and any degree fraction is a multiple of 1/180000.

Degrees North and East are positive while degrees South and West are negative.

Example: North 45 degrees, 59 minutes, 30 seconds is 8278500.

Example: North 0 degrees, 30 minutes, 0 seconds is 90000.

Airport record data format

Each airport record consists of a fixed length data section followed by a flexible length list of frequencies applicable for this airport, followed by a flexible length of runway information for this airport, finally followed by an optional data section that can contain almost any type of data. This is intended to store image data such as approach plates and airport layouts (for example a graphic showing runways, taxiways and parking areas), however, it is flexible enough to store any kind of data.

Airport record fixed portion

Runways Pointer:	32 bit signed integer	Pointer to runway data
Data Pointer:	32 bit signed integer	Pointer to other data
Altitude:	16 bit signed integer	Altitude in feet. May be negative.
NumberOfFrequencies:	byte	Number of frequencies in this record
NumberOfRunways:	byte	Number of runways in this record
NumberOfOtherData:	byte;	Number of other data sections in this record

This is immediately followed by the list of frequencies if 1 or more frequencies is defined.

If no frequencies are defined, this is followed by the runway data if at least 1 runway is defined otherwise the “other data” follows here.

Airport record frequencies

The frequency section starts with a list of relative pointers to each frequency defined. Each pointer is relative to the position of the last byte of the fixed record portion +1 (i.e. The target address is the address of the first relative pointer plus the offset found in the pointer of interest).

For each frequency, a flexible length record is inserted. Each record is pointed to by a relative pointer.

Frequency:	32 bit unsigned integer	Frequency in Hz
Type:	1 byte length of string (1..4), followed by four ASCII characters	
Description:	1 byte length of string (0..50), followed by 50 ASCII characters	

The type string should follow recognizable abbreviations for the type of frequency used, for example "APP", "TWR", "CTAF", "GND" and so forth.

The descriptor may be any text, for example: "Tower, 07:00-17:00, 124.8Mhz other hours"

Airport record runway data

The runway section starts with a list of relative pointers to each runway defined. Each pointer is relative to the position of the last byte of the fixed record portion +1 (i.e. The target address is the address of the first relative pointer in the frequency section (if any) plus the offset found in the pointer of interest).

Designation:	16 bits	Designation code (see description below)
Length:	unsigned 16 bits	Runway length in feet
Width:	unsigned 16 bits	Runway width in feet
Bearing:	16 bits unsigned	True GPS approach bearing relative to first designation heading. 0XFFFF if no GPS approach data
Surface:	1 byte length of string (0..8), followed by eight ASCII characters	
Latitude1,Longitude1:	32 bit signed	position of first designation threshold
Latitude2,Longitude2:	16 bit signed	relative position of second designation threshold (relative to first designation threshold)
Altitude1,Altitude2:	signed 16 bits	altitude of threshold 1 and threshold 2 in feet

Surface is a ASCII string describing the surface of the runway. Examples: "TAR", "SAND", "GRASS".

Bearing bits 0-8 contains a bearing value of 0-359 true which is the exact bearing of the runway centerline to be used for GPS aided approaches. The bearing is the bearing onto the first designation runway heading. Reciprocal bearing is used for the second designation

heading. If no GPS approach data is available, this value is to set to 0xFFFF.

If bit 9 of the Bearing value is set, the 16 bytes from Latitude1 to Altitude2 contain localizer and glide slope data. The format of this has not been decided on at time of writing this document.

Latitude1/longitude1 contains the exact position of the runway threshold of the first designation heading in Enigma position format. Latitude2/Longitude2 specifies a relative offset for the second designation runway heading. Simply add these values to latitude1/longitude1 to obtain the position for the second threshold.

Altitude1 and Altitude2 specify the threshold altitudes in feet.

If GPS approach data is included, this data can be used by the instrument to guide the pilot down a glide slope path towards the runway.

Runway designation data format

The designation is stored as a 16 bit value.

If the value is greater or equal to 0x8000, the heading is expressed in cardinal points (often used for water “runways” for seaplanes). In this case the least significant 3 bits are used as an index to the heading as:

'N','NE','E','SE','S','SW','W','NW'.

The resultant designation is taken as index to the heading + '/' + reciprocal heading. For example 'NE/SW'.

If the value is less than 0x8000 then bits 12,13 and 14 define a designation type from 0 to 7:

Type 0: designation is formed from bits 0..5 and a reciprocal is calculated. Result is for example: 12/30. This designation is stored as value 0x000C.

Type 1: unused

Type 2: similar to Type 0 but used with dual runways with the first heading being the left runway. For example: 12L/30R

Type 3: As type 2 but first heading is the right runway. For example: 12R/30L.

Type 4: unused

Type 5: unused

Type 6: Helicopter pad. Bits 0..11 is the number of the pad from 1 to 4095. The value 0 is not used. Example: H1

Type 7: Water runway for seaplanes. Similar to type 0. Example: 12W/30W

Data sections

Each airport record may have zero to many data sections. Data sections may contain any data. Data types are identified by a single numeric in the index.

If at least one data section is identified, the data section starts with a table of a 16 bit type identifier followed by a 32 bit pointers to each data section. Please note that these pointers are absolute addresses in the data file and not relative as with frequencies and runways.

Each index entry for a data section is defined as follows:

Pointer:	32 bits unsigned	Absolute pointer to data
Type:	word	Type of data in this section

Currently the following types are defined:

- | | |
|---|---|
| 0 | Data is ASCII text. Characters 0x20 to 0x7F. 0x7F is the degree symbol, all other characters are standard ASCII. 0x0C,0x0A is used to terminate lines. Font is non-proportional allowing simple ASCII style layout. Any other value is interpreted as end-of-text.
Please note: Not more than 40 characters per line. Fixed 24 lines per page. Multiple pages may be used.
This data is mainly intended to be used as a brief description of the airport, cautions and other information relevant to a pilot using the airport. |
| 1 | Data is typeless image in Enigma MIF format |
| 2 | Data is approach plate in Enigma MIF format |
| 3 | Data is airport layout in Enigma MIF format |
| 4 | Data is parking bay layout in Enigma MIF format |
| 5 | Data is 8 bit/pixel Windows BMP image |
| 6 | Data is 8 bit/pixel Windows JPG image |
| 7 | Data is text/vector graphics in Enigma document format |

Comments

In order to find out the number of airport records, read the first 4 bytes in the file which is a pointer to the start of the data section (points one byte past the end of the index).

Subtract 4 from this value and divide the result by 20 (the size of each index entry). The result is the number of airport records.