

Airport, navigation aid and IFR intersection data in X-Plane

By Robin Peel, July 26th, 2000.
robin@cpwd.com
Version 1.3

Purpose of this FAQ.....	1
Who am I & what do I do?	1
What is the “master database”?.....	1
How is the data created, updated and generated for X-Plane?.....	2
How complete is the data?.....	2
Where do I find the latest data?	2
How can I correct errors or omissions I find in the data?	2
Where is the data stored in X-Plane and what data does each file contain?	3
General structure of apt.dat, nav.dat and fix.dat.....	3
Details – what do the entries in apt.dat mean?	3
Details – what do the entries in nav.dat mean?	5
Details – what do the entries in fix.dat mean?.....	6
Where are the localiser and glideslope aerals positioned in the “real world”?	6
I don’t see any DMEs in the nav.dat file. Where are they?	7
How do I convert my data to decimal degrees?.....	7
How do I calculate the position of somewhere in relation to somewhere else?.....	7
Why are some runway and localiser headings very precise (to three decimal places)?.....	8
Are there limits on the numbers of airports, runways and taxiways in X-Plane?.....	8

Purpose of this FAQ

This FAQ describes the contents and structure of the files that store airport, nav-aid (NDB, VOR and ILS) and IFR intersection data within the X-Plane flight simulator (published by Laminar Research). It is not a tutorial on how to fly an aeroplane within X-Plane – please see the many other excellent FAQs for advice upon how to use NDBs, VORs, ILSs, intersections, etc.

Who am I & what do I do?

Sometime in 1997, I volunteered to Austin Meyer (the owner/developer of X-Plane) to try and improve upon the very basic airport data included in X-Plane version 3.x. I built an application in Microsoft Access 97 (now converted to Access 2000) to store and manipulate the airport, nav-aid and intersection data. I have worked with Austin to add additional airport and nav-aid details to X-Plane (such as custom taxiways, airports outside the boundaries of the USA, more accurate runway markings, etc) and during this time the master database has grown to over 20MB in size.

I perform this task as a volunteer. The data is published as a free resource to the X-Plane community (and also for the FlightGear project), and is also the default data published with retail versions of X-Plane. But if you really appreciate my efforts, please make a contribution to your local humane society or dogs’ home (and let me know – I would like to keep tally)!

At present, I have around 400hrs of “real” flying experience, gained since 1989. I have a UK Private Pilot’s Licence (gained at White Waltham, EGLM), a US Private Certificate and a US Instrument Rating. I currently fly a rented C-172 SP out of Albuquerque Double Eagle II (KAEG) or more elderly C-172s out of Santa Fe (KSAF).

What is the “master database”?

The master database is my Microsoft Access 2000 database that stores all the X-Plane airport, nav-aid and intersection data. In database parlance, it is highly “normalized” and consists of over 35 Access tables accessed from a custom-designed Access 2000 application. The master database currently contains over 22,000 airports.

How is the data created, updated and generated for X-Plane?

I regularly import updates and amendments to the master database using routines that will automatically read fragments of new data sent to me by other users, and save the new data into my Access tables. Similarly, I have routines that will generate the data required by X-Plane in the necessary formats for X-Plane Classic or version 5.23 and beyond. I also use the master database to generate data for the FlightGear freeware project. These export routines sometimes have to prioritize the data in the master database to suit X-Plane's limitations (eg. currently, X-Plane cannot deal with more than 100 airports in a 2x3 degree latitude/longitude area). I try to generate a new version of the data every month or so – depending upon other commitments.

How complete is the data?

The data is pretty complete for the USA (based upon FAA data sources). Quality data for other countries is harder to obtain, but dedicated X-Plane users have sent me data for much of Canada, Western Europe (with some gaps), Japan and Australia. Some other random major airports also exist (eg. in Central and South America).

Where do I find the latest data?

I usually upload the data as a zip file to the official X-Plane FTP site at:

<ftp://ftp.x-plane.com/pub/scenery/>

The filename is typically "AptNavXXX.zip". The XXX refers to my version number for the data, currently 2.23. The zip file usually contains the latest copies of apt.dat, nav.dat, fix.dat, a ReadMe.txt file, and a copy of the FAQ in Adobe Acrobat (.pdf format).

I will announce new versions of the X-Plane data on the Announcement e-mail list (for details, see www.x-plane.com), or you may check the FTP site periodically for new data. I will also periodically update this FAQ and place it on the same FTP site as AptNavFAQ.zip and as AptNavFAQ.pdf.

How can I correct errors or omissions I find in the data?

First, you need to find a good source of information. An official chart of the airport (such as those published by Jeppesen) is a good starting point – they often have a helpful latitude/longitude scale around the edges that you can use to figure out the exact positions of runways and taxiways. Remember that these scales appear to be backwards for western longitudes and southern latitudes!

Then you need to get the data into the apt.dat and nav.dat files. X-Plane comes with a tool (WorldMaker) that includes modules for editing airports and nav-aids. Instructions for WorldMaker are in the X-Plane manual. Sometimes it is easier to edit the apt.dat and nav.dat files directly (using a text editor) rather than by using WorldMaker. Here are some tips for using WworldMaker:

- In the airport editor, taxiways are created just like runways, but taxiways must have their "runway number" set to "99" to identify them to X-Plane (and WorldMaker) as taxiways.
- Save your work often! It is easy to accidentally select (and then delete or move) a runway/taxiway that you just spent ages positioning correctly.
- Remember to set the lighting for taxiway edge lights, if you so wish.

I am afraid that this FAQ does not further address the issues associated with using WorldMaker. Queries should be answered by the X-Plane manual or by Laminar Research (Austin@x-plane.com).

Finally, send errors, corrections and additions to me (robin@cpwd.com). The easiest way to send new or corrected data is to copy just the appropriate lines from your apt.dat, nav.dat and fix.dat files into a plain text file, and attach the text file to an e-mail. All three of these files are just plain text files, and can be edited with any text editor (or a word processor operating in "text-only" mode). Note that in Windows, the Notepad editor sometimes balks at the size of these files (it seems to be a problem in Windows 95/98, but not in Windows 2000).

Please do not send me:

- Your entire apt.dat file (with your corrections embedded somewhere in its midst) – these are impossible for me to sort through!
- Scanned copies of charts in the hope that I will spend several hours trying to plot the positions of runways and taxiways for you.

Where is the data stored in X-Plane and what data does each file contain?

The following files contain airport, nav-aid and intersection data in X-Plane:

Apt.dat	Airports and heliports, with their runways and taxiways.
Nav.dat	NDBs, VORs and individual ILS elements.
Fix.dat	IFR intersections (often referred to as “fixes”).

These files are all installed in the “additional nav data\earth” folder of your X-Plane installation. They are all plain text files that can be viewed or edited with any text editor (such as Notepad on a Windows PC). Be warned that, on occasions, a file created on Macintosh may not display carriage returns correctly on a PC.

General structure of apt.dat, nav.dat and fix.dat

Features common to each file are:

- The first line of each file indicates if the file was generated on a PC (“I” for Intel) or a Macintosh (“A” for Apple). I believe that X-Plane uses this code to help deal with the different ways in which PCs and Macs manipulate carriage returns in text files.
- The second line contains an internal version number used by X-Plane, along with my own copyright message (that also includes the version number of the data).
- The very last line of each file is marked by a “99”.
- The order in which data is stored in the files is conceptually unimportant ... but it does determine the sort order of the data in X-Plane’s menus and selection screens. By default the data is sorted as follows:
 - Airports (apt.dat) are sorted by their name (and usually the name is prefixed by the nearest city, though this practice varies by country). Closed airports (name prefixed with an “X”) and helipads (name prefixed with an “H”) are sorted to the end of the list
 - Nav aids (nav.dat) are sorted first by type (NDB, VOR, LOC, GS etc) and then by their name.
 - IFR intersections (fix.dat) are sorted by their name.

IMPORTANT: All headings referenced in the files are **true** (not magnetic). X-Plane has an internal model of magnetic variation that will be used to properly align VORs, etc.

Data is stored on a line-by-line basis. The type of data stored on each line in apt.dat and nav.dat is indicated by the first integer value on each line.

The meanings of these line codes in apt.dat are:

1	Airport header data
10	Runway or taxiway at an airport

The meanings of these line codes in nav.dat are:

2	NDB
3	VOR
4	Localiser that is part of a full ILS (inc. IGS)
5	Stand-alone localiser (inc. LDA and SDF).
6	Glideslope
7	Outer marker (OM)
8	Middle Marker (MM)
9	Inner Marker (IM)

Since the fix.dat file contains only IFR intersections, no codes are used to differentiate the data

Individual data elements on each line can be separated by any number of spaces – however, it is a good plan to keep them aligned in columns – it is easier to spot silly errors in this way.

Details – what do the entries in apt.dat mean?

Each airport has a header line (code 1) and one or more runway/taxiway lines (code 10). The runways for each airport **MUST** follow the airport header line. Taxiways should follow the runways. A blank line can be used to separate different airports – this helps improve readability.

Here is a simplified fragment for part of an airport in apt.dat:

```
1  5352 1 0 KABQ Albuquerque International Sunport
10 35.044209 -106.598560 08 x 90.428 13775 150 2252231
10 35.032077 -106.618983 03 x 44.513 10000 150 2352321
10 35.044022 -106.611983 17 x 183.135 10000 150 1231231
10 35.039105 -106.614064 12 x 128.839 5142 150 1121331
10 35.034420 -106.607850 99 x 218.800 200 100 2161161
```

This shows one airport header line, four runways and one taxiway. The order of the data on the airport header line is (using the above example):

1 = This is an airport header line
5352 = Airport elevation in feet (above MSL).
1 = Airport has a control tower (1=yes, 0=no). Used by X-Plane's ATC system.
0 = Display X-Plane's default airport buildings (1=yes, 0=no).
KABQ = ICAO code for the airport. All airports need a unique ICAO code.
"Albuquerque International Sunport" = airport name.

The data on the runway/taxiway lines is a little more complex. Using the first runway in the above example data:

10 = This is data line for a runway or taxiway.
35.044209 = Latitude (in decimal degrees) of runway center.
-106.598560 = Longitude (in decimal degrees) of runway center.
08 = Runway number (integers only). "99" implies a taxiway.
x = Runway number suffix (eg. "L", "R" or "C"). "x" implies no suffix.
90.428 = **True** (not magnetic) heading of the runway in degrees.
13775 = Runway length in feet.
150 = Runway width in feet.
2252231 = Runway surface type and lighting. These are the most complex codes. The first digit refers to the runway surface for the entire runway. The following three digits define the lighting for the runway as seen when approached from the direction implied by the runway number (08 in our example), and the final three define the lighting for the runway as seen when approached from the opposite end (26 in our example).

The meaning of the runway surface and lighting codes are:

2 = Runway surface type (1= "asphalt", 2= "concrete", etc – see below for details)
2 = Runway end "A" (08): Visual approach path (VASI / PAPI etc.) lighting. See below for details.
5 = Runway end "A" (08): Runway lighting. See below for details.
2 = Runway end "A" (08): Approach lighting. See below for details..
2 = Other runway end (26): Visual approach path (VASI / PAPI etc.) lighting.
3 = Other runway end (26): Runway lighting.
1 = Other runway end (26): Approach lighting.

Runway surface types:

1 = Asphalt
2 = Concrete
3 = Turf/grass
4 = Dirt
5 = Gravel
6 = Asphalt helipad (big "H" in the middle!)
7 = Concrete helipad (big "H" in the middle!)
8 = Turf helipad (big "H" in the middle!)
9 = Dirt helipad (big "H" in the middle!)

Visual approach path codes:

1 = No visual approach path lighting
2 = VASI (Visual Approach Slope Indicator)
3 = PAPI (Precision Approach Path Indicator)
4 = Space Shuttle Landing PAPI (20 degree glide path) – used is very rare!

Runway lighting codes:

- 1 = No runway lighting
- 2 = Runway edge lights
- 3 = Also has REIL (Runway End Identification Lights)
- 4 = Also has CLL (Center Line Lighting)
- 5 = Also has TDZ (Touch Down Zone) lighting
- 6 = Only has blue taxiway edge lights (use on taxiways!).

[Note that for values 1-5, the codes are cumulative]

Approach lighting codes:

- 1 = No approach lights
- 2 = SSALS (Simplified short approach light system)
- 3 = SALSF (Short approach light system with sequenced flashing lights)
- 4 = ALSF-I (Approach light system with sequenced flashing lights)
- 5 = ALSF-II (Approach light system with sequenced flashing lights and red side bar lights the last 1000')
- 6 = ODALS (Omni-directional approach light system)

[Note – this does not cover all the possible real-world approach light systems. Obscure types of approach lighting, such as the British “Calvert”, will need to be mapped to the nearest match]

Details – what do the entries in nav.dat mean?

Each nav-aid type is on a separate line, usually sorted by the nav-aid name within each nav-aid type. Unfortunately, this separates the many elements of an ILS (eg. LOC, GS, OM, MM, IM) and can cause some confusion.

Here are some example lines, showing selected nav-aids in the Albuquerque area:

2	34.987022	-106.620384	5304	247	0.000	ILT	Isleta	NDB
3	35.043796	-106.816312	5740	11320	0.000	ABQ	Albuquerque	VORTAC
4	35.044026	-106.570548	0	11190	90.428	ISPT	KABQ 08	ILS
6	35.043212	-106.614641	5352	11190	300090.428	ISPT	KABQ 08	GS
7	35.046352	-106.742583	0	0	90.428	----	KABQ 08	OM
8	35.044686	-106.628247	0	0	90.428	----	KABQ 08	MM

The meaning of this data for the NDBs (code 2) and VORs (Code 3) are:

2 = Type of nav-aid.

34.987022 = Latitude of nav-aid in decimal degrees.

-106.620384 = Longitude of nav-aid in decimal degrees.

5304 = Elevation of nav-aid in feet above MSL.

247 = Frequency of nav-aid. (Always an integer – so VOR and localiser frequencies multiplied by 100).

0.000 = Not currently used except to indicate the heading of a localiser in **true** (not magnetic) degrees.

ILT = Identification of nav-aid (broadcast in morse code in X-Plane).

“Isleta NDB” = Name of nav-aid.

Localisers (code 4 and 5) are very similar, except that:

- They include the heading of the localiser (as described above).
- The nav-aid name is replaced by the airport code, runway number and ILS component type (eg. “KABQ 08 ILS”). This data is not used by X-Plane, but is used by my import programmes that allocate the ILS elements to a valid runway in the master database. I cannot import ILS data without these codes.

Glideslopes (code 6) have very similar data to localisers, except that the glideslope angle (typically 3 degrees) is combined with the heading in a slightly complex fashion. A glideslope of 3 degrees on a heading of 90.428 (true) will be listed as “300090.428”. This is calculated by:

- Rounding the glideslope angle (3 degrees in our example data) to two decimal places (3.00).
- Multiplying the result by 100,000 (giving 300000).
- Adding the result to the heading (300000 + 90.428 = 300090.428).

ILS marker beacons (codes, 7, 8 and 9) are formatted just like localisers, but note that:

- They have no associated frequency (use “0”).
- The heading is important, to ensure that they are oriented and displayed correctly on X-Plane’s approach charts.
- Marker beacons do not have a Morse code identification, so “----” is used as a default.
- For a Locator Outer Marker (LOM), which is an NDB co-located with an OM, the NDB must be added to nav.dat as a separate, stand-alone NDB (at the same location!). The name of the OM must remain in the “KABQ 08 ILS” format – but the NDB may have the appropriate name (eg. “Wobin LOM”).

Details – what do the entries in fix.dat mean?

This is the easiest file to interpret! Each intersection is on a separate line. Here is an example line:

```
35.162472 -106.646500 WOBIN
```

The meaning of this data is:

35.162472 = Latitude in decimal degrees.

-106.646500 = Longitude in decimal degrees

WOBIN = Intersection name (always five characters and must be unique).

Currently, intersections are displayed on X-Planes charts as little triangles. If you zoom in quite close, the intersection name will appear. There are plans afoot to add intersections to X-Plane’s GPS.

Where are the localiser and glideslope aerals positioned in the “real world”?

Flight simulator pilots often get confused about where the aerals that form the components of an ILS are positioned in relation to the runway. Here is a simple example for a fictitious ILS for runway 09.

The localiser (which provides left-right guidance to the pilot) is usually positioned just beyond (500 – 1000 feet) the far end of the runway it serves (ie. beyond the eastern end of our example runway). The localiser’s beam points back down the runway (westward in our example) towards an approaching plane – the center of the beam passes through the runway’s touch down zone. You can see the localiser aerial at your nearest major airport – the aerial is a wide, flat thing, usually painted red, and hidden amongst the forest of approach lighting for the opposite runway (runway 27 in our example). It is usually the first valuable thing destroyed by an aeroplane that over-runs the end of a runway – or by an aeroplane the approaches the opposite end of the runway a little low.

Some localisers exist in isolation from other components of an ILS. These form part of Localiser (LOC), Localiser Directional Aid (LDA) or Simplified Directional Facility (SDF) approaches. Such aerals can be positioned wherever is most useful ... an extreme example is the LOC on top of a mountain near Aspen, Colorado (KASE) that is used to provide guidance for departures.

The glideslope (GS) aerial (which provides up-down guidance to the pilot) is usually positioned just to one side of the runway’s touch down zone (TDZ), which is about 1,000’ along the runway from the threshold. Typically, a GS aerial might be 200 – 300 feet to the left (north in our example for runway 09) of the TDZ. Again, you can see this vertical aerial at your local airport – it often has a small shack close by housing the electrical gear. The shack is often painted in lurid red/white or orange – presumably in an attempt to stop errant from aeroplanes flying (or taxiing) into it. The beam from the glideslope is typically angled up from the TDZ at 3 degrees, though this may vary. Steeper angles may not sound significant (say 5 degrees) but they are surprisingly disconcerting to nervous passengers.

The marker beacons (which provide information to a pilot about the distance from the runway) are placed on the ground at certain distances from the runway’s threshold. The Inner Marker (IM) is usually very close to (or at) the runway threshold. The Middle Marker (MM) is typically 3,500’ out from the threshold (to the west in our example) and usually indicates a point at which an approaching aeroplane on the glideslope should be 200’ above the TDZ elevation. The Outer Marker (OM) varies in location, but is usually 4 – 7 nautical miles for the runway. Not all ILS approaches have the full complement of marker beacons – inner markers are relatively rare.

Here is a summary for an ILS for our example runway 09:

```
OM      MM      IM      #  
( )      ( )      ( ) 09=====27  @
```

Where:

```
# - Glideslope aerial  
@ - Localiser aerial  
= - Runway 09/27  
( ) - Marker beacons
```

I don't see any DMEs in the nav.dat file. Where are they?

Currently, X-Plane does not support stand-alone DMEs. X-Plane incorrectly assumes that every VOR and every ILS has a co-located DME.

Though most VORs / VORTACs in the USA *do* have a co-located DME, this is often *not* the case in the rest of the world. An NDB with a co-located DME is not uncommon in Europe. Note also, the positioning of a “real-world” DMEs associated with an ILS varies widely – the DME might be co-located with the localiser or glideslope aerals ... or it may be somewhere else, or it may not exist at all. A study of some approach charts will reveal the variations, and the title of the approach will include the phrase “DME” if DME forms a required part of the approach (eg. “ILS/DME 08”).

I do have some DME data in my master database ... hopefully this can be more effectively utilized in future X-Plane releases.

How do I convert my data to decimal degrees?

X-Plane defines all positions as “decimal degrees” to six decimal places (eg. -123.456789). This makes mathematical calculations faster.

But remember from your basic school geometry that a degree is *traditionally* subdivided into 60 minutes, and that a minute can be further subdivided in 60 seconds. Some aviation data sources choose not to use the “seconds” – instead they use decimal parts of a minute. Other sources use data defined in degrees and the decimal part of degrees, just as in X-Plane. Here are some example data formats (all refer to the same position):

```
N35.5000 W106.5000 (Decimal degrees, or dd.dddd)  
35 30.00N 106 30.00W (Decimal minutes, or dd mm.mm)  
35 30 00N 106 30 00W (Degrees, minutes and seconds, or dd mm ss)
```

X-Plane has a convention that western longitudes and southern latitudes are negative numbers when converted to decimal degrees. So data for the USA will have positive latitudes and negative longitudes (see all the example data quoted above).

So, to convert a dd mm.mm format (eg. 35 30.00N) to decimal degrees, you need to:

- Divide the minutes by 60 (in our example: $30.00/60 = 0.5$).
- Add this result to the degrees (in our example: $35 + 0.5 = 35.5$).
- Check the sign – south or west is negative (in our example, north is positive).
- And so the converted answer is: 35.50. How do I calculate the position of somewhere in relation to somewhere else?

[From JJ Brennan] For those who don't like to do the math, there is a very useful little program called LLCALC that will do latitude/longitude calculations. It's also very useful if you wish to locate a point in some relation to another point (like placing an ILS GS transmitter alongside a runway, or finding the end points of a runway from its center point).

A copy can be found at:

<ftp://ftp.kingmont.com/pub/kingmont/x-plane/llcalc.zip>

Why are some runway and localiser headings very precise (to three decimal places)?

There are two reasons why runways, taxiways and localisers have very “precise” headings in X-Plane:

- The data reflects the “real world” headings very accurately.
- Small corrections have been applied to correct for X-Plane’s modeling of the “round world”.

Most “real-world” runway and localiser headings are quoted on charts to the nearest whole degree (magnetic), since this is usually more than accurate enough for day-to-day flying. But, of course, runway and localisers are not always laid out neatly in whole degree increments ... and X-Plane needs to have precise headings (in **true** degrees, not magnetic) in order to make localiser alignments work, and ensure that taxiways join up neatly with each other and with the runways. An error of 0.5 degrees in a runway heading can misplace the runway threshold by up to 50 feet on a 12,000’ runway ... a significant margin if you are trying to land a Boeing 747.

Also, X-Plane still has some significant internal coding issues that need to be resolved. One of these issues is the way in which the real-world “round” (actually, an oblate spheroid) earth is treated. X-Plane has its own internal co-ordinate system (a “flat earth” model) that does not accurately portray the real-world precise alignment of the headings of runways, taxiways and localisers. Headings are skewed in an anti-clockwise direction (in the northern hemispheres). The effect is worse toward the western edge of each degree of longitude, and is also much worse in extreme northern or southern latitudes.

At the mid-latitudes the errors are small, but are still sufficient to cause localisers to be significantly mis-aligned from the runway centerline at the runway touch-down zone. In an effort to correct for these errors, the routines that export data from my master database to the X-Plane file formats will attempt to calculate the error that X-Plane will introduce. This small correction (usually less than half of a degree) is applied to the headings of localisers, runways and taxiways. The resulting headings are saved in the apt.dat and nav.dat files with a precision of three decimal places.

Laminar Research has committed to resolving this “flat earth” problem in X-Plane in the release of version 6.0, at which point a true “round-world” model will be used (that will be compliant with the WGS-84 standard).

Are there limits on the numbers of airports, runways and taxiways in X-Plane?

The short answer is “yes”. The limits have changed between X-Plane versions. Here is a summary based upon X-Plane version 5.31 (except where noted).

Description	Limit
Total number of runways and taxiways (combined)	24,000
Total number of airports in a 2x3 degree latitude/longitude region	100
Total number of runways/taxiways in a 2x3 degree latitude/longitude region	400 (600 in 5.32 beta 8)
Total number of runways/taxiways at an airport	400 (600 in 5.32 beta 8)