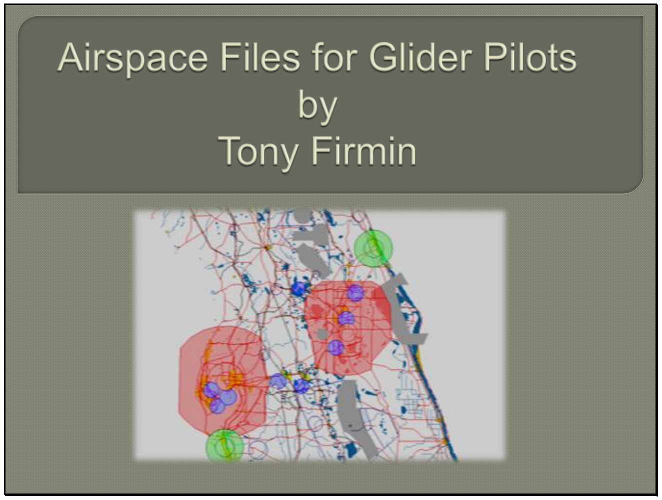
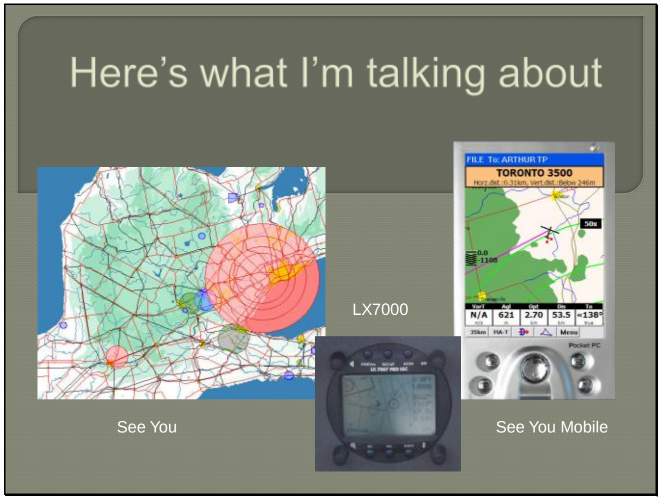


Airspace Files for Glider Pilots
by
Tony Firmin

A survey of Airspace files available for glider pilots and some useful background information.



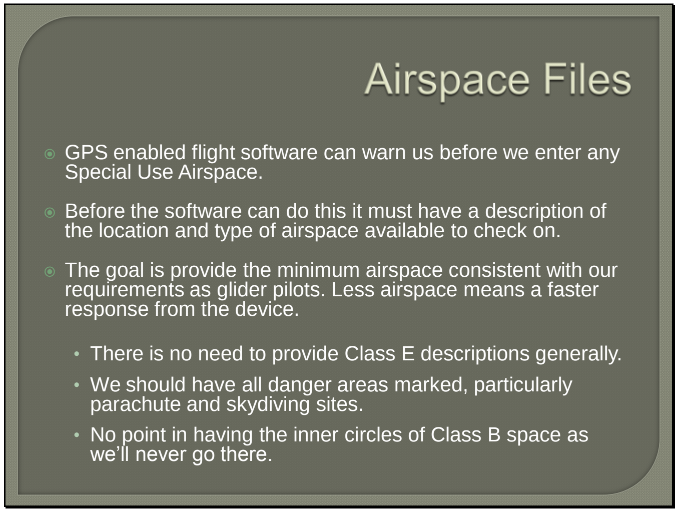
Here's what I'm talking about

See You

LX7000

See You Mobile

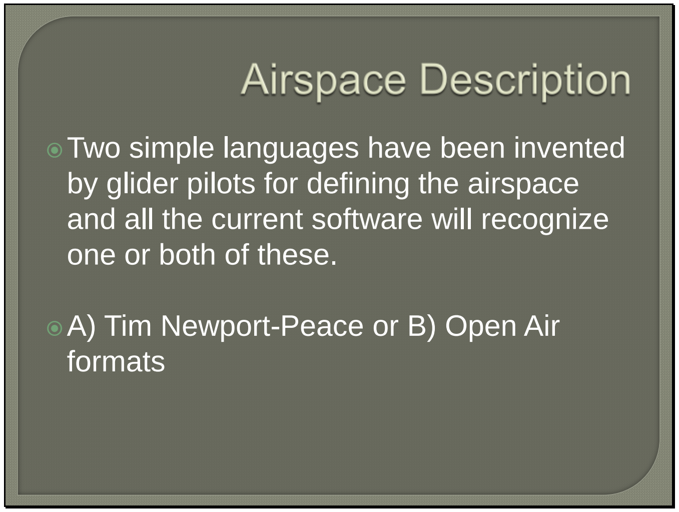
PC software, PDA software and some instruments will display Controlled and Special Use airspace. It is common today to use a PDA coupled to a GPS to keep track of where the glider is, relative to the defined airspace.



## Airspace Files

- GPS enabled flight software can warn us before we enter any Special Use Airspace.
- Before the software can do this it must have a description of the location and type of airspace available to check on.
- The goal is provide the minimum airspace consistent with our requirements as glider pilots. Less airspace means a faster response from the device.
  - There is no need to provide Class E descriptions generally.
  - We should have all danger areas marked, particularly parachute and skydiving sites.
  - No point in having the inner circles of Class B space as we'll never go there.

In order for this airspace description to be available to the software, a description needs to be provided. As far as glider pilots are concerned the less information needed the better, as being warned unnecessarily is distracting and slows the programs response. So generally we can ignore Class E definitions as this is largely an issue for IFR pilots. We don't need the inner most circles of Class B space as we are never going to go there. But we do need Danger areas where for example parachutes may be in use or in some cases model aircraft or model rockets. Because the best knowledge of the location where the pilot flies, is best known to the local pilots, these files are probably best "tuned" up by people with this local knowledge.



## Airspace Description

- Two simple languages have been invented by glider pilots for defining the airspace and all the current software will recognize one or both of these.
- A) Tim Newport-Peace or B) Open Air formats

Two (unfortunately) simple languages have been invented to describe the location and description of the controlled space. The first by Tim Newport-Peace around 1996 and then by the authors of WinPilot around 1998 the latter is called the Open Air format. Both are text based languages that can be interpreted by any software that cares to use them.

## Airspace Files

- Files for countries around the world have been collected on a site hosted by John Leibacher. http://soaringweb.org/
- You may also obtain these from fellow pilots in your club or maybe your club website.
- The files for Southern Ontario are called sontario_2009.sua and .txt

Files for North America and around the world have been contributed in either of these formats by various people to John Leibacher's famous website http://soaringweb.org/ where this and turnpoint data are available. These files are also circulated locally through the grapevine or club websites. In Southern Ontario the latest data is now available on Soaringweb site as sontario_2009.sua (and txt)

## Tim Newport-Peace

- The first airspace description language was defined by Tim Newport-Peace in the UK around 1996. As a result it was influenced by their requirements.
- The full description of this language can be found on Leibacher's site
- Each program has then to interpret this, which hopefully it will do correctly !

Typical Statements

TITLE=

TYPE= CTA/CTR; RESTRICTED; DANGER etc

CLASS= A,B,C,D etc

POINT= Lat/Lon

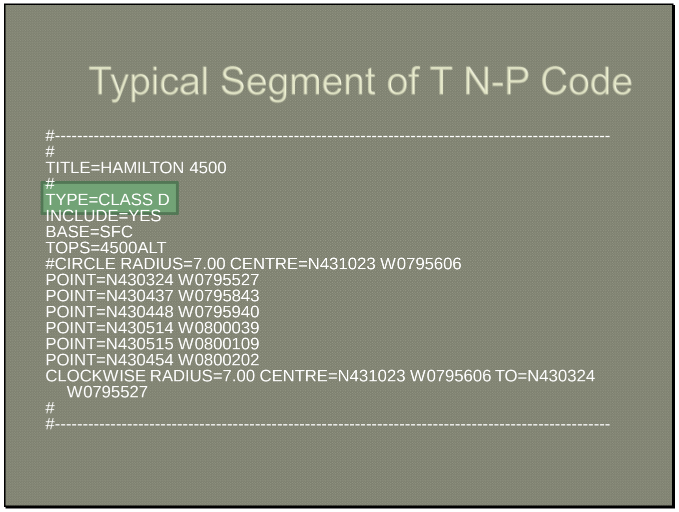
CLOCKWISE RADIUS= Lat/Lon

TOPS= height
BASE= height

Quickly looking at the T N-P definition the language is of the form Command= followed by some relevant data. A block of airspace is supposed to start with the TITLE= statement which indicates the commencement of a new block then followed by a number of lines that detail the space. This commences with a TYPE= statement which may be followed by a number of preset words such as CTA/CTR or RESTRICTED. If the CTA/CTR parameter is used then the classification of the airspace is given by the CLASS= statement followed by a letter A-G.

A full description of the syntax is available on John Leibacher's site.

Each program has to be written to interpret this into whatever graphics it subsequently displays.

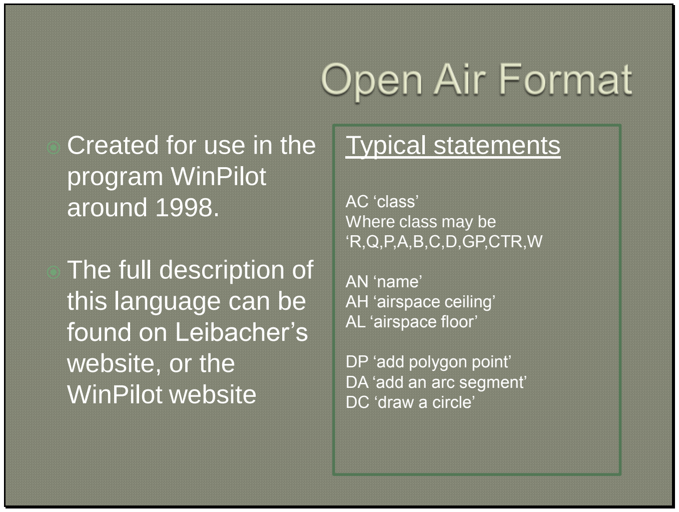


## Typical Segment of T N-P Code

```
#--------------------------------------------------------------------------------------------------
#
TITLE=HAMILTON 4500
#
TYPE=CLASS D
INCLUDE=YES
BASE=SFC
TOPS=4500ALT
#CIRCLE RADIUS=7.00 CENTRE=N431023 W0795606
POINT=N430324 W0795527
POINT=N430437 W0795843
POINT=N430448 W0795940
POINT=N430514 W0800039
POINT=N430515 W0800109
POINT=N430454 W0800202
CLOCKWISE RADIUS=7.00 CENTRE=N431023 W0795606 TO=N430324
    W0795527
#
#--------------------------------------------------------------------------------------------------
```

An example of a typical block of airspace shows how this is typically done. The # character means the rest of the line will be ignored, so will be taken as a comment. A straight line is drawn between each POINT statement and the CLOCKWISE command defines a circular arc of a given radius and centre between to end points.

**Note however:** TYPE=CLASS is not how T N-P intended this to be defined and is a new statement that has crept into the language.



## Open Air Format

- Created for use in the program WinPilot around 1998.
- The full description of this language can be found on Leibacher's website, or the WinPilot website

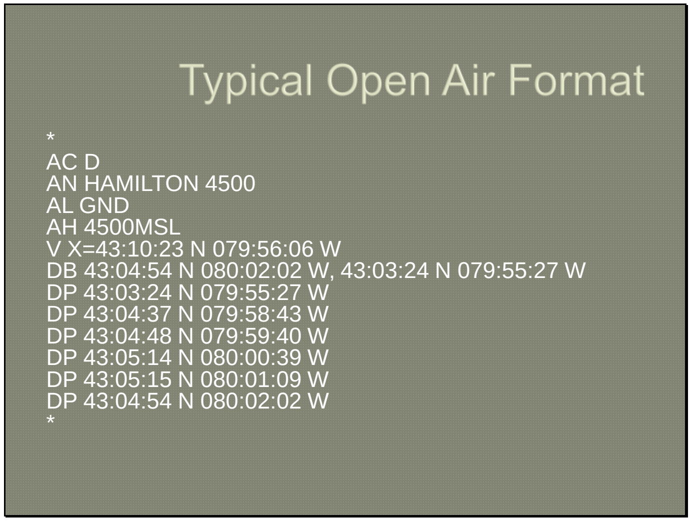
Typical statements

AC 'class'
Where class may be
'R,Q,P,A,B,C,D,GP,CTR,W

AN 'name'
AH 'airspace ceiling'
AL 'airspace floor'

DP 'add polygon point'
DA 'add an arc segment'
DC 'draw a circle'

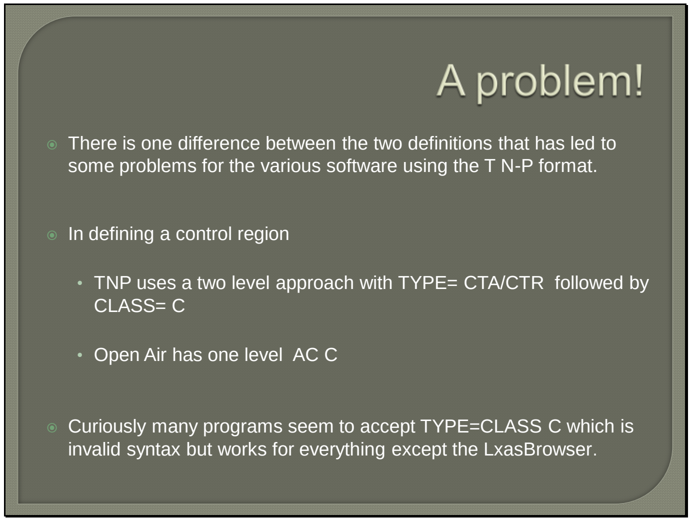
Moving on to the OpenAir format, this was produced about 1998 for the American program WinPilot. This takes into account the somewhat simpler airspace definitions used in North America. In particular it has a single level definition of the type of airspace given by the AC statement followed by letters indicating whether its A-D (note no E) and Q means danger, R restricted and so on. The rest of the description is very similar to T N-P though more cryptic. Again this is available on John Leibacher's site.



## Typical Open Air Format

```
*
AC D
AN HAMILTON 4500
AL GND
AH 4500MSL
V X=43:10:23 N 079:56:06 W
DB 43:04:54 N 080:02:02 W, 43:03:24 N 079:55:27 W
DP 43:03:24 N 079:55:27 W
DP 43:04:37 N 079:58:43 W
DP 43:04:48 N 079:59:40 W
DP 43:05:14 N 080:00:39 W
DP 43:05:15 N 080:01:09 W
DP 43:04:54 N 080:02:02 W
*
```

An example for the same description given by the T N-P format previously shows the command that define this block. Note that the V command followed by X= sets the centre for the DB arc definition.

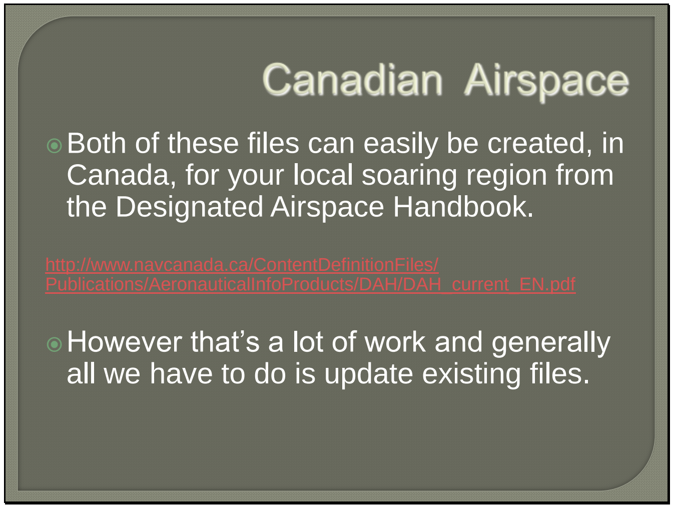


## A problem!

- There is one difference between the two definitions that has led to some problems for the various software using the T N-P format.
- In defining a control region
  - TNP uses a two level approach with TYPE= CTA/CTR followed by CLASS= C
  - Open Air has one level AC C
- Curiously many programs seem to accept TYPE=CLASS C which is invalid syntax but works for everything except the LxasBrowser.

The problem that is evident from this is that the T N-P syntax that uses the potential for two levels of definition of airspace is in conflict with the single level described with the Open Air format. When we look at the software SeeYou and StrePla we see that they two effectively subscribe to this single layer approach as, for example, they cannot show an Airway with a Class attached. Apparently in the UK an airway can change Class along its length.

The best compromise that exists currently is to structure the .sua file with the TYPE=CLASS statements where appropriate and recommend that someone wanting to use the Lxasbrowser uses the .txt file. In order to create the file needed for input to the LX, SeeYou may also be used.



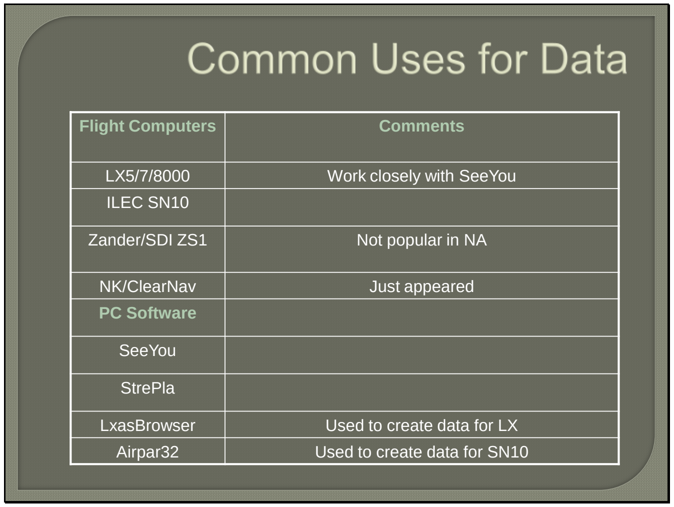
## Canadian Airspace

- Both of these files can easily be created, in Canada, for your local soaring region from the Designated Airspace Handbook.

http://www.navcanada.ca/ContentDefinitionFiles/Publications/AeronauticalInfoProducts/DAH/DAH_current_EN.pdf

- However that's a lot of work and generally all we have to do is update existing files.

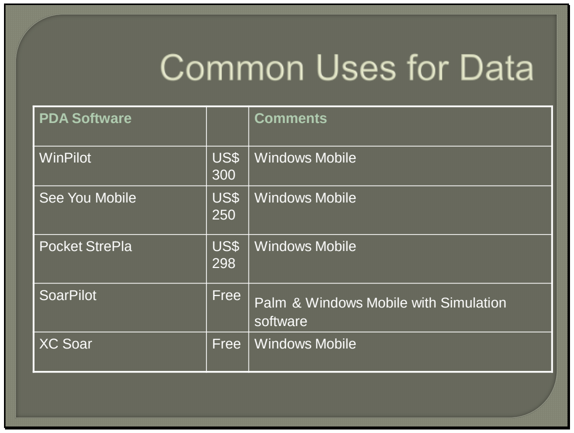
In Canada our airspace is spelled out in the Nav. Canada bible called the Designated Airspace Handbook. Here the descriptions are given in Lat/Long coordinates that can be immediately coded into either of these forms of syntax. This could be a lot of fairly tedious work but fortunately someone has already started this and most of the time we just need to keep these files up to date.



## Common Uses for Data

| Flight Computers | Comments |
|---|---|
| LX5/7/8000 | Work closely with SeeYou |
| ILEC SN10 | |
| Zander/SDI ZS1 | Not popular in NA |
| NK/ClearNav | Just appeared |
| PC Software | |
| SeeYou | |
| StrePla | |
| LxasBrowser | Used to create data for LX |
| Airpar32 | Used to create data for SN10 |

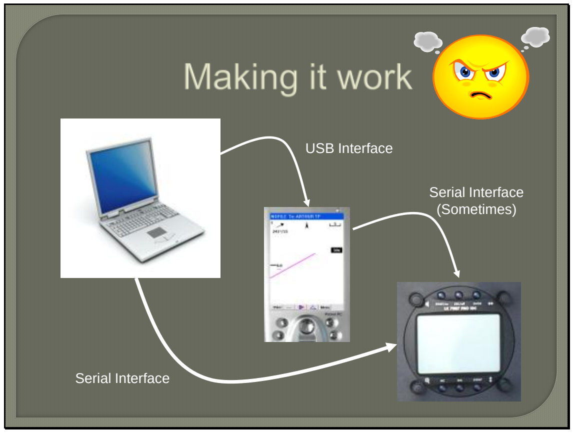
The flight computers and PC software listed above are those that provide this capability and are in common use today



## Common Uses for Data

| PDA Software | | Comments |
|---|---|---|
| WinPilot | US$ 300 | Windows Mobile |
| See You Mobile | US$ 250 | Windows Mobile |
| Pocket StrePla | US$ 298 | Windows Mobile |
| SoarPilot | Free | Palm & Windows Mobile with Simulation software |
| XC Soar | Free | Windows Mobile |

The common programs running on today's PDA's are listed along with their approximate prices. The last one is an attempt at an OpenSource software package.



## Making it work

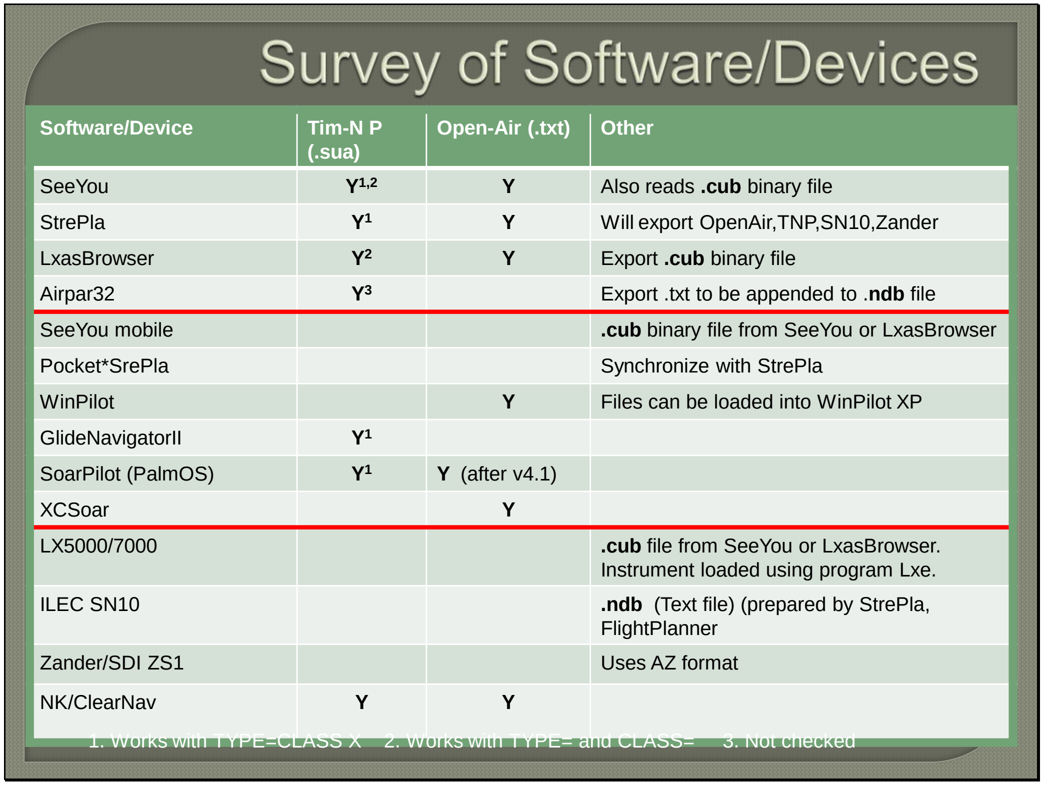
Now comes the frustrating part how do plug all this stuff together and make it work. First let me say it can be done!

There are some gotcha's hidden in the bushes unfortunately particularly, when trying to get the serial interfaces to connect correctly. In this sense by far the best solution is the one that LX came up with which is to provide a memory card reader in the instrument. This will be the standard in the near future I suspect.

However for the rest of us here are some of the show stopper's

- The instrument has a serial connection but you do not have a serial port on your shiny new laptop.
- You buy a USB serial port adapter and it doesn't work (curse Microsoft for their implementation)
- Synchronizing the PDA with the laptop is generally OK because HP/Compaq provide a USB connection but connecting to an instrument requires a serial cable.
- The PDA has to be set to the specific Baud rate the instrument requires, or it will not talk to it.
- Then of course there is a menu or two to negotiate.

Fortunately the glider pilot grapevine is pretty strong so all you need to do is to seek out someone who has figured this out before.



## Survey of Software/Devices

| Software/Device | Tim-N P (.sua) | Open-Air (.txt) | Other |
|---|---|---|---|
| SeeYou | **Y**[1,2] | **Y** | Also reads **.cub** binary file |
| StrePla | **Y**[1] | **Y** | Will export OpenAir,TNP,SN10,Zander |
| LxasBrowser | **Y**[2] | **Y** | Export **.cub** binary file |
| Airpar32 | **Y**[3] | | Export .txt to be appended to **.ndb** file |
| SeeYou mobile | | | **.cub** binary file from SeeYou or LxasBrowser |
| Pocket*SrePla | | | Synchronize with StrePla |
| WinPilot | | **Y** | Files can be loaded into WinPilot XP |
| GlideNavigatorII | **Y**[1] | | |
| SoarPilot (PalmOS) | **Y**[1] | **Y** (after v4.1) | |
| XCSoar | | **Y** | |
| LX5000/7000 | | | **.cub** file from SeeYou or LxasBrowser. Instrument loaded using program Lxe. |
| ILEC SN10 | | | **.ndb** (Text file) (prepared by StrePla, FlightPlanner |
| Zander/SDI ZS1 | | | Uses AZ format |
| NK/ClearNav | **Y** | **Y** | |

1. Works with TYPE=CLASS X   2. Works with TYPE= and CLASS=   3. Not checked

Finally here is the matrix which summarizes which files work with which software. Note the LX and SN10 need their own databases to be created, .cub and .ndb respectively.