AirScout

Aircraft Scatter Prediction

Copyright: DL2ALF
Version: 1.4.0.0
Last Edit: 2024/01/08

Table of Content

Table of Figures

1. Preface

The use of Aircraft Scatter (AS) to enlarge distances of ham radio contacts on VHF and above frequencies is being practiced since years. Especially in Australia the theory and practical use of AS has been described in many articles since the end of the 80s [2].

A new quality in prediction and online surveillance of AS contacts came up with the roll-out of ADS-B in 2006 (see Wikipedia). The automated, undirected beacon transmissions of aircrafts contain position reports which are primary intended to increase air traffic safety, but can be received by every person having a suitable receiver for 1090 MHz. Following the timetable of roll-out, in 2020 each civil aircraft must be equipped with ADS-B. In 2013 most of the civil aircrafts used for public transportation are already equipped. A description of receiving ADS-B signals with a special receiver and a documentation of extensive AS tests in Europe was published in 2006 by DF9IC [3].

Next step was the broadcast of received position reports via Internet. There is now a large community feeding a couple of web pages allowing other persons to see aircraft positions online on a map. This service is used by flight enthusiasts all over the world and is giving not just the position but also a lot of additional information, like altitude, speed, track etc.

The aim of AirScout is now to put all these information together in one easy applicable tool for ham radio operators. It provides the following functionality:

- Calculate a propagation path as a cross circle path between two QSO – partners
- Calculate a path profile between both QSO – partners using a Digital Elevation Model (DEM)
- Calculate the mutual visibility of an aircraft from both QSO – partners for each point on the path using their elevation and any possible obstruction between them
- Calculate a “hot area” in which an aircraft is mutually visible from both QSO – partners where a reflection is theoretically possible
- Show calculated path and aircrafts in real time on a map
- Predict Aircraft Scatter potential for each single aircraft according to position, track and altitude

The use of the software is free for personal, ham radio related use. It is copyrighted by me, DL2ALF. It has been developed using C# and Microsoft Visual Studio Express. It is released under the GNU Public License V3.0. It could not have been developed without the great help of Open Source software found on the Internet for almost all presentation and calculation tasks inside the software. You may use it “as it is” without any warranties. For further information see the disclaimer on the info tab. As I am not a skilled software engineer, the software may not fit your expectations in functionality and usability. Your feedback is appreciated, but you should remember that this is still a “fun project”.
The software does not claim to be a scientific tool. It is optimized for practical use and does not contain any attenuation calculation according to the bi-static radar equation.

The name of AirScout is derived from the following:

- The capital letters “AS” are short for the phenomenon of Aircraft Scatter
- “Air” is short for “Aircraft” or “Air traffic”
- “Scout” is a synonym for pathfinder, here: finding a path between two stations

The title photo is taken by me, DL2ALF, and is showing an old DC-3 (“Candy Bomber”) used to supply the West Berlin citizens during the Berlin Blockade in 1948/1949. This one is still operational at the Berlin-Schoenefeld Airport and can be booked for sightseeing flights.

The following document is a step by step introduction in the Aircraft Scatter theory and a description of the AirScout concepts and the use of the software itself.

2. Description of Aircraft Scatter Theory

2.1. Introduction

The Aircraft Scatter Theory is well known since the first RADAR systems came up at the end of World War II. Scattered signals are being used to detect an aircraft's position, altitude and speed. If transmitter and receiver are located at different places, the Bistatic Radar Equation (see Wikipedia) describes the situation very well. The scope of this equation is to calculate the path loss between TX and RX in case of a non-directional scatter from aircraft. As we know from several experiments is, that under special circumstances a significant enhancement of the received signal is obtained (20...30 dB). There is an ongoing discussion whether the enhancement is caused by forward scatter or by reflection. I don't want to join the discussion and want to make the following statements (valid for the whole document):

1. The following considerations are only valid for the period of the above mentioned signal enhancement
2. The received signal is considered as reflected by the aircraft
3. There is no calculation for path losses or possible signal enhancements included

The basic view of an aircraft scatter QSO is shown on Fig.1:

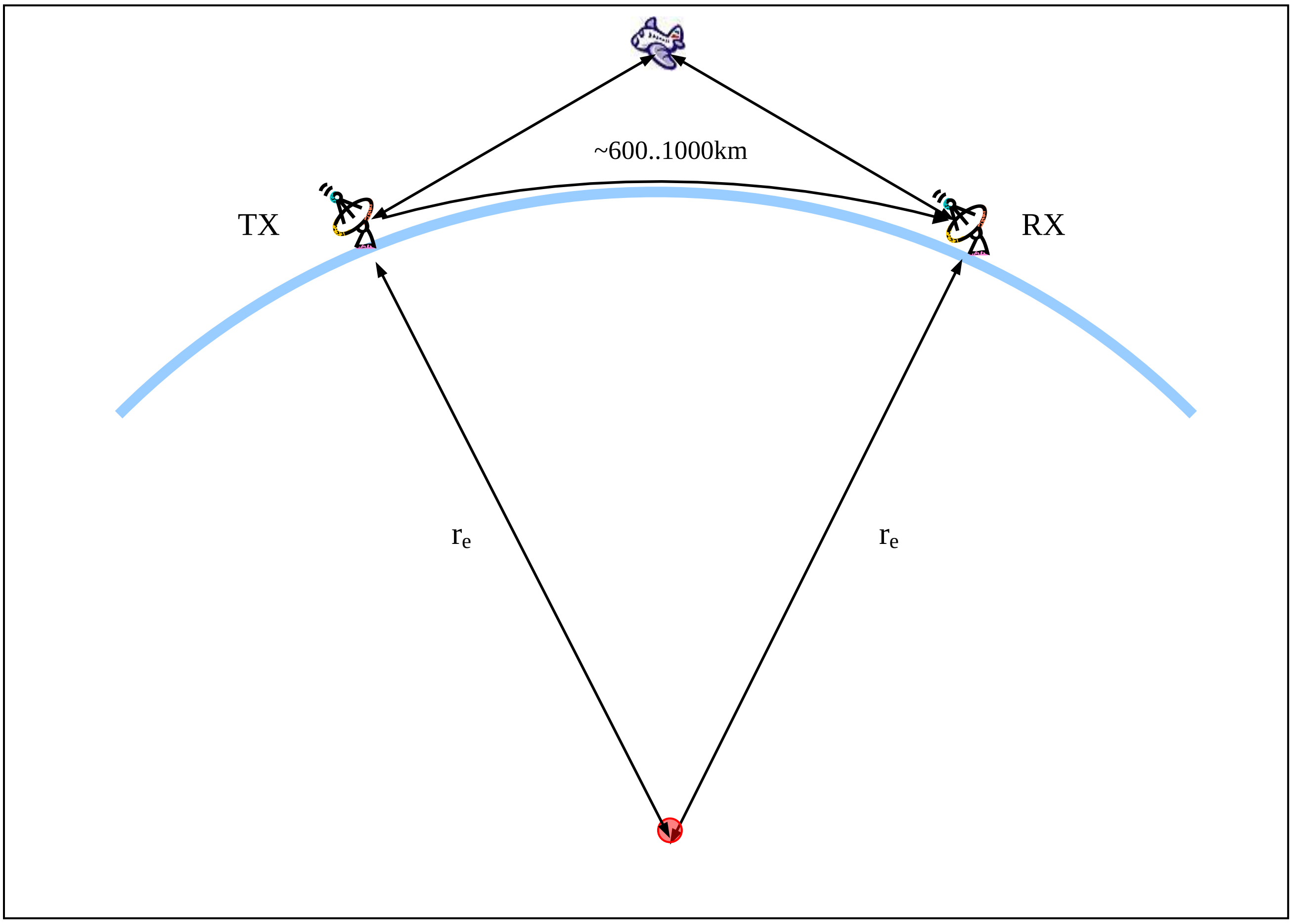


Figure 2-1 Schematic view of an aircraft scatter QSO

Based on these simplifications we can start now with some theory.
The most basic conditions for the use of Aircraft Scatter are very simple:

1. The aircraft must be in an unobstructed Line Of Sight (LOS) of both TX and RX.
2. The aircraft must be on or very near to the propagation path between both TX and RX

LOS does not mean optical LOS but “Radio”- LOS. Due to the fact that VHF waves are slightly bended back to earth, the “Radio”- LOS is larger than optical. To compensate this behaviour, the earth radius is scaled up by 4/3 in all related calculations. On an ideal spherical earth surface the LOS distance can be estimated by:

(1) $$d \approx 4.12\,(\sqrt{h_1}+\sqrt{h_2}) ,$$

Where: d is in [km],
h_1 and h_2 are in [m] above sea level

In reality, the current tropospheric situation is not constant and, especially at low VHF, frequency dependant. You can cover this by changing the K – Factor, but this will remain unpredictable forever.

Assuming that a RX/TX location is at sea level the maximum possible distance is shown in the following diagram:

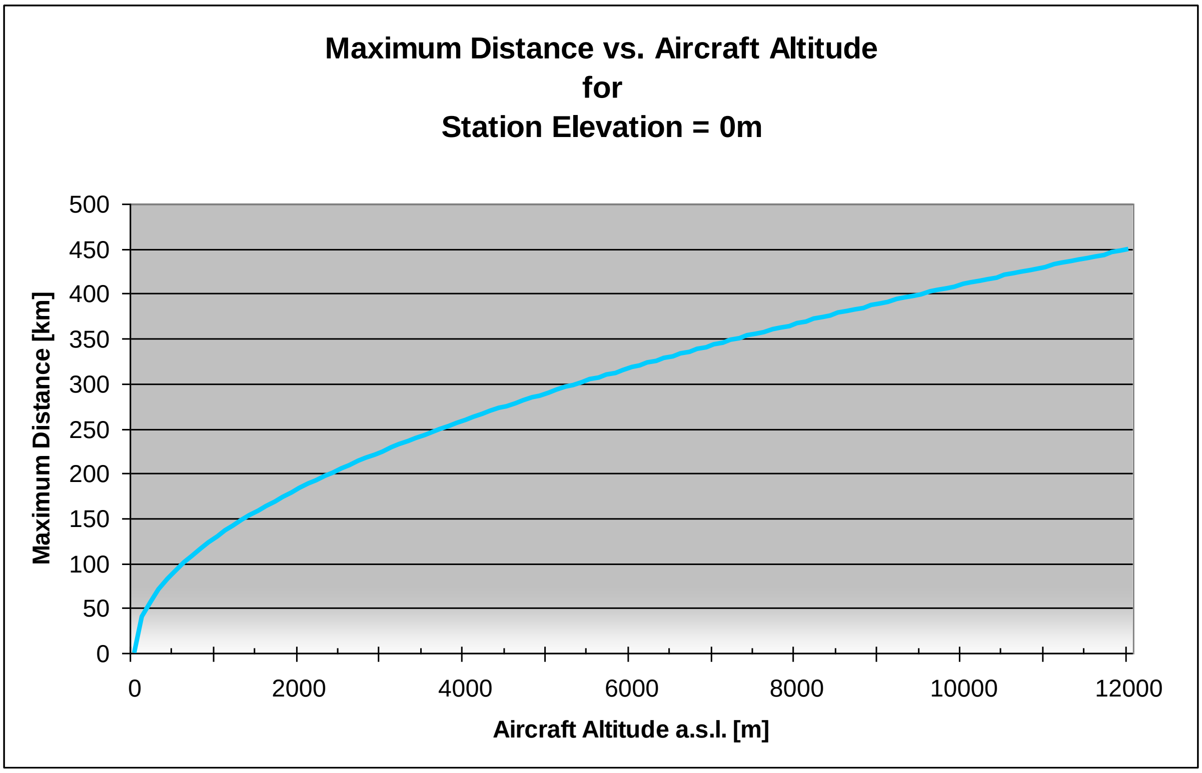


Figure 2-2 Maximum Distance vs. Aircraft Altitude

Assuming that the maximum cruising altitude of civil aircrafts is about 12000m we can calculate the maximum possible distances for typical TX/RX locations (elevated above sea level). Seen from either TX or RX this is shown in the following diagram:

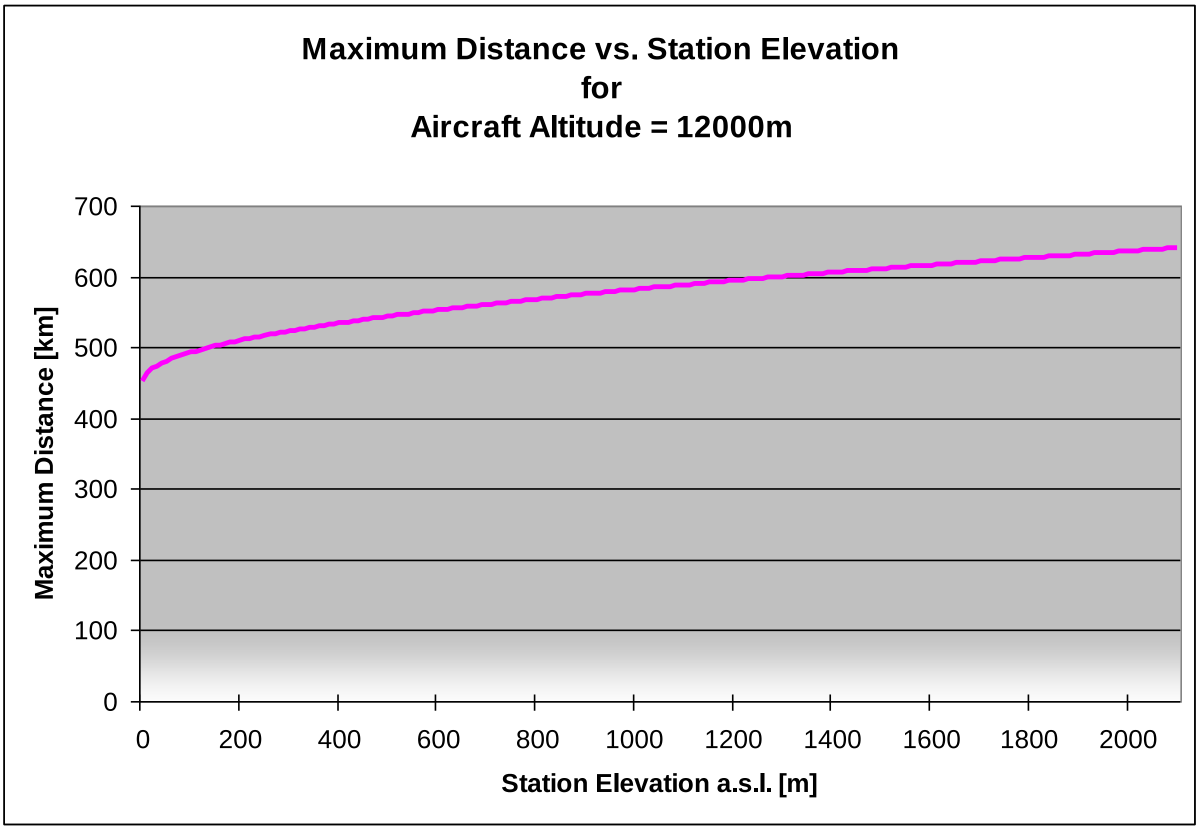


Figure 2-3 Maximum Distance vs. Station Elevation

To get the whole path length of an AS connection we have to sum up the distances for both TX and RX. We can expect a range of 900km (both stations at sea level) up to 1200km (both contest stations on mountains). In fact distances above 800km seem to be rather difficult and more investigation has to be taken on those experiences.

Remember that the equation above is only valid for an ideal earth surface. Limiting facts can minimize the distance dramatically, such as local or distant obstructions. To estimate this influence we need an elevation model (described later on).

The following calculations are not a overall description of airplane scatter theory. They only help to understand how the software works and what it can and what it cannot.

2.2. Basic Aircraft Scatter Geometry

To understand the basic key figures of an AS connection we will start with some geometry. The figure below is showing the situation at one end of a typical AS connection on a spherical earth (not true to scale).

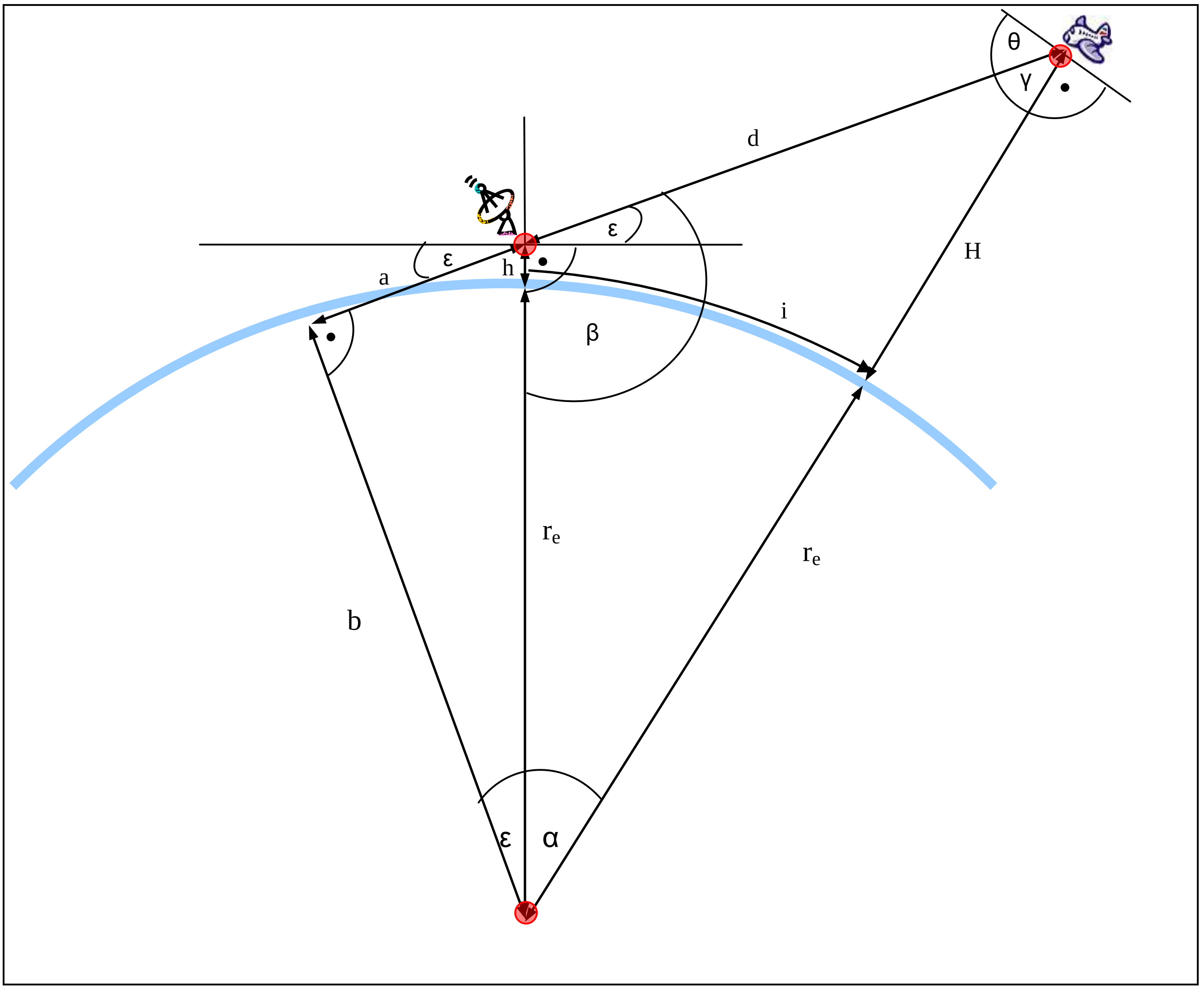


Figure 2-4 Basic aircraft scatter geometry

<u>*Legend:*</u>

*r_e = effective earth radius (4/3 * geometric Earth radius)*
h = transmitter elevation (above sea level)
H = aircraft altitude (above sea level)

i = great circle distance
d = slant range (which is very near but not equal to i for long distances)

ε = elevation angle (of aircraft seen from transmitter's horizontal plane)
θ = angle of incidence (angle between incoming wave front and aircrafts horizontal plane)

2.2.1. Calculation of Angles ε and θ

The two important angles to know are:

1. the angle of aircraft seen from the transmitter's horizontal plane, called ε
2. the angle of incidence of the transmitted wave front at the aircraft's horizontal plane, called θ

The figure below shows three triangles used to calculate these angles:

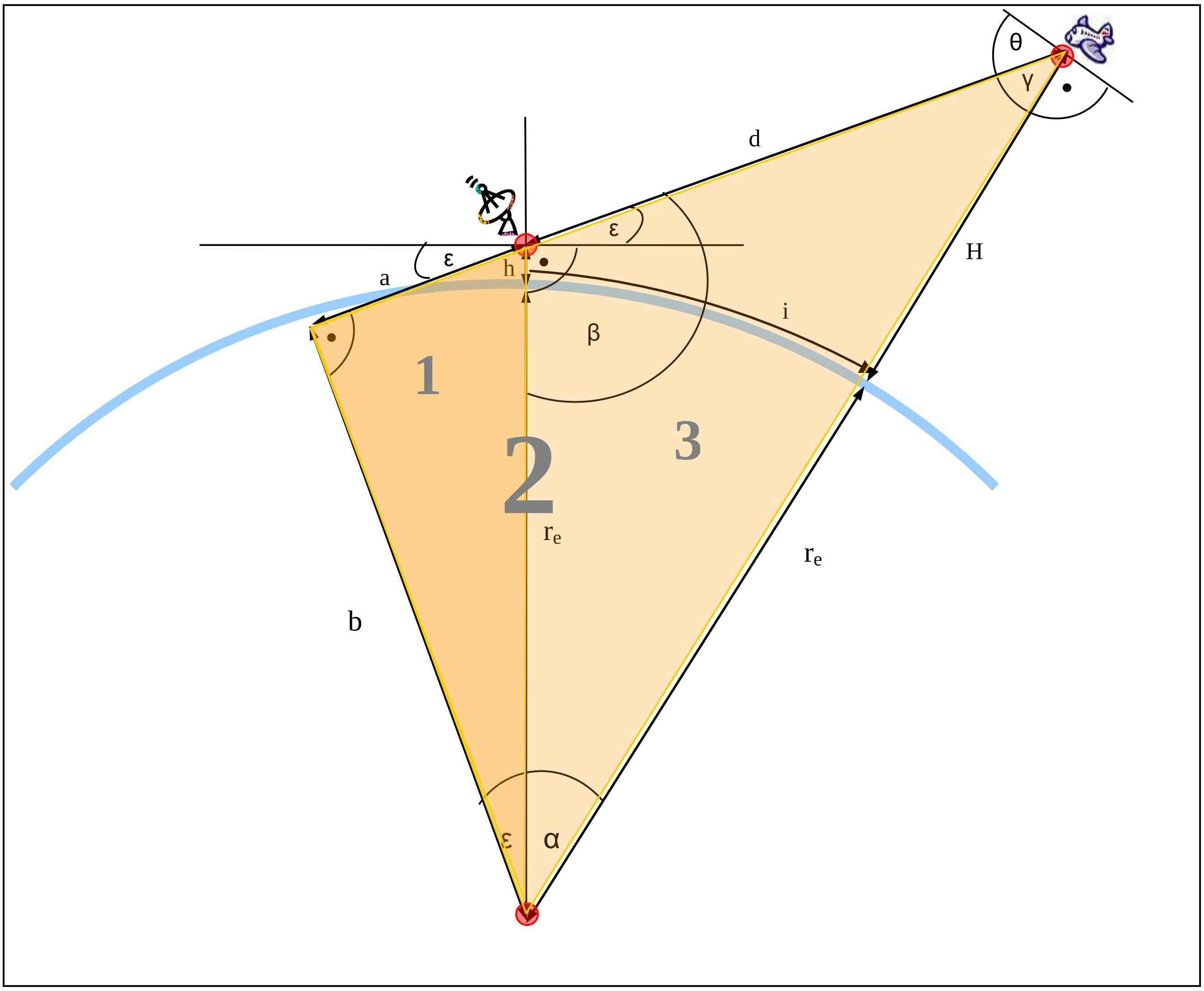


Figure 2-5 Calculation of Angles ε and θ

After understanding the geometry overview we can start now with calculations. Assuming that the position and altitude of TX and a distinct aircraft are known, the calculation of angles is possible. Idea and most of the following arithmetic is courtesy of Henning, DF9IC, adapted by Frank, DL2ALF. All formulas are optimized for software programming; some further simplifications might be possible.

Given: r_e, h, H, i

We start with basic triangulation functions of right-angled triangles (named 1 and 2):

(1) $a^2 + b^2 = (r_e + h)^2$; triangle 1

(2) $b^2 = (r_e + h)^2 - a^2$

(3) $(a + d)^2 + b^2 = (r_e + H)^2$; triangle 2

Under the assumption that the earth is a sphere, α simply represents the ratio between the great circle distance *i* and the complete circumference of earth:

(6) $$\alpha = \frac{i}{r_e}$$

Now we can calculate the slant range *d* by using the rule of cosine in triangle 3:

(4) $d^2 = (r_e + h)^2 + (r_e + H)^2 - 2(r_e + h)(r_e + H)\cos\alpha$; triangle 3

(5) $$d = \sqrt{(r_e + h)^2 + (r_e + H)^2 - 2(r_e + h)(r_e + H)\cos\alpha}$$

First, calculate *a* by replacing b in (3) with (2):

(7) $(a + d)^2 + (r_e + h)^2 - a^2 = (r_e + H)^2$

(8) $(a + d)^2 - a^2 = (r_e + H)^2 - (r_e + h)^2$

(9) $a^2 + 2ad + d^2 - a^2 = (r_e + H)^2 - (r_e + h)^2$

(10) $2ad + d^2 = (r_e + H)^2 - (r_e + h)^2$

(11) $2ad = (r_e + H)^2 - (r_e + h)^2 - d^2$

(12) $$a = \frac{(r_e + H)^2 - (r_e + h)^2 - d^2}{2d}$$

Next step is to calculate ε using trigonometric ratios in triangle 1:

(13) $$\sin\varepsilon = \frac{a}{(r_e + h)}$$

(14) $$\varepsilon = \arcsin\frac{a}{(r_e + h)}$$

And by replacing *a* by (8):

(15) $$\varepsilon = \frac{(r_e + H)^2 - (r_e + h)^2 - d^2}{2d(r_e + h)}$$

Next is to replace d in (15) with (5). As the formula is too complicated it is better to calculate d separately and use the result in (15).

Now we can calculate γ as an interim result using trigonometric ratio in triangle 2:

(16) $$\cos\gamma = \frac{a+d}{(r_e+H)}$$

(17) $$\gamma = \arccos\frac{a+d}{(r_e+H)}$$

Now it's easy to get θ finally:

(18) $$\theta = 90^\circ - \arccos\frac{a+d}{(r_e+H)}$$, or complementary:

(19) $$\theta = \arcsin\frac{a+d}{(r_e+H)}$$

2.2.2. Calculation of Minimum Usable Elevation Angle ε_{min}

Same calculation as shown above is done for any obstruction caused by terrain on the path between RX and TX to get the minimum usable elevation angle. On the figure below we see an obstructed view from TX with a minimum usable angle. The red framed aircraft would be in the LOS theoretically but is obstructed by mountains and therefore unusable for AS. The procedure is quite similar to ray tracing which is used to calculate shadows in computer generated graphics (see Wikipedia).
With an elevation model the task is now to calculate ε for each single step on the whole path. To get ε_{min} we simply search for the **Maximum** of all ε. This represents the highest obstruction on the path respecting the earth curvature and ground elevation of the obstruction.
As you can see on the figure below on the path from TX to obstruction a lower ε is possible but makes no sense, as there a no planes on such low altitudes.
If the TX elevation is high (h >> 0) and the view is mostly unobstructed, the minimum angle can have negative values as well ($\varepsilon_{min} < 0$).

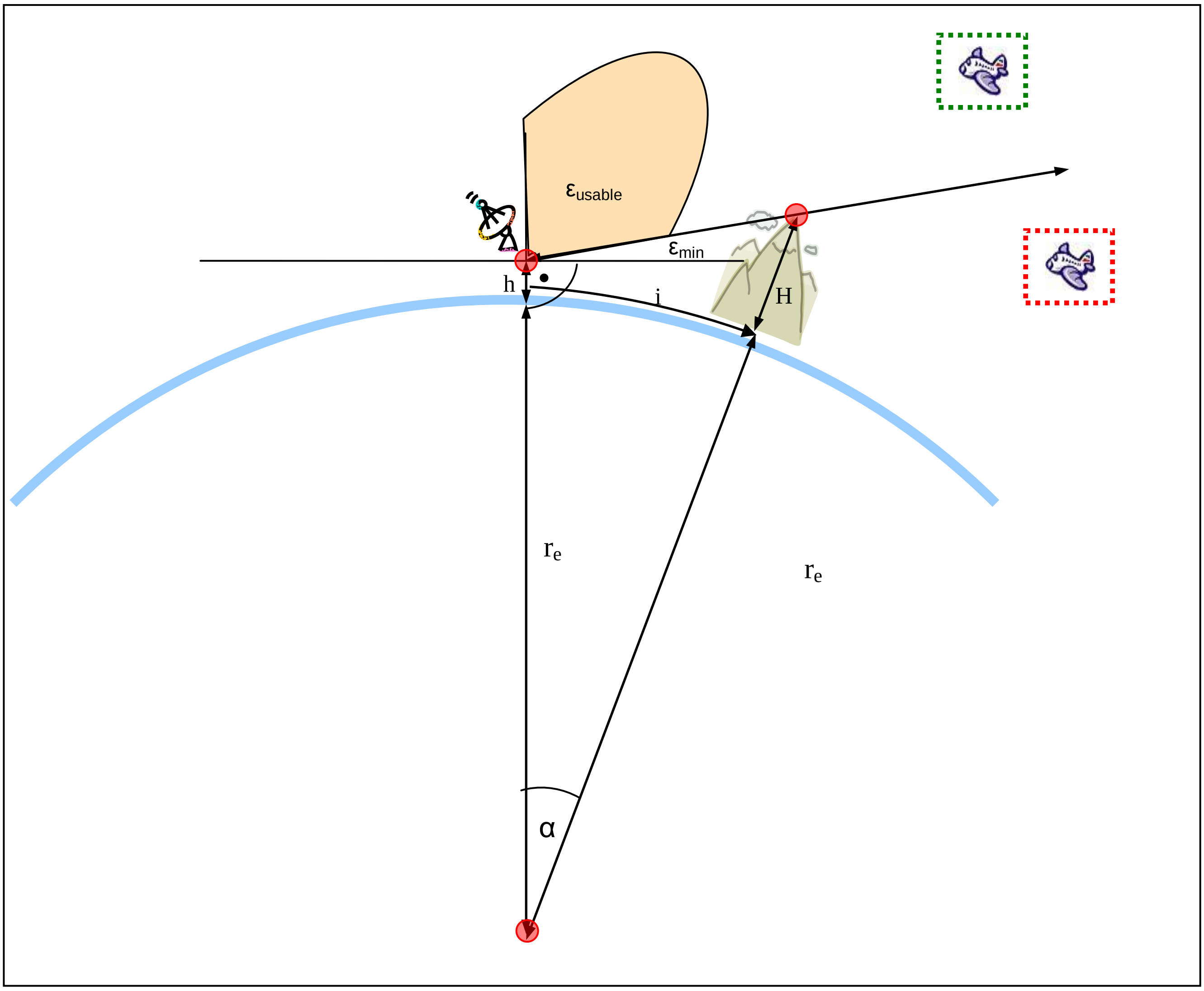


Figure 2-6 Calculation of Minimum Usable Elevation εmin

2.2.3. *Calculation of Minimum Usable Aircraft Altitude Hmin*

With a given ε_{min} we can now calculate the minimum altitude H_{min} an aircraft must have along each step on the whole path.

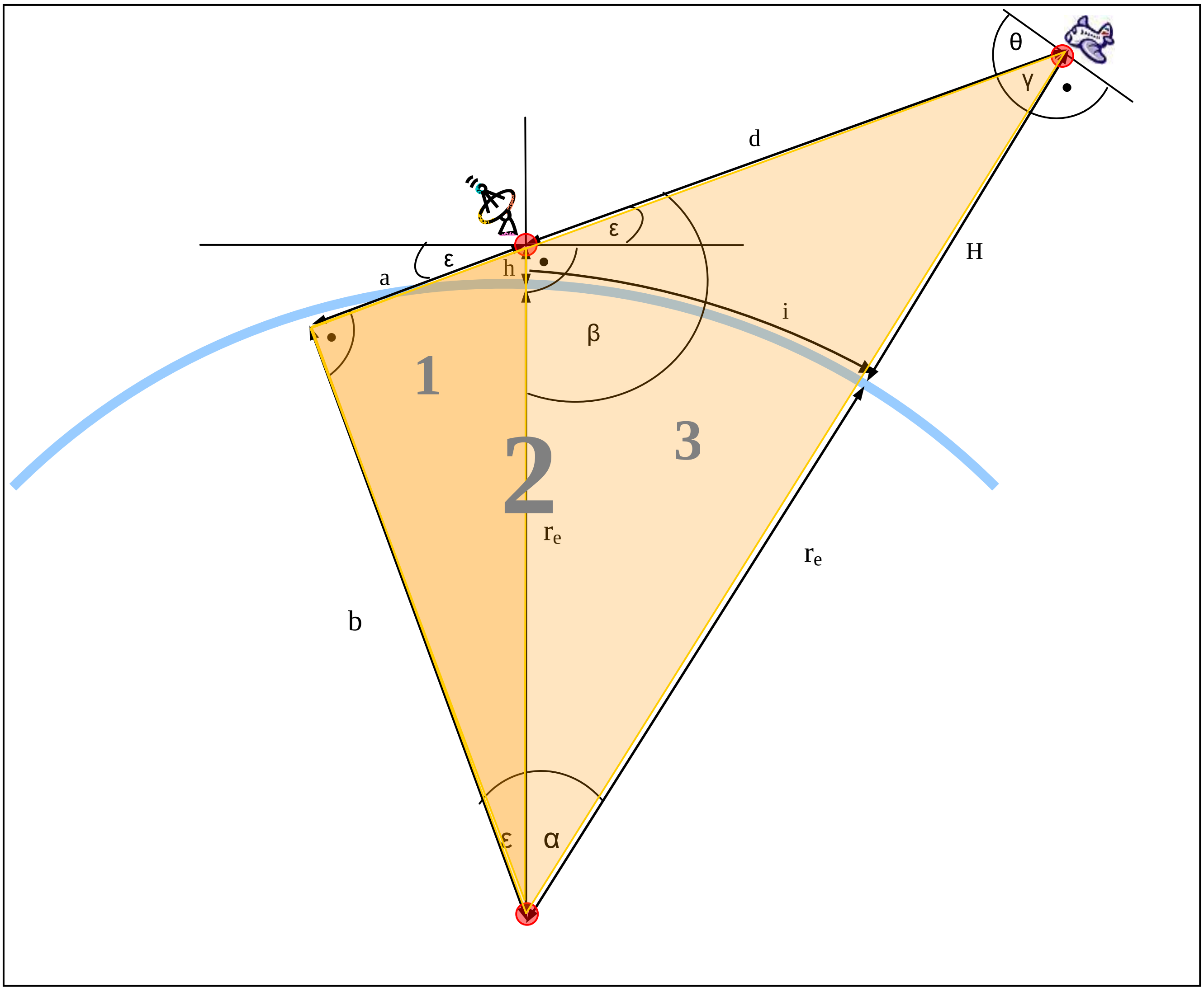


Figure 2-7 Calculation of Minimum Usable Aircraft Altitude Hmin

We start again with calculation on triangle 3.

Given: $r_{e,}$ h, i, ε

As shown in Chapter 2.2.1:

(1) $\alpha = \frac{i}{r_e}$

And with known ε we can calculate the missing two angles as:

(2) $\beta = 90° + \varepsilon$

(3) $\gamma = 180° - \alpha - \beta$

Following the rule of sine we get:

(4) $$\frac{(r_e + h)}{\gamma} = \frac{(r_e + H)}{\beta}$$, which is solved to H as

(5) $$H = \frac{(r_e + h)\beta}{\gamma} - r_e$$

Using the given ε_{min} we can iterate i now through the whole path and get a H_{min} for each step, means each great circle distance from the location of TX or RX. In other words, we are now able to calculate the minimum altitude an aircraft must have to be in an unobstructed LOS from TX as well as from RX.

2.2.4. Calculation of the zone of mutual visibility ("Hot Area")

Doing the above calculations for both ends of the path (TX and RX) and putting both curves on a diagram we can see the intersection easily. This intersection is capped on the upper side by the maximum cruising altitude a civil aircraft can have (abt. 12000m). The resulting triangle is an area in which an aircraft is "seen" from both ends of the path (mutual visibility) and is below the maximum cruising altitude.
This area is called the "Hot Area", in which an AS connection is (theoretically) possible. Projected on a flat earth surface it looks like the picture below.
Depending on the elevation of both TX and RX and the given obstructions, this area is usually asymmetric to the half way of the path. This will get more importance later on.
If the distance is too long or the LOS is strongly obstructed the "Hot Area" might be empty as well. In this case an AS connection is most unlikely.

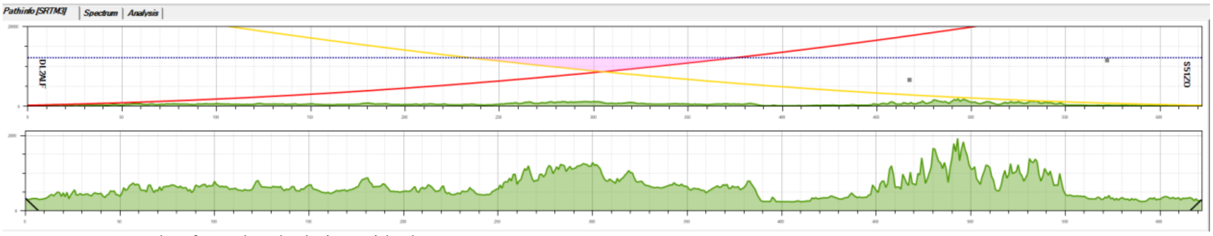


Figure 2-8 Example of a path calculation with „hot area"

2.3. Basic Parameters for Path Calculation

The basic parameters for path calculation are described below. Since most of the parameters are frequency dependant they can be set per band.

2.3.1. K-Factor

The K-Factor is used to extend the Earth radius due to the fact that radio waves are slightly bended back to Earth. Set to 1.33 by default, the effect of bending is decreasing with higher frequencies. You can set it to higher values on 144/432 MHz and to lower values on > 5.7 GHz. The default values for the K-Factor in V1.4 are:

QRG	K-Factor
50M	1.6
70M	1.6
144M	1.5
432M	1.4
1.2G	1.33
2.3G	1.33
3.4G	1.33
5.7G	1.33
10G	1.33
24G	1.33
47G	1.33
76G	1.33

2.3.2. F1-Clearance

The space around a travelling wave front is divided in so called Fresnel Zones (see Wikipedia). The most important is the F1 Fresnel Zone. Any obstruction inside the F1 Fresnel Zone will increase the attenuation along the path therefore a minimum clearance must be respected. The Fresnel Zone extent itself is clearly frequency dependant. It seems that the value of the necessary F1 Zone clearance also includes additional frequency dependencies. It is set to 0.6 by default, but you can adjust to lower values on 144/432 MHz. The default values for the F1-Clearance in V1.4 are:

QRG	F1-Clearance
50M	0.1
70M	0.1
144M	0.2
432M	0.4
1.2G	0.6
2.3G	0.6

QRG	F1-Clearance
3.4G	0.6
5.7G	0.6
10G	0.6
24G	0.6
47G	0.6
76G	0.6

2.3.3. Ground Clearance

The ground clearance is intended to be used for a correction of the altitude values given by the elevation model (primarily the GLOBE model). This may be necessary to respect the existing urban development or vegetation outside. Since the SRTM3 – data seem to deal already with this fact the parameter is more or less obsolete. It is set to 0 by default.

2.3.4. Maximum Distance (from Path)

The maximum distance is used to indicate a maximum lateral distance an airplane can have to consider it "on the path". Since this is not really a constant value along the path it should be replaced by a maximum lateral angle in future. It is set to 10km by default.

2.3.5. Maximum Squint Angle

Reserved for further use

2.3.6. Maximum Elevation Angle

Reserved for further use

2.4. Using Elevation Models

2.4.1. Introduction

The use of a Digital Elevation Model (DEM) is essential for:

- Getting elevation of TX and RX location
- Getting elevation of any obstruction along the path

Fortunately, there are several elevation models available for download on the internet. They contain more or less pre-processed elevation data for almost each part of the earth surface in several resolutions. Most of them are free of charge although restricted to personal use or copyrighted.
Starting with V1.3, AirScout is now consuming pre-processed elevation tiles in a 6digit Maidenhead structure. The tiles are categorized per DEM and are ready for download at the AirScout website. A huge amount of calculations was necessary to create all the data.

The DEMs used by AirScout are described below with advantages and disadvantages.

2.4.2. Global Land One-km Base Elevation (GLOBE)

The GLOBE model is provided by NGDC (part of NOAA) and is one of the most convenient and easy-to-use elevation models. Is has a resolution of 30 arc seconds (1km) and contains very well pre-processed and quality-controlled data. Most of the covered areas are free of copyright and can be redistributed as well. The download is free, users can download compressed data file “as a whole” or single tiles for any selected area. The file size is 100 - 120 MB (< 30MB zipped). Coverage is the whole earth. The only disadvantage is the low resolution which does not allow an accurate modelling in the near field of TX and RX.

2.4.3. Shuttle Radar Topography Mission 3 arc seconds (SRTM3)

The SRTM is probably the most popular and successful global survey mission. In 2000 a specialized Radar system aboard the Space Shuttle Endeavour was performing a near-global scale scan of the Earth surface. The processed elevation data are now used as the basis of almost all elevation models. The data are available in high resolution of 3 arc seconds (roughly 90m x 90m). In the current version 2_1 most of the artefacts are wiped out by averaging and comparing with other elevation models. The data files are organized in 1” x 1” tiles which are free for download as Public Domain. According to the characteristics of the Space Shuttle orbit the data are only available for latitudes of abt. -60° to +60°.
Another disadvantage is the large amount of files and data which cannot be held in memory. To get reasonable access, a caching algorithm is necessary.
The use of SRTM3 elevation data is optional in AirScout. Due to the huge amount of data they cannot be distributed along with the software. An automatic download of needed tiles is implemented.

2.4.4. Shuttle Radar Topography Mission 1 arc seconds (SRTM1)

SRTM1 data are provided basically the same way as SRTM3 data, and since 2015 they are available for the same areas as SRTM3. The data are available in very high resolution of 1 arc second (roughly 30m x 30m).
The use of SRTM1 elevation data is optional in AirScout. Due to the huge amount of data they cannot be distributed along with the software. An automatic download of needed tiles is implemented.

2.4.5. Advanced Space Borne Thermal Emission and Reflection (ASTER)

The ASTER DEM was initially released for public in 2016 by METI of Japan and NASA jointly and is available as V3 since 2019. It is one of the youngest elevation models and contains very high resolution data of 1 arc second (roughly 30m x 30m). Therefore it is treated as an alternative choice to SRTM1 data. At present time the data were completely downloaded by DL2ALF and are processed that they are available in the same format as the SRTM data. The implementation is considered as experimental.

2.4.6. ASTER 3 arc seconds (ASTER3)

The ASTER3 DEM was created by calculating a mean value in a 3 x 3 point raster of the original ASTER elevation data and by converting the results into the SRTM data format. The intention was to shrink the size of database and to eliminate artefacts. Furthermore a floating average by 5 values is used on path calculation.

2.4.7. ASTER 1 arc seconds (ASTER3)

The ASTER1 DEM was created from the original ASTER elevation data and converted into the SRTM data format. A floating average by 5 values is used on path calculation.

2.5. Using Real Time Aircraft Information

2.5.1. Introduction

After calculating of the path between two stations and of the “hot area” we need to set aircraft positions, tracks and altitudes in relation to this. There are several ways to get such information in real time.

2.5.2. Using an ADS-B Receiver

Receiving the ADS-B information send by the aircraft can be done with specialized receivers on 1090 MHz. There are some commercial products available as well as some low cost projects using a DVB-T stick. The problem of both opportunities is the coverage area: Depending on receiver sensitivity and antenna gain, only an area <200km around the receiver’s location can be covered. Taken into consideration that a usual AS distance is above 600km, the coverage area is far from enough. “Best practise” for the use of a dedicated receiver is to feed the aircraft positions to an Internet server and use the help of the community.

CAUTION: Running an ADS-B receiver and/or sharing the received information may not be legal in some countries!

2.5.3. Using a Plane Feed from Internet

The easiest way to get plane position information is the use of an internet feed. There are several community projects of flight enthusiasts who are operating dedicated servers. Most of them are delivering websites which are showing real time aircraft positions on a map.
Some of them are proving text feeds for special purpose in addition (often in JSON format). With the help of those text feeds it’s easy to supply AirScout software with the information needed.

IMPORTANT: Most of the text feeds are not intended for public use and can be loaded only via “deep link” (see Wikipedia). The use may not be permitted in some countries. Due to legal issues the AirScout user is explicitly asked to accept that he does this on his own risk. The feeds can be changed in URL and data format frequently and without notice. Furthermore it is not guaranteed to be a free service in future.

At present, AirScout can only handle different text feeds from internet, local files and raw data from ADS-B receivers.

2.5.4. Essential Data Fields of ADS-B Information

Once we have got plane information via ADS-B we can use the data fields for our own calculations. The following data fields are used:

- Latitude
- Longitude
- Altitude
- Ground speed
- Track
- Type identifier
- Time of transmission (UTC)

2.5.5. Storing aircraft positions and history

The latest received position of each aircraft during the runtime of AirScout is kept in memory. AirScout is fetching position information from internal database every second using current time when in “Play Mode” by estimating the position. For this the youngest timestamp found in memory is used. If the last position record is too old (older than 5mins by default) the estimation is getting too inaccurate and therefore null is returned.

Furthermore, AirScout can record a timeline with aircraft position in a database. In order to keep the amount of data small, the recorded positions are restricted to:

- An area around the own QTH defined by coverage area (see Options/General)
- A range of altitude suitable for Aircraft Scatter, starting at 5000m by default and reaching the maximum cruising altitude of a civil aircraft at 12200m

Recording will produce files extraordinary in size. A database maintenance procedure clears up data older than 7 days by default.

2.5.6. Aircraft Scatter Prediction Procedure

Next step is to collect aircraft information in an area around the propagation path.

1. Estimate each aircraft's position at this time using last reported position information (lat/lon, altitude, track, time of transmission)
2. Check if aircraft is on or near "hot path" and aircraft's altitude is above the calculated minimum altitude H at the shortest distance to path = <u>AS now</u>!
3. If not, calculate any intersection between "hot path" and aircraft's predicted route in future and check if current altitude is above the calculated minimum altitude H at the intersection point (using lat/lon track and altitude) = <u>AS in future</u>, calculate time remaining to crossing (using current position and current speed information)

3. Aircraft Scatter Operating Procedure

3.1. Recommendations for a QSO Procedure

A significant signal enhancement during a single aircraft crossing is lasting only about 30sec. Therefore a special operating procedure is necessary. It should be very near to the Airplane Scatter Procedure described in the VHF Manager's Handbook [1], ch. 7.6, p. 106. A fixed time schedule (even/odd minutes) similar to a MS procedure may help with random contacts; in a scheduled QSO with AirScout this kind of operating is obsolete.
The recommended QSO procedure is described below:

Calling

The QSO starts with one station calling CQ or calling other station by sending both call signs in case of a scheduled QSO. A calling sequence should be kept as short as possible. Do not use alternate calling, only one station is calling, the other is listening until receiving the first calls.

Reporting system

The report system is the standard RS(T) system (e.g. 59, 599).

Reporting procedure

A report is sent when the operator has positive evidence of having received the correspondent's or his own call sign or parts of them. The report must not be changed during a contact even though signal strength or duration might well justify it.

[Standard]
The report should be sent at least twice between each set of call signs.

[Contest]
Extend the report with additional information requested by the contest rules (e.g. contest exchange or locator). It is recommended to send report and exchange at least twice and locator once followed by a break.

Keep sending information with breaks until you receive an "R" from your QSO partner. Do not send call signs anymore when first copied correctly.

Confirmation procedure

As soon as either operator copies both call signs and a report he may start sending a confirmation. This means that all letters and figures have been correctly received. The message can be pieced together from fragments, but it is up to the operator to ensure that it is done correctly and unambiguously.

[Standard]
Confirmation is given by inserting an R before the report.

[Contest]
Confirmation is given by inserting "RRR" before starting to send report and additional information.

When one operator receives a confirmation message, and all required information is complete he must confirm with a string of R's. When the other operator has received the R's, the contact is complete and he may respond in the same manner. It is recommended to send “73” or “gl” to let the other operator know that you’re done.

Requirement for a complete QSO

Both operators must have copied both call signs, the report and a confirmation that the other operator has done the same. This confirmation can either be an "R" preceding the report or a string of minimum three consecutive "RRR".

3.2. Arranging a Sked

Aircraft scatter contacts are scheduled in almost all cases. The skeds are arranged via ON4KST chat usually. Good practise is to ask someone for sked with clear arrangements on:

- The frequency for sked
- Who is calling first
- Time left for aircraft to cross the path

Remember: it is OK to arrange a sked but it is not allowed to publish any information concerning the QSO unless you are complete.
Here are some examples for ON4KST announcements:

OK:

“Pse AS test on 23cm. We CQ at 1296.240 ur dir. Next AP in 5mins.”
“Tnx fb AS qso es new odx. 73 es gl!”

OK, but:

“NC! Sigs faded out. Next chance in 10mins. Wl meep u short before.” –
but: You have to restart the procedure!

Not OK:

“Hrd u 559. pse continue clg.”
“Mni tnx for 010. 73 es gl!”
„Pse number agn!“

3.3. Hints for Successful Operation

While setting new DX records and contest high scores on the bands please do not forget the ham spirit. Some hints for clean and successful operation:

- Remember that all information for a valid QSO must be sent and confirmed via HF

- Any conversation by any means other than HF communication unless you are complete invalidates the QSO and you must start over again with full procedure
- Do not send redundant information that is known to be received by your QSO partner already (e.g. do not send call signs when already called with your call sign by your QSO partner)
- While repeating a message send only requested information by your QSO partner (e.g. do not send grid square twice when requested to send the report)
- You don’t need a QSO partner for your first tries. If you are living in Europe you can listen to beacons as well, especially those with high ERP. I made my best experiences with GB3MHZ on 23cm and with OZ7IGY on 2m and 70cm. An SDR with waterfall display helps to find the traces even when they are not audible.
- You don’t need to turn your antenna during a QSO. On usual DX interesting planes are well inside the beam width of your antenna. Just beam directly to your QSO partner except you want to experiment with side scatter.
- You don’t need to compensate a Doppler shift except on 10 GHz or above, if the aircraft is on or near the path, the Doppler shifts of the incoming and reflected wave front are compensation each other
- You don’t need big equipment; any average configuration will do the job. An EME rig will help for sure, but a lot of contacts were made with 20W into a single yagi.
- Try to improve your high speed CW skills. Contacts on CW are working on lower SNR than on SSB but need more time for information interchange. Use faster CW helps to complete the QSO within one aircraft pass.
- Don’t give up when an aircraft pass does not cause a significant signal enhancement. Sometimes reflections are unpredictable. Remember the experimental part of our hobby and try again with next aircraft.

4. Software Manual

4.1. Basic Concepts

4.1.1. User Interface

The user interface should be as simple as possible, allowing the user to work with the main window only. In most cases the input of a desired DX call and pressing the “Play/Pause” button would be sufficient.

Starting with V1.3 AirScout is providing two basic operation modes:

- Single-Path: works with one single DX call like in previous versions
- Multi-Path: observe path and aircrafts to multiple stations selected from a watch list

4.1.2. Single-Path Operation

The Single-Path operation is designed to analyze a path to a single DX station in detail. AirScout is not only showing path and aircrafts on map but also charts with both propagation paths and elevation path in detail.

4.1.3. Multi-Path Operation

The Multi-Path operation is designed for contest-like operation. Not necessary to see path details but observe multiple DX stations simultaneously and co-ordinate skeds according to the current "global" aircraft situation. Calculating a path is a time consuming process and cannot be performed at runtime for more than a couple of stations. Therefore a extensive pre-calculation process is installed and will be described later on in this document.

4.1.4. Options

All settings can be modified in a separate “Options” window which is available only when the software is in “Pause” mode. Before opening the window, all background threads are stopped to avoid side effects when changing settings. Separate tabs contain settings for each topic. On closing the window the background threads are restarted.

4.1.5. Covered Area

The covered area is the basic area in which all observations should take place. It is used to:

- Limit the aircraft position recording on database
- Limit the automatic download of elevation data files

The latitude / longitude values can be entered via “Options/General”. Please choose an area around your current location; a suitable value would be around 20 degrees in each direction.

4.1.6. Elevation Data

The use of elevation data is mandatory for the work with AirScout. They are necessary for:

- Getting the elevation above sea level of both QSO partners
- Calculating the path between and find obstructions

Both topics are connected closely to each other and must taken from the same elevation model (regardless which one). It makes no sense to enter the elevation information manually or to "correct" incorrect elevation with increasing the antenna height. Otherwise the calculation of the path and its obstructions will fail. If more than one elevation data source is enabled the search order (e.g. the fallback) for a distinct location is

1. SRTM1 data (if enabled)
2. SRTM3 data (if enabled)
3. GLOBE data
4. set elevation value to 0 and mark the information as invalid

4.1.7. Working with precise locations

The precise location of both QSO partners is extremely helpful for a proper path calculation. As an example you can see the difference between midpoint and precise location at OK1TEH here:

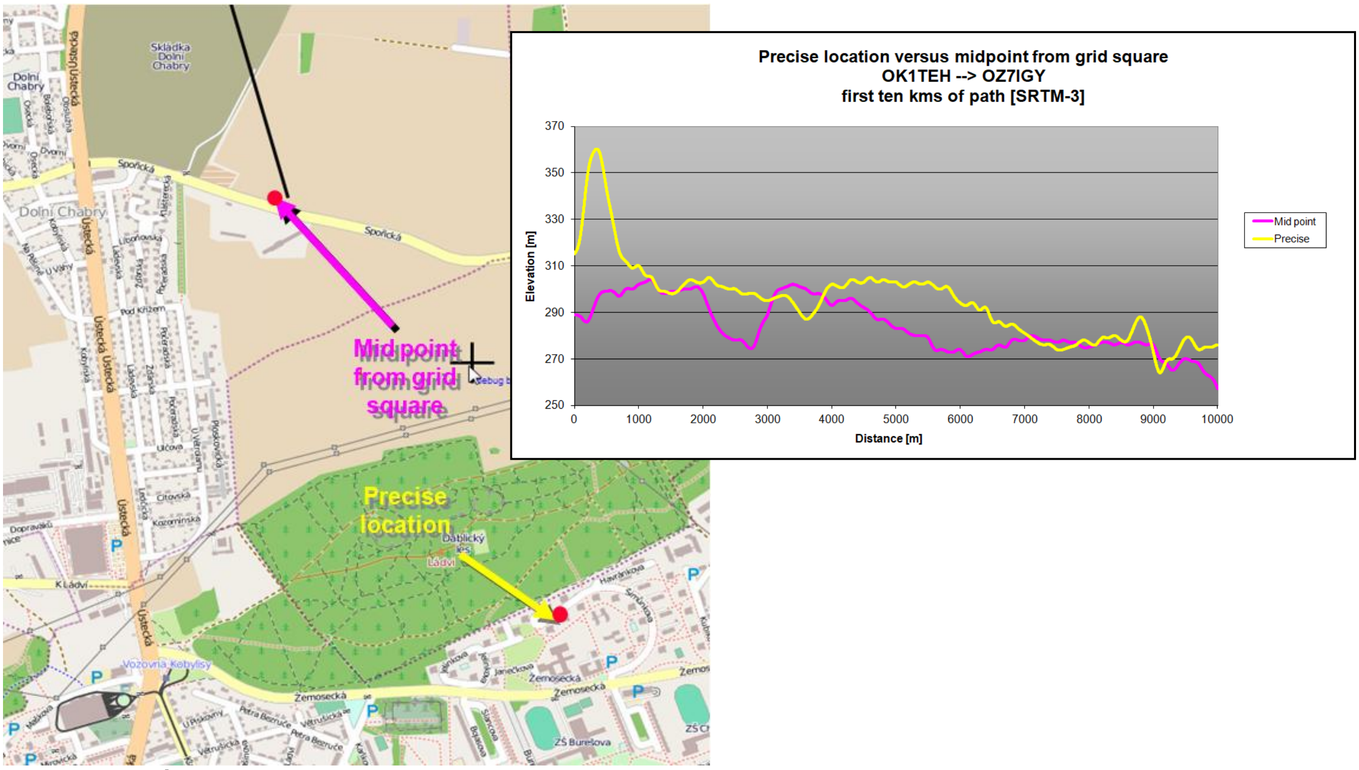


Figure 4-1 Precise location

As you can see, a massive obstruction in direction NW would not be taken into account when using midpoint from grid square (see yellow curve in chart). A path calculation would give nonsense results with such precautions.

Therefore the use of precise locations is strongly recommended whenever available.

- Users are asked to input their precise location at program's first run
- Users can enter/edit a precise location for themselves or for others in via Options/Stations (by entering numeric, dragging needle on map, QRZ.com request)

On the other side AirScout should work with a minimum of 6digit Maidenhead locator. To solve this, the software is using the following strategy depending on the option:

"Use best case elevation from grid square if precise location is unknown"

Option set (Default): assuming that a VHF amateur is using the highest possible location inside a given grid square

Option not set: use the midpoint of a given grid square

Updated locations can be sent to a web database and after review they will be available for the public via update mechanism.

4.1.8. Monitoring Spectrum

For educational purpose AirScout is support a Level Meter to visualize the signal enhancement during an aircraft pass. This is working only in co-operation with Spectrum Lab Software by DL4YHF. Spectrum Lab is capturing data from a NF source like a soundcard connected to a VHF receiver. This is basically the same technology like WSJT or other digimodes are running in a ham radio shack. AirScout is getting FFT data from Spectrum Lab server and is searching the maximum level in the selected spectrum. This maximum is plotted in a diagram with a 10 minutes history showing the signal enhancements during that time.

To get this working, do the following:

1. Tune your receiver and antenna to a suitable beacon (e.g. GB3MHL, OZ7IGY). Mode SSB with -800Hz offset is recommended.
2. Connect your RX – NF to your computer soundcard input or use a station controller like microHAM.
3. Start Spectrum Lab software
4. Select the sound card for capturing in Spectrum Lab and check the NF level
5. Ensure that you see the captured spectrum at Spectrum Lab’s waterfall display
6. Switch to the “Spectrum” tab in AirScout’s main window to see the level display.

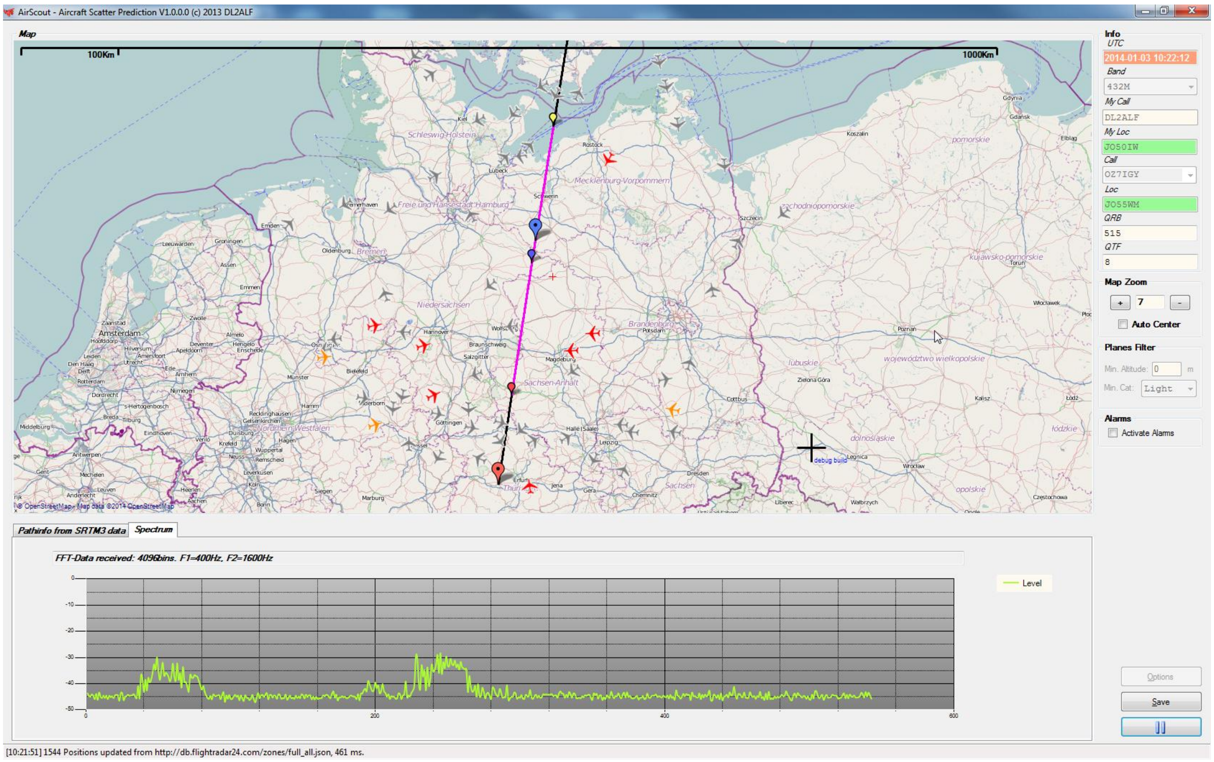


Figure 4-2 Program's main window with Spectrum view

Some additional hints for successful operation:

- Watch the status messages in the "Spectrum" tab. Error messages will show up if something is wrong with the data or the connection
- Adjust the NF to get about -40dB noise level
- Try to adjust the RX frequency so that you see the beacon carrier around 800Hz
- Narrow the captured spectrum as much as you can using the F1 and F2 settings in the AirScout Options tab to avoid false maximum triggered by birdies. Assuming the carrier is at 800Hz, set the frequencies as follows:
  F1 keyed beacon: F1 = 400Hz F2 = 1000Hz
  PI-4 keyed beacon: F1 = 400Hz F2 = 1600Hz
  WSJT keyed beacon: F1 = 400Hz F2 = 3000Hz
- Check the frequency edges using the waterfall display of Spectrum Lab. Ensure that you are capturing the whole beacon transmissions in any mode. Be sure that there are no birdies inside the captured spectrum that will cause false maxima.
- Spectrum Lab power users can play with filter settings to filter out the carriers to get more sophisticated results

4.1.9. Dealing with big data

AirScout is dealing with real big data which ends up in large files and high CPU load. The aim of the software architecture is to present a responsive user interface on one hand and to have all information needed available quickly.

The majority of information clearly comes from the elevation model and (if activated) from aircraft position history. To give you an idea of how large files can be, this are file sizes of a typical AirScout configuration with scope Middle Europe and all elevation models activated:

GLOBE elevation database:	*20 MB*
ASTER3/SRTM3 elevation database:	*2 GB*
ASTER1/SRTM1 elevation database:	*20 GB*
Aircraft position history:	*several GB / day very much depending on traffic*

and compared to this:

Station database:	*15 MB*

Be sure to have sufficient space on your local disk!

As this huge amount of data cannot be distributed along with the software, all information must be gathered on first run from AirScout web database.

CAUTION: Depending on your Internet connection speed and PC capabilities the first update process will take several hours to complete! It does not make much sense to work with AirScout in that state.

After updating all necessary data from web resources the software does as much pre-calculations as possible in the background. This will again last some hours but you can work with AirScout already.

A modern database concept helps to fulfil all of these requirements, the information once created is stored in the database for further use.

4.1.10. Database Overview

As mentioned above almost all information needed for AirScout's operation is stored in databases. The main advantages of a modern database are:

- structured data storage and query
- fast and flexible data access
- thread safety for simultaneous operation
- proven libraries for all operating systems available

AirScout is using multiple SQLite databases for different kind of data, the GreatMaps.NET library is also using this format to store map tiles.

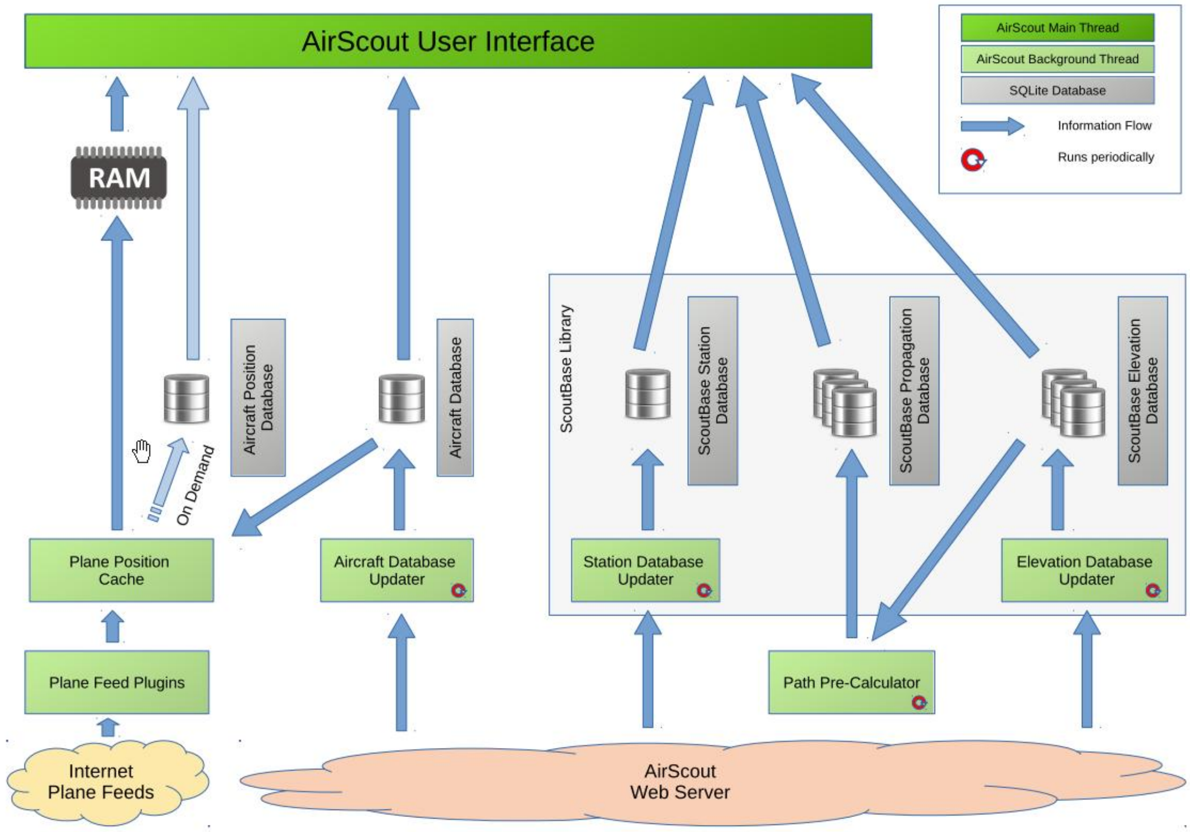


Figure 4-3 AirScout's internal database structure

4.1.11. Database update strategy

AirScout is providing all information for worldwide use as a web service. Depending on your scope of interest and elevation model the software is determining which information is needed, tries to download all files in temporary folders and updates the internal database. Once downloaded the database is kept up-to-date over the time by checking time stamps or version numbers against latest changes. The period for checking can be chosen via Options/Database, default is on program startup.

4.1.12. ScoutBase Station Database

Database file: <AppData>\Local\DL2ALF\Scoutbase\StationData\stations.db3

Contains all basic station information:

- Station location
- Station QRV info

The ScoutBase station database is intended to use with other software as well.

Table Location

Contains station location information unique per call sign & locator, so a station with one call sign can deal with different locations.

Column Name	Column Type	Description
Call	TEXT	Station's call sign
Loc	TEXT	Station's 6digit Maidenhead locator (upper case)
Lat	REAL	Latitude of location
Lon	REAL	Longitude of location
Source	INT32	Source of information (see * below)
Hits	INT32	Reserved for further use
LastUpdated	INT32	Time stamp of last update (UNIX)

*) Source of Information is one of the following GEOSOURCE constants:

- UNKNOWN 0 source is unknown
- FROMLOC 1 location is calculated as midpoint of locator
- FROMUSER 2 location is entered by user (usually precise)
- FROMBEST 3 location is calculated as point of highest elevation inside the given locator (best case elevation)

Table QRV

Contains station QRV information unique per call sign, locator & band, so a station with one call sign can deal with different locations and bands.

Column Name	Column Type	Description
Call	TEXT	Station's call sign
Loc	TEXT	Station's 6digit Maidenhead locator (upper case)
Band	INT32	Band [MHz]
AntennaHeight	REAL	Antenna height [m]
AntennaGain	REAL	Anntena gain [dBd], reserved for further use
Power	REAL	Output power [W], reserved for further use
LastUpdated	INT32	Time stamp of last update (UNIX)

4.1.13. ScoutBase Elevation Database

Database file: <AppData>\Local\DL2ALF\Scoutbase\ElevationData\<model>.db3

Contains all elevation related information:

- Elevation tiles as a worldwide map organized in 6digit Maidenhead locators
- Elevation path between two locations
- Elevation horizon in 1° steps around a location

The database is split in single database files with similar structure, one file per elevation model:

- GLOBE: globe.db3
- STRM3: strm3.db3
- STRM1: strm1.db3

The ScoutBase elevation database is intended to use with other software as well.

Table Elevation

Contains elevation tile information unique per tile index, which is in fact the 6digit -Maidenhead Locator. Tiles are pre-calculated and contain min/max info beside the elevation information.

Column Name	Column Type	Description
TileIndex	TEXT	Index of elevation tile
MinLat	DOUBLE	Latitude of lower left corner of tile
MinLon	DOUBLE	Longitude of lower left corner of tile
MinElv	DOUBLE	Lowest elevation inside the tile
MaxLat	DOUBLE	Latitude of upper right corner of tile
MaxLon	DOUBLE	Longitude of upper right corner of tile
MaxElv	DOUBLE	Maximum elevation inside the tile
Rows	INT32	Number of rows in elevation matrix
Columns	INT32	Number of columns in elevation matrix
Elv	BLOB	Elevation matrix, contains elevation per point [m]
LastUpdated	INT32	Time stamp of last update (UNIX)

Table ElevationPath

Contains elevation path information between two locations and is unique per:

- Lat1
- Lon1
- Lat2
- Lon2
- Calculation step width

The elevation path, of course, does not depend on a call sign, but on two distinct geographical locations, calculation step width and the elevation model.

Column Name	Column Type	Description
Lat1	DOUBLE	Latitude of first location
Lon1	DOUBLE	Longitude of first location
Lat2	DOUBLE	Latitude of second location
Lon2	DOUBLE	Longitude of second location
StepWidth	DOUBLE	Calculation step width
Count	INT32	Number of waypoints in path
Path	BLOB	Elevation path
LastUpdated	INT32	Time stamp of last update (UNIX)

Table ElevationHorizon

Contains information about the elevation horizon around a location and is unique per:

- Lat
- Lon
- Distance
- Calculation step width

The elevation horizion, of course, does not depend on a call sign, but on a distinct geographical location, calculation step width and the elevation model. As the following propagation path calculation depends on the path distance, this is also part of the table index. The elevation horizon is always calculated in 1° steps.

Column Name	Column Type	Description
Lat	DOUBLE	Latitude of location
Lon	DOUBLE	Longitude of location
Distance	DOUBLE	Latitude of second location
Lon2	DOUBLE	Longitude of second location
StepWidth	DOUBLE	Calculation step width
Count	INT32	Number of elevation paths
Paths	BLOB	Array of elevation paths
LastUpdated	INT32	Time stamp of last update (UNIX)

Table LocalObstructions

Sometimes the elevation model does not represent local obstructions like higher building which are blocking a distinct direction around a location. Therefore the definition of such local obstructions is possible in the elevation database. It is unique per latitude & longitude and is always calculated in 1° steps around the location.

Column Name	Column Type	Description
Lat	DOUBLE	Latitude of location
Lon	DOUBLE	Longitude of location
Distance	BLOB	Array of distances to obstructions [m]
Height	BLOB	Array of heights of obstructions [m]
LastUpdated	INT32	Time stamp of last update (UNIX)

4.1.14. ScoutBase Propagation Database

Database file: <AppData>\Local\DL2ALF\Scoutbase\PropagationData\<model>.db3

Contains all propagation related information:

- Propagation path between two locations
- Propagation horizon in 1° steps around a location

As the basis for a propagation calculation is the elevation model, the database is split in single database files with similar structure, one file per elevation model:

- GLOBE: globe.db3
- STRM3: strm3.db3
- STRM1: strm1.db3

The ScoutBase propagation database is intended to use with other software as well.

Table PropagationPath

Contains propagation path information between two locations and depends on a number of variables which can be set in Options/Path. It is unique per:

- Lat1
- Lon1
- h1
- Lat2
- Lon2
- h2
- QRG
- Effective earth radius
- Fresnel zone F1 clearance
- Calculation step width

The propagation path, of course, does not depend on a call sign, but on two distinct geographical locations, calc step width and the elevation model. It is not necessary to store the whole path in database but only the resulting minimum elevation levels for both ends of the path!

Column Name	Column Type	Description
Lat1	DOUBLE	Latitude of first location
Lon1	DOUBLE	Longitude of first location
h1	DOUBLE	Height of first location (incl. antenna height) [m]
Lat2	DOUBLE	Latitude of second location
Lon2	DOUBLE	Longitude of second location
h2	DOUBLE	Height of second location (incl. antenna height) [m]
QRG	DOUBLE	Frequency [GHz]
Radius	DOUBLE	Effective earth radius [km]
F1_Clearance	DOUBLE	Fresnel zone F1 clearance
StepWidth	DOUBLE	Calculation step width
Eps1_min	DOUBLE	Minimum elevation for first location
Eps_min	DOUBLE	Minimum elevation for second location
LastUpdated	INT32	Time stamp of last update (UNIX)

Table PropagationHorizon

Contains information about the propagation horizon around a location and depends on a number of variables which can be set in Options/Path. It is unique per:

- Lat
- Lon
- h
- Distance
- QRG
- Effective earth radius
- Fresnel zone F1 clearance
- Calculation step width

The propagation path, of course, does not depend on a call sign, but on a distinct geographical locations, calculation step width and the elevation model. As the Fresnel zone F1 depends on distance between two locations and a second location is not available, a synthetic distance must be assumed for calculations.

Column Name	Column Type	Description
Lat	DOUBLE	Latitude of location
Lon	DOUBLE	Longitude of location
H	DOUBLE	Height of location (incl. antenna height) [m]
QRG	DOUBLE	Frequency [GHz]
Radius	DOUBLE	Effective earth radius [km]
F1_Clearance	DOUBLE	Fresnel zone F1 clearance
StepWidth	DOUBLE	Calculation step width
Horizon	BLOB	Horizon array
LastUpdated	INT32	Time stamp of last update (UNIX)

4.1.15. AirScout Aircraft Database

Database file: <AppData>\Local\DL2ALF\AirScout\AircraftData\aircrafts.db3

Contains all aircraft and air traffic related information

- Aircraft info (Call, Registration, Hex, Type)
- Aircraft type info
- Aircraft registration info
- Airline info
- Airport info

Each time above mentioned information is altered, the database is updated. The information stored can be used to complete missing parts of received plane feeds.

Table Aircrafts

Contains basic information about existing aircafts and is unique per HEX, which is the ICAO-24bit unique aircraft identifier. The identifier is considered to be forever linked to a distinct aircraft. In fact is not and can be changed by maintenance or other reasons; this could affect historical considerations but is ignored in AirScout. Other possible identifiers like call sign or registrations will be changed more frequently.

Column Name	Column Type	Description
Hex	*TEXT*	HEX identifier of aircraft
Call	TEXT	Call sign of aircraft
Reg	TEXT	Registration of aircraft
TypeCode	TEXT	Type code of aircraft
LastUpdated	INT32	Time stamp of last update (UNIX)

Table AircraftTypes

Contains information about existing aircaft types and their classification. Unfortunately there are two naming systems IATA and ICAO which both must be considered. Therefore the index is unique per IATA & ICAO. The most important information of this table is the category which primarily characterizes the wake turbulence behind an aircraft. Indirectly connected with aircraft size it gives us the possible aircraft scatter potential.

Column Name	Column Type	Description
ICAO	*TEXT*	ICAO type code
IATA	*TEXT*	IATA type code
Manufacturer	TEXT	Manufacturer of aircraft type
Model	TEXT	Model designator of aircraft type
Category	INT32	Category of aircraft type (* see below)
LastUpdated	INT32	Time stamp of last update (UNIX)

*) Category is one of the following PLANECATEGORY constants:

- NONE 0 not categorized
- LIGHT 1 light
- MEDIUM 2 medium
- HEAVY 3 heavy
- SUPERHEAVY 4 super heavy

Table AircraftRegistrations

Contains additional information about ICAO assigned aircraft registration prefixes and is unique per prefix. The prefix stored here is not connected to an individual aircraft. It helps to identify data fields in not yet known JSON plane feeds from internet.

Column Name	Column Type	Description
Prefix	*TEXT*	Prefix of registration
Country	TEXT	Country of registration
Remarks	TEXT	Remarks
LastUpdated	INT32	Time stamp of last update (UNIX)

Table Airlines

Contains information about existing airlines. Unfortunately there are two naming systems IATA and ICAO which both must be considered. Therefore the index is unique per IATA & ICAO. The airline information stored here is not used to identify an individual aircraft. It helps to identify data fields in not yet known JSON plane feeds from internet

Column Name	Column Type	Description
ICAO	*TEXT*	ICAO code of airline
IATA	*TEXT*	IATA code of airline
Airline	TEXT	Name of airline
Country	TEXT	Country of airline
LastUpdated	INT32	Time stamp of last update (UNIX)

Table Airports

Contains information about existing airports. Unfortunately there are two naming systems IATA and ICAO which both must be considered. Therefore the index is unique per IATA & ICAO. The airport information stored here is used to show airports on the AirScout main map.

Column Name	Column Type	Description
ICAO	*TEXT*	ICAO code of airport
IATA	*TEXT*	IATA code of airport
Lat	DOUBLE	Latitude of airport
Lon	DOUBLE	Longitude of airport
Alt	DOUBLE	Altitude of airport
Airport	TEXT	Name of airport
Country	TEXT	Country of airport
LastUpdated	INT32	Time stamp of last update (UNIX)

4.2. Plane Feeds

Plane feeds are essential to get aircraft positions online. Several types of plane feeds are implemented.

CAUTION: Running an ADS-B receiver and/or sharing the received information may not be legal in some countries!

The text feeds from web servers are not intended for public use and can be loaded only via "deep link" (see Wikipedia). The use may not be permitted in some countries. Due to legal issues the AirScout user is explicitly asked to accept that he does this on his own risk. The feeds can be changed in URL and data format frequently and without notice. Furthermore it is not guaranteed to be a free service in future.

Starting with V1.3 AirScout is using a plug in concept for plane feeds. New/updated feeds can be made available online without updating the whole AirScout version.

4.2.1. Plane feed location and update procedure

Plugin Initial Directory:	<Program's main directory>\Plugin
Plugin Working Directory:	<AppData>\Local\DL2ALF\AirScout\Plugin

The update procedure is as follows:

1. Create the plugin directory if not already created
2. Check the AirScout's initial plugins against the plugins in working directory (version and time of creation)
3. Copy them over to working directory if not exist or newer
4. Check the AirScout web repository for new plugins (version and time of creation)
5. Copy them over if not exist or newer
6. Try to activate each single plugin, ignore those with incompatible versions or errors on start up

If you open lane feed settings via Options/Planes, a configuration file is created and saved into the plugin working directory.

4.2.2. [Web Feed] Open Sky

This is the standard web feed selected at first run. This a community feed; the feed can be used with the following restrictions:

- anonymous: 100 fetches /day
- with registration: 1000 fetches/day

You should consider to support the community by operating your own ADS-B receiver and feed the data into the community server landscape.

Parameters: Yes

Parameter	Description	Default
URL	Server URL	https://opensky-network.org/api/states/all?lamin=%MINLAT%&lomin=%MINLON%&lamax=%MAXLAT%&lomax=%MAXLON%
Timeout	Response timeout	30sec
SaveToFile	Save received data to file	False

Import/Export: None

4.2.3. [Web Feed] VRS Web Server

Working as a community project the feed is dedicated to work with Virtual Radar Server instances (see https://www.virtualradarserver.co.uk).
So far, there is one feed operated by Thomas, OV3T, which delivers data exclusively for AirScout users. You should consider to support the community by operating your own ADS-B receiver and feed the data into the community server landscape.

To avoid misuse the feed is secured by a username/password. You can get your access data by registering here.

Parameters: Yes

Parameter	Description	Default
URL	Server URL	http://airscatter.dk:8890/VirtualRadar/AircraftList.json?ldv=%LASTDV%&stm=%UNIXTIME%&lat=%MYLAT%&lng=%MYLON%&fDstL=%MINDISTKM%&fDstU=%MAXDISTKM%&fAltL=%MINALTFT%&fAltU=%MAXALTFT%
Username	User name	
Password	Password	
Timeout	Response timeout	30sec
URL2	Redundant Server URL	http://airscatter.dudez.no:18080/VirtualRadar/AircraftList.json?ldv=%LASTDV%&stm=%UNIXTIME%&lat=%MYLAT%&lng=%MYLON%&fDstL=%MINDISTKM%&fDstU=%MAXDISTKM%&fAltL=%MINALTFT%&fAltU=%MAXALTFT%
Username2	Redundant user name	
Password2	Redundant password	
UseGeoAlt	Use geodetic altitude instead of barometric	False
LoadShare	Load share between both servers	True
SaveToFile	Save received data to file	False

4.2.4. [Raw Data] RTL1090 Data

This feed reads plane from a simple ADSB receiver (DVB-T dongle) via TCP connection. Pre-configured to work with RTL1090 software. Latest Version 2 is available from here. This feed is confirmed to be working with RTL1090 Version 2 (Build 103).

Parameters: Yes

Parameter	Description	Default
Server	***Server URL or IP address***	***localhost***
Port	Server TCP port	31001
Binary	Use binary protocol or AVR text protocol	true
Report Messages	Report received ADS-B messages to AirScout main window (slow!)	true
Mark Local	Mark locally received planes with "@" to avoid conflicts with position messages from web feeds	true
SaveToFile	Saves received messages to file	false

Import/Export: None

Alternative:

You can also run dump1090.exe which is considered to slightly better in decoding ADS-B messages. The Windows version of this software and installation procedure are available here. The feed is confirmed to be working with version 1.10.3010.14 for Windows. Be sure that you have the following command inside your dump1090.bat:

dump1090.exe --interactive --net --net-beast --mlat --gain -10

Furthermore, you must change your parameters as follows:

Server	**localhost**
Port	**30005**
Interval	**60**
Binary	**True**
ReportMessages	**True**
MarkLocal	**True**

4.2.5. [Web Feed] AirScout Server

This feed reads plane positions from an AirScout web server via HTTP. The web server can be a dedicated AirScout web server or AirScout software configured as a server (see "Options/Network").
This is useful to aggregate other plane feeds and provide it to one or more AirScout client computers. It may also save bandwidth in multi-client environments. The webserver will deliver a JSON file as described in the Appendix.

Parameters: Yes

Parameter	Description	Default
URL	AirScout Server URL	http://localhost:9880/planes.json
Interval	Update interval	60sec

Import/Export: None

4.2.6. [Web Feed] Flexible JSON

This feed is the "Swiss Army Knife" of web feeds. It is intended to read plane feeds from web resources <u>not yet discovered and unknown in structure</u>.

If a valid URL is entered, AirScout is trying the following:

- download the JSON file from the given URL
- detect the data structure
- convert it into a data table
- auto-index the columns to find all necessary data

As a result, the discovered indices are saved in the settings and can be altered by the user.
To use this feed:

- set all fields to default value by clicking on "Default" button
- enter a valid URL for download
- close settings and wait for at least one download interval
- check which indices were found (<> -1), Index_Hex and Index_UTC are mandantory
- adjust rest of indices if necessary

To find out the right indices you may check the file "AutoJSON_Values.csv" in the applications temp directory.

CAUTION: Using this plane feed may violate copyrights in your country. You must explicitly agree that you are using the feed at your own risk.

Parameters: Yes

Parameter	Description	Default
URL	Server URL	[none]
Index_UTC	Index of the UTC column	-1
Index_Hex	Index of the HEX column	-1
Index_Call	Index of the Call column	-1
Index_Lat	Index of the Latitude column	-1
Index_Lon	Index of the Longitude column	-1
Index_Alt	Index of the Altitude column	-1
Units_Alt	Units of Altitude column [ft] or [m]	[ft]
Index_Speed	Index of the Speed column	-1
Units_Speed	Units of Speed column [kts], [km/h] or [m/s]	[kts]
Index_Track	Index of the Track/Heading column	-1
Index_Type	Index of the Type Code column	-1
Index_Squawk	Index of the Squawk column	-1
Index_Reg	Index of the Registration column	-1
MinElements	Minimum number of elements in the JSON file (to avoid false detects)	10
MinPlanes	Minimum number of planes in the JSON file (to avoid false detects)	50
SaveToFile	Saves received data to file	False

Import/Export: Yes

4.3. Minimum Hardware and Software Requirements

The use of AirScout requires sufficient hardware resources; therefore an up-to-date hardware is strongly recommended. The following configuration is considered as the minimum:

- Dual core processor 1.2GHz
- 2GB RAM
- 20GB hard disk space
- Screen resolution 1024 x 786 (1280 x 1024 recommended)
- Microsoft Windows XP(*), Vista(*), Win7, Win8, Win10, Linux/Mono(*)
- Microsoft .NET 4.0 (usually already installed on systems > = Vista)
- Internet connection

(*) reduced functionality, not confirmed to be working with V1.4.x

If you need to install the Microsoft .NET4.0 runtime you can download it from here:
https://www.microsoft.com/en-US/download/details.aspx?id=17718

4.4. Directory structure

After the installation is completed the resulting directory structure in the user's Application Data Directory is created automatically.

<Program directory>	Contains all main components (incl. AirScout.exe)

<AppDataDirectory>\Local\DL2ALF\ScoutBase

StationData:	Contains the ScoutBase station database
ElevationData:	Contains the ScoutBase elevation databases
GLOBE:	Contains cached GLOBE elevation data
SRTM1:	Contains cached SRTM1 elevation data
SRTM3:	Contains cached SRTM3 elevation data
\ASTER1:	Contains cached ASTER1 elevation data
\ASTER3:	Contains cached ASTER3 elevation data
\PropagationData:	Contains the ScoutBase propagation database

<AppDataDirectory>\Local\DL2ALF\AirScout

AircraftData:	Contains the AirScout aircraft database
Log:	Contains the AirScout log files
Plugin:	Contains the AirScout plane feed plugins and configs
Tmp:	Contains all temporary files, hardcopies, exports etc.

Some other directories may be created in order to get backward compatibility to previous versions.

4.5. Installation procedure

AirScout installation comes without an installer in a single ZIP – file. In order to get the software running please do the following:

1) Download the ZIP file from the website
2) Unzip the file to a directory of your choice (be sure to extract it with all subdirectories)
3) Check that your PC clock is running synchronized and in the right time zone
4) Run AirScout.exe
5) A wizard will guide you through basic settings
6) After finishing the Wizard the program will start immediately
7) Press "Run/Pause" button to see path on map and planes moving
8) Click on "Options" button for further options when on "Pause"

If you are experiencing problems especially with further BETA versions of AirScout you can run the program with an cleanup routine at start up. This will wipe your settings, data base and other temp files.

Run "AirScout_clean.cmd" to activate the clean start and choose cleanup functions in the dialog box:

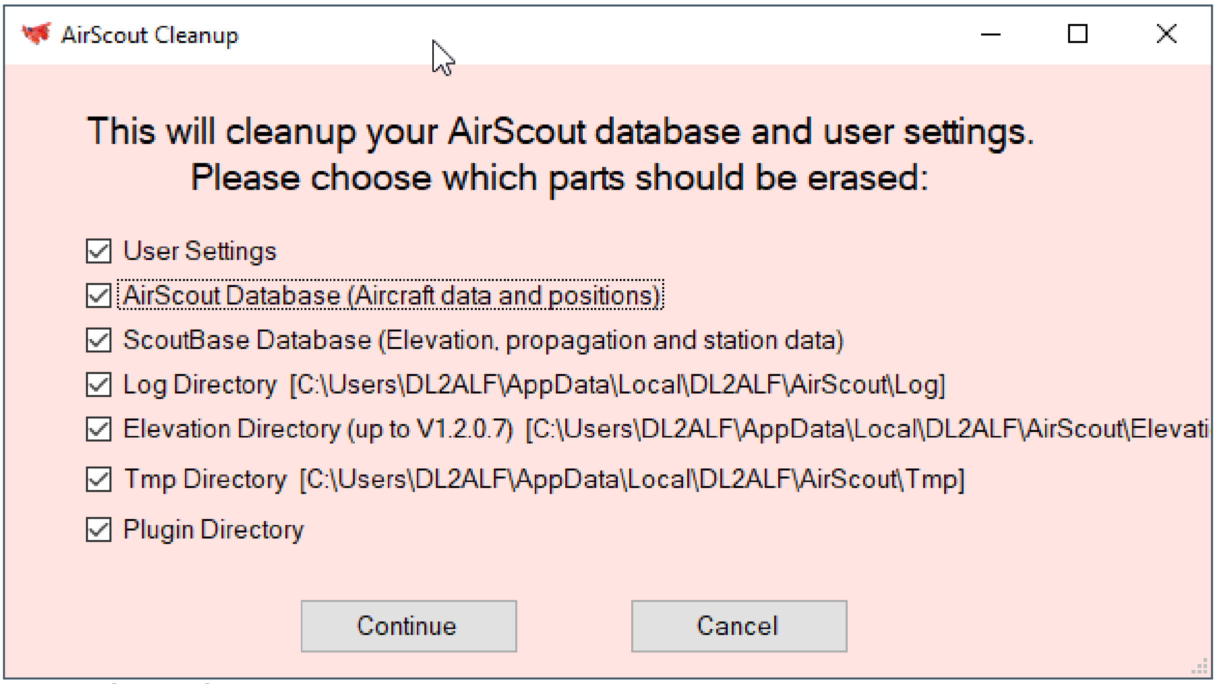


Figure 4-4 Cleanup Dialog

4.6. First Run

Starting with V1.2 a database structure must be created on first run. An internet connection is necessary to download data files from web resource. While this will take some time you will see the splash screen with status messages.

Figure 4-5 Splash window while initializing database on first run

When the database setup is finished successfully, a wizard will guide you through the basic settings on first run. This will ensure that all precautions were made before running AirScout for the first time. This will also happen on each new AirScout version although most settings are kept during first run.

The navigation is simple: just use "Back" and "Next" to navigate through the tabs. The "Finish" button only shows up on the last page. Pressing "Cancel" will cancel the wizard and close the program. The procedure will start again from the beginning next time.

4.6.1. *Tab "Terms and Conditions"*

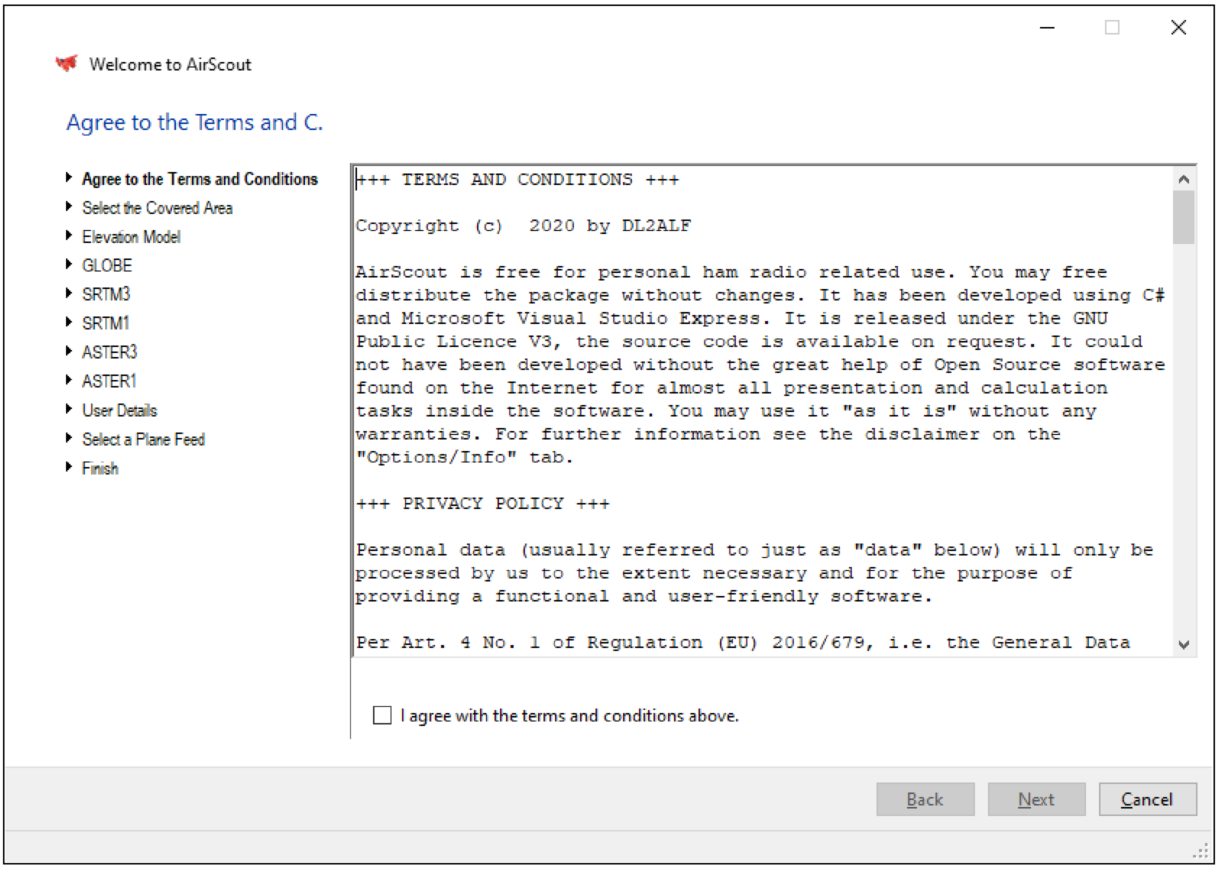


Figure 4-6 Terms and Conditions

You will see the terms and conditions of AirScout here. Please read it carefully. You must agree with the terms and conditions to continue.

4.6.2. *Tab "Select the covered area"*

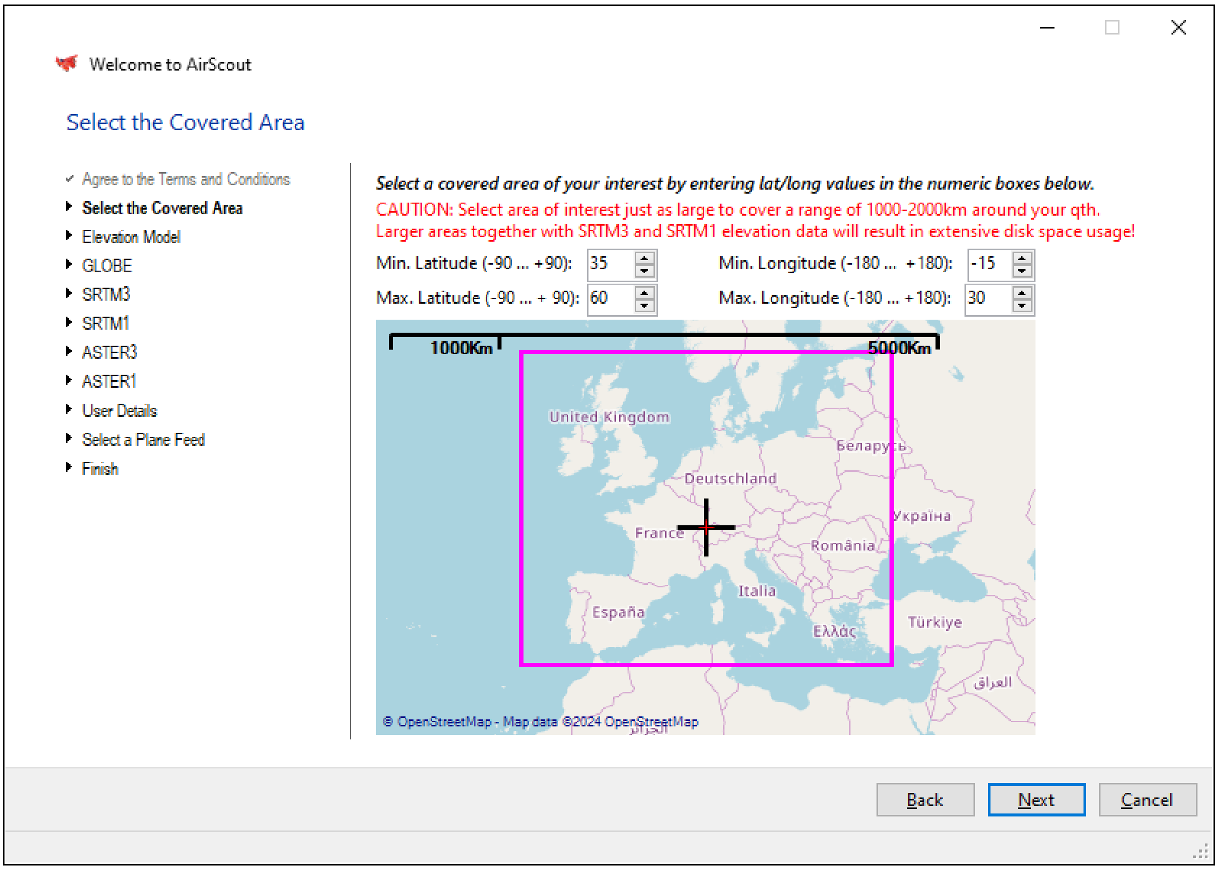


Figure 4-7 Covered area

The covered area is representing the “basic scope” of AirScout.

- Flight traffic is only recorded inside this area. The covered area is acting as a basic filter already when fetching data from web server.
- AirScout determines the elevation tiles needed for this area and tries to download them automatically.

You can adjust the area by entering numeric values in the boxes or by clicking on the up/down buttons next to the values. The frame on the map will be updated immediately.

4.6.3. Tab "Select the elevation model"

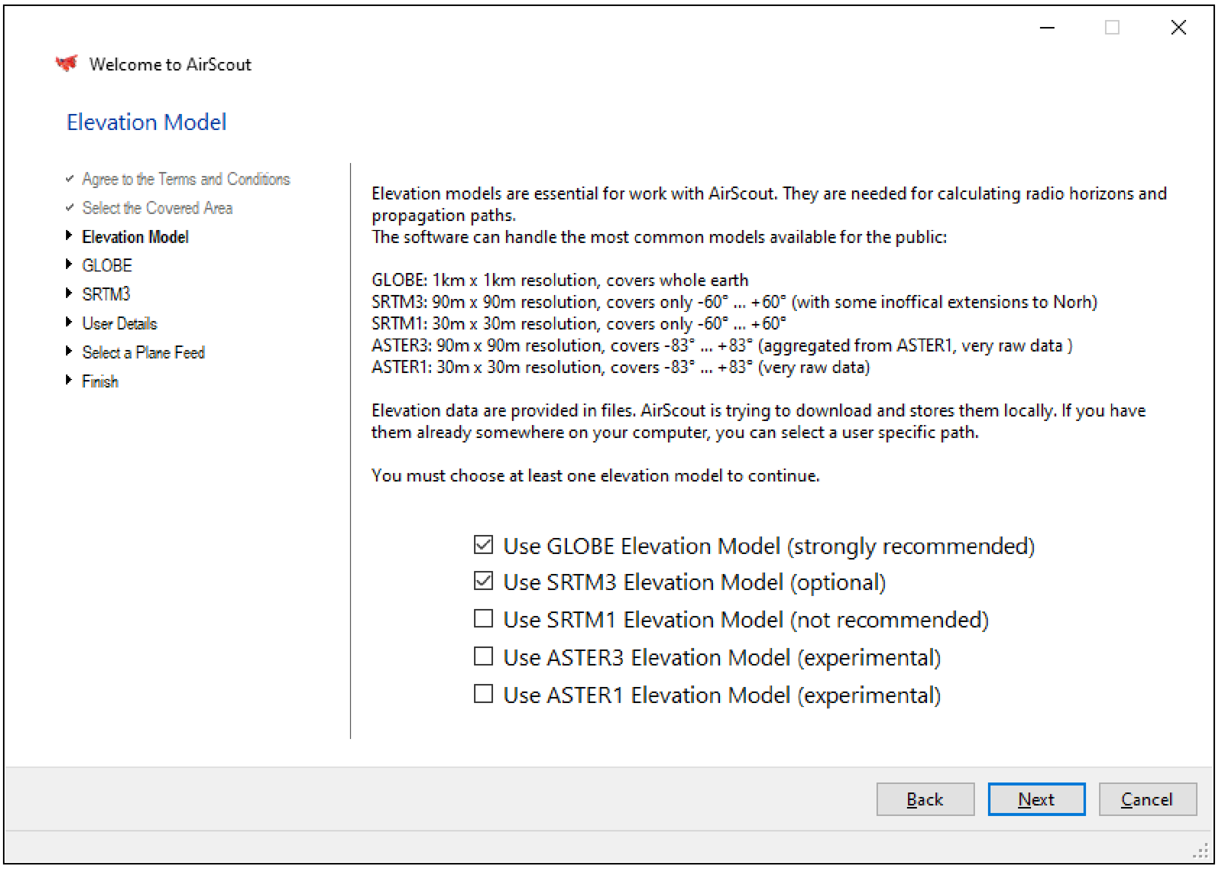


Figure 4-8 Elevation model

Elevation models are essential for work with AirScout. They are needed for calculating radio horizons and propagation paths. The software can handle the three most common models available for the public:

GLOBE: 1km x 1km resolution, covers whole earth
SRTM3: 90m x 90m resolution, covers only -60° ... +60°
SRTM1: 30m x 30m resolution, covers only -60° ... +60°
ASTER3: 90m x 90m resolution, covers only -80° ... +80°
ASTER1: 30m x 30m resolution, covers only -80° ... +80°

Elevation data are provided in files index by 6digit Maidenhead locator. AirScout is trying to download and stores them into local database.

You must choose at least one elevation model to continue.

4.6.4. Tab "GLOBE elevation model"

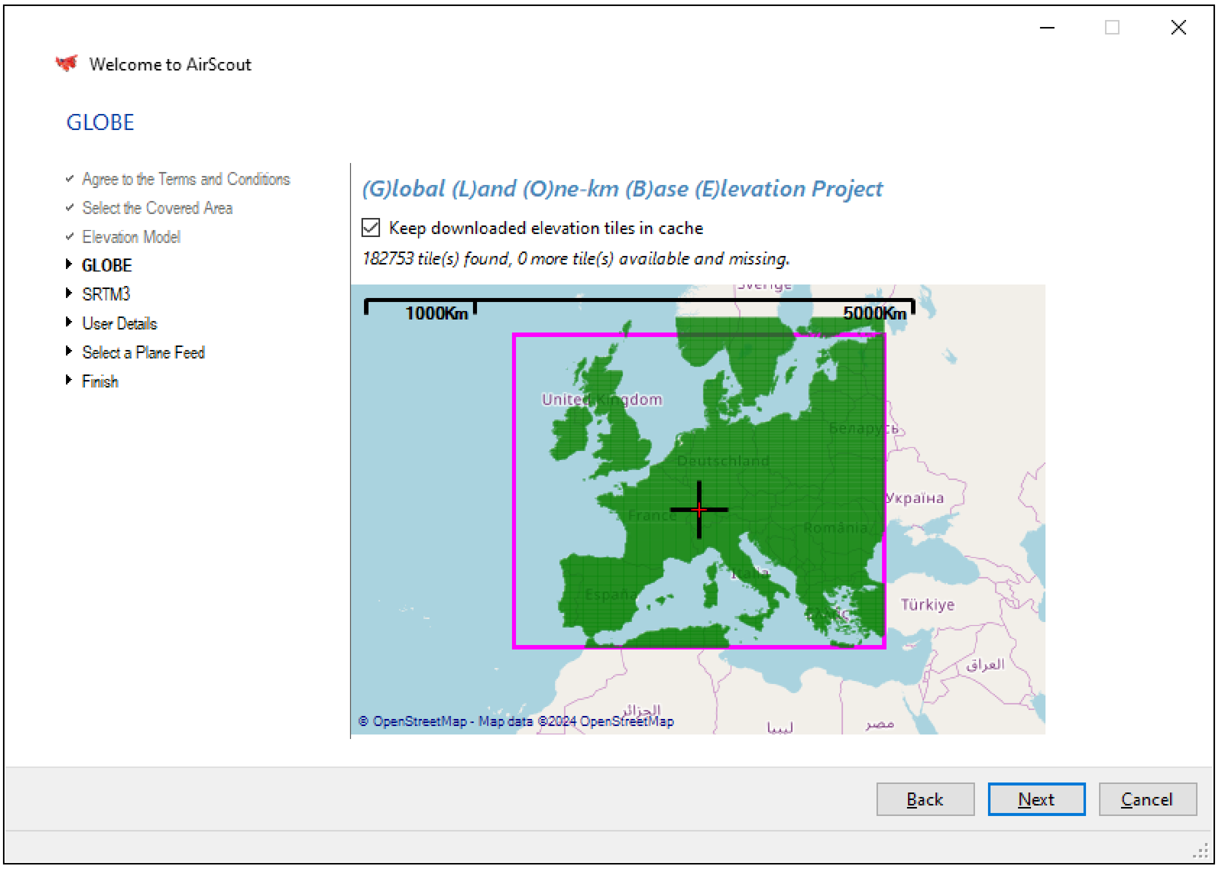


Figure 4-9 GLOBE Elevation Model

When opening this tab AirScout is calculating the amount of tiles needed to cover your area of interest. Then the software is scanning your local database and is trying to find tiles on a web catalogue if they are missing. After the process is finished the tiles are shown on the map using the following colour code:

- Tiles found already in local database are shown in green
- Tiles needed and found on web catalogue are shown in red

A status message is shown while the scan is running. You may keep the downloaded ZIP files with tiles in your local storage (Default = on)

4.6.5. Tab "SRTM3 elevation model"

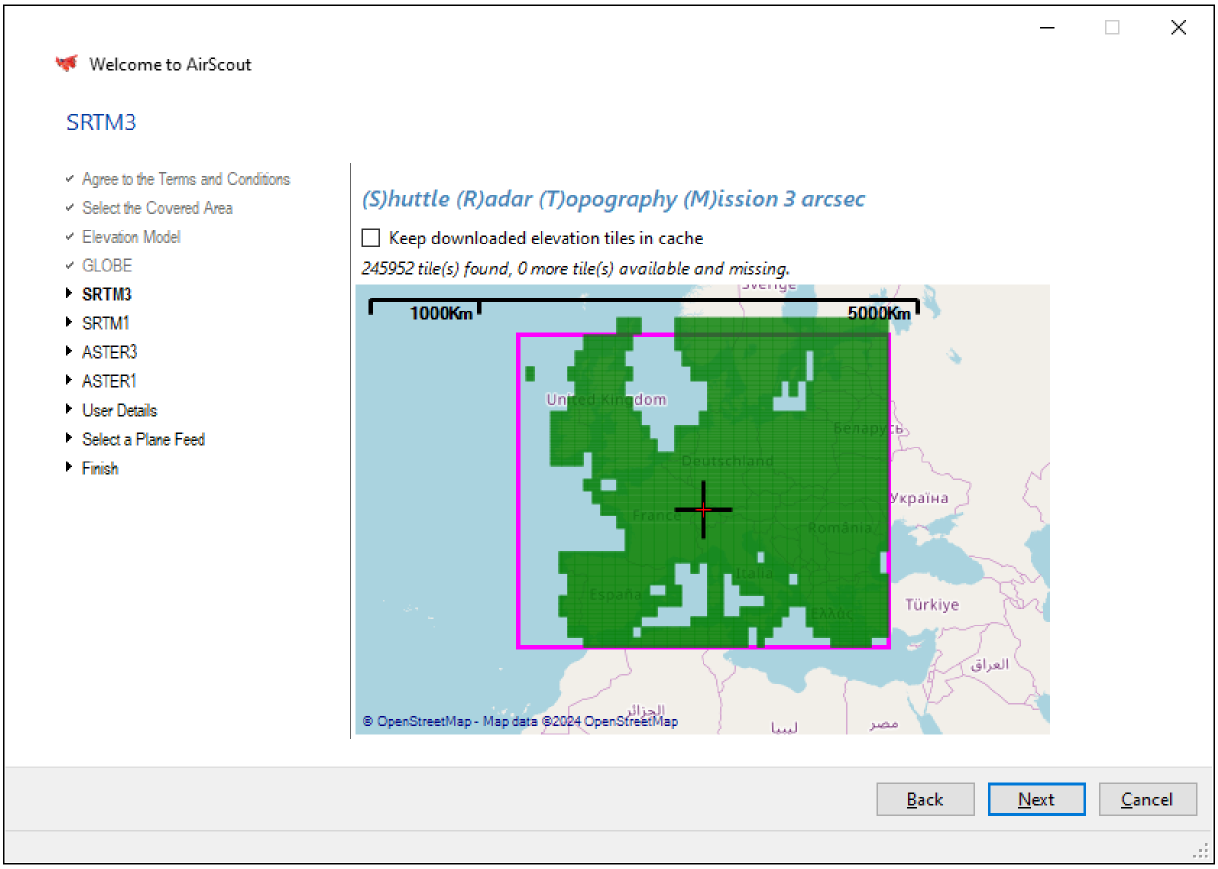


Figure 4-10 SRTM3 Elevation Model

When opening this tab AirScout is calculating the amount of tiles needed to cover your area of interest. Then the software is scanning your local database and is trying to find tiles on a web catalogue if they are missing. After the process is finished the tiles are shown on the map using the following colour code:

- Tiles found already in local database are shown in green
- Tiles needed and found on web catalogue are shown in red

A status message is shown while the scan is running. You may keep the downloaded ZIP files with tiles in your local storage (Default = off).

4.6.6. *Tab "SRTM1 elevation model"*

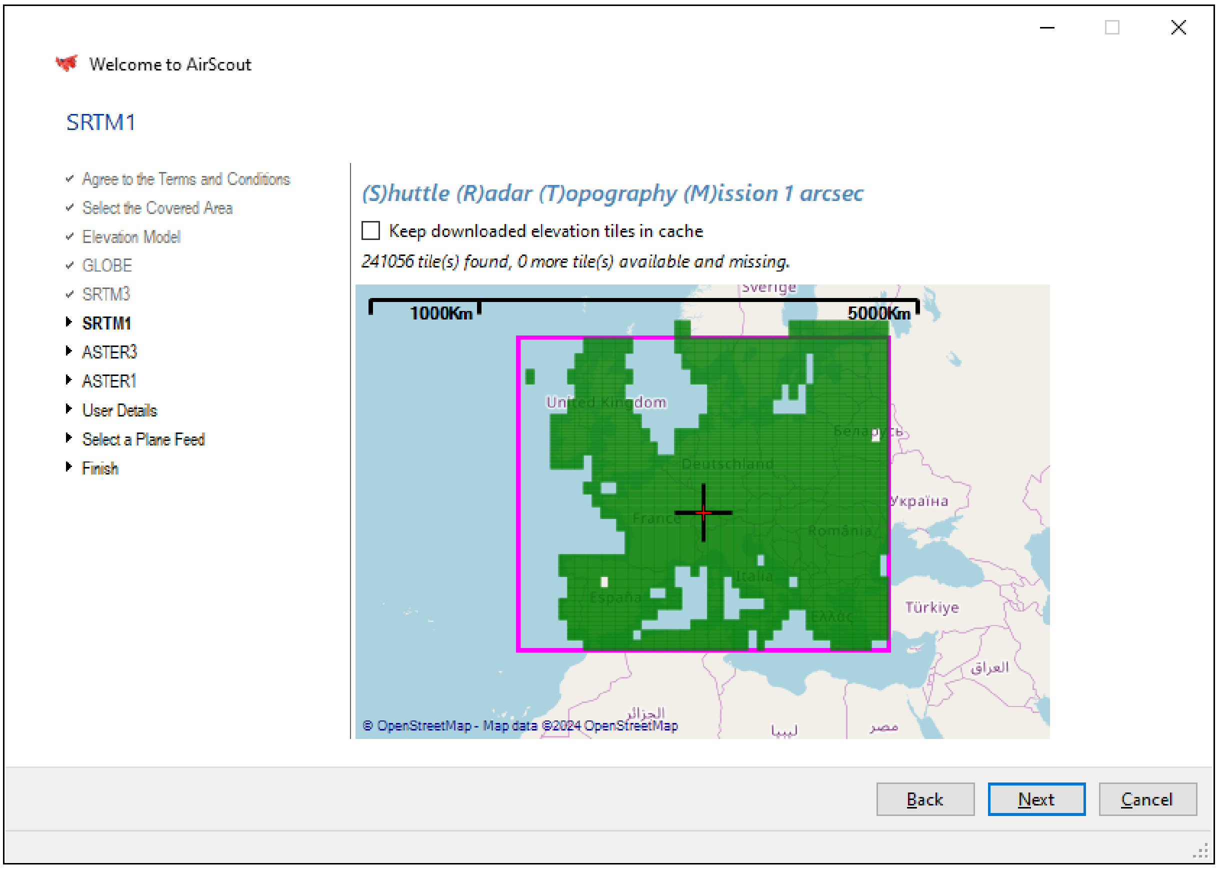


Figure 4-11 SRTM1 Elevation Model

When opening this tab AirScout is calculating the amount of tiles needed to cover your area of interest. Then the software is scanning your local database and is trying to find tiles on a web catalogue if they are missing. After the process is finished the tiles are shown on the map using the following colour code:

- Tiles found already in local database are shown in green
- Tiles needed and found on web catalogue are shown in red

A status message is shown while the scan is running. You may keep the downloaded ZIP files with tiles in your local storage (Default = off).

4.6.7. Tab "ASTER3 elevation model"

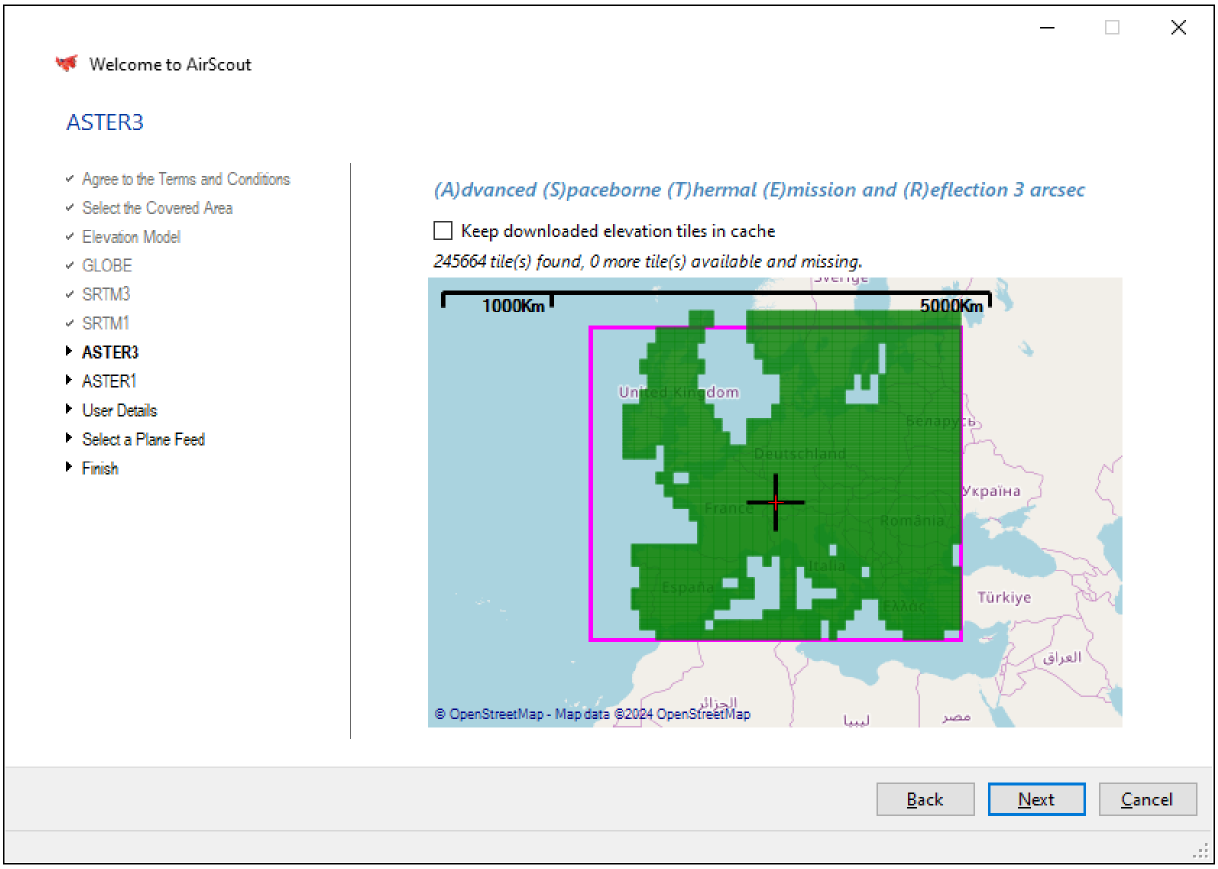


Figure 4-12 ASTER3 Elevation Model

When opening this tab AirScout is calculating the amount of tiles needed to cover your area of interest. Then the software is scanning your local database and is trying to find tiles on a web catalogue if they are missing. After the process is finished the tiles are shown on the map using the following colour code:

- Tiles found already in local database are shown in green
- Tiles needed and found on web catalogue are shown in red

A status message is shown while the scan is running. You may keep the downloaded ZIP files with tiles in your local storage (Default = off).

4.6.8. Tab "ASTER1 elevation model"

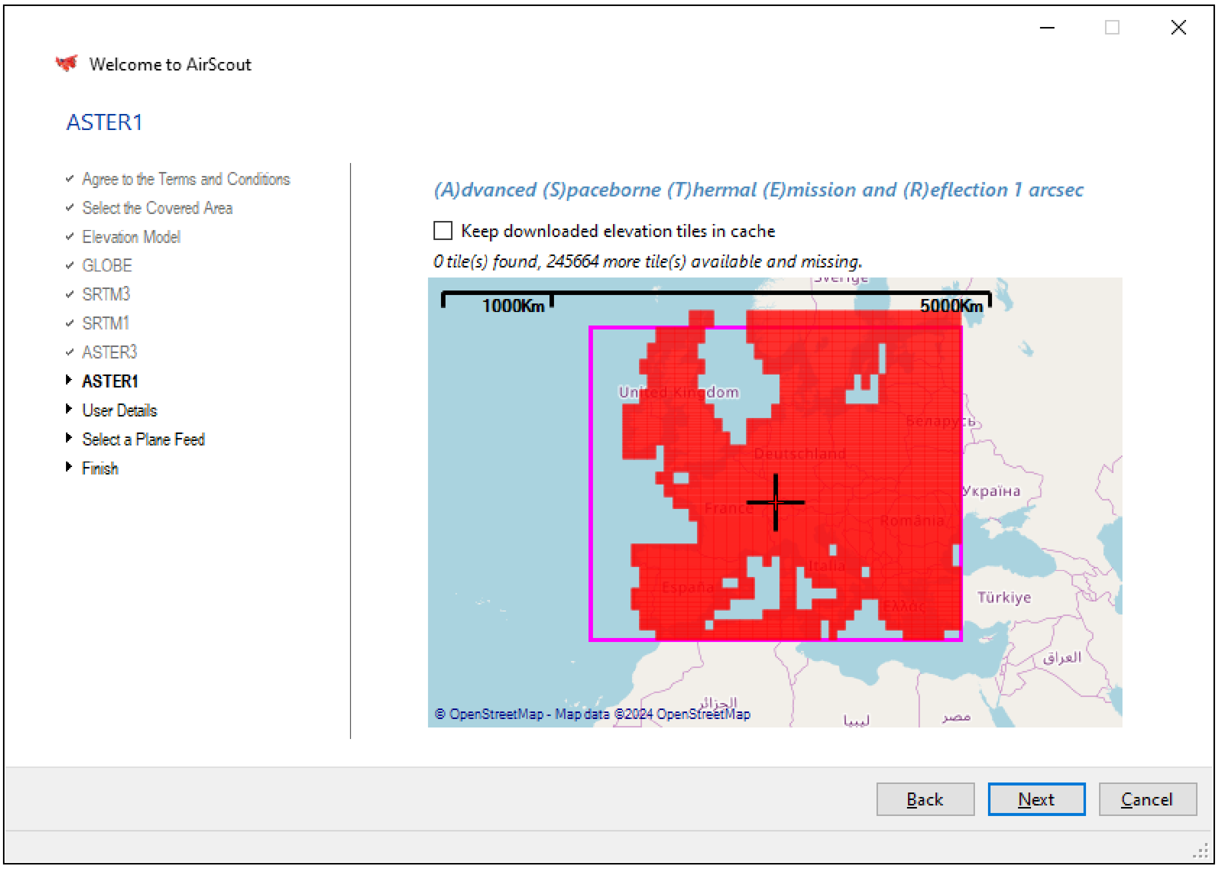


Figure 4-13 ASTER1 Elevation Model

When opening this tab AirScout is calculating the amount of tiles needed to cover your area of interest. Then the software is scanning your local database and is trying to find tiles on a web catalogue if they are missing. After the process is finished the tiles are shown on the map using the following colour code:

- Tiles found already in local database are shown in green
- Tiles needed and found on web catalogue are shown in red

A status message is shown while the scan is running. You may keep the downloaded ZIP files with tiles in your local storage (Default = off).

4.6.9. *Tab "User details"*

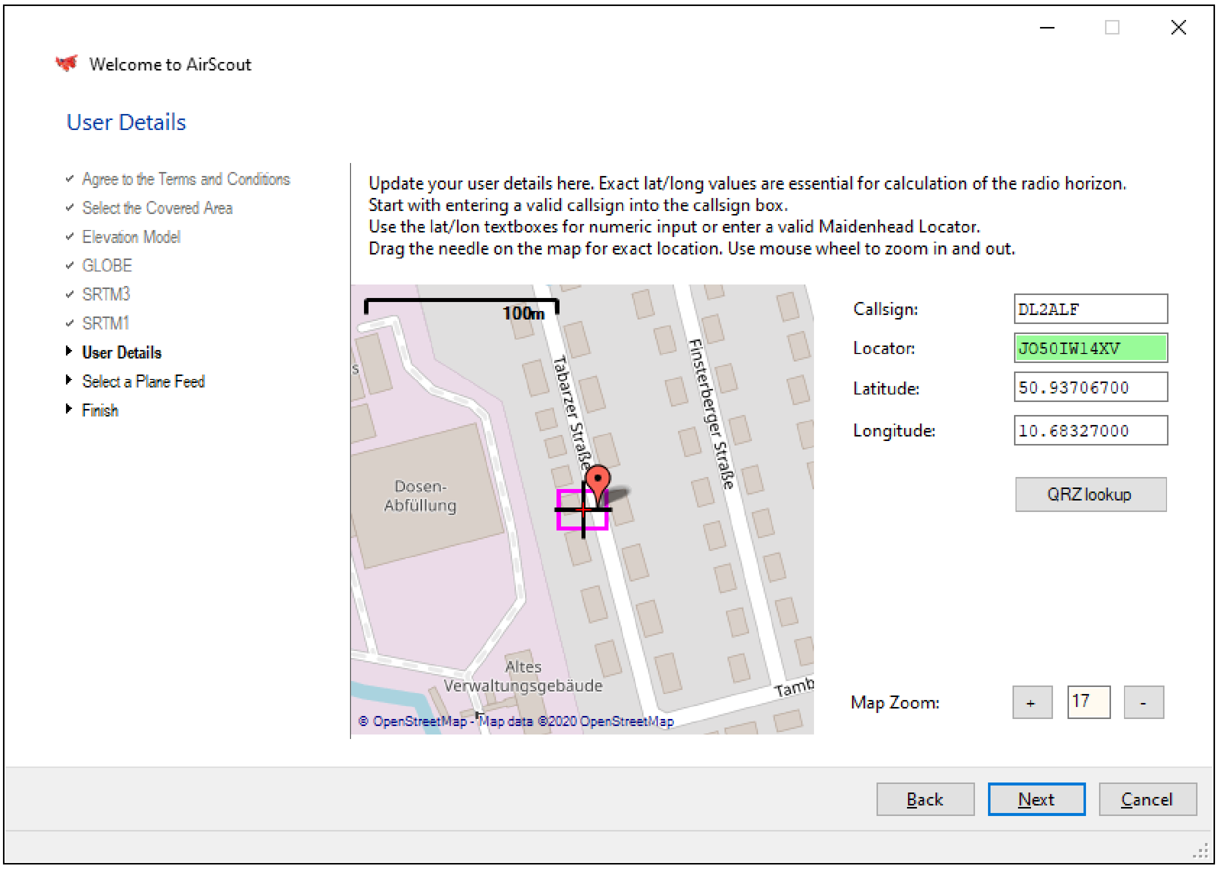


Figure 4-14 User Details

You must update your user details here. Exact lat/long values are essential for calculation of the radio horizon. To complete the information do the following.

1. Always start with entering a valid call sign into the call sign box. If found in the call sign database the location is updated automatically.
2. Update the lat/lon information by one of the following methods:

   - enter numeric values into the lat/lon textboxes (English notation)
   - enter a valid Maidenhead Locator (6..14 digits, default = 10)
   - Click and hold mouse left to drag the needle on the map for exact location. Use mouse wheel or +/- buttons to zoom in and out.

If your user details are listed in the QRZ.com database you can fetch them also by clicking on "QRZ lookup" button. The exact position is then taking from their database in the case the 6digit Maidenhead locators are matching.

It is strongly recommended that you have a QRZ.com account and maintain your exact location there. It is the most common database for amateur radio call signs and location details on the Internet. As long as there is no distinct AirScout web database is available, QRZ.com is the only way to publish your exact location to other AirScout users!

4.6.10. Tab "Plane Feeds"

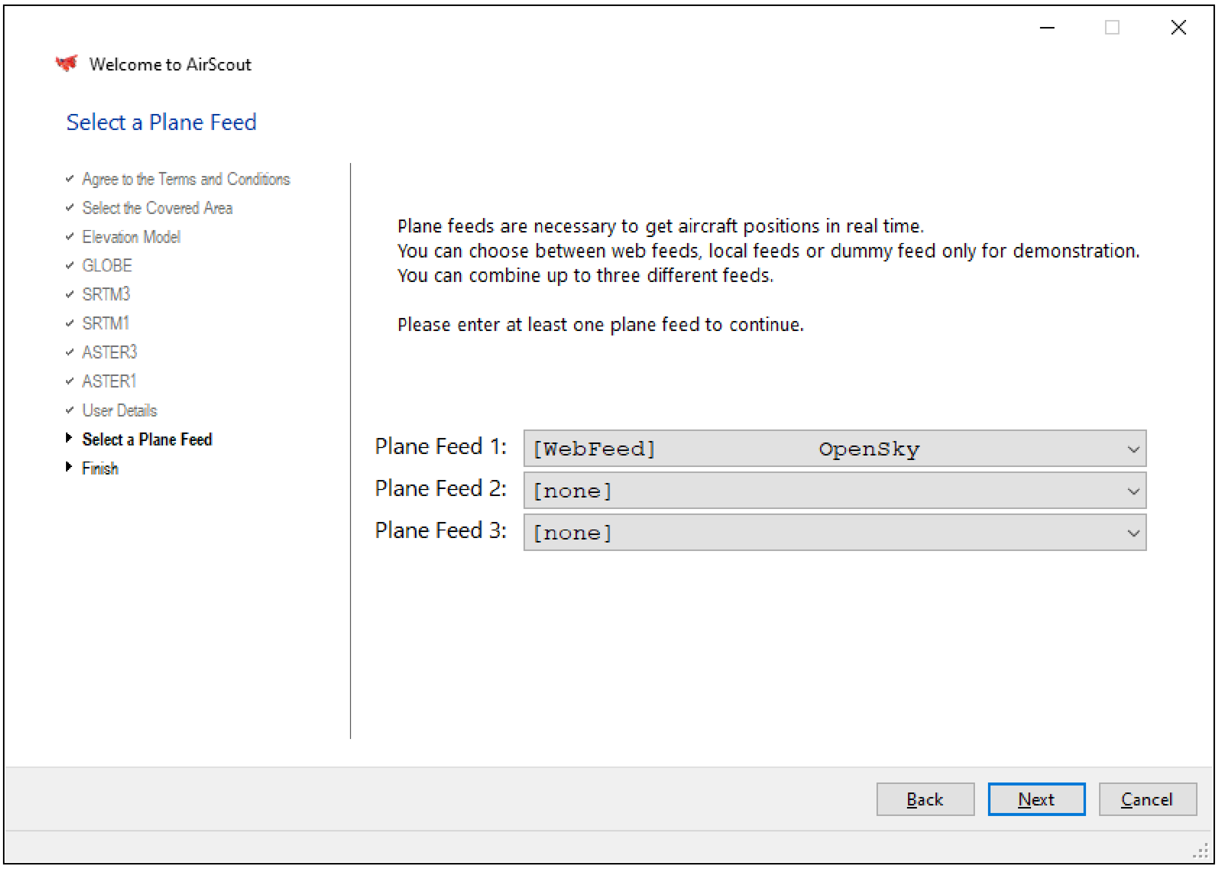


Figure 4-15 Plane Feeds

Plane feeds are necessary to get aircraft positions in real time. You can choose between:

- web feeds from several servers
- local feeds with your own ADS-B receiver

You can combine up to three different feeds here. The feeds are running with default settings guaranteed to be working. You can change settings later via "Options/Planes". Normally the simultaneous use of more than one web stream is producing best results. Anyway, sometimes inconsistencies between two feed may occur and you will see doubled plane icons with slightly different positions or call signs. In this case or to reduce network traffic it is recommended to disable multiple feeds. One single feed will produce a traffic of about 2MB/min.

Please enter at least one plane feed to continue.

4.6.11. Tab "Finish"

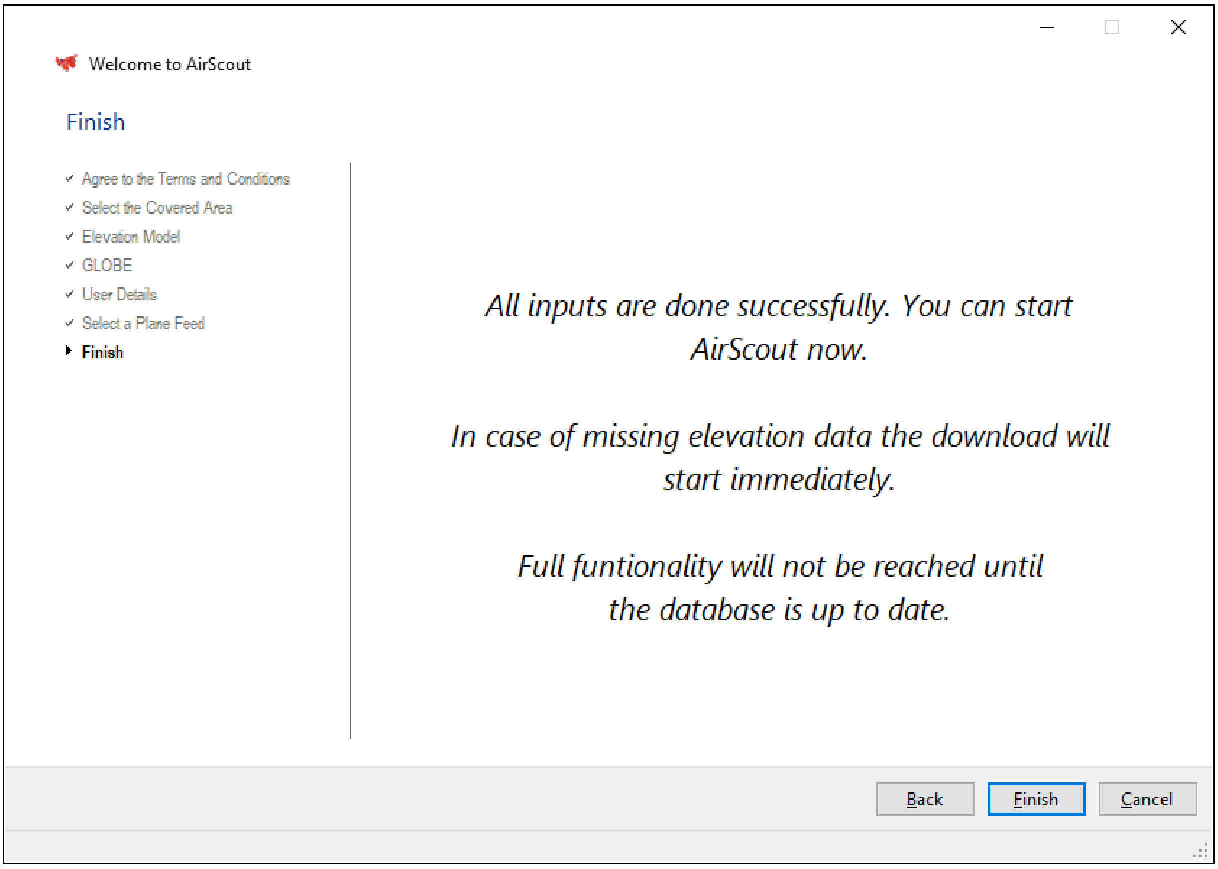


Figure 4-16 Finish

Congratulations! You are finished with all precautions and can start AirScout now by clicking on the "Finish" button. The AirScout main window will show up and all update processes will start immediately.

4.7. Program's Main Window and its Elements

See the programs main window and its elements below:

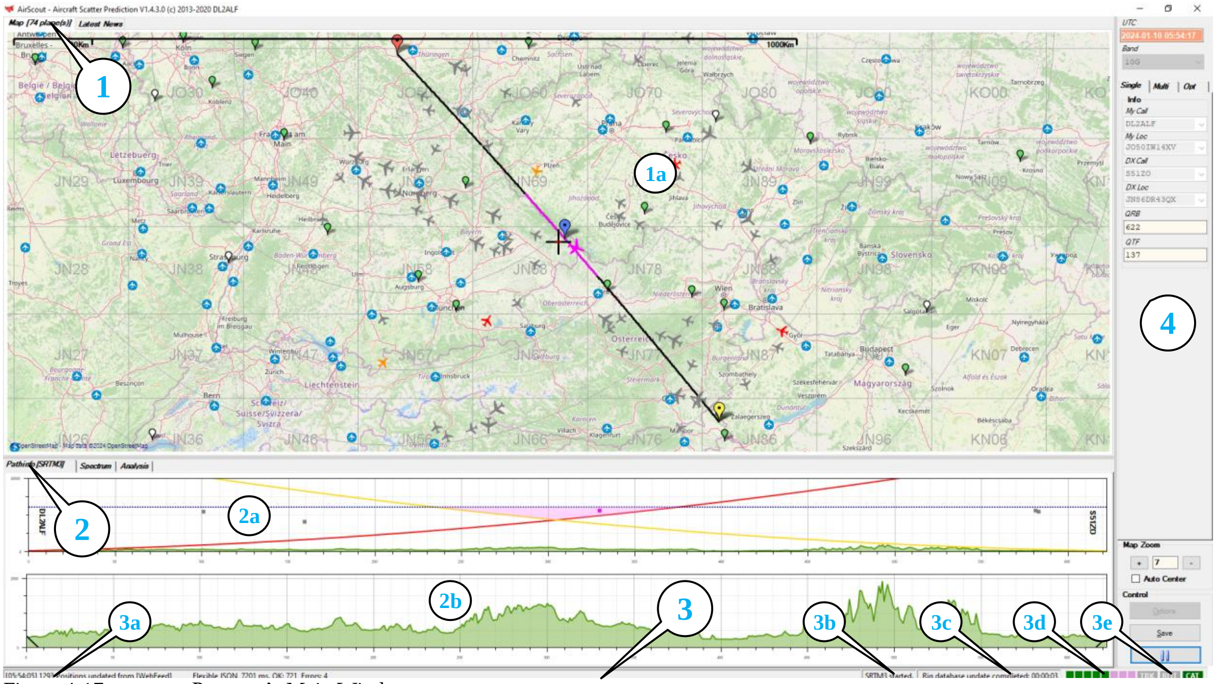


Figure 4-17 Program's Main Window

4.7.1. *Main Map Tab Control (1)*

Combines the main map and the browser window with AirScout news. AirScout counts the planes currently visible and shows it on the tab title bar (only on Windows). The browser tab has no function on Linux/Mono.

Main Map (1a)

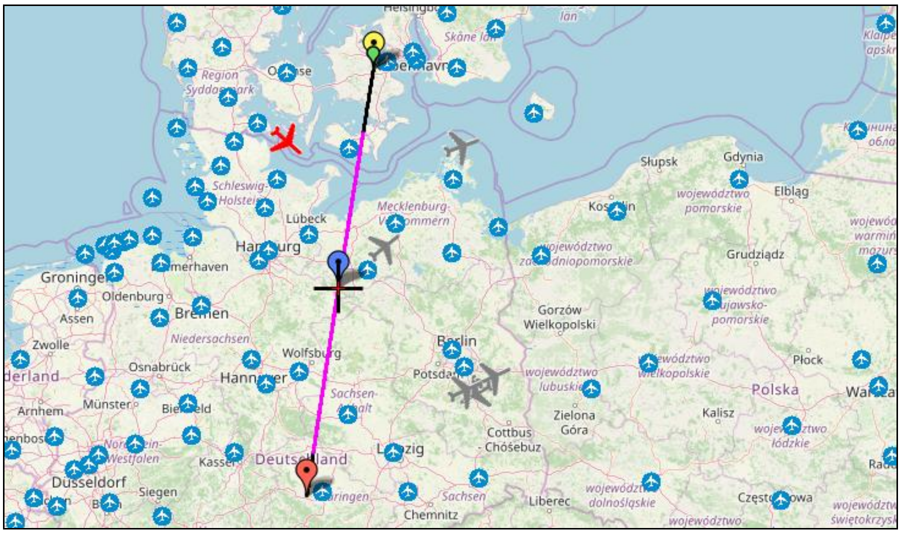


Figure 4-18 Main Map

Is showing:

- the map (Open Street Map by default)
- the calculated path (in black)
- the “hot path” (in magenta)
- big red marker for MyLoc
- big yellow marker for DXLoc
- big blue marker for midpoint between
- small red marker for begin of “hot path”
- small yellow marker for end of “hot path”
- small blue marker for midpoint of “hot path”
- small green marker for location of stations from watch list with precise location
- small white marker for location of stations from watch list without precise location
- coloured planes with real time positions

4.7.2. Path Info Tab Control (2)

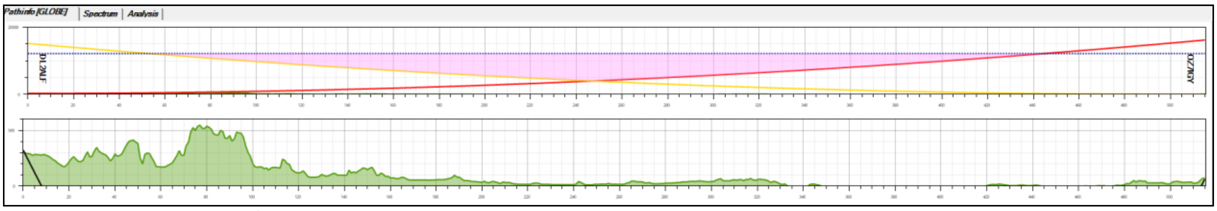


Figure 4-19 Path Info

Path Info Diagrams

Is showing in upper half (**2a**)

- red line for minimum aircraft altitude seen from MyLoc
- yellow line for minimum aircraft altitude seen from DXLoc
- blue line for maximum aircraft altitude (12200m by default)
- magenta triangle for “hot area”, zone of mutual visibility
- gray points for aircrafts on or near path with not suitable altitude
- magenta points for aircrafts on or near path with suitable altitude
- green area with elevation profile (mostly not seen because of scaling)
- MyCall
- DXCall

Is showing in lower half (2b):

- green area for elevation profile with auto scaling
- black line for pure Line Of Sight (LOS)

Spectrum Diagram (hidden)

Is showing a 10 mins history of the NF – spectrum when running Spectrum Lab by DL4YHF as a server and nearest plane in a small map most probably causing the signal enhancement. Can be used to detect signal enhancements when aircrafts are passing.

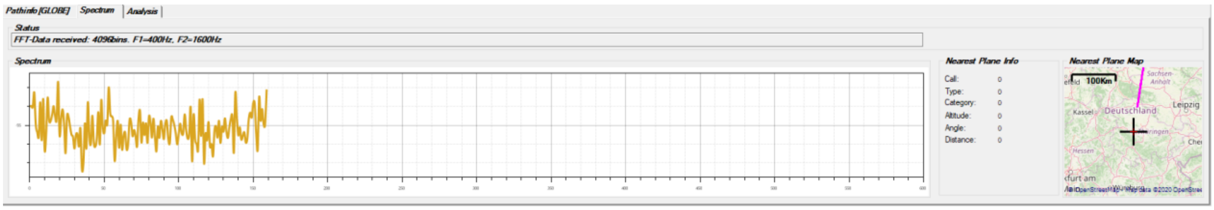


Figure 4-20 Spectrum Diagram

History Analysis (hidden)

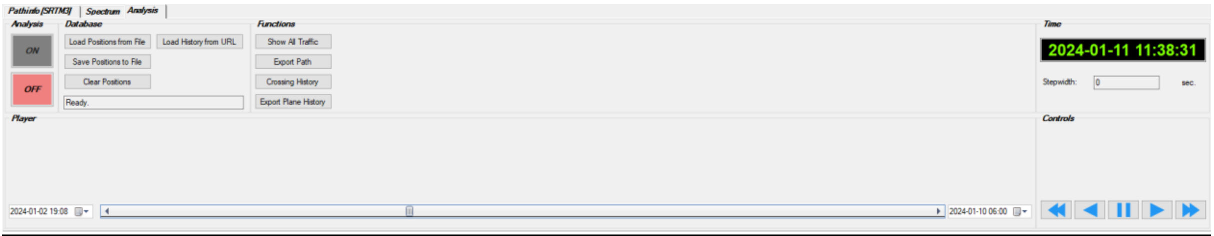


Figure 4-21 History Analysis Player

Is showing controls and player for history analysis. You can load aircraft positions from file or from database and play past movements on the map. Furthermore you can analyse the whole traffic in the Covered Area or check the crossing history for a distinct path.

4.7.3. *Status Line (3)*

Is showing messages and status info of subprocesses and devices for a quick overview.

General Messages (3a)

Is showing general messages like startup, plane feeds working, frequency control and others.

Background Calculations (3b)

Is showing messages from the background calculation processes.

Database Updater (3c)

Is showing messages from the background database updater.

Database Status LEDs (3d)

Is showing status of databases. The color code of the LEDs is as follows:

	Not used
	Unknown or not used
	Updating
	Complete but not confirmed up to date
	Complete and up to date

Tracking and External Device Status (3e)

Is showing status of tracking and external devices (rotator, rig). The color code of the status indicators is as follows:

CAT	Disabled
CAT	Enabled but not active
CAT	Enabled and active
CAT	Error occurred

4.7.4. Sidebar (4)

Contains all user input elements, information about the current path and tabs for mode and quick options.

UTC

Is showing UTC and is updated only when in “Play” mode.

Band

Is showing the currently selected band. You can change it using the drop down list box but only when in “Pause” mode.

Simple Tab

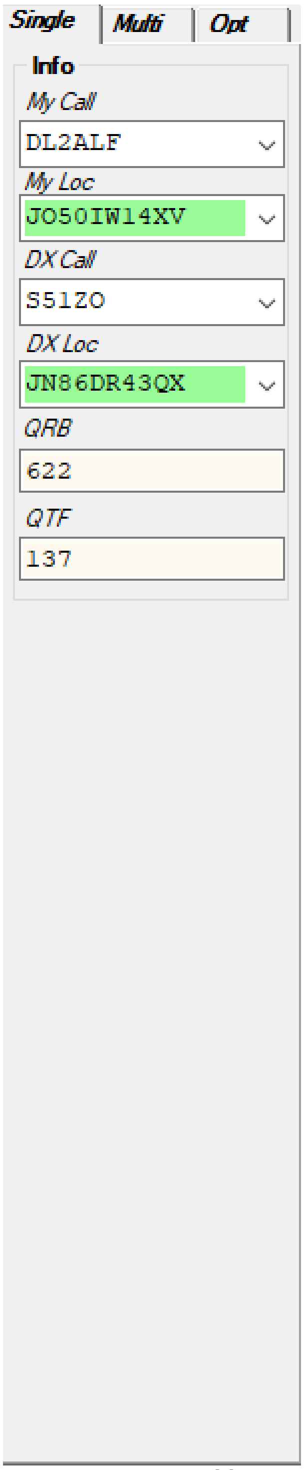


Figure 4-22 Sidebar

MyCall

Is showing MyCall. You can enter a new call only when in "Pause" mode.

MyLoc

Is showing MyLoc as a 6 digits Maidenhead locator. You can enter a new locator only when in "Pause" mode. Green background indicates that the position used is more precise than the midpoint of the 6 digit locator.

DXCall

Is showing the DXCall. You can enter a new call or quick choose a previous on via drop down but only when in "Pause" mode. If the new call is found in database, the according position is loaded and shown in the DXLoc box below.

DXLoc

Is showing the DXLoc as a 6 digit Maidenhead locator. You can enter a new locator only when in "Pause" mode. Green background indicates that the position used is more precise than the midpoint of the 6 digit locator.

QRB Box

Is showing the distance between MyLoc and DXLoc and is updated when in "Play" mode. You cannot enter anything here.

QTF Box

Is showing the direction from MyLoc to DXLoc and is updated when in "Play" mode. You cannot enter anything here.

Changes of tab or inputs are only allowed when in "Pause" mode.

Multi Tab

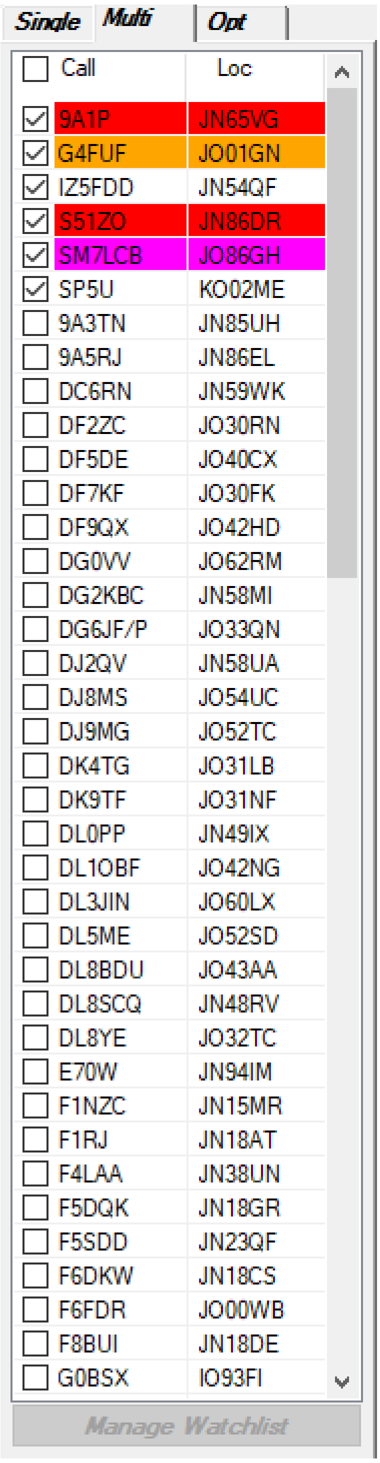


Figure 4-23 Watchlist

Shows a watch list of pre-selected call signs and locators where the user can select one or more to show multiple paths on the map.

The list can be filled manually at Options/Watchlist or can be synchronized from external sources like wtKST with ON4KST users.

The Multi Tab makes it easy to supervise more than one path and to estimate the general situation in relation to wanted QSO partners.

Background color of entries is showing the AS potential or the best AS potential in case of multiple path crossings.

Changes of tab or change of selections are only allowed when in "Pause" mode.

Opt Tab

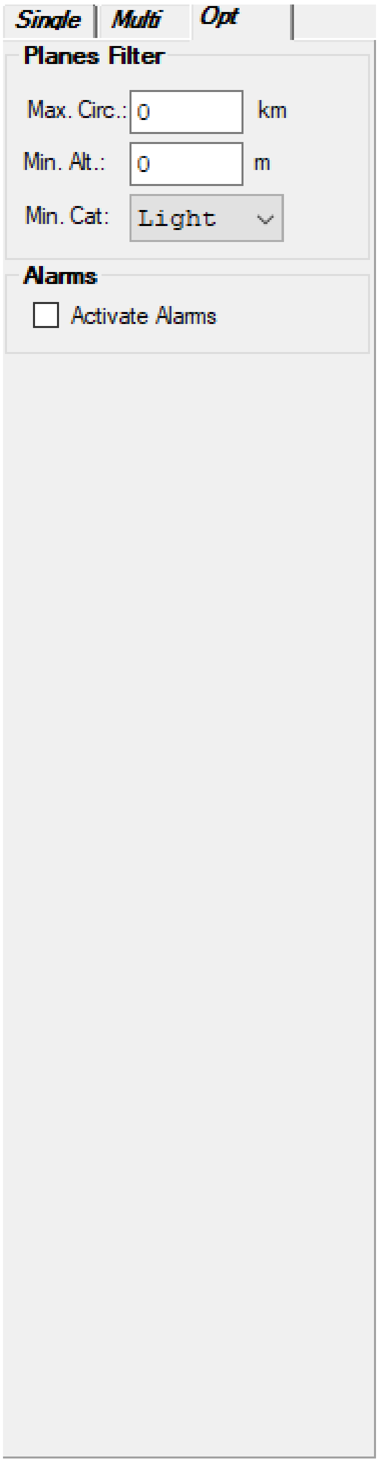


Figure 4-24 Quick Options

Shows some quick filter options for planes showing on map. You can enter a minimum altitude and a minimum category an aircraft must have to be shown on the map.

If a filter other than default is selected the box appears in plume colour

Furthermore, the alarm can be activated/deactivated here.

Changes are only allowed when in "Pause" mode.

Map Zoom Box

Is showing the current map zoom level. You can change it either with the “+” or “-“ buttons or using the mouse scroll bar when over the map area. When checking the “Auto Center” option the map is centered to the midpoint between MyLoc and DXLoc when in “Play” mode.

Options Button

Opens the Options dialog box. All background processes were stopped until the dialog box is closed. Available only in "Pause" mode.

Save Button

Makes a hardcopy of the current main window. Available only in "Pause" mode.

Play/Pause Button

Toogles between "Play" and "Pause" mode. When in "Play" mode, most of the user interactions are blocked until back in "Pause" mode.

4.8. Program's Options window and its tabs

All options and settings are shown in the "Options" Dialog box.

4.8.1. *Tab "General"*

The "General" tab contains information about general settings.

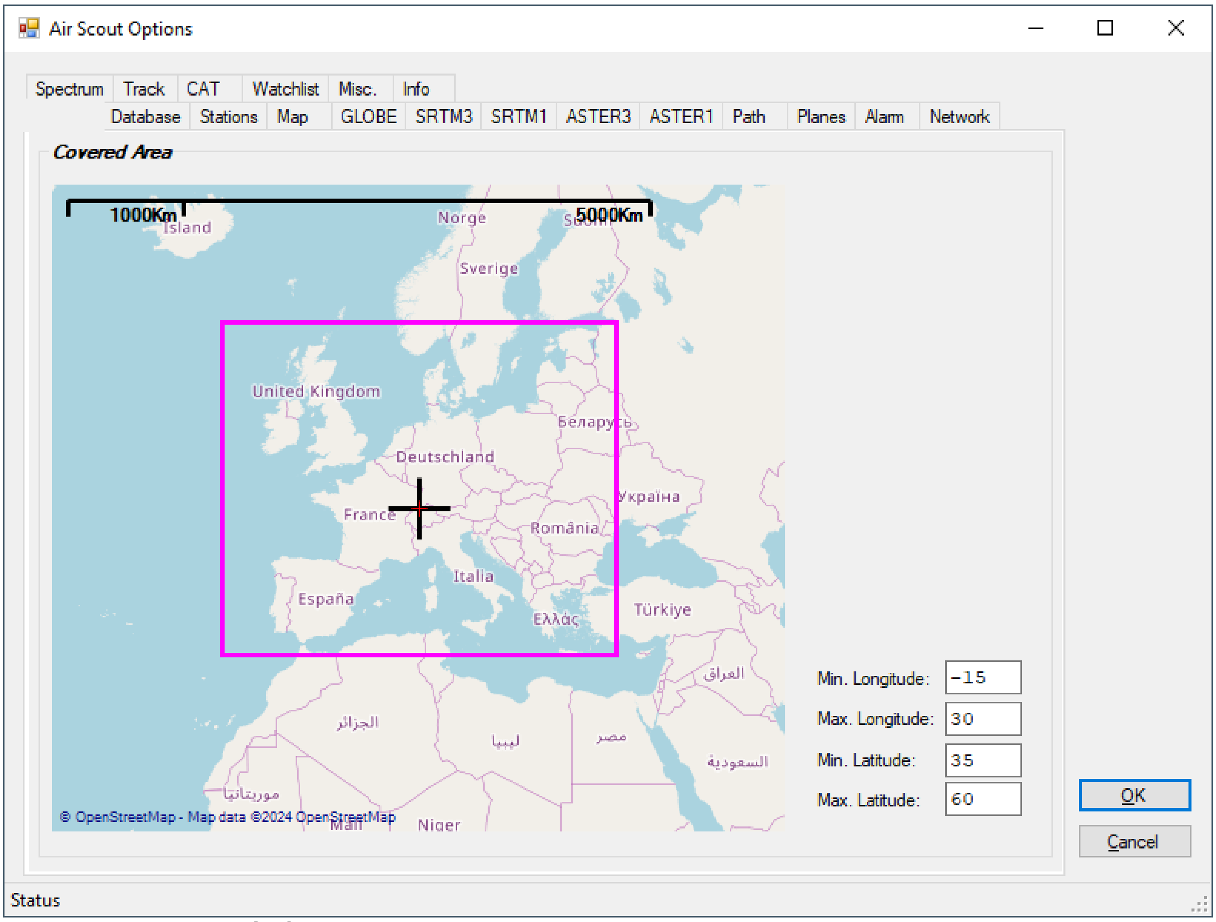


Figure 4-25 General Tab

<u>*Covered Area*</u>

The covered area is representing the "basic scope" of AirScout. Flight traffic is only recorded inside this area. Furthermore the area is used for automatic download of elevation data files. You can adjust the area by entering numeric values in the boxes on the right.

4.8.2. *Tab "Database"*

The database tab is showing basic information of the SQLite databases used since AirScout V1.2.

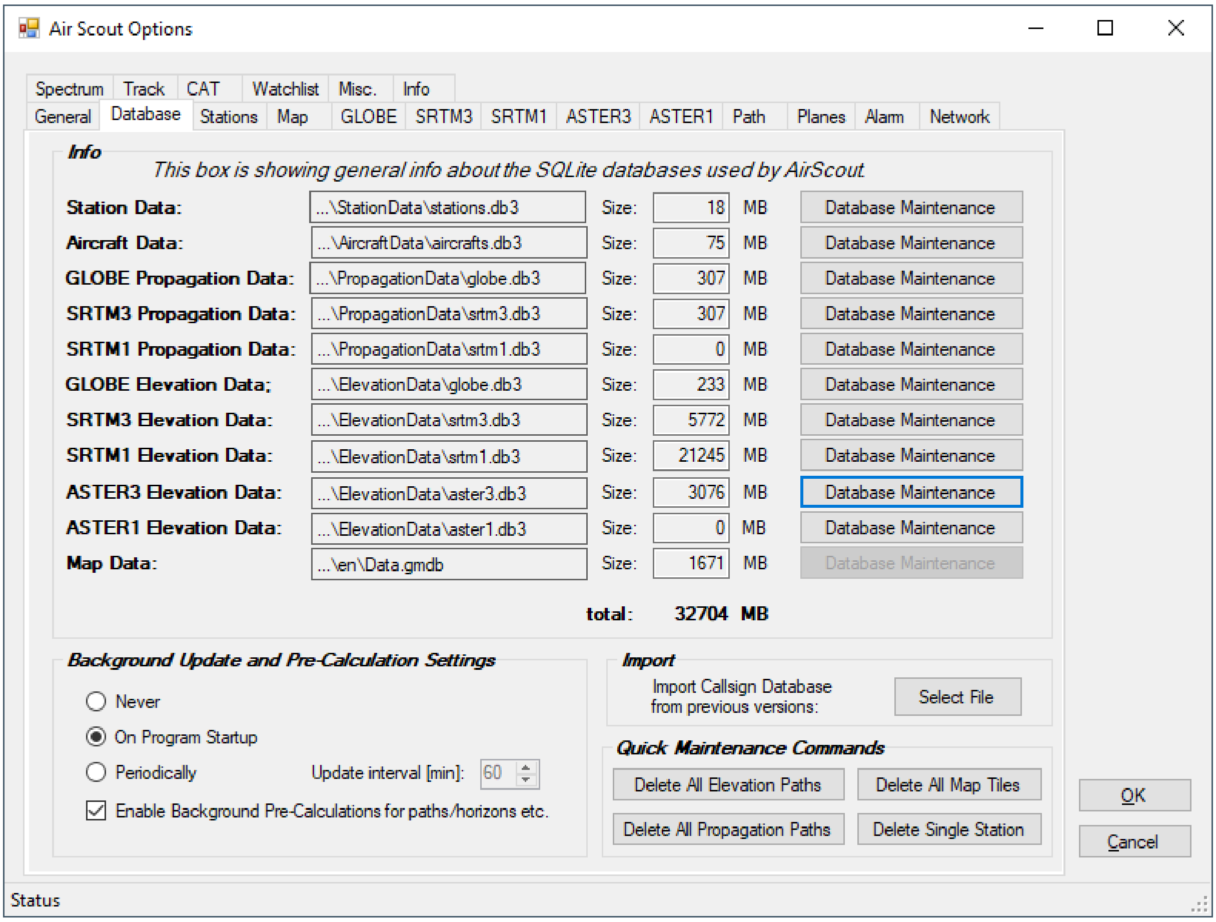


Figure 4-26 Database Tab

The "Database" tab is providing information about databases used in AirScout and maintenance and update functions.

Info

Is showing statistical information and maintenance buttons for all databases used.

Background Update and Pre-Calculation Settings

Controls the background update an pre-calculation procedures.
AirScout can update its databases from web resources on program startup by default. You can change this behavoir here.
Furthermore, AirScout is pre-calculating paths and horizons in the background to allow quick access when needed. You can enable/disable this function here.

Import

You can import callsign database files from early versions of AIrScout here.

Quick Maintenance Commands

Is used to quickly delete some pre-calculated database content to allow recalculation.

4.8.3. *Tab "Stations"*

The “Stations” tab contains information about the two QSO – partners. Normally, it is sufficient to fill in a call sign here. If the call sign is found in database, all other information is updated automatically. Green background indicates that a precise location is available. If the call sign is unknown, the update strategy depends on which field is filled (see below). The description of the fields and buttons is valid for both “My Station” and “DX Station”.

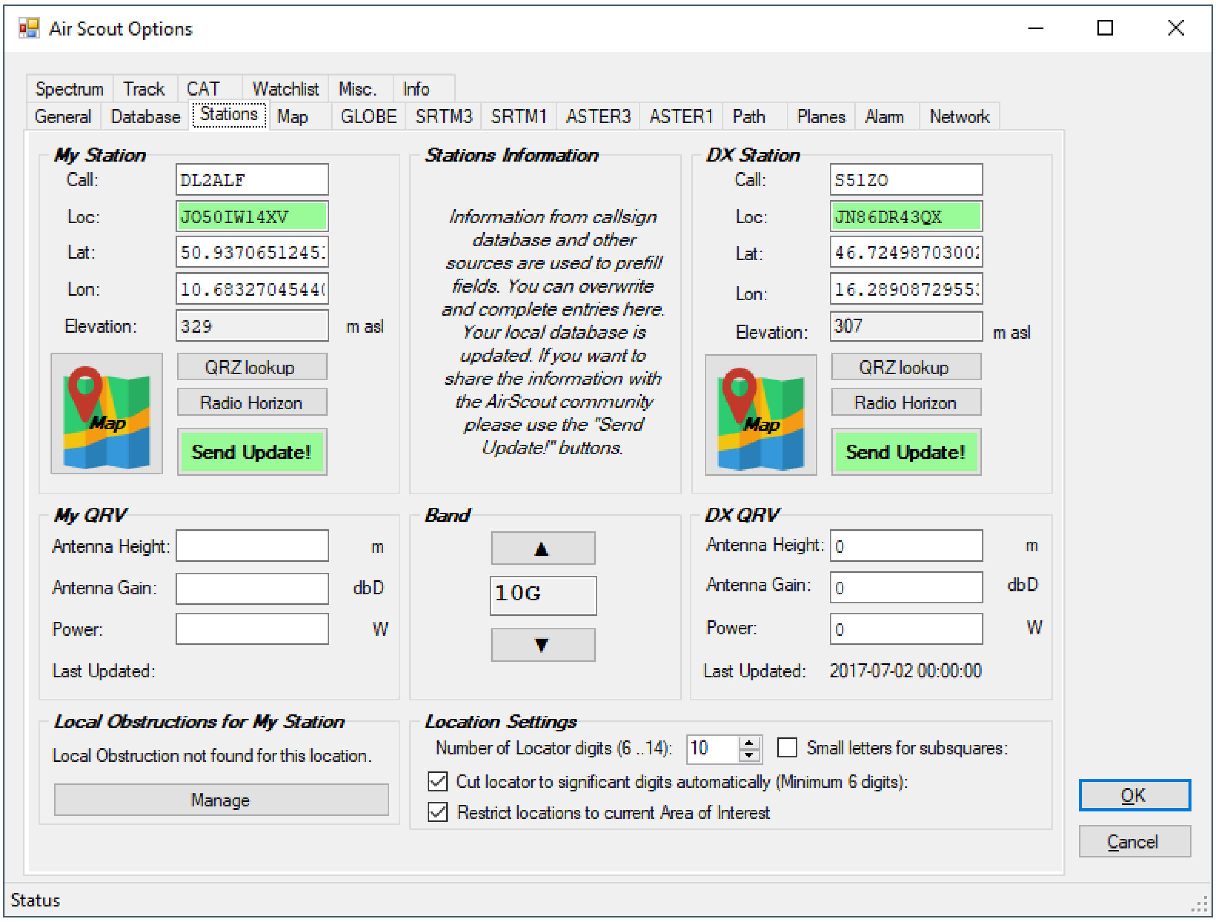


Figure 4-27 Stations Tab

__*Call*__

Enter a valid call sign here. If the call sign is found in database, the lat/lon and locator information is updated automatically. If not, the lat/lon and locator boxes are cleared.

__*Loc*__

Please enter a valid 6digit or more Maidenhead locator here.
The lat/lon boxes are updated with the co-ordinates of the locator midpoint.
The background is set to white on a 6digit locator, which indicates that no precise location is available.
It turns green when a position is more precise than a 6digit locator is entered.

__*Lat*__

Enter valid Latitudes here. The range is:

Format:	floating point, English notation	
Minimum:	-90 deg	= 90°S
Maximum:	+90 deg	= 90°N
Example:	10.6458333	

The locator field is updated automatically. A green background indicates that this is a precise location.

Lon

Enter valid Longitudes here (manually, do not use Cut & Paste). The range is:

Format:	floating point, English notation	
Minimum:	-180 deg	= 180°W
Maximum:	+180 deg	= 180°E
Example:	10.97166667	

The locator field is updated automatically. A green background indicates that this is a precise location.

Elevation

It is showing the elevation of the station in meters above sea level. You cannot enter an elevation here. The value is updated according to the current elevation model whenever the location has changed.

QRZ lookup

The QRZ.com lookup is an easy way to get exact location if there is an entry in their database. To make sure that the location is correct, you must enter a valid locator in the loc box. When the button is pressed AirScout is trying to get the information from the QRZ.com website. If found and the locator from QRZ.com matches with the locator in the locator box, the location information is updated. In all other cases an error message will occur.

Map

As an alternative, you can enter the location on a map by dragging the needle to the exact position.

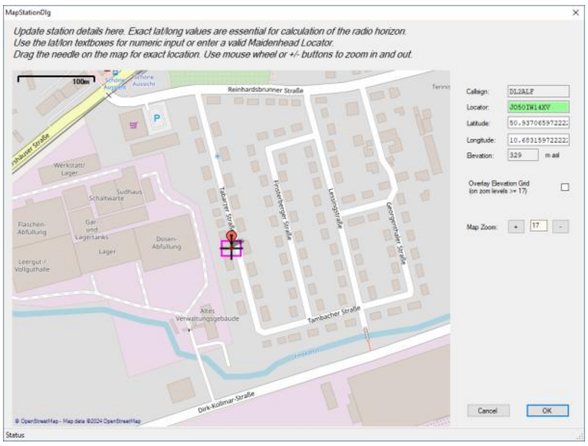


Figure 4-28 Station Location Map

Radio Horizon

The radio horizon is calculated from the current elevation model in 1° steps around the station. The calculation is strongly influenced by the selected elevation model and the precise location. Therefore it does not make sense to use it with the locator midpoint only.
To get the diagrams filled, choose the type of diagram (Cartesian/Polar) and press the calculate button. Furthermore, the horizon distance can be plotted on a map.

The **Minimum Elevation Angle** is showing the minimum usable elevation in each direction around the station. The value should be 0 for an unobstructed view and >0 if any obstruction is in sight.

The **Horizon Distance** is showing the distance to the according obstruction. For an unobstructed view the distance should be equivalent to the radio horizon as a theoretical maximum and should be nearer if any obstruction is in sight.

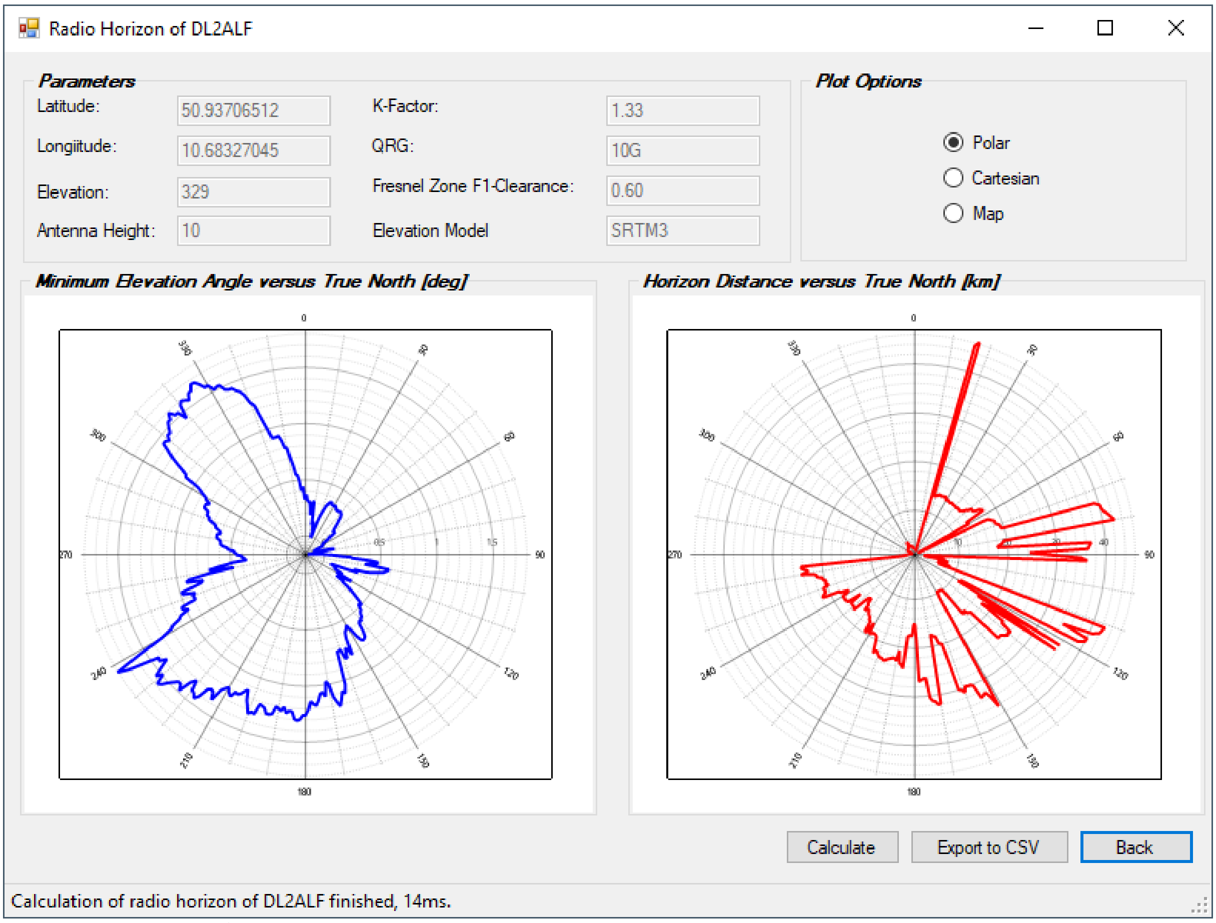


Figure 4-29 Station's Radio Horizon

Once calculated you can export the calculation to a CSV file by pressing the "Export to CSV" button.

Send Update!

Use this button if you are sure about the entered information and to send the data to AirScout's web repository. After validation the data will be part of the web repository for common use.

The QRV information can be different and therefore is available per band. So complete the information as follows:

If the station is not QRV on this band:	*leave all fields empty*
If the station is QRV on this band but with unknown setup:	*fill in Zero on unknown fields*
If the station is QRV on this band with known setup:	*fill in values on all fields*

If an information is not available for a distinct station, AirScout is using default values per band for calculations.

Antenna Height

Enter the antenna height of the station in meters here. Do not correct any elevation incorrectness with the antenna height as the resulting elevation + antenna height does not fit the path calculation.

Antenna Gain

Enter the antenna gain of the station in dBD here. This values is only used to estimate the beam width in horizontal plane which of course has its limitations. Fill in as follows:

single antenna, parabolic dishes:	*enter gain of single antenna*
vertically stocked:	*enter gain of a single antenna*
horizontally stocked:	*enter resulting gain of whole array*
H- arrays:	*enter resulting gain of all antennas stacked horizontal*

Power

Not used in AirScout so far.

Local Obstructions

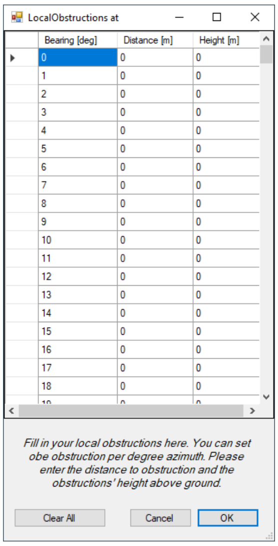


Figure 4-30 Local Obstructions

Can be used to define local obstructions not covered by the Digital Elevation Model, e.g. buildings.

Enter bearing, distance and height here and AirScout will use the values in the path calculations instead of the values from DEM.

Locator Settings

You can maintain some general Maidenhead Locator settings here.

1. The maximum number of digits shown on the map, textboxes etc. and allowed for input.
   Possible values: 6digits ... 14digits, default = 10digits
2. Cut locator automatically to the number of relevant digits by checking the midpoint of a given locator accuracy.
   Example: if a location is only known by a 6digit Maidenhead locator, lat/lon values are set to the midpoint when created. AirScout is detecting this and will show only 6digits even if > 6digits are chosen in the settings.
3. Use small letters for subsquares will result in showing "JO50iw14xv" instead of "JO50IW14XV".
4. Restrict locations to the current Area of Interest = Coverd Area. Enabled by default, this option reduces the amount of background calculations to potential QSO partners.

4.8.4. Tab "Map"

The “Map” tab contains basic settings for map source and map information showing on the main window.

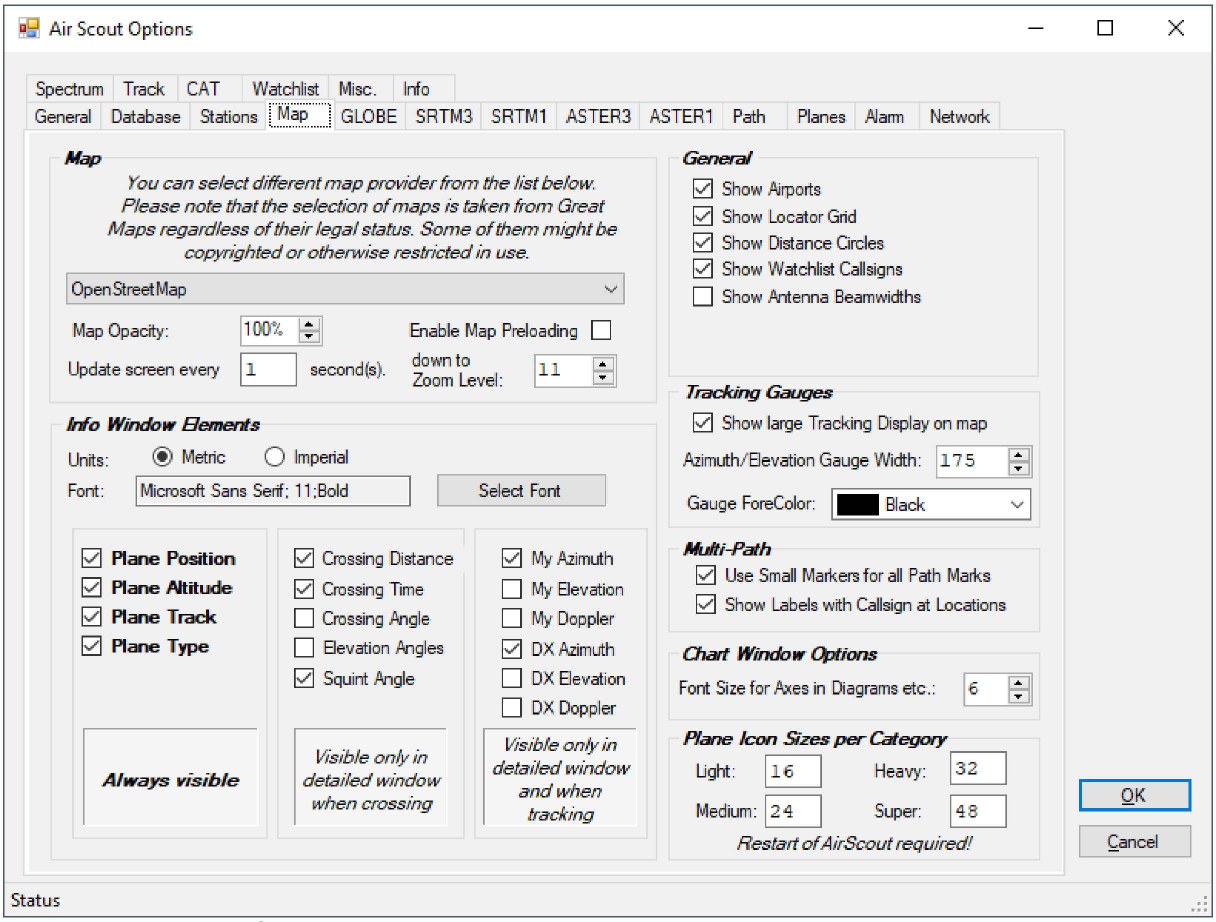


Figure 4-31 Map Tab

Map

You can select different map sources here. This is a functionality of Great Maps library and not a part of AirScout itself. The default map source is OpenStreetMap. Some of the map sources may not be available for your area. Some of the map sources may be copyrighted or otherwise restricted in use. If you are choosing another map source than OpenStreetMap you were asked to agree with the terms of use first.

CAUTION: The use of all map sources is on your own risk. The author will not be responsible for any copyright violation in any case.

Map Opacity:	sets the map opacity (helps on map with intense colors)
Screen Update:	sets the update cycle for map and all content
Map Preloading:	enables/disables map tile preloading and storing down to a distinct zoom level

General

You can select general options here, mostly which objects/helpers are shown on the map.

Show Airports:	enables/disables airport layer on the map
Show Locator Grid:	draws/hides a locator grid overlay
Show Distance Circles:	draws/hides distance circles overlay (100km steps)
Show Watchlist Locations:	shows/hides markers form all locations in the watchlist
Show Antenna Beamwidths:	shows/hides antenna beamwidth overlay

Tracking Gauges

You can select general options for tracking gauges when in tracking mode.

Show Large Gauges:	shows/hides an overlay with large gauges
Gauge Width:	defines the width of the gauges
Gauges Fore Color:	defines the fore color of the gauges

Info Window Elements

You can select the font family and size for the Info Window. Furthermore, you can toggle between metric or imperial units.

You can choose the basic elements shown in the Info Window tooltips. The settings are split depending on category and tracking mode. They should be self-descriptive.

Multi-Path

You can select some options when in Multi-Path mode:

Use Small Markers:	reduces the size of markers on the map for better view
Labels With Callsigns:	shows labels with callsigns on all markers

Chart Window Options

You can choose the font size of axes on all diagrams.

Plane Icon Sizes per Category

You can choose the icon sizes per plane category here. As the icons were calculated only at program start up a restart of AirScout is necessary.

4.8.5. Tabs "GLOBE", "SRTM3", "SRTM1", "ASTER3", "ASTER1"

All those tabs are showing similar information about the available Digital Elevation Models (DEMs). Activate the use of a distinct DEM by checking the appropriate checkbox. You can choose whether to keep the downloaded tiles in local storage to save download time. A map is showing you the general coverage, tiles available on the web, tiles needed for coverage of your area and tiles found on your local store. Tiles in AirScout are handled on a 6digit Maidenhead locator basis. The color code is as follows:

	Tile not available
	Tile available but not in local database
	Tile available and in local database

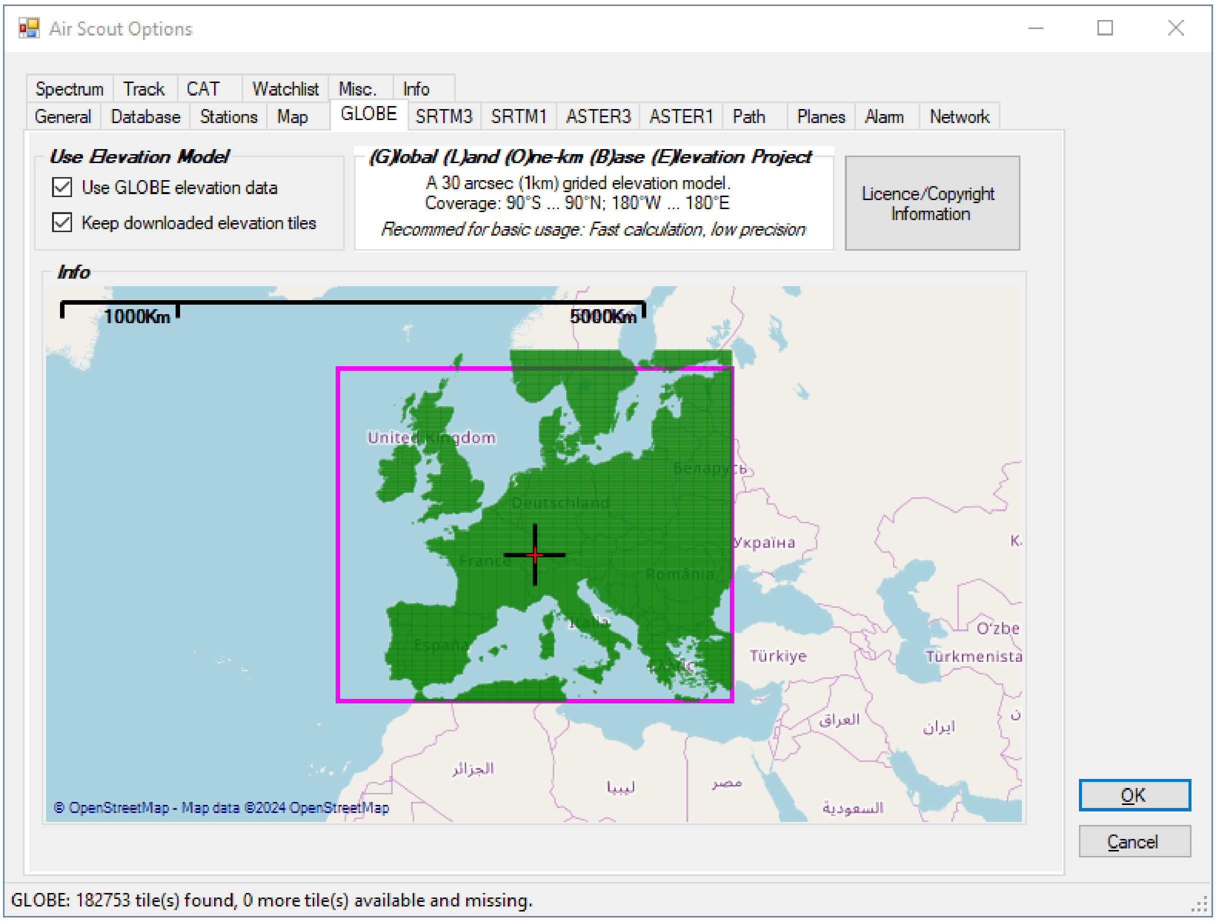


Figure 4-32 GLOBE Tab

4.8.6. Tab "Path"

The “Path” tab is showing currently selected path parameters. These can be set for each band separately. There are:

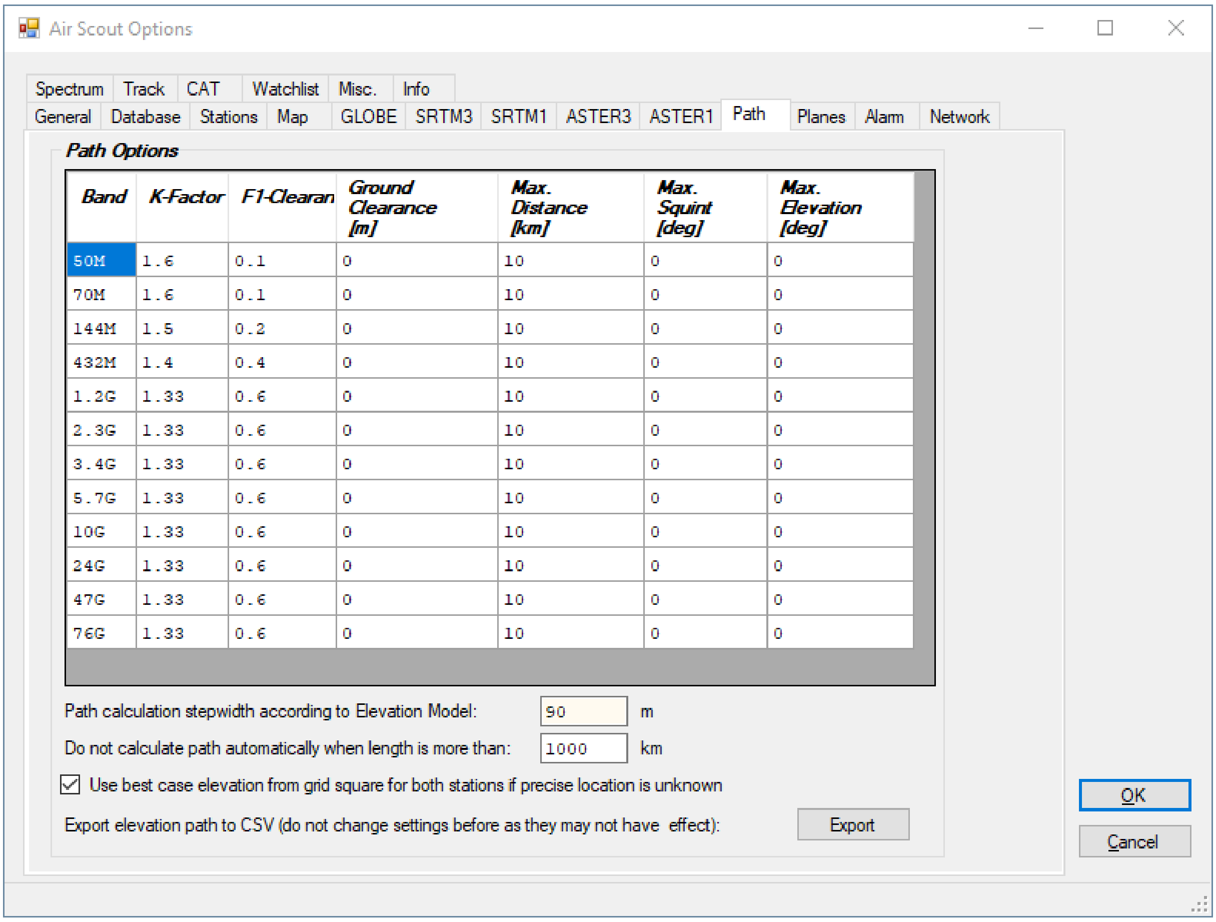


Figure 4-33 Path Tab

Path Options

Per band available set of path parameters as described in chapter "Basic Parameters for Path Calculations".

So far, only K-Factor and F1-Clearance are used for calculations.
Be careful! Use EN notation only. There is no plausibility check for all parameters right now.

Calculation Step Width

You can see the calculation step width for the path calculation here according to the chosen elevation model.

Use Best Case Elevation

With SRTM3 or SRTM1 elevation data enabled the use of precise locations is strongly recommended. If only 6digit Maidenhead locators are available the software will set the location to the centre of that grid square automatically. The dynamic range of SRTM data is very high so that the calculated centred location can be very near to an obstruction, especially in a hilly environment. This will result in a completely wrong path calculation!

The option "Use best case elevation" will use the highest elevation within the given grid square instead of the centre automatically if a precise location is not available. The path calculation is too optimistic then in most cases but it helps to avoid false results.
The use of the much smoother GLOBE elevation model does not course such severe errors.

Export to CSV

You can export the calculated path to CSV for further considerations. The structure is described in the Appendix.

4.8.7. Tab "Planes"

The “Planes” tab is showing basic plane feeds and plane database options.

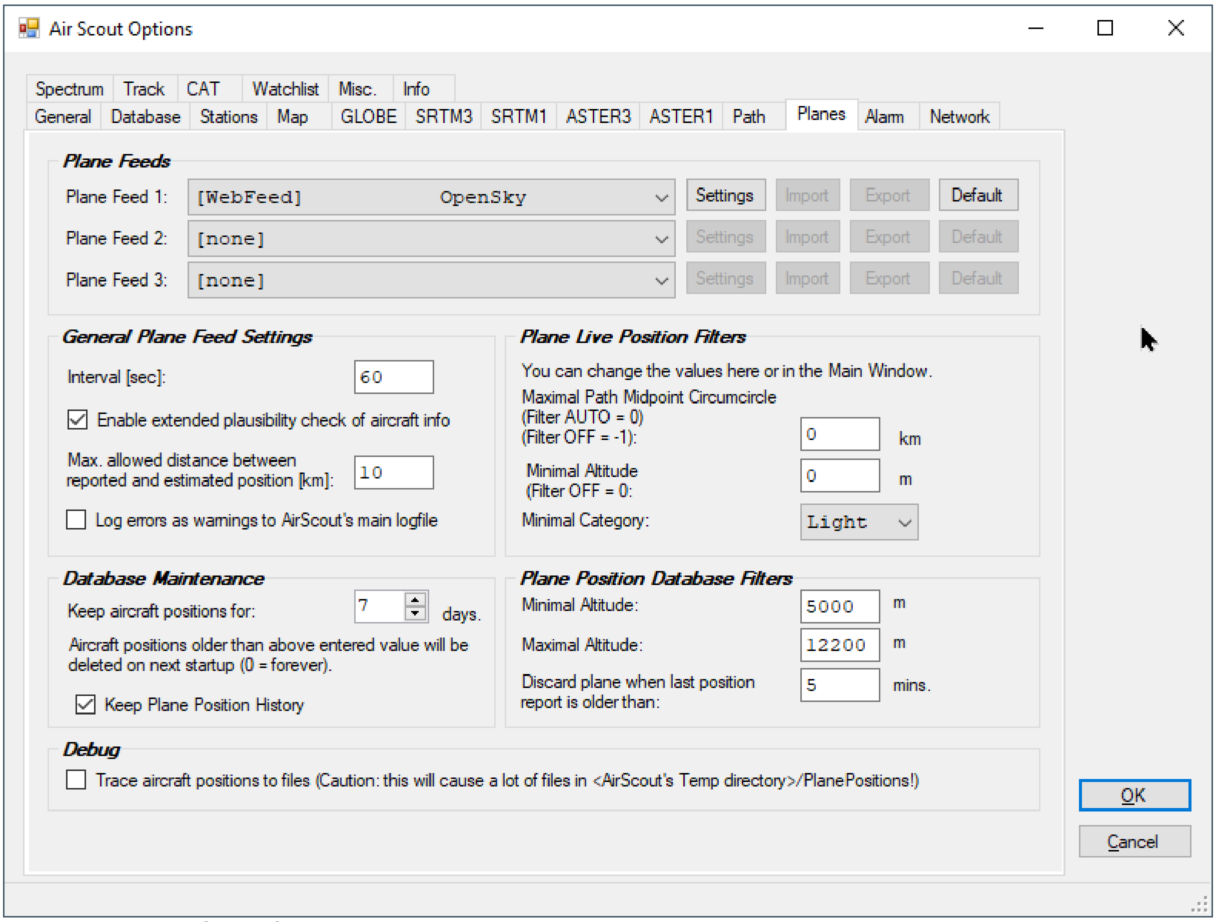


Figure 4-34 Planes Tab

Plane Feeds

You can enable/disable plane feeds here. Furthermore, multiple Internet feeds from different URLs with different formats can be activated simultaneously. Some feeds have setting to change it to special needs. Do not change them unless you are exactly knowing, what you are doing. Special keywords are supported in the URLs to fit the request to the covered area. See appendix for details. Some feeds can export/import their settings. You can use this to interchange your settings with others.

Normally the simultaneous use of more than one stream is producing best results. Anyway, sometimes inconsistencies between two feed may occur and you will see doubled plane icons with slightly different positions or call signs. In this case or to reduce network traffic it is recommended to disable multiple feeds. One single feed will produce a traffic of about 2MB/min.

Plane Live Positions Filter

You can enter live filter options here for showing planes on the map. The filtering will apply after the database filter and is only for display. The settings are similar to these on the program's main page.

Maximum Path Midpoint Circumcircle

Is used to define the maximum distance away from the path a plane can have to show it on the map. The distance is calculated from the midpoint of the current path. Except entering a distinct value into the box you have two more options:

Value = 0 (Default):	Planes are shown in a radius of half the distance between both QSO - partners around the midpoint
Value = -1:	All planes inside the covered area are shown regardless of the radius

Minimal Altitude

Is used to define the minimum altitude plane must have to show it on the map. Except entering a distinct value into the box you have one more option:

Value = 0 (Default):	All planes are shown regardless of their altitude

Minimal Category

Is used to define the minimum category plane must have to show it on the map. Default is "LOW" means all planes are shown on the map.

Plane Positions Database Filter

You can enter filter options here for position database. The filtering will apply before the plane positions are stored in the database. All planes sorted out by this filter are lost and cannot be restored later when changing the filter settings. Using this filters will keep the database small. Some web feeds allow to set the filter as parameters on the request.

Minimal Altitude

Is used to define the minimum altitude plane must have to store it in the database. Default is 5000m.

Maximal Altitude

Is used to define the maximum altitude plane can have to store it in the database. Default is 12200m.

Minimal Category

Is used to define the minimum category plane can have to show it on the map. Default is "LOW" means all planes are shown on the map.

Discard plane when last position report is older than

You can enter a plane positions “Time To Live” here. If the last position report is too old to estimate a current position the plane object is discarded and not used anymore. Set to 5 minutes by Default it is sometimes recommended to extend the time span. This will cause bigger estimation errors but is necessary when plane position reporting is delayed (e.g. by federal law in the U.S.).

General Plane Feed Settings

You can enter some general plane feed settings here.

Interval

Defines the cycle when AirScout is fetching plane data from the plane feed background threads.

Extended Plausibility check

You can enable/disable an extended plausibility check for plane feed positions. If enabled, AirScout estimates the plane position based on last messages and compares it with the newly fetched position. New position is discarded when difference in position is bigger than the maximum value given here. You can log these errors to AirScout's main log.

Database Maintenance

You can store positions in a special database for history analysis. In order to avoid too many entries, an automatic cleaning can be set up here.

Debug

You can store positions on a per-plane-basis in files for debug purpose. This will help to find errors in position reporting and estimation but will cause a lot of files in the Temp directory.

4.8.8. Tab "Alarm"

The “Alarm” tab is showing basic alarm settings. To generate an alarm, the software must be in “Run” mode and at last one action must be defined in “Alarm Settings”. It is not possible to watch multiple paths to different QSO partners in background.

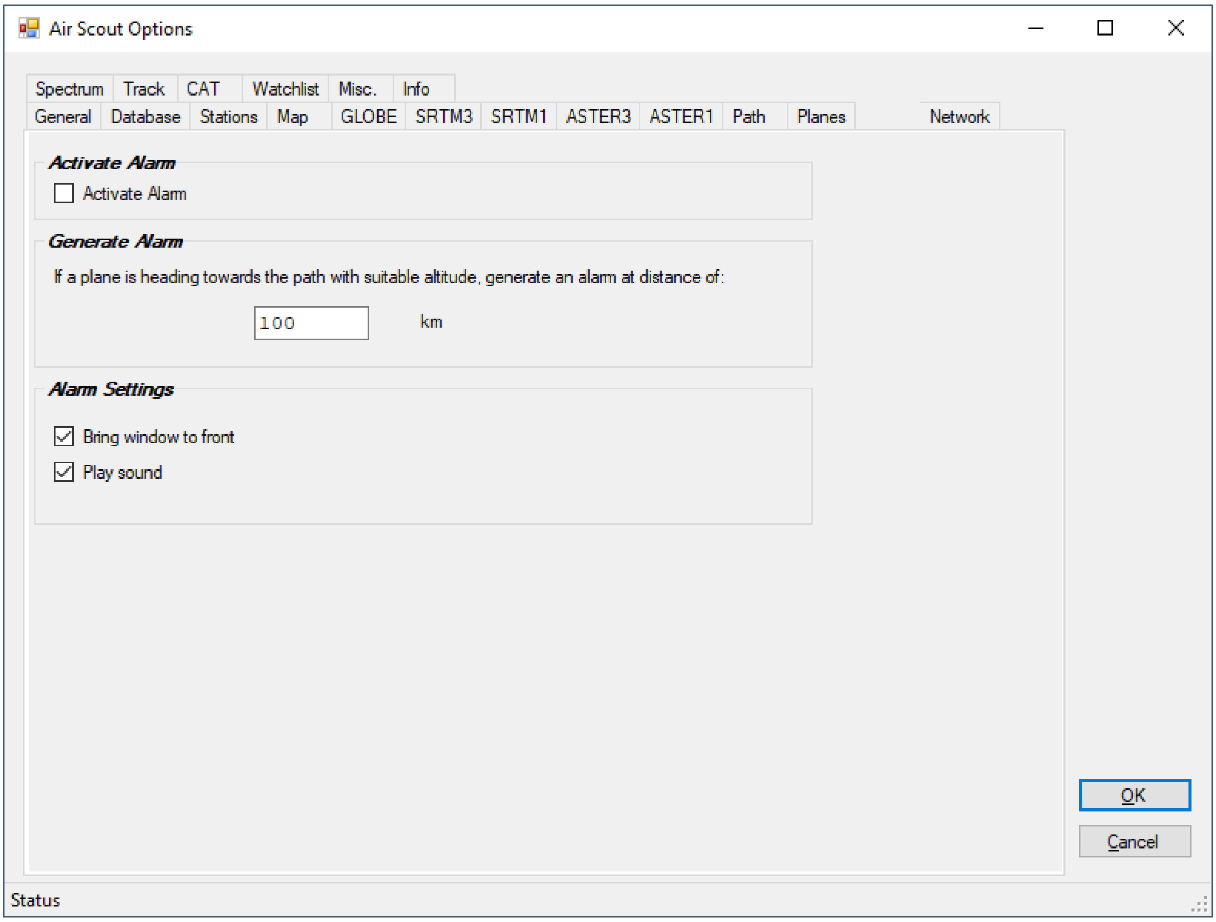


Figure 4-35 Alarm Tab

Activate Alarm

You can enable/disable an alarm here. The same function is available on the main program window.

Generate Alarm

You can set a minimum distance an aircraft must have to generate an alarm. An alarm is generated if one aircraft is within the given range. There is no acknowledgement to the alarm; simply switch it off. AirScout cannot handle multiple alarms or alarm on a single aircraft so far.

Alarm Settings

You can set the behaviour of the software in case of an active alarm. You can bring the program's main window to front (may cause some issues under Windows XP) or play a system sound in an endless loop.

4.8.9. Tab "Network"

The "Network" is showing basic network settings for use with wtKST and other software.

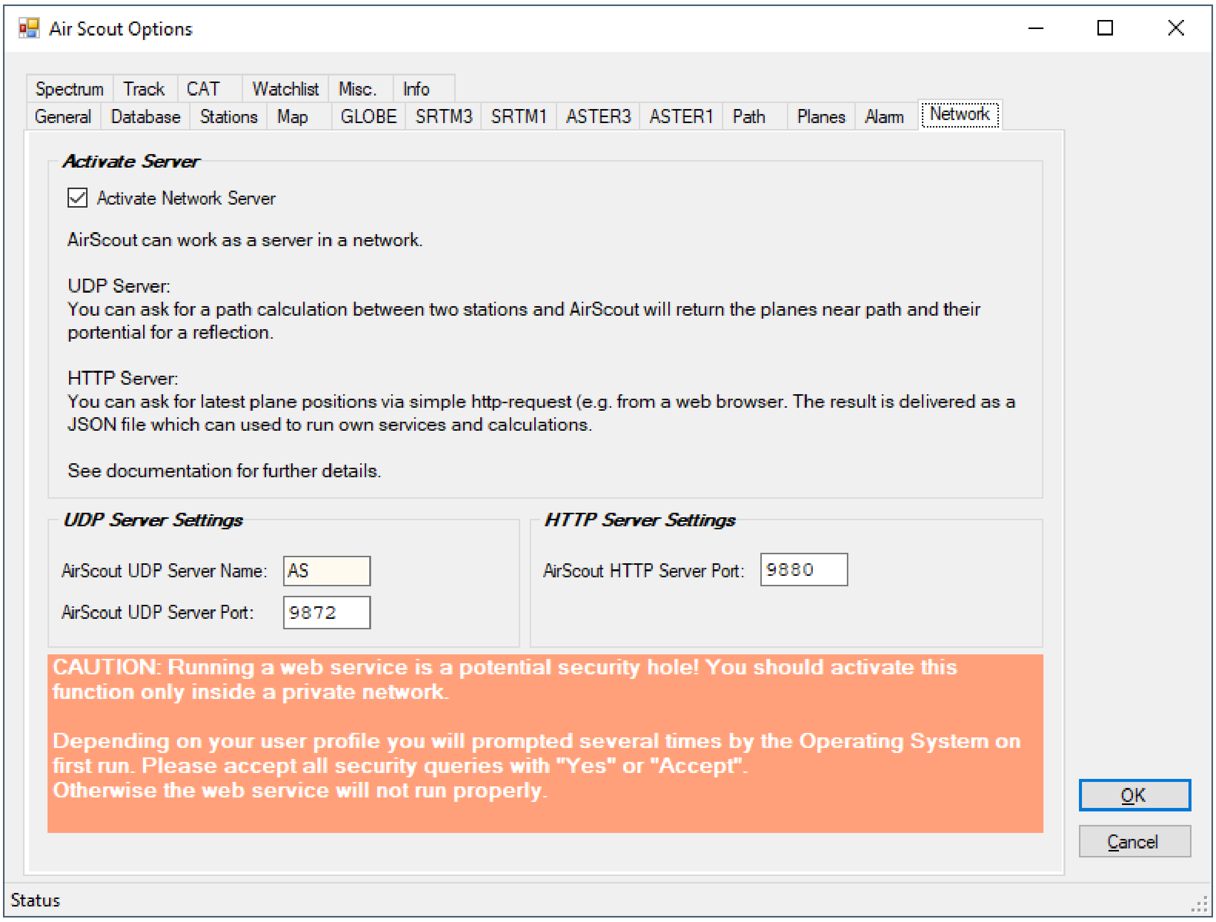


Figure 4-36 Network Tab

Activate Server

This allows AirScout to act as a server in the network.

UDP Server Settings

The UDP communication is a broadcast similar to the Win-Test contest logger communication (see Appendix for protocol description). You can have more than one server/client active in the same network.

AirScout UDP Server Name

Set a unique server name here for your AirScout server. The name is used to identify different servers in the network. The name must match with all clients which want to communicate with a server.

AirScout UDP Server Port

You can set the UDP port for broadcasting here. The port number must match with all participants in the network and must not collide with any other network communication. Do not change it unless you exactly know what you are doing here.

HTTP Server Settings

The HTTP communication is used to provide aircraft positions via JSON file on request. This is similar to plane feeds on web. Use "http://<IP address of AirScout server>/planes.json" to get the JSON file.

4.8.10. Tab "Spectrum Lab"

The "Spectrum Lab" tab is showing basic settings to interact with [Spectrum Lab Software by DL4YHF](). Starting from V1.0.0.0 AirScout can capture a NF spectrum from a receiver with the help of Spectrum Lab.

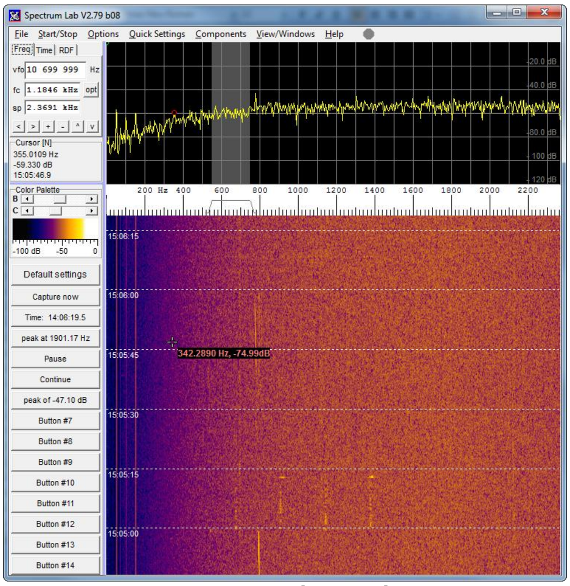


Figure 4-37 Spectrum Lab Screenshot

AirScout can get FFT data from Spectrum Lab via Network using basically the same technology as position data (JSON files). You must have Spectrum Lab software running using <u>default settings</u>. To communicate with AirScout you have to activate the http – server functionality in Spectrum Lab options. There is not much error checking right now, so don't wonder if you get unexpected results.

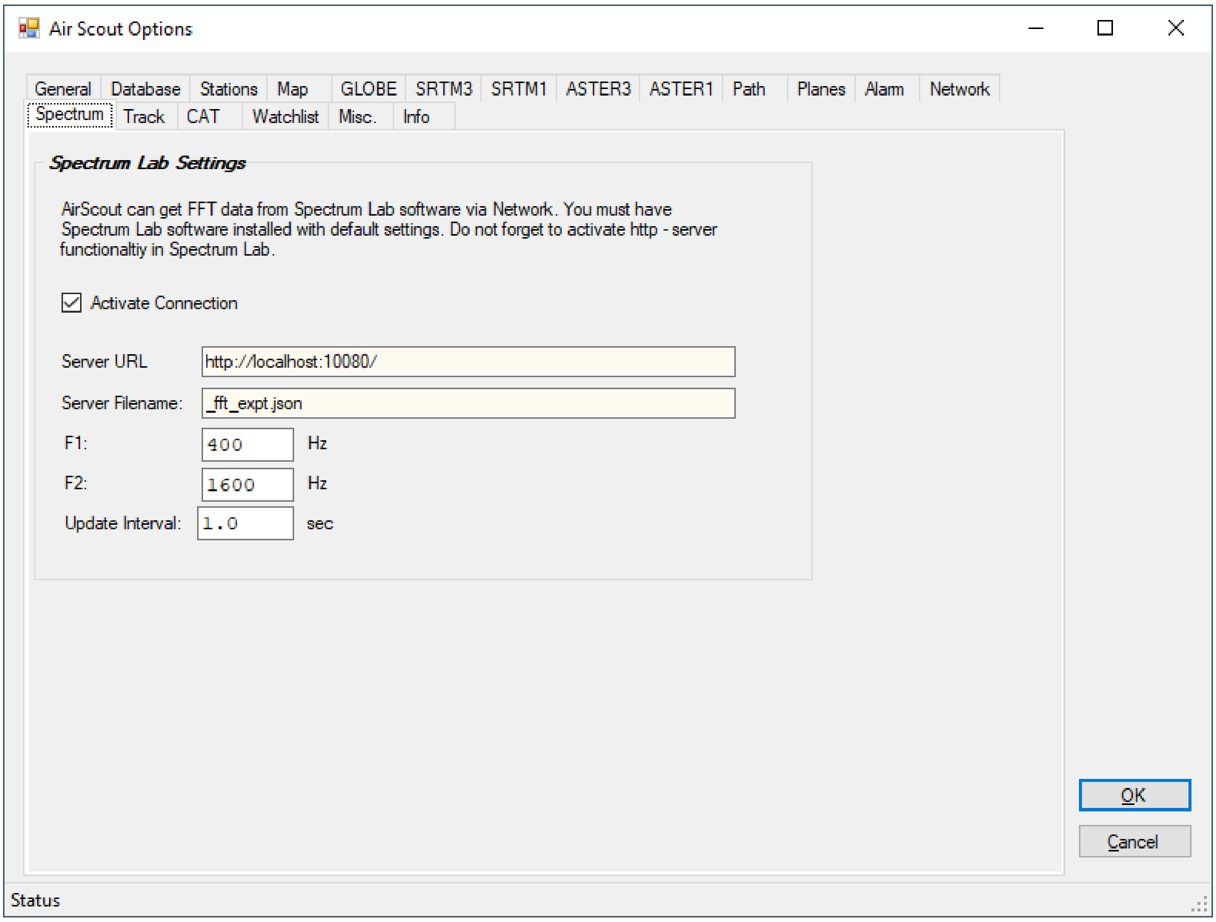


Figure 4-38 Spectrum Lab Tab

Activate Connection

You can activate the connection with Spectrum Lab here.

Server URL

This is the Spectrum Lab server URL which is "localhost" by Default.

Server Filename

This is the FileName for the FFT data from Spectrum Lab. Do not change this unless you know what you are doing.

F1

This is the lower bound of the recorded frequency range

Minimum: 0 Hz
Maximum: 3000 Hz

F2

This is the upper bound of the recorded frequency range.

Minimum: 0 Hz

Maximum:	3000 Hz

<u>Update Interval</u>

This is the update interval AirScout is asking for new FFT data (Default = 1sec).

Minimum:	1 sec
Maximum:	60 sec

4.8.11. Tab "Track"

AirScout has an antenna tracking and Doppler correction function. When activated you can track a single plane which is selected by left clicking on it. Multiple selections are not allowed. AirScout is then showing two instruments on the lower right corner of the map with current azimuth/elevation values. There are several tracking outputs which can be used simultaneously.

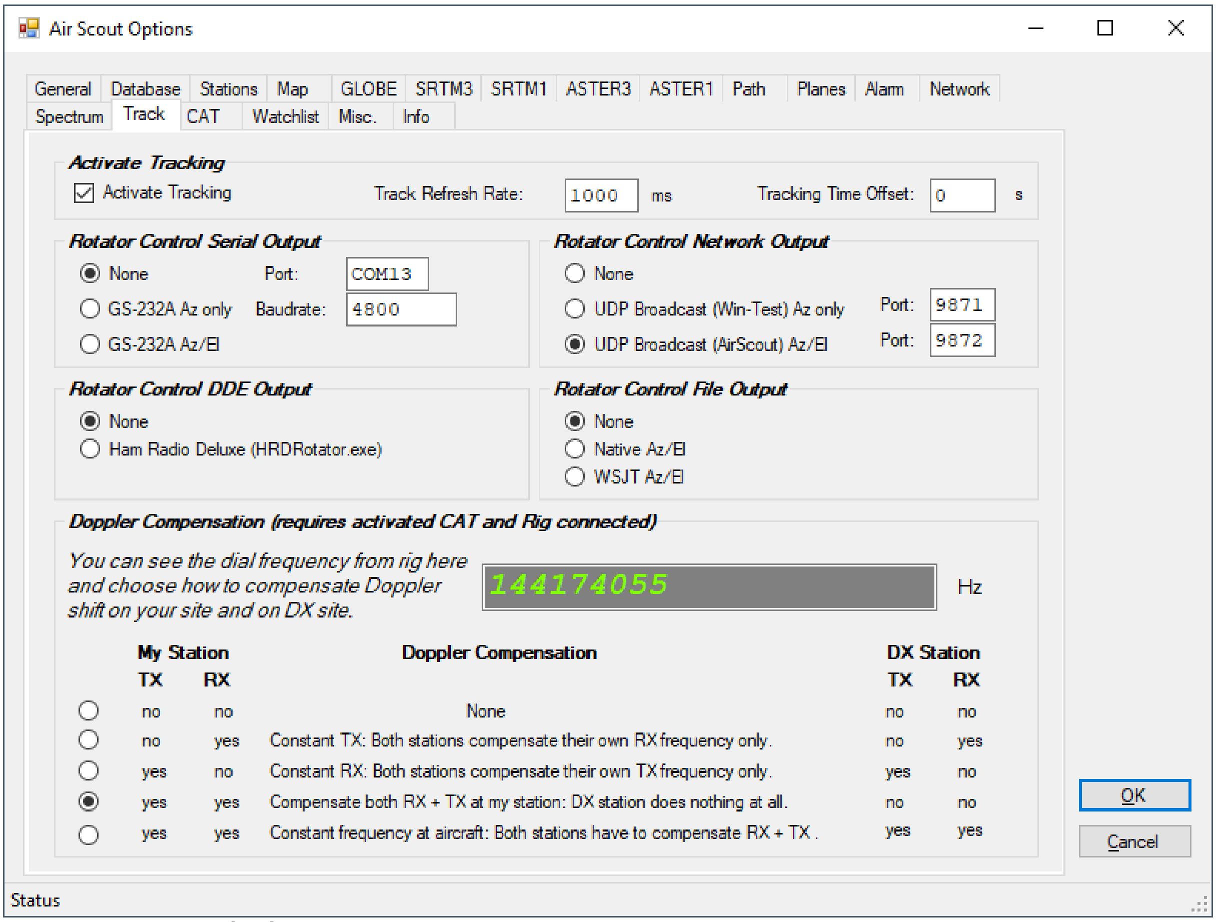


Figure 4-39	Track Tab

<u>Activate Antenna Tracking</u>

Activates the antenna tracking. You can select only one plane on the map when activated.

<u>Track Refresh Rate</u>

Set the refresh rate for tracking here. It is not recommend to go below 1000 ms.

Tracking Offset Time

In case your antenna tracking is not as fast as needed in reality you can experiment with a tracking offset time which looks a bit forward in the future to give the system more time for beaming. As this is very dynamic and depends on several variables this is only for playing around with.

Rotator Control Serial Output

Generates a serial output for direct control of a Yaesu/Kenpro rotator GS-232A. No other rotator type is supported so far.

Rotator Control Network Output

Generates a UDP output to interact with Win-Test wtRotator software (Az only).
Second option is a AirScout protocol with Az/El values for special purpose.

Rotator Control DDE Output

Uses a DDE communictaion with Ham Radio Deluxe rotator control software (Az/El possible).

Rotator Control File Output

Generates a simple file output periodically which can be read by any other program. Second option is file output in WSJT style which can be read by several rotator control software.

Doppler Compensation

Allows to correct frequency drift caused by Doppler effect when tracking. Requires a rig connected via CAT. The current rig frequency is showing as information here.

CAUTION! The amount of Doppler correction is always calculated according to AirScout's current band selection. No need to see the real frequency in CAT response.

You have to choose a Doppler compensation strategy here:

None:	no correction at all
Constant TX:	both stations compensate their RX only
Constant RX:	both stations compensate their TX only
Both RX/TX:	you compensate both RX + TX. DX station does nothing.
Constant frequency at aircraft:	both stations compensate both RX + TX

It is recommended that you compensate both RX + TX in normal operation if you can and let the DX station do nothing. Another way is to agree on compensation via chat (like for EME).

Doppler correction can be used without antenna tracking.

4.8.12. *Tab "CAT"*

Is used to control the CAT settings. You can activate/deactivate the CAT interface here. The refresh time is recommended to remain on the default of 500 ms.

You can choose between two CAT engines:

Connection to an OmniRig instance

Works on Windows only and needs V1.19 or V2.x of OmniRig installed. In this case you can select one of Rig1, Rig2 or (Rig3, Rig4) of virtual OmniRigs and use it in co-operation with other programs. This is the recommended setting because there is no need to switch over the rig or to stop other programs when using the CAT in AirScout.

Direct control with Scoutbase.CAT

Works stand alone with AirScout and controls the rig directly via COM port. Works on both Windows and Linux/Mono but need exclusive port access. In this case you must select one of the supported rigs from the dropdown and set the correct COM port parameters.

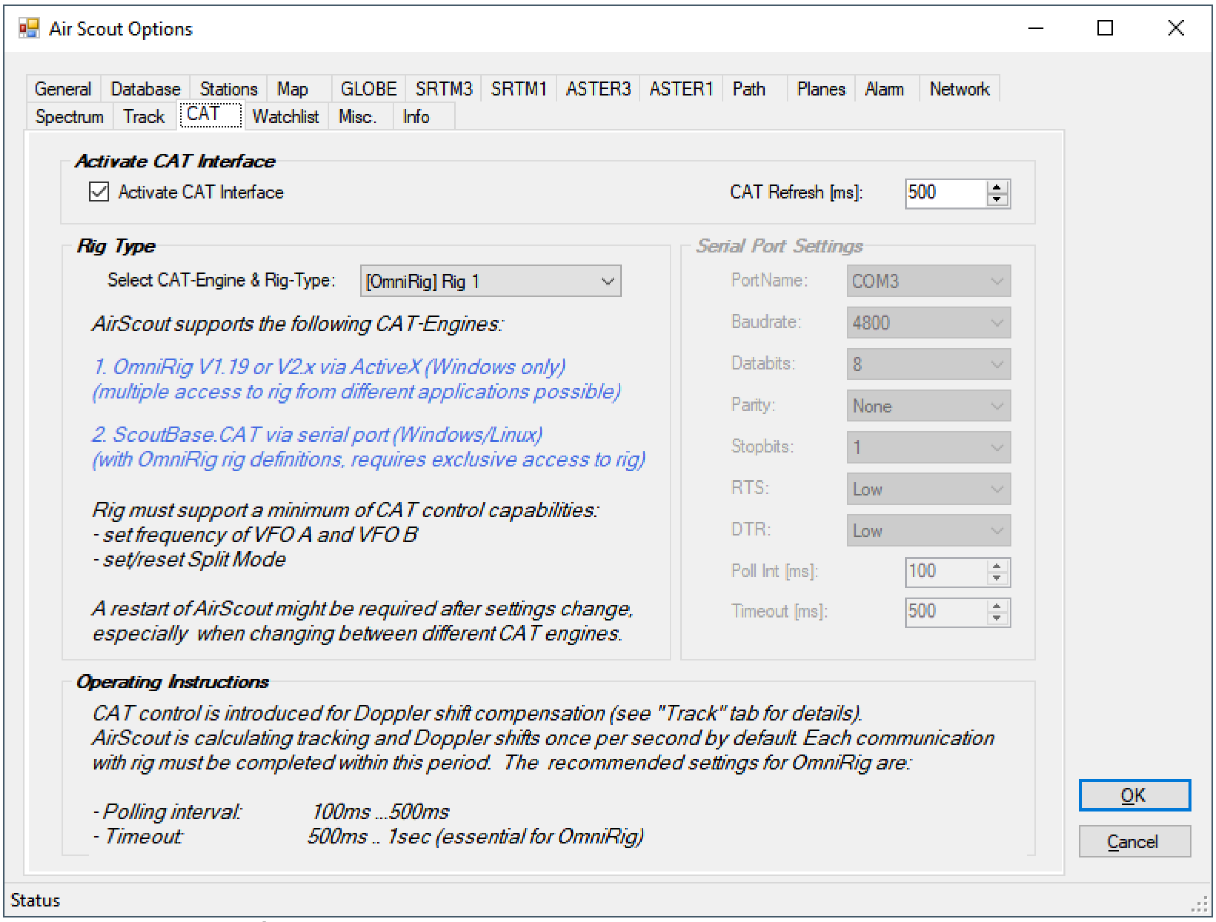


Figure 4-40 CAT Tab

4.8.13. *Tab "Watchlist"*

The call watch list keeps the call signs recently entered in the DXCall boxes or of your special interest. You can set the maximum number of entries (default = 1000) and you can manage the list by pressing the "Manage Watchlist" button. Starting with AirScout V1.2 and wtKST V3.1 you can synchronize the watch list with the list of ON4KST users (the chat you are currently logged in). Network functions must be activated to use this feature. Watch list calls are shown on the map, clicking on it changes the path to this call sign.

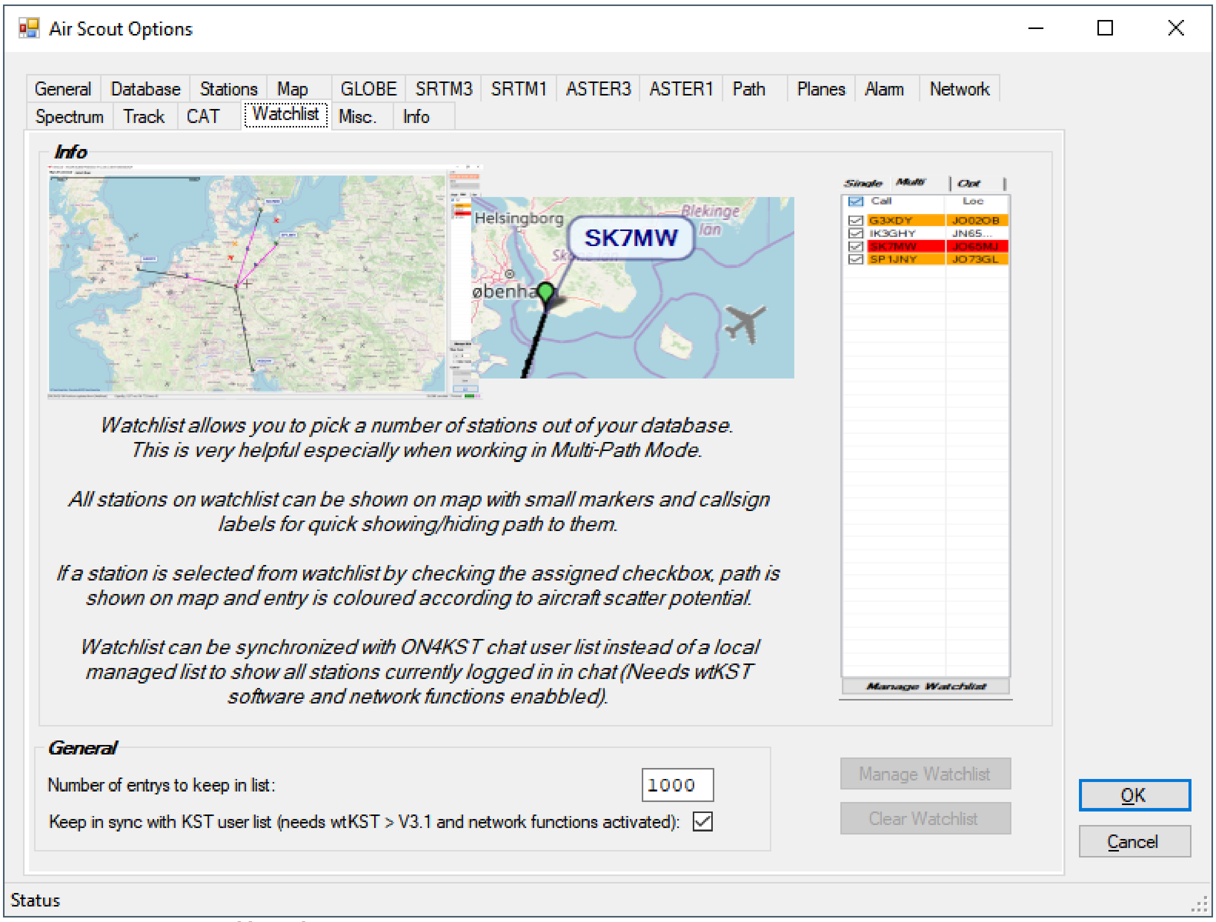


Figure 4-41 Watchlist Tab

4.8.14. *Tab "Misc"*

Provides miscellaneous options, so far only:

- buttons to show some AirScout directories in Explorer
- a Donate button
- enable/disable notifications from www.airscout.eu

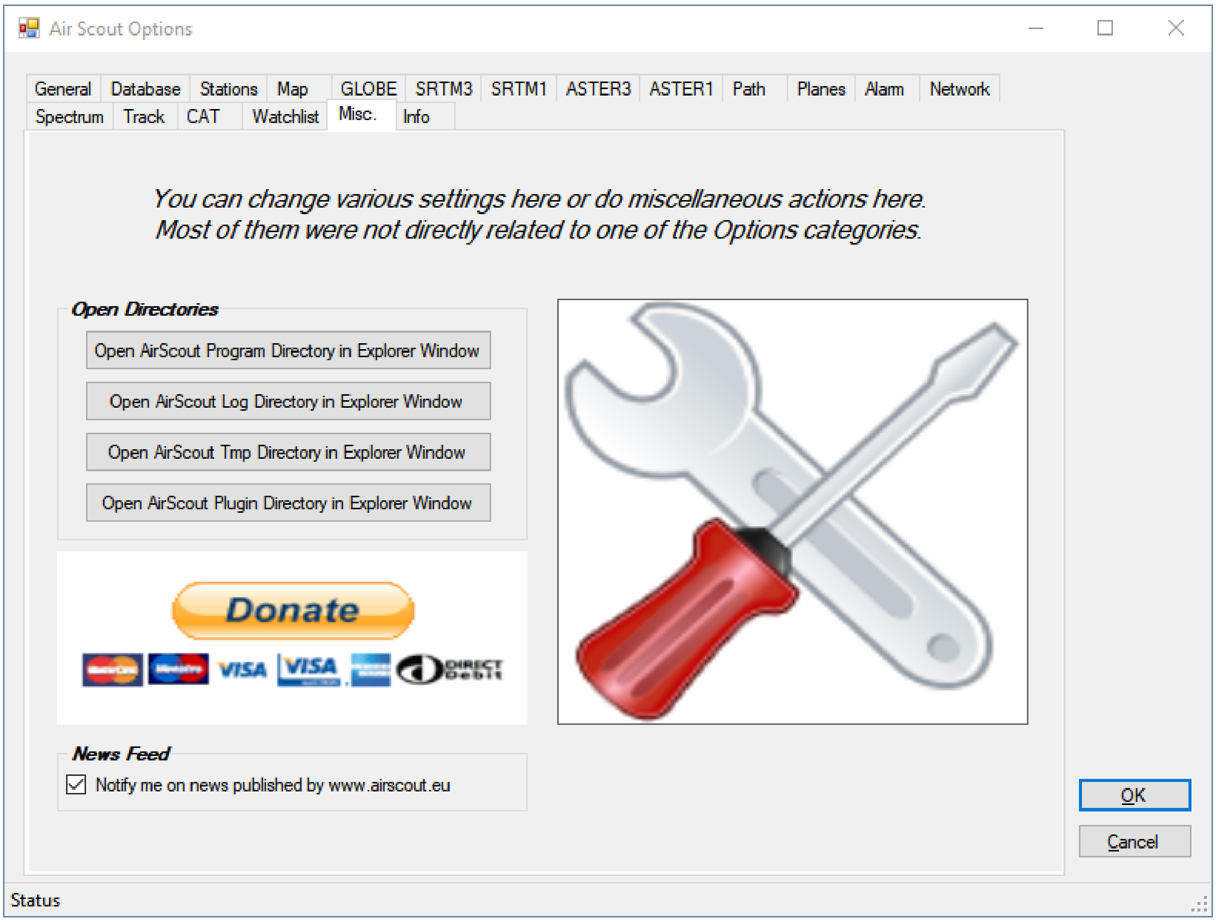


Figure 4-42 Misc Tab

4.8.15. *Tab "Info"*

The "Info" tab is showing program and version information, copyrights and acknowledgements.

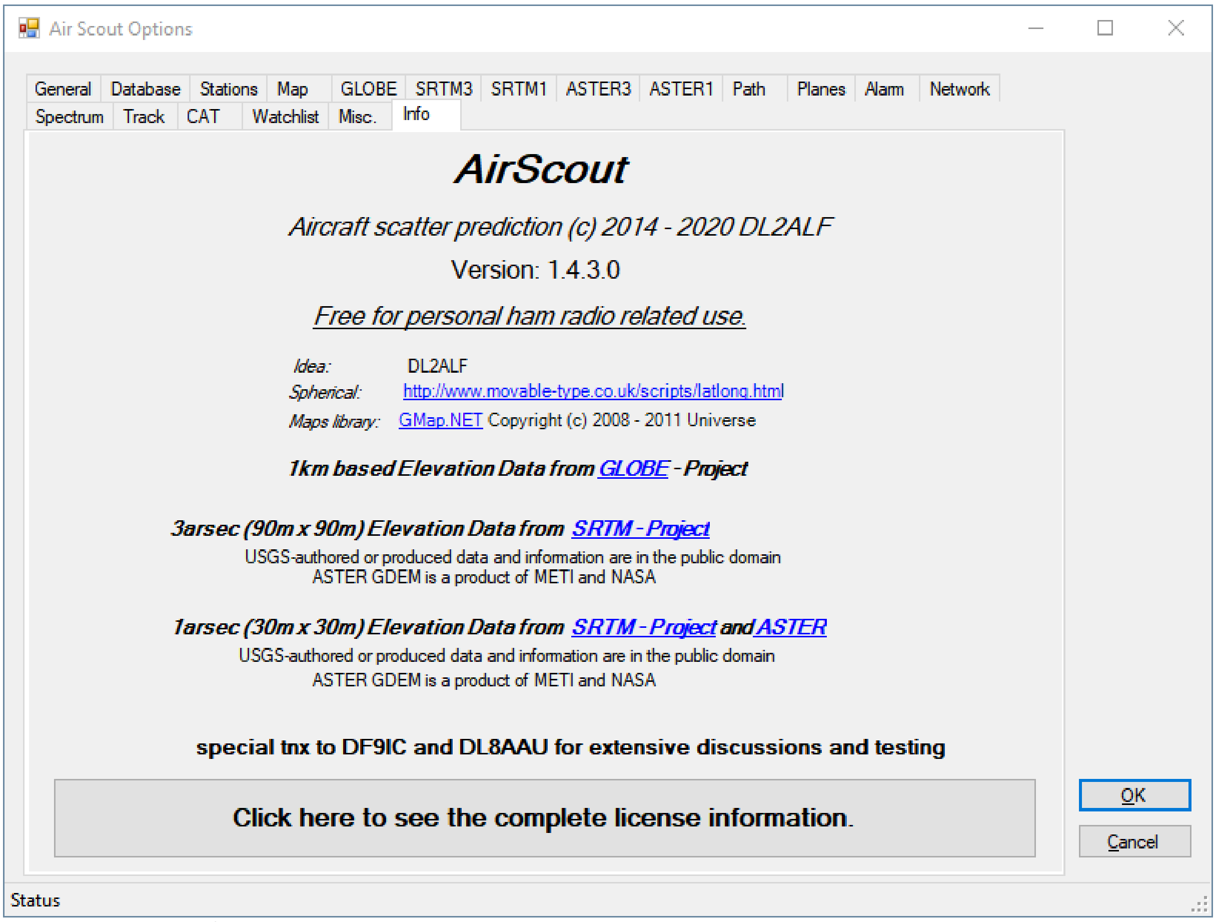


Figure 4-43 Info Tab

4.9. Further Uutlook

There are some possible enhancements already identified as they are:

- Include an antenna beam width (az/el) per band in calculations to get a maximum allowed elevation angle and limit the static distance from path with an maximum allowed azimuth angle
- Include a maximum squint angle per band in calculations
- A global server based database with station locations (other than QRZ.COM)
- An server based map to see who is online with AirScout
- An own chat within AirScout users (additional to, not instead of ON4KST)

Currently, none of the enhancements is really scheduled. Further work will much depend on my personal time budget.

5. Acknowledgements

My special thanks goes to Henning, DF9IC, and Alex, DL8AAU, for their initial discussions with me on Aircraft Scatter topic and for their hints to basic calculations. Without their help the software would not have such a precise calculation model.

Thanks to all BETA – testers who helped to improve the functionality and usability of AirScout.

Thanks to all software developers, especially to *radioman* for that Great Circle Maps library, without them AirScout would not have such an expressive user interface.

Thanks to all aircraft enthusiasts all over the world feeding the servers with ADS-B data essential for the prediction and surveillance of aircraft scatter contacts.

And, last not least, big thanks to all airlines of the world populating the sky with heavy aircrafts setting up the basis for a successful aircraft scatter operation.

6. References

[1] VHF Manager Handbook Edition 6.00 (http://www.iaru-r1.org/index.php?option=com_content&view=article&id=914:vhfuhfmw-handbook-edition-600&catid=42:vhf&Itemid=100)

[2] VK3HZ, Collection of Aircraft Enhancement Articles (http://www.vk3hz.net/ae.htm)

[3] DF9IC, Microwave QSOs with the aid of airplane reflection (http://www.df9ic.de/doc/2006/sletten_2006/sletten06_airplane_reflection.ppt)

7. Appendix

7.1. AirScout Network Protocol

7.1.1. Basic Functionality

The AirScout Network Protocol allows other software to interact with the program. The current implementation includes the following functionality:

- The request of path information
- The request of nearest planes information
- Simple remote control of the AirScout user interface

The implementation is very similar to the network protocol used by Win-Test contest logger (© F5MZN). It is a simple ASCII – type communication and it does not require much network knowledge for use.

Type of telegrams: UDP broadcast
Default network port: 9872 (Default)

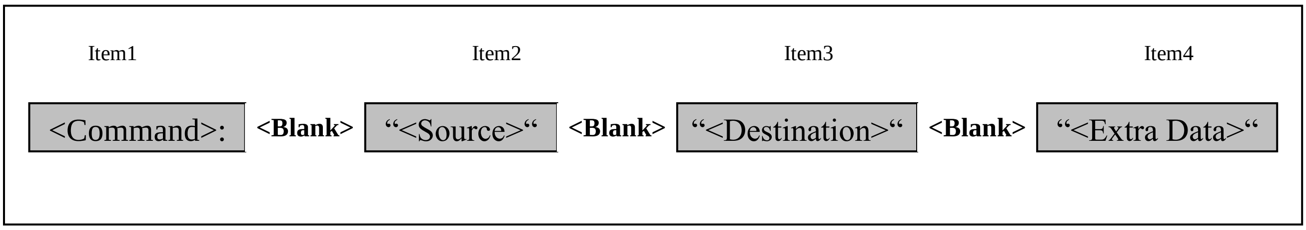
<u>Basic telegram structure:</u>

<table>
<tr><th colspan="17">Data</th><th colspan="8">Checksum</th></tr>
<tr><td colspan="8">Byte 0</td><td rowspan="3">…</td><td colspan="8">Byte n-1</td><td colspan="8">Byte n</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="7">7bit ASCII data</td><td>0</td><td colspan="7">7bit ASCII data</td><td>1</td><td colspan="7">7bit Checksum</td></tr>
</table>

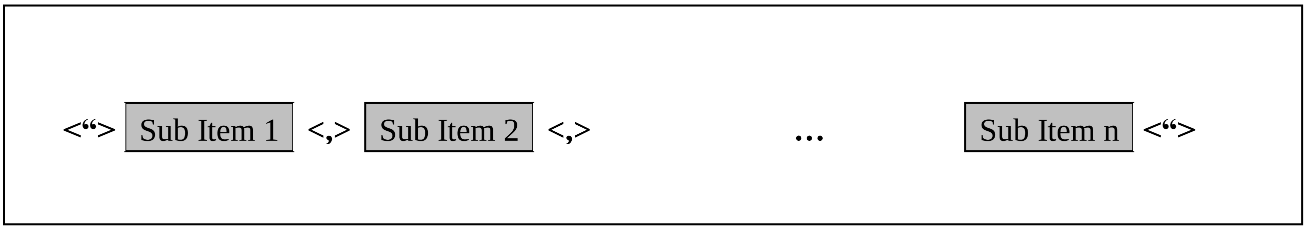
The data structure is containing several items. Each item is separated by a <Blank>. Strings must be included in <”>.
AirScout is working with max. 4 items. Except the command, all other items are treated as strings.

Item1 (mandatory): Pre-defined command string,
All chars in upper case, immediately followed by a <:>
Command is <u>not</u> included in <”>
Item2 (mandatory): Source of information
Station name as string, included in <”>
Item3 (mandatory): Destination of information
Station name as string, included in <”>
Item4 (optional): Extra data if necessary
All data is treated as one common string and is included in <”>
Sub items can be separated with <,>
No <Blank> between sub items allowed

Data structure (Item1 – Item4):



Extra data structure (Item4):



All AirScout messages start with “AS” in the command keyword, followed by source and destination information. The message structure is different for the directions client-> server and server->client. AirScout is always acting as a server.

Common data for all messages are described as follows.

Command: One of the pre-defined AirScout commands

Source: Station name for the source of information
(use short and plain ASCII names)

Destination: Station name for the destination of information,
(use short and plain ASCII names)
An empty destination string is used for broadcast info

Band: Information of the selected band;
frequencies are in 100Hz steps according to Win-Test communication,
can be one of the following strings:

500000,
700000,
1440000,
4320000,
12960000,
23200000,
34000000,
57600000,
103680000,
240480000,
470880000,

	760320000
MyCall, DXCall:	plain call signs, do not use any appendices, like “CS7/” or “/P”.
MyLoc, DXLoc:	6digit Maidenhead locator, more precise information is used for calculations if available in the AirScout data base

A typical telegram would look like this:

```
ASSHOWPATH: ”KST_23” ”AS_23” ”12960000,DL2ALF,JO50IW,GB3MHL,JO02PB”
```

7.1.2. The ASGETPATH message

Not yet implemented

7.1.3. The ASSETPATH message

This message is used to request information about all aircrafts near the given path. AirScout will:

- Calculate the new requested path (in background operation)
- Return information about aircrafts near the new path

The client should wait for the ASNEAREST respond from the server before sending a new ASSETPATH message.

Client -> Server

Command:	ASSETPATH
Source:	Name of Client
Destination:	Name of Server
Band:	Band information
MyCall:	My call sign
MyLoc:	My grid locator
DXCall:	DX call sign
DXLoc:	DX grid locator

Example:

```
ASSETPATH: ”KST_23” ”AS_23” ”12960000,DL2ALF,JO50IW,GB3MHL,JO02PB”
```

Server -> Client

The server should respond with a ASNEAREST message.

7.1.4. The ASNEAREST message

This is the response from a server to a ASSETPATH message from client.

Server -> Client

Command: ASNEAREST
Source: Name of Client
Destination: Name of Server

UTC: Timestamp of calculation finished (YYYY-MM-DD HH:mm:ssZ)
MyCall: My call sign
MyLoc: My grid locator
DXCall: DX call sign
DXLoc: DX grid locator
Count: Number of aircraft data records to follow [0 .. n]
Data: Aircraft data records, each record as follows:

Call: Call of aircraft

Category: Category of aircraft according to AirScout's category code:

L = Light,
M = Medium,
H = Heavy,
S = Super heavy

Distance: Distance to path crossing [km]

Potential: Current reflection potential according to AirScout's colour code:

100 = magenta,
75 = red,
50 = orange,
<50 = grey

Minutes: Time left to path crossing [mins]

Example (without line breaks, only 3 of 25 data sets shown):

```
ASNEAREST: ”KST_23” ”AS_23” ”2015-08-19
08:07:06Z,DL2ALF,JO50IW,GB3MHL,JO02PB,25,AFR1044,M,294,75,20,DLH507,H,253,5
0,17, ... ,WZZ933,M,28,75,2”
```

7.1.5. The ASSHOWPATH message

This command is used to control the AirScout user interface. It will:

- Stop all current playback on the AirScout main window

- Calculate and show the new requested path
- Start playback with new path and plane data

Client -> Server

Command:	ASSHOWPATH
Source:	Name of Client
Destination:	Name of Server
Band:	Band information
MyCall:	My call sign
MyLoc:	My grid locator
DXCall:	DX call sign
DXLoc:	DX grid locator

Example:

```
ASSHOWPATH: ”KST_23” ”AS_23” ”12960000,DL2ALF,JO50IW,GB3MHL,JO02PB”
```

Server -> Client

There is no response from the server. All reaction is seen on the AirScout screen.

7.1.6. The ASWATCHLIST message

This command is used to maintain AirScout's watch list. The primary idea is to synchronize the watch list with a list of ON4KST users, e.g. the chat you are currently logged in.

Client -> Server

Command:	ASWATCHLIST
Source:	Name of Client
Destination:	Name of Server
Band:	Band information
Data:	watch list data: call sign , 6digit locator, ..., call sign, 6digit locator

Example:

```
ASWATCHLIST: ”KST_23” ”AS_23” ”12960000,DL2ALF,JO50IW,G3XDY,JO02PB...”
```

7.2. The Horizon export file structure

The horizon calculation export file is a text file with comma separated values. It has the following structure:

Bearing[deg];eps[deg];Distance[km]

Bearing: the bearing in 1deg steps
eps: the minimum possible elevation angle in deg
Distance: the distance at where the minimum possible elevation occurs

7.3. The Path Calculation export file structure

The path calculation export file is a text file with comma separated values. It has the following structure:

Distance[km];Lat[deg];Lon[deg];Elevation[m]; Min_h1[m]; Min_h2[m]; Min_h[m]; Max_h[m]; F1[m]; Min_a1; Min_a2; Min_da; Max_a1; Max_a2; Max_da;eps1[deg];eps2[deg]

7.4. Webserver API and JSON File Structure

If enabled, AirScout is running a web server with a small API and delivers JSON files for further use. The syntax follows the JavaScript Object Notation description (see Wikipedia).

You can simply enter one of the URLs in your browser window to see a response for testing purpose.

CAUTION! There is no authentication mechanism, so anybody can get a response. Be sure to limit access to your local network or ensure to restrict access otherwise.

Requests always starts with this base URL:

http://<IP of AirScout Webserver>:<Port of AirScout Webserver>

7.4.1. Aircraft List

URL: /AircraftList.json
Parameters: none
Response: Aircraft list compatible to VRS Webserver
(see https://www.virtualradarserver.co.uk/Documentation/Formats/AircraftList.aspx)

7.4.2. Plane List

URL: /planes.json
Parameters: none
Response: Plane list with internal format

This JSON structure is used to read/write plane positions from/into a text file. A further purpose is to deliver this file on a web request.

General Attributes

Attribute Name	*Attribute Value*
full_count	Number of positions to follow

version File version

Plane Position Attributes

Attribute Name	*Attribute Value*
"nnnnnnnn"	Index
[.....]	Plane position array
	0: HEX
	1: Latitude[deg]
	2: Longitude[deg]
	3: Track/Heading[deg]
	4: Altitude[ft]
	5: Speed[kn]
	6: Squawk
	7: Radar
	8: Type Code
	9: Registration
	10: Timestamp [UNIX time]
	11: Departure airport
	12: Destination airport
	13: Flight number
	14: not used
	15: not used
	16: Call sign
	17: not used

Example:


```
{"full_count":1382,"version":1,
"00000000":["39840F",42.3184,3.0518,162,39000,472,"3980","","A320","F-HBAP",1475084126,"","","",0,0,"AAF221",0],
"00000001":["AACAA5",52.9520,-6.4292,295,38000,406,"2358","","B772","N795AN",1475084126,"","","",0,0,"AAL141",0]
...
}
```


7.4.3. AirScout Version

URL: /version.json
Parameters: none
Response: Version string

Example:

"1.4.3.0"

7.4.4. AirScout Settings

URL: /settings.json
Parameters: none
Response: Current settings

Example:


```
{
  "MyCall": "DL2ALF",
```

"DXCall": "S51ZO",
"Band": 10368,
"Planes_Update": 60,
"Planes_MinAlt": 5000,
"Map_AutoCenter": false,
"Planes_Draw_Path": true,
"Path_GroundClearance": 0.0,
"Planes_MaxAlt": 12200,
"MyHeight": 10.0,
"DXHeight": 10.0,
"Watchlist_MaxCount": 1000,
"Planes_Filter_Min_Alt": 0,
"Planes_Filter_Min_Category": 1,
"MinLon": -15.0,
"MaxLon": 30.0,
"MinLat": 35.0,
"MaxLat": 60.0,
...
}

7.4.5. AirScout Bands

URL: /bands.json
Parameters: none
Response: Available bands by name

Example:

["50M","70M","144M","432M","1.2G","2.3G","3.4G","5.7G","10G","24G","47G","76G"]

7.4.6. AirScout Band Values

URL: /bandvalues.json
Parameters: none
Response: Available name/value pairs for bands

Example:

[{"Name":"50M","Value":50},{"Name":"70M","Value":70},{"Name":"144M","Value":144},{"Name":"432M","Value":432},{"Name":"1.2G","Value":1296},{"Name":"2.3G","Value":2320},{"Name":"3.4G","Value":3400},{"Name":"5.7G","Value":5760},{"Name":"10G","Value":10368},{"Name":"24G","Value":24048},{"Name":"47G","Value":47088},{"Name":"76G","Value":76032}]

7.4.7. AirScout Aircraft Categories

URL: /bandvalues.json
Parameters: none
Response: Available name/value pairs for aircraft categories

Example:

[{"Name":"None","Value":0},{"Name":"Light","Value":1},{"Name":"Medium","Value":2},{"Name":"Heavy","Value":3},{"Name":"Superheavy","Value":4}]

7.4.8. Station location

Get location details for a call sign from database.

URL: /location.json?call=<call>&loc=<loc>&bestcaseelevation=<true/false>
Parameters:

call:	callsign		
loc:	empty	-->	last recent dataset
	locator	-->	details for exact this locator
	ALL	-->	all locators for this call found in database
bestcaseelevation:		use best case elevation strategy if no exact location	

Response: location details

Example:


```
{
  "Elevation": 543,
  "BestCaseElevation": true,
  "Call": "DL2ALF",
  "Loc": "JO60KG",
  "Lat": 50.283333333333331,
  "Lon": 12.9,
  "Source": 3,
  "Hits": 0,
  "LastUpdated": "2024-01-13T16:18:42.3175216Z"
}
```


7.4.9. Station QRV Information

Get QRV information for a call sign from database.

URL: /qrv.json?call=<call>&loc=<loc>&band=<band>
Parameters:

call:	callsign		
loc:	locator		
band:	empty	-->	current band selected in AirScout
	NONE	-->	none
	band value	-->	for this band only
	ALL	-->	for all bands found for this station in database

Response: QRV details

Example:


```
[
  {
    "Call": "DL2ALF",
    "Loc": "JO50IW",
    "Band": 144,
    "AntennaHeight": 0.0,
    "AntennaGain": 0.0,
    "Power": 0.0,
    "LastUpdated": "2015-11-08T00:00:00Z"
  },
  {
    "Call": "DL2ALF",
    "Loc": "JO50IW",
    "Band": 432,
    "AntennaHeight": 0.0,
    "AntennaGain": 0.0,
    "Power": 0.0,
    "LastUpdated": "2015-11-08T00:00:00Z"
```

```
 },
 {
   "Call": "DL2ALF",
   "Loc": "JO50IW",
   "Band": 1296,
   "AntennaHeight": 14.0,
   "AntennaGain": 18.0,
   "Power": 600.0,
   "LastUpdated": "2024-01-08T13:34:50Z"
 }
]
```


7.4.10. Elevation Path

Get an elevation path between two stations. You must enter at least both callsigns. All other parameters are optional. The strategy is as follows:

1. lat/lon given --> use this values
2. loc given --> try to find exact location for this call/loc in database and use it, use midpoint or best case elevation if not found
3. nothing more given --> try to find last recent location for this call in database and use it, error message if not found

URL: /elevationpath.json?mycall=<mycall>&mylat=<mylat>&mylon=<mylon>&myloc=< myloc>& dxcall=<dxcall>&dxlat=<dxlat>&dxlon=<dxlon>&dxloc=<dxloc>

Parameters:

mycall:	my callsign
mylat:	my latitude
mylon:	my longitude
myloc:	my locator
dxcall:	dx callsign
dxlat:	dx latitude
dxlon:	dx longitude
dxloc:	dx locator

Response: Elevation path in internal format

Example:


```
{
 "Valid": true,
 "Selected": false,
 "Location1": {
   "Elevation": 329,
   "BestCaseElevation": false,
   "Call": "DL2ALF",
   "Loc": "JO50IW",
   "Lat": 50.937065124511719,
   "Lon": 10.683270454406738,
   "Source": 2,
   "Hits": 0,
   "LastUpdated": "2021-12-21T16:09:44Z"
 },
 "Location2": {
   "Elevation": 29,
   "BestCaseElevation": false,
   "Call": "GB3MHZ",
   "Loc": "JO02PB",
```

"Lat": 52.056259155273438,
"Lon": 1.2802290916442871,
"Source": 2,
"Hits": 0,
"LastUpdated": "2021-09-03T08:25:32Z"
},
"QRV1": {
"Call": "DL2ALF",
"Loc": "JO50IW",
"Band": 10368,
"AntennaHeight": 10.0,
"AntennaGain": 30.0,
"Power": 5.0,
"LastUpdated": "2024-01-13T16:53:02.2802849Z"
},
"QRV2": {
"Call": "GB3MHZ",
"Loc": "JO02PB",
"Band": 10368,
"AntennaHeight": 10.0,
"AntennaGain": 30.0,
"Power": 5.0,
"LastUpdated": "2018-08-22T18:25:46Z"
},
"Lat1": 50.937065124511719,
"Lon1": 10.683270454406738,
"Lat2": 52.056259155273438,
"Lon2": 1.2802290916442871,
"StepWidth": 90.0,
"Bearing12": 284.48228289925061,
"Bearing21": 97.1169740722575,
"Distance": 662.2221258567464,
"Count": 7359,
"Path": [
329,
327,
327,
329,
...
}

7.4.11. Propagation Path

Get an propagation path between two stations. You must enter at least both callsigns. All other parameters are optional. The strategy is as follows:

1. lat/lon given --> use this values
2. loc given --> try to find exact location for this call/loc in database and use it, use midpoint or best case elevation if not found
3. nothing more given --> try to find last recent location for this call in database and use it, error message if not found

All other relevant settings for band, K-Factor etc. are taken from AirScout's current settings.

URL: /propagationpath.json?mycall=<mycall>&mylat=<mylat>&mylon=<mylon>&myloc=<myloc>& dxcall=<dxcall>&dxlat=<dxlat>&dxlon=<dxlon>&dxloc=<dxloc>

Parameters:

mycall:	my callsign
mylat:	my latitude
mylon:	my longitude
myloc:	my locator
dxcall:	dx callsign

dxlat: dx latitude
dxlon: dx longitude
dxloc: dx locator

Response: Propagation path in internal format

Example:


```
{
  "Valid": true,
  "Selected": false,
  "LocalObstructed": false,
  "LocalObstruction": -1.5707963267948966,
  "Location1": {
    "Elevation": 329,
    "BestCaseElevation": false,
    "Call": "DL2ALF",
    "Loc": "JO50IW",
    "Lat": 50.937065124511719,
    "Lon": 10.683270454406738,
    "Source": 2,
    "Hits": 0,
    "LastUpdated": "2021-12-21T16:09:44Z"
  },
  "Location2": {
    "Elevation": 29,
    "BestCaseElevation": false,
    "Call": "GB3MHZ",
    "Loc": "JO02PB",
    "Lat": 52.056259155273438,
    "Lon": 1.2802290916442871,
    "Source": 2,
    "Hits": 0,
    "LastUpdated": "2021-09-03T08:25:32Z"
  },
  "QRV1": {
    "Call": "DL2ALF",
    "Loc": "JO50IW",
    "Band": 10368,
    "AntennaHeight": 10.0,
    "AntennaGain": 30.0,
    "Power": 5.0,
    "LastUpdated": "2024-01-13T16:59:44.66169Z"
  },
  "QRV2": {
    "Call": "GB3MHZ",
    "Loc": "JO02PB",
    "Band": 10368,
    "AntennaHeight": 10.0,
    "AntennaGain": 30.0,
    "Power": 5.0,
    "LastUpdated": "2018-08-22T18:25:46Z"
  },
  "Lat1": 50.937065124511719,
  "Lon1": 10.683270454406738,
  "h1": 339.0,
  "Lat2": 52.056259155273438,
  "Lon2": 1.2802290916442871,
  "h2": 39.0,
  "QRG": 10.368,
  "Radius": 8473.43,
  "F1_Clearance": 0.6,
  "StepWidth": 90.0,
  "Bearing12": 284.48228289925061,
  "Bearing21": 97.1169740722575,
  "Distance": 662.2221258567464,
  "Eps1_Min": 0.010697514990364717,
  "Eps2_Min": -0.0011855823307549685,
  "LastUpdated": "2021-09-03T06:41:59Z"
}
```

7.4.12. Get Nearest Planes

Get nearest planes for a path between two stations. You must enter at least both callsigns. All other parameters are optional. The strategy is as follows:

1. lat/lon given --> use this values
2. loc given --> try to find exact location for this call/loc in database and use it, use midpoint or best case elevation if not found
3. nothing more given --> try to find last recent location for this call in database and use it, error message if not found

All other relevant settings for band, K-Factor etc. are taken from AirScout's current settings.

URL: /nearestplanes.json?mycall=<mycall>&mylat=<mylat>&mylon=<mylon>&myloc=<myloc>& dxcall=<dxcall>&dxlat=<dxlat>&dxlon=<dxlon>&dxloc=<dxloc>

Parameters:

mycall:	my callsign
mylat:	my latitude
mylon:	my longitude
myloc:	my locator
dxcall:	dx callsign
dxlat:	dx latitude
dxlon:	dx longitude
dxloc:	dx locator

Response: Nearest planes in internal format

Example:


```
[
 {
   "Track": 111.0,
   "IntPoint": {
     "Lat": 51.541619962779478,
     "Lon": 6.4450853499762637
   },
   "IntQRB": 185.54935982979651,
   "AltDiff": 2277.0582088453557,
   "Potential": 75,
   "Eps1": 0.018205882357031057,
   "Eps2": 0.0099464577827096372,
   "Theta1": 0.0,
   "Theta2": 0.0,
   "Angle": 0.0,
   "Squint": 0.023320500552348293,
   "SignalStrength": -1.7976931348623157E+308,
   "Ambiguous": false,
   "Comment": "",
   "OldTime": "0001-01-01T00:00:00Z",
   "OldLat": 0.0,
   "OldLon": 0.0,
   "OldAlt": 0.0,
   "OldTrack": 0.0,
   "OldSpeed": 0.0,
   "Time": "2024-01-13T17:01:56.1657973Z",
   "Call": "ARN122",
   "Reg": "F-EGSH2",
   "Hex": "00A2E4",
   "Lat": 52.166450872839924,
   "Lon": 3.9398383218682471,
```

"Alt": 36950.0,
"Alt_m": 11262.359639604492,
"Speed": 503.375543964,
"Speed_kmh": 932.251507421328,
"Type": "B763",
"Manufacturer": "BOEING",
"Model": "767-300",
"Category": 3,
"From": null,
"To": null,
"VSpeed": 0
},
{
"Track": 117.0,
...
]

7.4.13. AirScout Aiports

Get a list of airports from AirScout database.

URL: /airports.json
Parameters: none
Response: A list of airports in internal format

Example:

[{"ICAO":"EKYT","IATA":"AAL","Lat":57.092781,"Lon":9.849164,"Alt":10.0,"Airport":"Aalborg Airport","Country":"Denmark","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"OMAL","IATA":"AAN","Lat":24.26166,"Lon":55.609161,"Alt":869.0,"Airport":"Al Ain International Airport","Country":"United Arab Emirates","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"URKA","IATA":"AAQ","Lat":45.00209,"Lon":37.347271,"Alt":174.0,"Airport":"Anapa Airport","Country":"Russia","LastUpdated":"2019-03-08T19:05:46Z"},{"ICAO":"EKAH","IATA":"AAR","Lat":56.300011,"Lon":10.619,"Alt":82.0,"Airport":"Aarhus Airport","Country":"Denmark","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"UNAA","IATA":"ABA","Lat":53.740002,"Lon":91.385002,"Alt":831.0,"Airport":"Abakan International Airport","Country":"Russia","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"OIAA","IATA":"ABD","Lat":30.371111,"Lon":48.228329,"Alt":19.0,"Airport":"Abadan Airport","Country":"Iran","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"KABE","IATA":"ABE","Lat":40.652302,"Lon":-75.440399,"Alt":393.0,"Airport":"Allentown Lehigh Valley International Airport","Country":"United States","LastUpdated":"2019-03-08T19:05:46Z"},{"ICAO":"KABI","IATA":"ABI","Lat":32.411301,"Lon":-99.681801,"Alt":1790.0,"Airport":"Abilene Regional Airport","Country":"United States","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"DIAP","IATA":"ABJ","Lat":5.261386,"Lon":-3.92629,"Alt":21.0,"Airport":"Abidjan Port Bouet Airport","Country":"Cote D'ivoire (Ivory Coast)","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"KABQ","IATA":"ABQ","Lat":35.040218,"Lon":-106.609001,"Alt":5355.0,"Airport":"Albuquerque International Airport","Country":"United States","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"YMAY","IATA":"ABX","Lat":-36.067699,"Lon":146.957993,"Alt":539.0,"Airport":"Albury Airport","Country":"Australia","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"EGPD","IATA":"ABZ","Lat":57.201939,"Lon":-2.19777,"Alt":215.0,"Airport":"Aberdeen International Airport","Country":"United Kingdom","LastUpdated":"2019-03-08T19:05:45Z"},{"ICAO":"MMAA","IATA":"ACA","Lat":16.7
...
]

7.4.14. Plane Feed Settings

Get a current plane feed settings.

URL: /planefeedsettings.json
Parameters: none
Response: A list of planefeed settings in internal format

Example:

[{"Name":"[WebFeed]	VRS Web Server","Value":{"DisclaimerAccepted":"","SaveToFile":false,"URL":"http://airscatter.dk:8890/VirtualRadar/AircraftList.json?ldv=%LASTDV%&stm=%UNIXTIME%&lat=%MYLAT%&lng=%MYLON%&fDstL=%MINDISTKM%&fDstU=%MAXDISTKM%&fAltL=%MINALTFT%&fAltU=%MAXALTFT%","Username":"dl2alf","Password":"airscout","URL2":"http://airscatter.dudez.no:18080/VirtualRadar/AircraftList.json?ldv=%LASTDV%&stm=%UNIXTIME%&lat=%MYLAT%&lng=%MYLON%&fDstL=%MINDISTKM%&fDstU=%MAXDISTK

M%&fAltL=%MINALTFT%&fAltU=%MAXALTFT%","Username2":"dl2alf","Password2":"airscout","LoadShare":true,"UseGeoAlt":false ,"Timeout":30}}]

7.5. Special keywords used in URLs for plane feeds

The following keywords can be used in URL strings for plane feeds. They can be placed at any position of the URL and will be replaced with online values before sending the request to the web server.

You can customize your covered area, flight altitudes this way. Be careful, a syntax checking is provided and will throw an Exception if something is wrong. The following keywords are allowed so far:

`%APPDIR%`	AirScout's program directory
`%DATADIR%`	AirScout's local application data directory
`%LOGDIR%`	AirScout's log file directory
`%DATABASEDIR%`	AirScout's database directory
`%MINLAT%`	latitude of the lower left corner of your covered area
`%MINLON%`	longitude of the lower left corner of your covered area
`%MAXLAT%`	latitude of the upper right corner of your covered area
`%MAXLON%`	longitude of the upper right corner of your covered area
`%MINALTM%`	minimum altitude of aircrafts in m
`%MAXALTM%`	maximum altitude of aircrafts in m
`%MINALTFT%`	minimum altitude of aircrafts in feet
`%MAXALTFT%`	maximum altitude of aircrafts in feet