Flight Simulator Adventure Creator

Command-line version

Source code (C++), betas, and latest data files:
https://github.com/MaximumOctopus/FlightSimAdventureCreator

Latest version zip download:
https://flightsim.to/file/41477/flight-simulator-adventure-creator

Last Revision: 0.4, October 18th 2022

Introduction

Flight Simulator Adventure Creator (FSAC) is a powerful command-line tool for generating random routes for flight simulators (currently optimised for Microsoft Flight Simulator 2020).

It offers the ability to create MSFS plan files (.pln), so it's easy to load your newly generated route into MSFS 2020.

There are many options available for customising the type of route that will be built.

If you have any questions or suggestions for new features, then don't hesitate to get in touch.

By default, with no options selected, FSAC will randomly select an aircraft and start location for you. It will even give you a suggested job, a reason for your journey.

Adding more options will allow for a more controlled experience, still random, but tweaked towards your preferences.

FSAC has a database of around 23200 airports. All of these are valid within MSFS 2020, although its airport list is much larger (around 35000). It's currently very difficult to get hold of a definitive MSFS airport list. I'll add to the FSAC list as and when airports are found to be missing.

Included Files and Customisation Options

Every file that comes with FSAC is editable and customisation is encouraged. Keep a backup of your edits, as new updates will overwrite any changes.

`config_readme.txt`
Details on creating and customising configuration files. These are an alternative to using FSAC with a set of command-line parameters. See the *Configuration File Handling* section below.

`custom_aircraft.txt`
Add your purchased and free aircraft to this file. Details for editing it can be found inside.

Alternatively, use the `/fsacca` option to automatically generate the file from the contents of your community folder.

`default_aircraft.txt`
This file contains information on every aircraft that comes with the various versions of MSFS 2020.

`favourites.txt`
A list of favourite airports specified using ICAO codes. This comes populated with a list of interesting airports. Feel free to add your own.

Installation

FSAC doesn't need installing in the normal sense. Download the zip from either *github* or *FlightSim.to* and extract it to a folder of your choice.

This is a command-line tool, you'll need to use it from a Windows console. Navigate to the folder where it's installed (in Windows Explorer), click in the folder path area (so the entire folder path is highlighted, type "cmd", then press enter. You'll now have a new console window set to the location of FCAS!

Command Line Parameters

To get live help run FSAC with the `/?` Parameter:

```
fsac /?
```

This will show you all the available commands and some detail on their use.

You find a set of example parameters at the end of this document.

All comments and suggestions are gratefully received, please email them to me at paul@freshney.org

Aircraft Selection

These settings affect the pool of aircraft that the random aircraft is selected from. Without setting any options, every aircraft contained in the `default_aircraft.txt` and `custom_aircraft.txt` files will be included.

`/aircrafttype:`*type*
Limits the randomly selected aircraft to a specific type, where *type* must be one of the following options:

0	Props
1	Jets
2	Helicopters
3	Gliders
4	Twin props
5	Turbo props
6	Twin turbo props
7	Balloons

`/noairliner`
Removes airliners from the list of aircraft that can be selected randomly.

`/nomilitary`
Removes military aircraft from the list of aircraft that can be selected randomly.

`/msfsversion:`*version*
Select the random aircraft based on the version of MSFS 2020 they are available in. Where version can be one of these:

0	All (includes third party aircraft)
1	Deluxe version and above
2	Premium deluxe

`/maxspeed:`*speed*
Set a maximum cruise for your random aircraft. Specify the speed in knots.

`/minspeed:`*speed*
Set a minimum cruise for your random aircraft. Specify the speed in knots.

`/ga`
Select only GA aircraft. All props, turbo props, twins. No military, and no airliners.

`/ja`
Select only jet airliners. Jets, airliners, no military.

`/props`
Only props and twin props.

`/seaplanes`
Disables all aircraft except seaplanes, and enables the seaplane base airport type.

`/twins`
Only twin props and twin turbo props.

`/turbos`
Only turbo props and twin turbo props.

`/nocustom`
Ignore aircraft from the `custom_aircraft.txt` file.

`/nodefault`
Ignore aircraft from the `default_aircraft.txt` file.

Airport Selection

The following options affect the pool of airports from which your random route is created.

`/continent:`*`ico_code`*
Filters available airports by continent. The ICO code must be one of these:

AF	Africa
AN	Antarctica
AS	Asia
EU	Europe
NA	North America
OC	Oceania
SA	South America

`/country:`*`ico_code`*
Filters available airports by country. Must be a value ICO country code. A complete list can be found here:

https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2\n

`/region:`*`ico_code`*
Filters available airports by specific region (eg. US-FL for Florida). Must be a valid ICO region code. A complete list can be found here:

https://en.wikipedia.org/wiki/ISO_3166-2

`/elevation:`*`value`*
Allows only airports above this value, in feet.

`/latfrom:`*`angle_degrees`*
Limit airport location by latitude. Set the from angle in degrees. 0° is the Greenwich Meridian. 180° is the international date line.

From -> *to* should go clockwise around the Earth, as soon from the North Pole.

`/latto:`*`angle_degrees`*
Limit airport location by latitude. Set the from angle in degrees. 0° is the Greenwich Meridian. 180° is the international date line.

From -> *to* should go clockwise around the Earth, as soon from the North Pole.

`/longfrom:`*`angle_degrees`*
Limit airport location by longitude. Set the angle in degrees. 0° is the equator, +90° is the North Pole, -90° is the South Pole.

`From -> to` should go anti-clockwise around the Earth, as soon from side-on with 0° latitude (the Greenwich Meridian) to the left.

`/longto:`*`angle_degrees`*
Limit airport location by longitude. Set the angle in degrees. 0° is the equator, +90° is the North Pole, -90° is the South Pole.

`From -> to` should go anti-clockwise around the Earth, as soon from side-on with 0° latitude (the Greenwich Meridian) to the left.

`/nosmall`
No small airports.

`/nomedium`
No medium airports.

`/nolarge`
No large airports.

`/noheliports`
No heliports.

`/onlysmall`
Only small airports.

`/onlymedium`
Only medium airports

`/onlylarge`
Only large airports.

`/onlyheliports`
Only heliports. This option also sets aircraft selection to helicopters only.

`/day`
Limit airports to those currently in daylight. This is a beta feature, improvements are incoming.

`/night`
Limit airports to those currently in darkness. This is a beta feature, improvements are incoming.

`/list`
List all loaded airports (those in the random selection pool). Not useful unless you know there’s going be a small number to show.

Route Creation

`/atob`
Informs FSAC that you want to create a route of at least one leg (A->B).

If FSAC detects that you want a route, but you've forgotten this parameter (I've done it several times while testing!), then it will add it for you.

`/legs:`*`number`*
The number of legs comprising your journey. Each leg will be a new airport to visit. The default value is 1.

`/range:`*`number`*
The length in nautical miles of each leg of your journey. The default is 100 nm. FSAC will add +10% and -10% of this value when searching for airports (so with 100nm, it will search from 90 to 110nm).

If you want to specify the range in kilometres, then append a K to the end of your value, eg.:

```
/range:100k
```

`/startairport:`*`icao`*
Your route will start at the specified airport. The ICAO code supplied must be an airport that exists in the FSAC airport database.

`/endairport:`*`icao`*
Your route will end at the specified airport. The ICAO code supplied must be an airport that exists in the FSAC airport database.

`/favourite`
Select the start airport from one of the airports in the favourites file (`favourites.txt`).

`/number:value`
The number of multi-leg routes to generate. Default is 1.

`/simple:value`
The number of single leg routes to generate. Default is 5.

`/bearing:`*`compass_bearing`*
Set a direction for you journey using a compass bearing. Valid values are:

```
N, NNE, NE, ENE, E, ESE, SE, SSE, S, SSW, SW, WSW, W, WNW, NW, NNW
```

`/direction:`*`angle`*
Set an angle to travel for your journey. *`angle`* should be between 0 and 360.

`/time:`*`minutes`*
Instead of using *range*, set the length of each leg based on the cruise speed of the selected aircraft. Where *minutes* is the number of minutes each leg of the journey should take.

The calculated distance value will have the same margins applied as if using the `/range` option.

`/useaircraftrange`
Use the selected aircraft's range as the basis of the leg length. By default FSAC uses 75% of the aircraft's maximum range. This percentage can be altered with the option below.

`/aircraftrange:`*`percent`*
Sets the aircraft percent modifier to the value of *`percent`*. The specified value must be between 1 and 100.

```
/allowexcessrange
```

Allow routes longer than the selected aircraft's maximum range. By default, the maximum leg length will be adjusted to fit. But don’t forget to refuel!

```
/route:start_airport_icao:range:legs:direction_or_bearing
```

This is a new option in 0.4 that allows for several of the most popular settings to be entered in a single parameter. No option is mandatary, don’t enter anything.

```
/route:EGNX::2
```

Sets the start airport to EGNX (East Midlands airport), no range is specified, so the default of 100 nm is used, 2 legs, and doesn’t specify a route direction.

```
/route::120:3:NE
```

No start airport, so one will be selected at random, 120nm range, 3 legs, and a route direction of NE.

```
/route:CYPU:50:2:180
```

Start at CYPU in Canada, range of 50nm, 2 legs, and a route direction of 180 (South).

```
/keeptrying
```

There are plenty of places in the world where 20 miles between airports isn't going to cause a problem. There are plenty of places where <5 will work! But this isn't true for many remote locations. So, rather than not getting a route at all (or if you’re having trouble completing routes in certain locations), try using `/keeptrying`.

If a search finds no airports within range, it will search again but will multiply range by 1.25 and (if set) increase the direction angle by 5 degrees plus and minus. If it still can't find an airport it will try again, modifying range and direction by the same amounts. It will give-up after 21 tries...

Export Options

```
/exportmsfs
```

Export each route to an MSFS 2020 plan file (.pln). These are stored in the *plans* folder in the following format:

```
\plans\icao_from_to_icao_to_yyyymmdd_hhmmss.pln
```

Where *yyyymmdd* is the current date, and *hhmmss* is the current time.

```
/exporttext
```

Export each route to a text itinerary file. These are stored in the *reports* folder in the following format:

```
\reports\icao_from_to_icao_to_yyyymmdd_hhmmss.txt
```

Where *yyyymmdd* is the current date, and *hhmmss* is the current time.

Configuration File Handling

If you find yourself running FSAC regularly with the same options, then you can make life easier by saving those options into a config file and loading at your leisure.

`/loadconfig:`*`file_name`*
Load a config file with the specified *file_name*.

`/saveconfig:`*`file_name`*
Save a config file to the *file_name* specified.

Miscellaneous Options

`/fsacca`
Generates a `custom_aircraft.txt` file based on the contents of your MSFS 2020 community folder. It's currently not 100% accurate in determining whether an aircraft is military or not.

It can't populate the minimum runway required field (this information isn't available in the aircraft configuration files), but this isn't that important *yet*.

This option must be used on its own, without any aircraft, airport, or route options.

It will overwrite the existing file, and you will lose any edits. So keep a backup.

`/q`
Silences some console output (particularly the full configuration list and loaded-data statistics).

`/cat`
This option must **never** be used.

Data Analysis

There is a lot of data available to FSAC: over 23000 airports, and over 25000 runways. I'll be adding interesting options over time that will allow you to interrogate the data, and maybe find something previously unknown.

Once run, these functions will close FSAC. They can't be used in conjunction with route or aircraft generation.

`/findnearest:icao_code`
Will show all airports within range (default 100nm) of the specified airport.

Combine this with other airport selection options to show only small airports within 50nm, for example:

```
/onlysmall /findnearest:EGNX /range:50
```

Examples

Not all options or combinations are demonstrated here; for a complete list use the help parameter, "FSAC /?" or see the section above.

```
fsac /prop
```

Will pick a random prop aircraft and random starting airport. Does not generate a route (/atob does that).

```
fsac /atob /continent:NA
```

Generate simple routes (from one airport to another) in the continent of North America.

```
fsac /atob /continent:AS /minspeed:300
```

Generate simple routes (from one airport to another) in the continent of Asia, with a plane that has a minimum cruise speed of 300 kts.

```
fsac /atob /continent:EU /nomedium
```

A simple route, in Europe, with medium airports disabled (leaving only large and small).

As mentioned below, the data available is larger than the number of valid airports in MSFS 2020. There is no definitive list of MSFS airports, but Brian Bernacki (many thanks) has done a lot of work to get this going. The data file for FSAC has around 23000 airports which is sizeable percentage of those advertised to exist in the sim (around 35000).

Generate simple routes (from one airport to another) in the continent of North America.

```
fsac /atob /legs:2
```

Generate a route of two legs. Starting location can be anywhere, maximum distance between legs is 100 nm (default).

```
fsac /atob /legs:2 /range:80
```

Generate a route of two leg. Starting location can be anywhere, maximum distance between legs is 80 nm.

```
fsac /atob /legs:2 /range:80 /ga
```

As above, but will now limit the random aircraft selection to either prop or twin prop, ignores military and airliners.

```
fsac /atob /legs:3 /range:200 /jet /noairliner
```

A route of 3 leg, where each leg is no more than 200 nm. Only jets, no airliners. Military jets allowed.

```
fsac /atob /legs:3 /range:200 /jet /noairliner /country:GB
```

As above, but limit airports to Great Britain.

A complete list of ISO country codes can be found here:

https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

A complete list of ISO region codes (for use with /region) can be found here:

https://en.wikipedia.org/wiki/ISO_3166-2

```
fsac /atob /useaircraftrange /legs:3
```

Creates a three-leg journey where each leg length is determined by the chosen aircraft's max range. A modifier is first applied to the aircraft's max range (the default is 75%) before dividing the range by the number of legs.

So if an aircraft is chosen with a max range of 1000 nm, this becomes 750 nm (75% of 1000), then this gets divided by three. So each leg can be a maximum length of 250 nm.

Use the option `/allowexcessrange` to disable this behaviour, just don't forget to refuel!

```
fsac /atob /range:50 /twins
```

A 50 mile route from airport A to B, and I only want to fly twin props or twin turbo props. Airliners and military are allowed.

```
fsac /atob /day
```

A->B route but only airports in daylight will be added to the random selection pool. `/night` does the opposite.

This is very much a beta feature. It'll likely work better for Northern Hemisphere locations (the airport, not you), but you might find yourself in an airport after dusk or before dawn. A better implementation will appear very soon!

```
fsac /atob /startairport:EKCH /bearing:N /legs:2 /range:200
```

So we start a 2-leg journey from Copenhagen (EKCH) and will fly north into Sweden.

```
fsac /startairport:EGKK /endairport:EGXW /range:50
```

Generate a route from Gatwick Airport to Waddington Airfield.

This will automatically set the number of legs based on distance from the start to end airports. If legs was specified instead, then the range would be automatically calculated.

Notice that `/atob` is missing. From version 0.3 onwards, the application will see that it's missing (providing some other settings have been set) and add it automatically.

Setting start and end is in beta at the moment. It seems to work well with testing. If you get any issues from it, then please send me details.

```
fsac /atob /turbos /favourite
```

As you can guess by now, this one plots an A->B route (a single leg) and specifies only turbo props or twin turbo props. It will pick the starting airport from one of the airports in the "`favourites.txt`" file. As with all FSAC files, this is editable, and I suggest you fill it full of your own favourite places. It comes pre-populated with a selection of interesting airports from across the world.

```
fsac /atob /props /range:20 /keeptrying
```

A nice short route in a slow-but-fun prop (probably). There are plenty of places in the world where 20 miles between airports isn't going to cause a problem. There are plenty of places where <5 will work! But this isn't true for many remote locations. So, rather than not getting a route at all, try using `/keeptrying`. If a search finds no airports, then it will search again but will multiply range by 1.25 and (if set) increase the direction angle by 5 degrees plus and minus. If it still can't find an airport it will try again, modifying range and direction by the same amounts. It will give-up after 21 tries...

```
fsac /startairport:YSSY /range:200k
```

A->B route `(/atob` added automatically as a start airport is specified). Range is specified in kilometers (which will be converted to nautical miles).

```
      /saveconfig:filename
```

This will save your route/aircraft configuration, using all of the options you've selected, to a file.

```
      /loadconfig:filename
```

This will load a previously saved or created configuration. You won't have to use command-line options again!

```
      /findnearest:ICAO /range:50
```

Show all airports within range of the specified airport. If the range parameter is omitted, then the default (100 nm) will be used.

This is one of a planned list of "data analysis" functions that will be added to regularly.

These can't be combined with any other route options. Once it's listed the nearest airports the application will exit.

Credits

Programming	Paul Alan Freshney
Development Cats	Rutherford, Freeman, and Maxwell `(maximumoctopus.com/developmentcats.htm)`
Airport/Runway data	`ourairports.com`
Icon	`https://icon-icons.com/`
Thanks to	Brian Bernacki for his MSFS airport data And *everyone* who has sent me feedback. Please keep it coming!

Written with C++ 20 in Visual Studio 2022.

All of my software is *free* and *open source*.

Please donate to your local cat charity or shelter, thanks.

paul@freshney.org

https://github.com/MaximumOctopus/FlightSimAdventureCreator