

XFSE User Manual

FSEconomy Client for X-Plane

The FSEconomy Team

Version 1.1.1, May 01, 2023

Table of Contents

1. Installation	1
1.1. Prerequisites	1
1.2. Installing the XFSE Application	2
1.3. Installing the Connector Plugin	3
1.4. Removing the Legacy Client	5
2. Application Overview	6
3. General Settings	7
3.1. FSEconomy Credentials	7
3.2. Simulator Connection	8
3.3. Application Settings	9
4. Aircraft Configuration	12
4.1. Aircraft Alias	13
4.2. Fuel Tank Configuration	14
4.3. Override Settings	18
5. Flying	19
5.1. Starting a Flight	19
5.2. Enroute Monitoring	21
5.3. Cancelling a Flight	25
5.4. Ending a Flight	26
6. Troubleshooting	27
6.1. XFSE Detected as Malware	27
6.2. Log Files	28
6.3. Configuration Files	30
7. Support Requests	31
7.1. Community Support	31
7.2. Official Support	31
A. Appendix	32
A.1. XFSE Changelog	32

Chapter 1. Installation

1.1. Prerequisites

1.1.1. Compatibility

Before installing, please check whether XFSE is compatible with your system. XFSE is compatible with these operating systems:

- Microsoft Windows 8 or newer (7 and older are not supported)
- macOS 10.15 “Catalina” or newer on Intel x86_64 systems
- macOS 11 “Big Sur” or newer on Apple Silicon (native support without Rosetta)
- Ubuntu Linux 20.04 LTS or newer (and derivatives, e.g. elementary OS, Mint, etc.)
- Arch Linux (and derivatives, e.g. Manjaro, Garuda etc., glibc 2.37 or newer)

XFSE works with X-Plane 9.50 and newer. Older versions of X-Plane are not supported.

NOTE

It is not required to run XFSE on the same computer as X-Plane. It is possible to run XFSE on a remote computer and let it “talk” to X-Plane via a local network connection.

1.1.2. Download

Before installing XFSE, you will need to download the following software:

- **All Platforms:** The latest version of XFSE, available here: <https://sites.google.com/site/fseoperationsguide/getting-started/using-the-fse-client/x-plane-client>
- **X-Plane 11.10 and older on Windows or macOS:** The XPlaneConnect plugin version 1.3rc6 or newer, available here: <https://github.com/fseconomy/XPlaneConnect/releases>
- **X-Plane 11.10 and older on Linux:** The XPlaneConnect plugin version 1.3rc6 or newer, available here: <https://github.com/fseconomy/XPlaneConnect/releases>

TIP

For X-Plane 11.20 and newer, the XPlaneConnect plugin is no longer required. XFSE ships with a bundled plugin called XfseConnect, offering better performance and enhanced features.

All downloads come as compressed archives, which you can download and store on your computer.

1.2. Installing the XFSE Application

XFSE itself is easy to install: simply extract the content of the downloaded archive to a suitable location on your computer. In order to run the XFSE application, these additional steps may be required on your system (depending on operating system, security settings and third party anti-malware):

- Exclude the folder from on-access scans by your virus scanner (some anti-malware have false positive hits with XFSE)
- Make sure the partition containing the XFSE executable is not mounted with `noexec` (Linux-specific), and the executable itself has the execution bit set in the file permissions
- Allow execution of the XFSE app in the macOS Security Center, even though it's not signed by Apple (required on macOS 11 and newer)

TIP

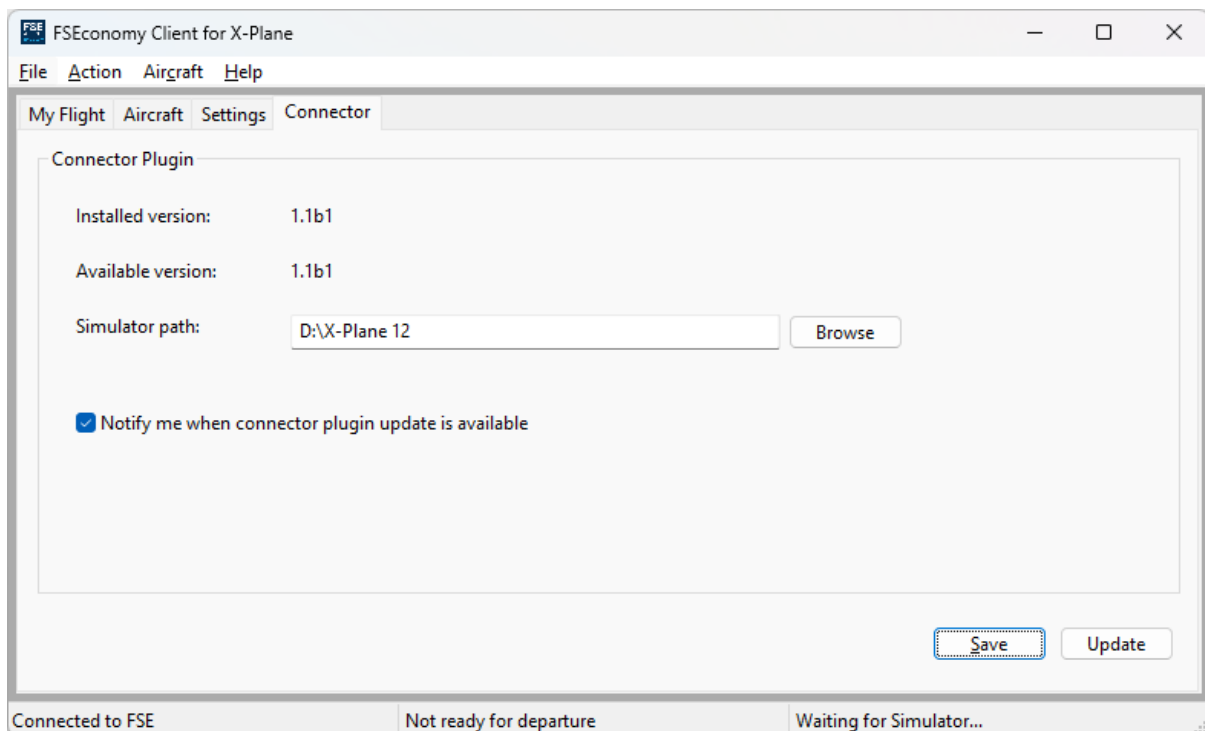
If your anti-malware software admonishes the XFSE executable as potentially malicious, please notify the FSEconomy staff. You can support us by submitting the executable as suspected false positive to the developer of your anti-malware software; they will analyze the file and — if found to be harmless — will add it to their internal whitelist.

1.3. Installing the Connector Plugin

1.3.1. X-Plane 11.20 and newer

Beginning with version 1.1, XFSE ships bundled with XfseConnect, a connector plugin specifically written for XFSE. To install the plugin, follow these steps:

- Run the XFSE application on the computer where your simulator is installed
- Make sure X-Plane is **not** running; otherwise the installation will fail
- Navigate to the *Connector* tab
- Select the root directory of your X-Plane installation
- Save the settings you made
- Click on *Update* to install the bundled plugin



When running XFSE for the first time, if no plugin has been installed so far, or if the installed plugin is older than the one bundled with XFSE, it will prompt you and ask whether it should update/install the plugin.

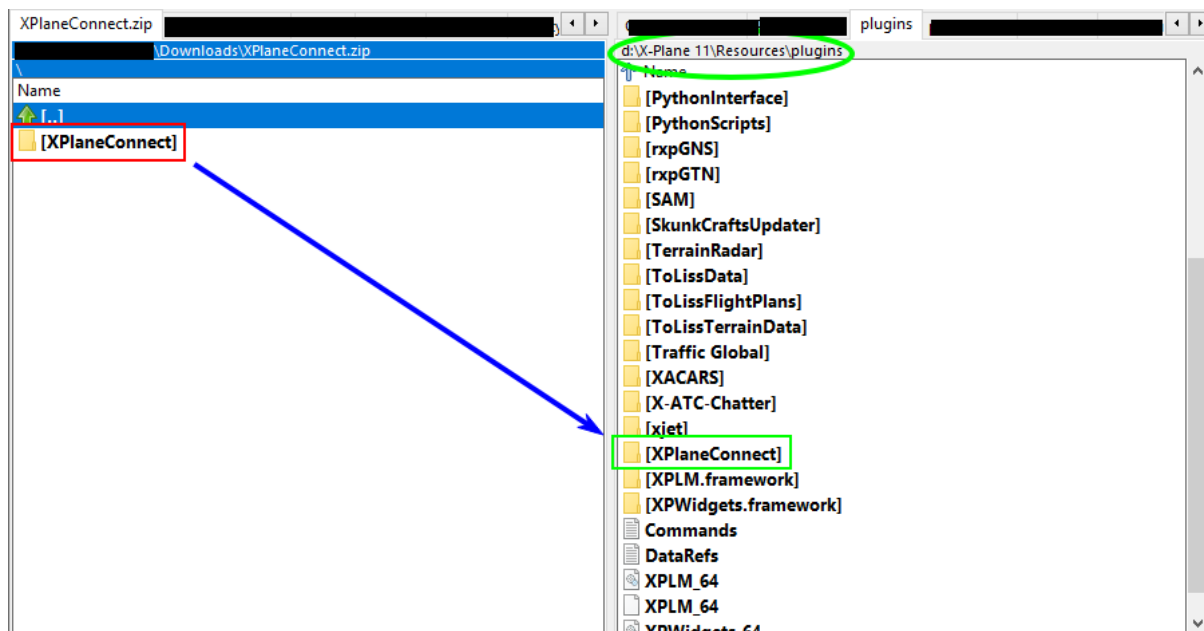
IMPORTANT

Even if XFSE can run on a remote computer, it cannot install the plugin remotely. To install the plugin, you have to run XFSE at least once on the computer where X-Plane is installed. On the remote computer, you can safely disable update notifications for the connector plugin.

1.3.2. X-Plane 11.10 and older

Install the *XPlaneConnect* plugin. It is a conventional X-Plane plugin. If you're familiar with X-Plane plugins, installing XPlaneConnect is straightforward:

- Extract the zip archive to a temporary folder on your computer
- Move the extracted XPlaneConnect folder into the **Resources/Plugins** directory of your X-Plane installation



1.4. Removing the Legacy Client

Using XFSE, you do not need the legacy client (*x-economy* 1.9.x series) or any of its dependencies anymore.

IMPORTANT

The *x-economy* legacy client has been decommissioned and is no longer operational. Therefore, we highly recommend to remove this client and its dependencies from your computer — it no longer serves any purpose, and only contributes to exposing your computer to potential security risks.

You can therefore safely remove:

From your `Resources/plugins/PythonScripts` folder:

- `PI_xfse.py` and `PI_xfse.pyc`
- `x-economy.ini`

NOTE

Always delete both, the `.py` (script) and `.pyc` (bytecode) file. If one of them is left behind, the Python plugin will continue trying to execute it.

If you have any of the optional scripts installed, please remember to also remove

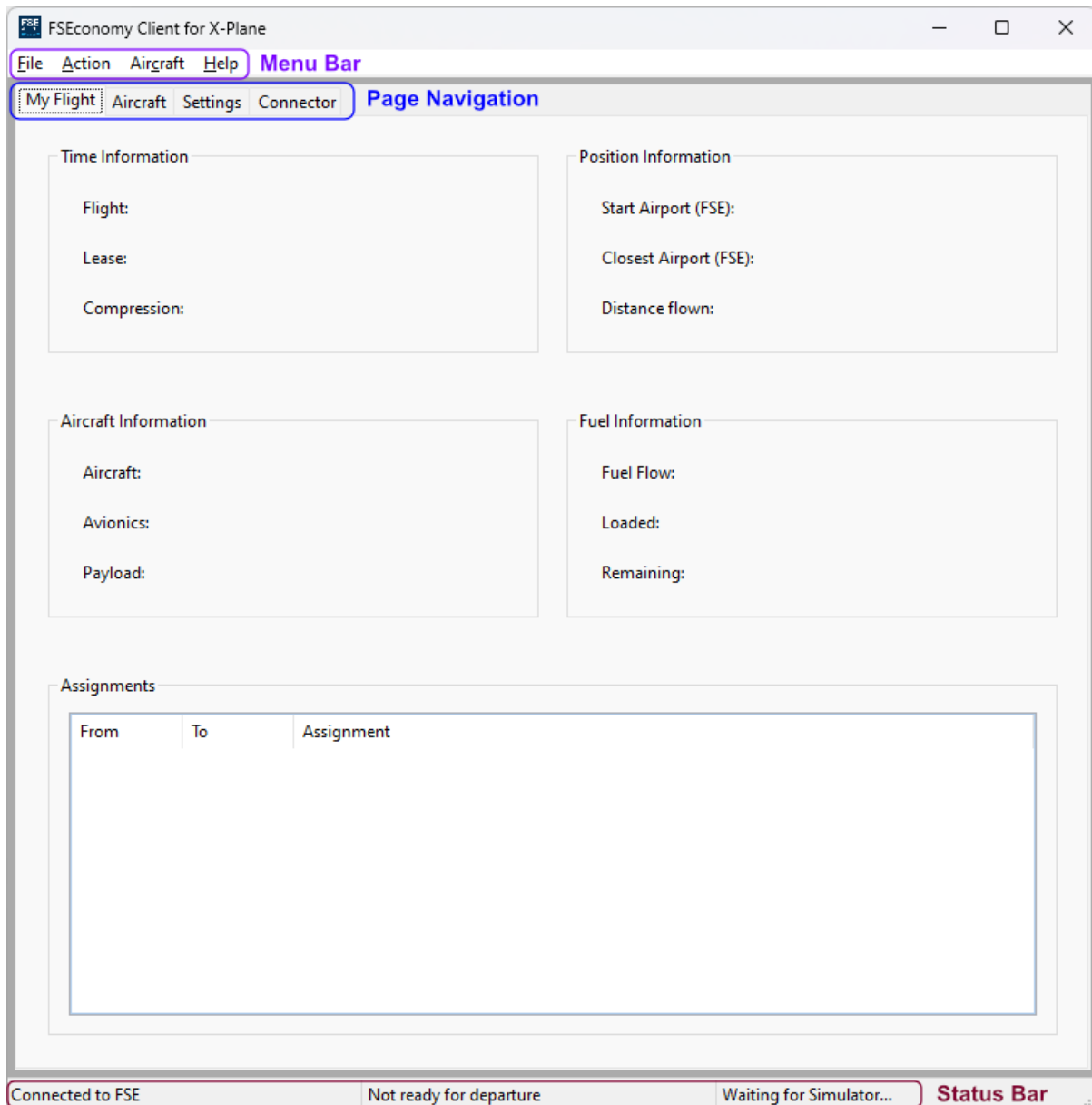
- from your `Resources/plugins/PythonScripts` folder:
 - `PI_xfse-fuel tanksconfig.py` and `PI_xfse-fuel tanksconfig.pyc`
 - `PI_xfse-interface.py` and `PI_xfse-interface.pyc`
- from your `Resources/plugins/FlyWithLua/Scripts` folder:
 - `FSE_Interface.lua`

If you did not use the Python Interface plugin for any other purpose than running the FSEconomy client, you can also

- remove these folders from your `resources/plugins` directory:
 - `PythonInterface`
 - `PythonScripts`
- uninstall Python 2.7 (only if you're sure you don't need it for any other software) — recommended since Python 2.7 is known to be vulnerable to certain attacks and should not be used any longer.

Chapter 2. Application Overview

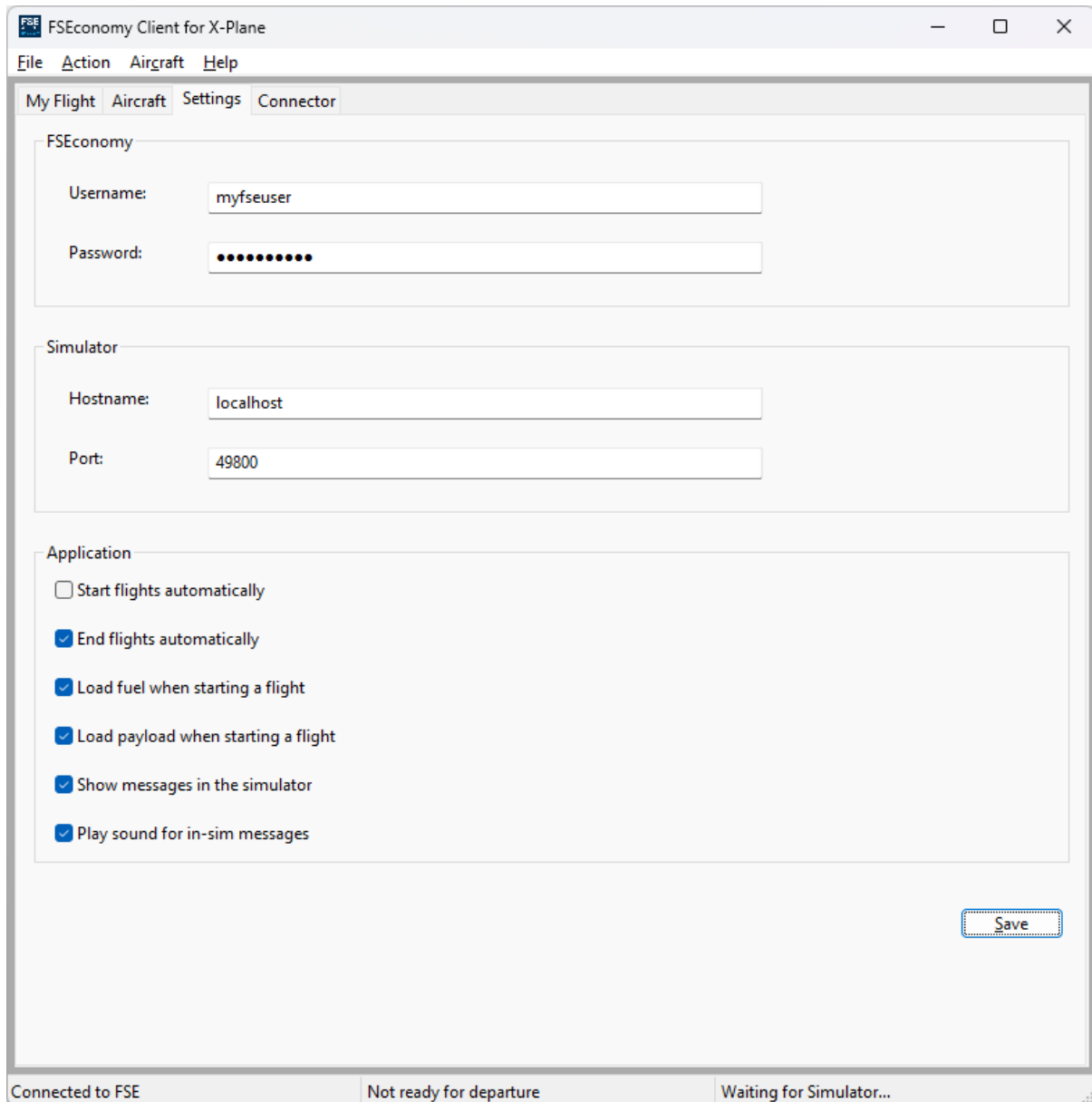
XFSE is a single window application, with a straightforward user interface:



The menu bar gives access to commands, such as exiting the application, and starting, ending or canceling a flight. The status bar gives a quick feedback on the current status (e.g. if the application could connect to FSEconomy or your simulator, and whether there's a flight in progress or not).

Chapter 3. General Settings

When starting the XFSE application for the first time, you will need to go through some basic configuration steps. To do so, select the Settings page from the page navigation:



The screenshot shows the 'FSEconomy Client for X-Plane' window with the 'Settings' tab selected. The window has a menu bar with 'File', 'Action', 'Aircraft', and 'Help'. Below the menu bar are four tabs: 'My Flight', 'Aircraft', 'Settings', and 'Connector'. The 'Settings' tab is active, displaying three sections: 'FSEconomy', 'Simulator', and 'Application'. The 'FSEconomy' section has 'Username' set to 'myfseuser' and 'Password' masked with dots. The 'Simulator' section has 'Hostname' set to 'localhost' and 'Port' set to '49800'. The 'Application' section has six checkboxes: 'Start flights automatically' (unchecked), 'End flights automatically' (checked), 'Load fuel when starting a flight' (checked), 'Load payload when starting a flight' (checked), 'Show messages in the simulator' (checked), and 'Play sound for in-sim messages' (checked). A 'Save' button is located at the bottom right of the settings area. At the bottom of the window, there is a status bar with three indicators: 'Connected to FSE', 'Not ready for departure', and 'Waiting for Simulator...'.

3.1. FSEconomy Credentials

In the FSEconomy section, you have to enter your FSEconomy game world username and password. Make sure they're spelled correctly; in particular the password is case-sensitive and will not be accepted by the server otherwise.

Click the *Save* button. This will store your login credentials permanently in an encrypted file to protect them from theft or manipulation, but at the same time sparing you the hassle of entering them over and over again.

Monitor the status message in the left part of the status bar after entering and saving your credentials. If the FSEconomy game world server is up and running, it should change to *Connected to FSE* a few moments after saving your credentials.

3.2. Simulator Connection

If you run the XFSE application and X-Plane on the same computer, keep the default settings for Hostname and Port.

IMPORTANT

When using XFSE with the older XPlaneConnect plugin, the port needs to be set to **49009**.

X-Plane and XFSE can be installed on different computers using a network connection. To configure this network connection, set the two options appropriately:

- **Hostname:** enter the IP address of the computer running X-Plane
- **Port:** enter the appropriate port number of the computer running X-Plane. Normally, this can remain unchanged from the default unless you modify the configuration of the XfseConnect plugin, or you have configured port redirects via a firewall or routing solution

Confirm the *Hostname* or *Port* settings by clicking on the *Save* button. This will permanently store the settings you entered and re-initialize the connection to the simulator.

The status of the connection to the simulator is shown in the right part of the status bar. However, the connection will only be shown as active if

- the simulator is up and running
- has the connector plugin installed and working correctly
- and is in flight mode (i.e. aircraft and scenery are loaded).

NOTE

While the simulator is in any configuration menu (flight setup, or general settings), the connector plugin is paused, so for the client it will look as if the simulator was not running.

Also, please note that the simulator will only reliably send data to the client as long as it runs above the magic threshold of 20 FPS or more. If the simulator runs slower, it doesn't work in real time anymore. The client can deal with stutters during a flight, but please do not attempt to issue a command (such as starting or ending a flight) while the simulator stutters — it will eventually fail.

3.3. Application Settings

These settings control the default behaviour of XFSE, and are also referred to as “global settings”. As with any setting on the *Settings* page, changing one of the checkboxes requires the Save button to be pushed; otherwise the setting change will not be recognized.

3.3.1. Start Flights Automatically

Default: Off

This setting lets you chose whether you want the XFSE application to automatically start a flight as soon as it detects the corresponding conditions:

- At least one engine must be running
- The parking brake must be released (ignored for aircraft with skids instead of wheels)
- The ground speed must exceed a threshold of 3 knots for more than 2 seconds

3.3.2. End Flights Automatically

Default: Off

This setting lets you choose whether you want the XFSE application to automatically report your flight to the server as soon as the client detects your flight as completed. If you previously used one of the Microsoft simulators with FSEconomy, this behavior will be familiar. The following criteria must be met:

- The total flight time must be longer than one minute
- The aircraft stands still on the ground
- The parking brake must be set for more than 5 seconds (ignored for aircraft with skids), or the engine(s) must be off
- The aircraft must have taken off before (left the ground)

The 4th criterion prevents the client from auto-reporting a flight while you’re doing the engine run-up check or holding short with the parking brake set.

NOTE

The auto end flight setting will not prevent you from manually ending & reporting a flight — this will always be possible as long as the first three criteria to end a flight are met.

3.3.3. Load Fuel When Starting a Flight

Default: On

When disabled, XFSE will no longer load fuel into your aircraft (in the simulator) when starting a flight. Fuel checks do however still apply, i.e. it is your responsibility to ensure the aircraft's tanks are filled appropriately prior to starting the flight.

NOTE

While this option is useful for aircraft with center of gravity issues or custom fuel systems, it also introduces the risk of losing big amounts of fuel on the FSEconomy side of things. Therefore, we recommend to ***not deactivate it globally***. Instead, opt out per aircraft where needed (cf. [Override Settings](#)).

3.3.4. Load Payload When Starting a Flight

Default: On

When disabled, XFSE will no longer load payload into your aircraft (in the simulator) when starting a flight. Currently, there are no payload checks in place. Consequently, it is up to you as responsible pilot how you manage payload loading.

This option can be helpful for aircraft with center of gravity issues — particularly since the introduction of load stations in X-Plane 12, CG issues have become more common when “blindly” adding bulk payloads.

3.3.5. Show Messages in the Simulator

Default: On

By default, the client will display important messages on screen in the flight simulator. This applies particularly to auto-ending or auto-starting a flight, and auto-canceling a flight.

These messages appear in green writing, usually appearing in the upper center area of your screen. These messages show for a duration of approximately 5 seconds. You can disable these messages by un-ticking the checkbox and saving the settings.

NOTE

When using the newer XfseConnect plugin, the messages are also shown in VR. The older XPlaneConnect plugin does not support VR.

3.3.6. Play Sound for in-Sim Messages

Default: On

Alongside with the visual messages described above, XFSE can also play a complementing sound. Positive messages are accompanied by a two-tone chime (“ding-dong”), while negative messages are marked by a triple chime (“breep-breep-breep”).

The main intention of these chime sounds is to give VR users some sonic guidance — a successful auto-start e.g. is signalled by the friendly two-tone chime. If you hear this sound when you expect autostart to kick in, there’s no need to visually check the XFSE app window (which means leaving VR).

NOTE

When disabling both, visual and sonic notification, XFSE will not notify you about success or failure of auto-starting or auto-ending a flight. You will need to check the XFSE app window’s display to figure out the current status.

Chapter 4. Aircraft Configuration

Before starting a flight, in most cases it's necessary and desirable to amend the aircraft configuration (alias and fuel tank mapping / priority). XFSE bundles this functionality in a convenient interface, accessible via the *Aircraft* page on the page navigation.

IMPORTANT

The Aircraft page needs an active simulator connection, with the targeted aircraft model loaded in your simulator. Also, you can only configure an aircraft model as long as there's no active flight in progress.

XFSE introduces one significant change vs. the old, legacy X-Plane client: while in the old world all aircraft configuration (alias and fuel tank priority) applied to a *folder* (and all aircraft *.acf* files within), XFSE identifies aircraft based on meta information set by the publishing studio or developer, namely:

- ICAO code
- aircraft description
- aircraft notes
- aircraft author

In addition, the loaded livery is taken into account, so it's possible to assign different aliases based on the livery loaded.

4.1. Aircraft Alias

Setting an alias will be required in most cases, since X-Plane aircraft model designations are usually not added to the FSEconomy alias database. Based on what information it detects from X-Plane, the XFSE application will propose an alias, which will likely need some amendment:

Aircraft Alias

① ICAO code: C172

② Aircraft Description: Cessna 172 SP Skyhawk - 180HP

③ Aircraft Author: Laminar Research - dmax3d.com

④ Livery: old_style

⑤ Alias: Cessna 172 SP Skyhawk - 180HP

The *Aircraft* field shows the aircraft identifier, combining various information to create a unique fingerprint, identifying an aircraft. It is auto-generated and cannot be modified.

- ICAO code (1)
- Aircraft Description (2)
- Aircraft Author (3)
- Livery (4)

The Alias field is editable. Initially the XFSE application automatically proposes the aircraft description (5), though this field needs to be amended to hold an alias defined in the FSEconomy aircraft database (cf. [Aircraft Models and Mapping](#) in the FSEconomy Operations Guide).

IMPORTANT

After changing the alias of an aircraft, don't forget to hit the *Save* button — this will make your edit permanent (the XFSE application will remember from now on), and also publish the change to other parts of the application (i.e. after having saved a new alias, it will be used when starting a flight with this aircraft). Without saving the alias, it will not be used when starting a flight, most likely resulting in a failed attempt to start the flight.

4.2. Fuel Tank Configuration

4.2.1. FSEconomy vs. X-Plane

FSEconomy uses a system of up to 11 fuel tanks, with

- three left and right tanks each (main, aux, tip),
- three center tanks (center, center 2 and center 3),
- and two external tanks (ext 1 and ext 2).

X-Plane on the other hand uses a system of up to 9 fuel tanks, of which none is dedicated to a specific purpose — they only have a numeric index (0 to 8) and can be flexibly used by aircraft developers for whatever purpose. So one developer could use the tank with index zero as the left main tank, while another developer uses tank index zero as the left tip or aux tank, or center tank, etc.

To complicate things a bit, X-Plane does not simply specify the size of each tank. Instead, X-Plane maintains a maximum total fuel, and an array of ratios — one for each tank. The ratios have to sum up to 1.0, so the fuel a tank can hold has to be calculated by multiplying the max total fuel with the tank's ratio. To add to this mayhem, X-Plane specifies tank capacity and fuel levels by weight, not by volume, whereas FSEconomy handles fuel (on the flight side of things) in gallons.

4.2.2. Automatic Configuration

The easiest way to deal with all this is having the client do it for you. In this case, there's no need to modify the proposed configuration:

Tank Index	Tank Size	Source (FSE)	Fill Priority
0	50% / 79.38 kgs	auto	0
1	50% 79.38 kgs	auto	0

- Tank Index in X-Plane (1)
- Tank Ratio — how much of the total fuel goes into this tank (2)
- Resulting max. fuel weight the tank can hold (3)

Just keep *Source* and *Fill Priority* set to **auto** and **0**, and the client will automatically distribute fuel evenly among the tanks when loading fuel into the aircraft (using the tank ratio). So without modifying the configuration, the behavior of the XFSE application is identical to the behavior of the old legacy client in that respect.

4.2.3. Control Refueling by Fill Priority

With some more complex tank configurations however, spreading the fuel out evenly among tanks might lead to an undesired result. Let's take for example a Boeing 737, coming with three tanks:

Tank Index	Tank Size	Source (FSE)	Fill Priority	
0	19% / 3,907.15 kgs	auto	0	Left Main
1	63% / 13,079.54 kgs	auto	0	Center Tank
2	19% / 3,907.15 kgs	auto	0	Right Main

If the client would now load fuel, it would load most of the fuel into the center tank — clearly not how you'd fuel up a Boeing 737. What we want here is first loading up the wing tanks, and only the fuel not fitting into them should go to the center tank.

Just as with the old client and the additional fuel tank configuration script, this can be achieved by giving the tanks a *Fill Priority*. Tanks with lower priority are filled first, tanks with higher fill priority are filled later with what's left.

NOTE

Priorities decrease with the numeric value representing a priority — think of it as the order in a “first come, first served” queue. Consequently, **0** is the *highest* priority available, **8** the *lowest*.

A good priority-based fuel tank configuration for the Boeing 737 could look like this:

Tank Index	Tank Size	Source (FSE)	Fill Priority	
0	19% / 3,907.15 kgs	auto	0	Left Main
1	63% / 13,079.54 kgs	auto	1	Center Tank
2	19% / 3,907.15 kgs	auto	0	Right Main

With this configuration, the client will take the fuel received from the server, and first fill up the left and right wing tanks (evenly). What's remaining will go into the center tank.

NOTE

The XFSE application allows you to define arbitrary priority levels. If you want to, you can set each of the 9 tanks on its own priority level. Since this could lead to a severe fuel imbalance, best research which tank index represents which tank, before going down that road. If in doubt, ask the aircraft developer or reach out to the community for help.

4.2.4. Control Refueling by Tank Mapping

The XFSE application offers an alternate way of controlling how much fuel will go into which tank. This method of controlling the refuel process was not present in the old legacy client. It is similar to how the Microsoft Flight Simulator clients manage fuel.

Going back to the example of the Boeing 737 from the previous chapter, you may have noticed I already scribbled the tank names next to the tank indices:

Fuel Tank Configuration				
Tank Index	Tank Size	Source (FSE)	Fill Priority	
0	19% / 3,907.15 kgs	auto	0	<i>Left Main</i>
1	63% / 13,079.54 kgs	auto	0	<i>Center Tank</i>
2	19% / 3,907.15 kgs	auto	0	<i>Right Main</i>

Coincidentally, the Boeing 737 has the very same configuration in the FSEconomy database:

Ext 1	0
L Tip	0
L Aux	0
L Main	1288
Center	4299
Center 2	0
Center 3	0
R Main	1288
R Aux	0
R Tip	0
Ext 2	0

In such a case, where the FSEconomy and the X-Plane tank configurations match 1:1, it is possible to define a mapping of tanks, by setting the *Source* field for each tank:

Fuel Tank Configuration				
Tank Index	Tank Size	Source (FSE)	Fill Priority	
0	19% / 3,907.15 kgs	Left Main	0	<i>Left Main</i>
1	63% / 13,079.54 kgs	Center	0	<i>Center Tank</i>
2	19% / 3,907.15 kgs	Right Main	0	<i>Right Main</i>

This will tell the XFSE application to load whatever it received from the server for the left

main tank into the tank index zero, what it received for the right main tank into tank index two, and what it received for the center tank into tank index 1.

NOTE

For tanks with a defined source, the application will ignore the fill priority, so the mapping overrules the fill priority order.

A word of caution: since the mapping overrides any other rule (fill priority, tank share, etc.), the refuel result will strongly depend on the distribution of fuel among the tanks as the FSEconomy server sees it. If the server sees most of the fuel in the center tank, the XFSE application will load fuel accordingly. If the server sees a severe imbalance between left and right wing tanks, the XFSE application will also oblige and produce the same imbalance in your simulator.

Also, please note that fuel not fitting into the mapped tank will be lost (spilled) when loading fuel into the aircraft. An incorrect mapping can therefore lead to a lot of fuel being spilled on the apron. Should that happen, it is recommended to cancel the flight — do not end & report it because **the fuel will be definitely lost and cannot be recovered!** Review the mapping configuration before starting it again.

4.2.5. Combining Fuel Control Methods

Some aircraft models come with highly complex fuel tank configurations. In these edge cases it may be interesting to combine the mapping and prioritization methods. You can do this, if you consider these rules:

- choose one method per tank index — either mapping, or prioritization (priority will be ignored for tank indices with an assigned mapping)
- mapped tanks will be filled first, and fuel not fitting will be lost
- remaining fuel (from not mapped tanks) will be allocated to the not mapped tank indices, respecting the indicated priority

NOTE

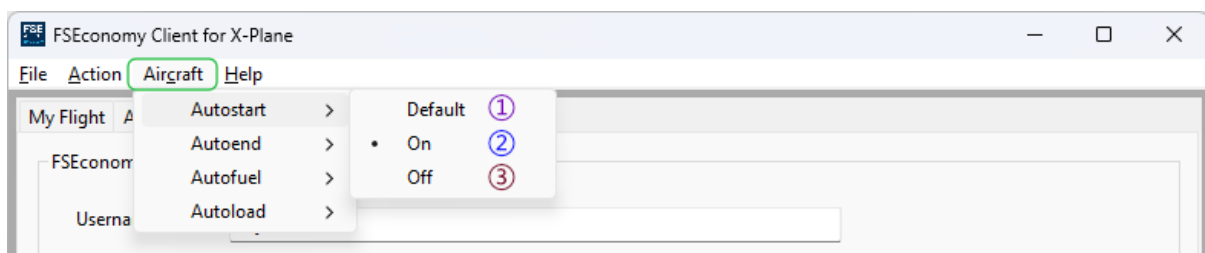
Whichever method or combination of methods you chose, don't forget to **save the configuration** — only after having saved it, the application will use it to determine fuel levels per tank when starting a flight.

4.3. Override Settings

XFSE offers pilots the option to override four global [Application Settings](#) for a specific aircraft, namely:

- Autostart — [Start Flights Automatically](#)
- Autoend — [End Flights Automatically](#)
- Autofuel — [Load Fuel When Starting a Flight](#)
- Autoload — [Load Payload When Starting a Flight](#)

The per-aircraft behavior towards these settings can be controlled via the *Aircraft* menu:



For each of the settings, there are three options available:

- **Default (1):** stick with the global setting (enabled by default)
- **On (2):** enable this setting, even if the feature is switched off globally
- **Off (3):** disable this setting, even if the feature is switched on globally

To manipulate these overrides, XFSE must be connected to the simulator, and the aircraft you want to set the overrides for must be loaded. Selecting an override in the menu immediately activates your choice; you don't need to explicitly save the settings. XFSE does remember these settings, i.e. you have to set them only once for an aircraft.

NOTE

These override options do what their name says—they *override* the default behavior of XFSE with selected aircraft. Please remember to double-check them before reporting a perceived inconsistent behavior of XFSE, since they might well be the root cause for the experienced inconsistency.

Chapter 5. Flying

5.1. Starting a Flight

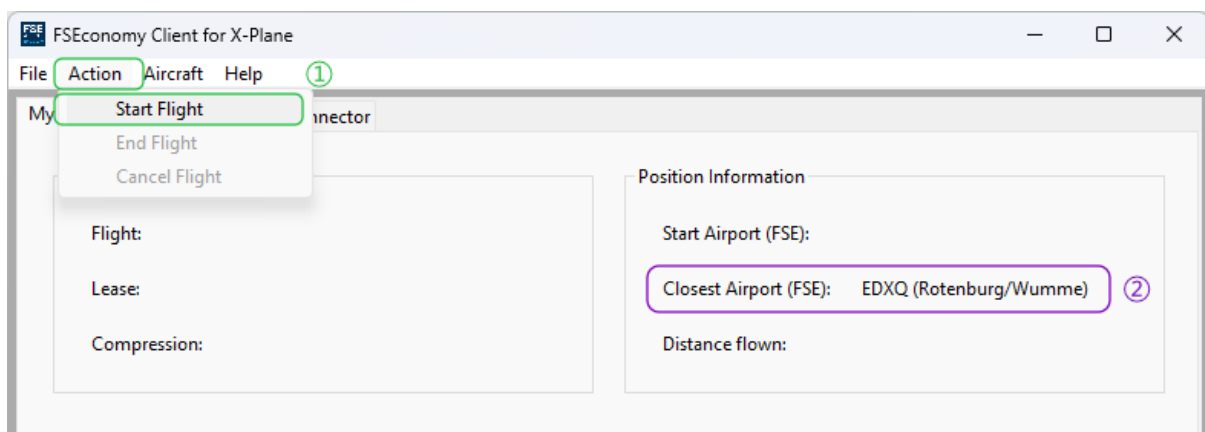
Before starting a flight, these prerequisites must be met:

- the client must be connected to the FSEconomy server (“Connected to FSE” in lower left corner)
- the client must be connected to X-Plane (“Connected to X-Plane” in lower right corner)

Also, please review the aircraft configuration page (alias and fuel tank settings) prior to starting a flight—this is particularly important when using an aircraft for the first time with FSEconomy.

5.1.1. Manual Start

If both criteria described above (simulator and FSEconomy server connection) are met, the *Start Flight* entry in the *Action* menu (1) will become available (otherwise it remains inactive and cannot be selected):



As soon as the client has established a connection to the simulator, it will start displaying *Position Information* (2). XFSE reads latitude and longitude data from the simulator, and uses them to calculate the *Closest Airport (FSE)*. It represents the airport FSEconomy will most likely recognize your aircraft to be at when starting or reporting the flight.

NOTE

Use the *Closest Airport (FSE)* information with a grain of salt. It can be a great help when navigating at airports split into different FSEconomy codes (e.g. PANC). And while it should be really close to what the FSE server would conclude, there might be edge cases where the client and server disagree due to computational limits in floating point precision.

5.1.2. Automatic Start

If the global [autostart](#) option or the corresponding [per-aircraft override](#) is enabled, XFSE will attempt to automatically start the flight if the following criteria are met:

- At least one engine must be running
- The parking brake must be released (ignored for aircraft with skids instead of wheels)
- The ground speed must exceed a threshold of 3 knots

Typically, you will meet these conditions when starting your taxi roll to the runway (when starting on the ramp), or when starting your takeoff roll (when starting on the runway).

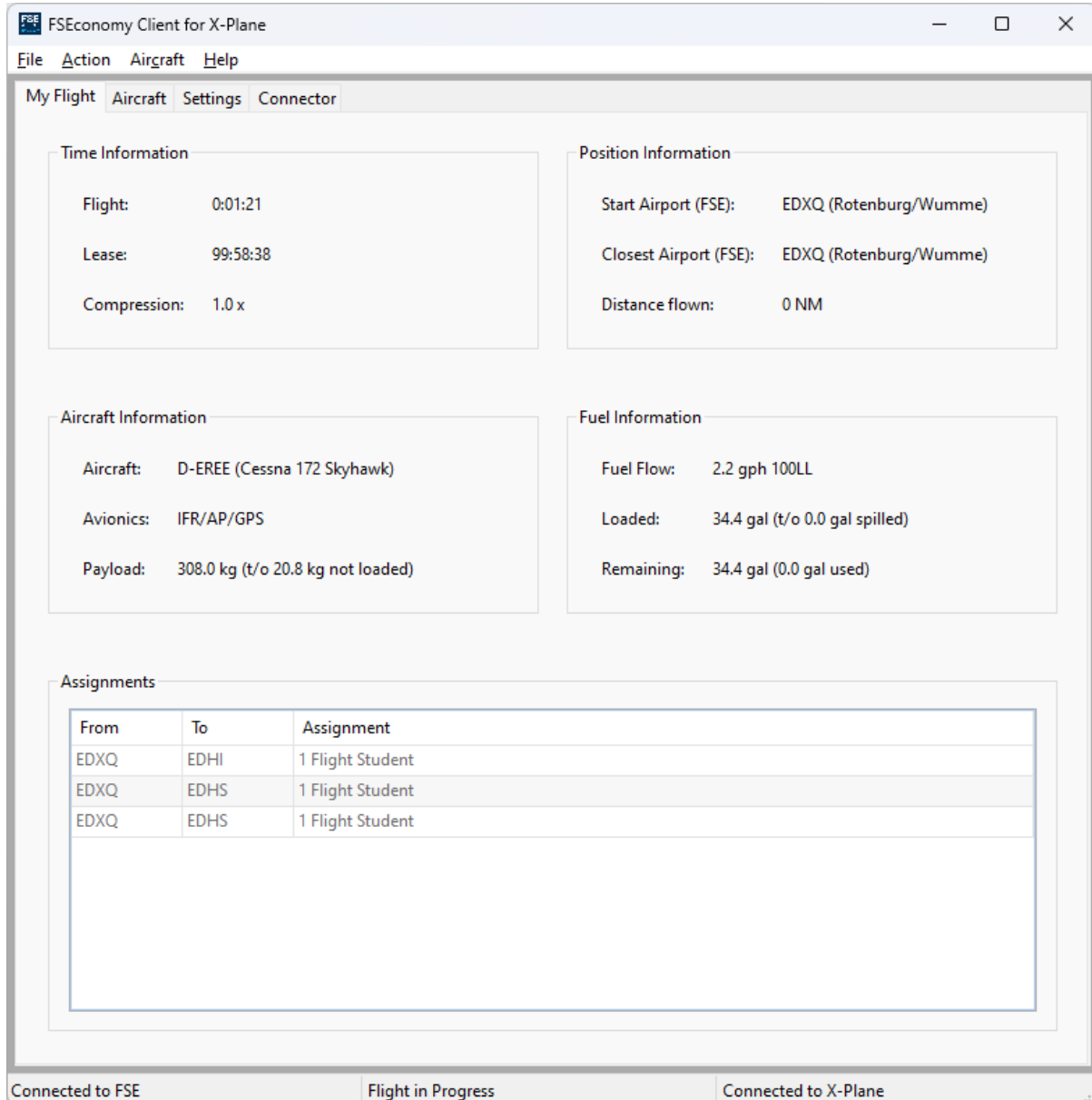
XFSE will send a notification to the simulator if the [corresponding notification settings](#) are enabled, confirming a successful start, or warning about a failed attempt.

NOTE

If a previous flight has been cancelled or otherwise terminated, XFSE will not attempt to auto-start another flight unless the user has acknowledged the corresponding message box in the XFSE user interface. While this intermediate step might seem annoying for VR users, it is crucial to avoid XFSE getting stuck in limbo between auto-starting and auto-cancelling a flight under certain conditions (e.g. if the aircraft in the simulator has too much fuel on board, and auto-fuel has been disabled).

5.2. Enroute Monitoring

Once a flight has been successfully started, you can use the *My Flight* page to monitor some information regarding your flight:



The screenshot shows the 'FSEconomy Client for X-Plane' window. It has a menu bar with 'File', 'Action', 'Aircraft', and 'Help'. Below the menu is a tabbed interface with 'My Flight', 'Aircraft', 'Settings', and 'Connector'. The 'My Flight' tab is active, displaying four panels: 'Time Information', 'Position Information', 'Aircraft Information', and 'Fuel Information'. At the bottom is an 'Assignments' table. The status bar at the very bottom shows 'Connected to FSE', 'Flight in Progress', and 'Connected to X-Plane'.

Time Information

Flight:	0:01:21
Lease:	99:58:38
Compression:	1.0 x

Position Information

Start Airport (FSE):	EDXQ (Rotenburg/Wumme)
Closest Airport (FSE):	EDXQ (Rotenburg/Wumme)
Distance flown:	0 NM

Aircraft Information

Aircraft:	D-EREE (Cessna 172 Skyhawk)
Avionics:	IFR/AP/GPS
Payload:	308.0 kg (t/o 20.8 kg not loaded)

Fuel Information

Fuel Flow:	2.2 gph 100LL
Loaded:	34.4 gal (t/o 0.0 gal spilled)
Remaining:	34.4 gal (0.0 gal used)

Assignments

From	To	Assignment
EDXQ	EDHI	1 Flight Student
EDXQ	EDHS	1 Flight Student
EDXQ	EDHS	1 Flight Student

Connected to FSE Flight in Progress Connected to X-Plane

5.2.1. Status Information

While airborne, the status bar will keep you informed on your connections to the FSEconomy server and the simulator, as well as the status of your flight.

The flight status is displayed in the center field of the status bar. *Flight in Progress* means the flight is airborne, and can currently not be ended (e.g. due to still being airborne, or other criteria to end a flight not yet being met). If all mandatory criteria to end a flight are met, the status will change to *Ready to Report*.

5.2.2. Time Information

The upper left box shows timing information:

- *Flight* indicates the elapsed flight time. This timer will pause while the simulator is paused, so it will only account for the net flight time. Once a flight gets reported, this timer constitutes what you see as flight time in your flight log in FSEconomy.
- *Lease* is a countdown, showing the remaining lease time (based on the max. allowed rental time for the aircraft currently flown). The lease timer will not pause while the sim is paused, so pay close attention to not running out of lease time. Once this timer reaches zero, the flight will be canceled.
- *Compression* is only indicated for convenience. It represents the currently commanded combined total compression (time acceleration and ground speed acceleration). The threshold for the commanded compression is 16x—be careful to never exceed this value (even if the actual compression might be lower), as it will lead to immediate cancellation of the flight.

5.2.3. Position Information

The upper right box shows information regarding the current position of the aircraft in your simulator:

- *Start Airport (FSE)* is the client's interpretation of latitude and longitude captured when starting the flight.
- *Closest Airport (FSE)* is the client's interpretation of current latitude and longitude captured. It represents the airport FSEconomy will most likely recognize your aircraft to be at when starting or reporting the flight.
- *Distance flown* is the great circle distance between the coordinates (latitude and longitude) captured when starting the flight, and the current aircraft position (**not** start airport and current airport!).

NOTE

Position data for the user interface are retrieved from the simulator and calculated asynchronously, meaning what you see here can be where the aircraft in the simulator was one or two seconds ago. If your aircraft is standing still and not moving (or the simulator gets paused), the information stabilizes and stops updating. Such a stable reading can be trusted and used to judge your actual position.

5.2.4. Aircraft Information

The lower left box shows information regarding the aircraft used to conduct the current flight.

- *Aircraft* indicates the registration and aircraft model as seen by FSEconomy (no matter what aircraft you actually use in your simulator).
- *Avionics* shows the equipment available to this aircraft as per FSEconomy's database. The client will block any avionics not installed from being used. Should you experience issues with the autopilot or GPS, check this information first.
- *Payload* shows the amount of payload (including crew weight, but without fuel) being loaded into your aircraft. If the aircraft you fly in X-Plane cannot take the full payload, the value in brackets will show you what was left behind. This information has no influence on what will be reported in FSEconomy, it's just for pilot's information, warning about operating at the MTOW limit of the aircraft loaded in the simulator.

NOTE

When [autoload](#) is disabled, *Payload* shows the amount of payload on board read from the simulator when starting the flight, alongside with the amount of payload loaded in FSEconomy.

5.2.5. Fuel Information

The lower right box shows information regarding fuel loaded and consumed for the flight.

- *Fuel Flow* indicates the current fuel flow and type of fuel used for this flight. The fuel flow displayed is a moving average of the past 5 readings from the simulator for *all* fuel consumers together (including e.g. a running APU). The fuel type displayed is not based on the aircraft loaded in X-Plane, but on the aircraft rented in FSEconomy. This information is useful when estimating the fuel weight, since both different fuel types use their own specific weight for X-Plane.
- *Loaded* indicates the amount of fuel loaded according to FSEconomy. The value in brackets indicates whether all the fuel sent by the server could be loaded into the aircraft, or whether some fuel has been spilled (due to fuel tank size mismatch or over-fueling).
- *Remaining* indicates how much fuel is currently still available in the aircraft's tanks (data retrieved from the simulator). The value in brackets shows how much fuel has already been consumed (excluding the spilled fuel).

NOTE

When [autofuel](#) is disabled, *Loaded* shows the amount of fuel on board read from the simulator when starting the flight, alongside with the amount of fuel loaded in FSEconomy.

Specific Fuel Weights Used by XFSE

100LL: 2.68735000 kg per gallon / 5.92459260 lb per gallon

JetA: 3.08447722 kg per gallon / 6.80010826 lb per gallon

Factor for weight conversion: 0.45359237 kg per lb

5.2.6. Effective Monitoring Strategy

You can use the information on the *My Flight* page to predict whether your flight will be accepted by the FSEconomy server, or if you risk a rejection due to speeding, exceeding the aircraft's range or a too low fuel consumption. This can be pretty useful, particularly when aliasing an aircraft that's faster or has better fuel economy than its FSEconomy equivalent. To achieve this, you will need to establish two pieces of information during your flight planning:

- The *Flight Time Envelope* can be calculated using this formula: $\text{Flight Time Envelope} = \text{GC Distance} / \text{Aircraft Speed}$, where *GC Distance* is the great circle distance between your start and destination airport, and *Aircraft Speed* is the aircraft's speed as recorded in FSEconomy's database.
- The *Fuel Consumption Envelope* can be calculated using this formula: $\text{Fuel Consumption Envelope} = \text{Flight Time Envelope} \times \text{Aircraft Fuel Flow}$, where *Aircraft Fuel Flow* is the aircraft's fuel flow as recorded in FSEconomy's database.

Monitor elapsed flight time and fuel used on the *My Flight* page — both display readings must be equal to or greater than the envelope values at the end of the flight.

5.3. Cancelling a Flight

5.3.1. Manual Cancellation

In case you need to cancel a flight, you can do so conveniently from the *Action* menu. Please note the *Cancel Flight* action will only become available while a flight is in progress. In case of a successful cancellation you will be notified by an information dialog and a message displayed in the simulator.

Canceling a flight through the client is equivalent to shutting down the client, canceling the flight on the FSEconomy game world website and starting the client up again.

5.3.2. Automatic Cancellation

In some situations, the client will initiate a cancellation of the current flight on its own. This is specifically the case when one of the client's anti-cheat mechanisms is triggered, or if the client loses the connection to the simulator and remains unsuccessful reestablishing it.

NOTE

A loss of connection to the simulator does not only happen when the simulator crashes or gets closed, but also when the simulator becomes unresponsive from the client's perspective. This may be the case when the simulator "hangs" (e.g. loading scenery), or when a configuration dialog in the simulator is opened. The client will gracefully allow for short disruptions to occur, but will not tolerate longer simulator outages. Therefore, please do not open configuration dialogs (e.g. to alter keyboard or control bindings) while a flight is in progress.

If the client cancels a flight on its own accord, you will be notified by a dialog box and a message appearing in the simulator, giving reason for the cancellation.

If the server is unresponsive while the client cancels a flight (manually or automatically), the client will continue to attempt canceling the flight also at the server. You can however shut the client down anytime, and cancel the flight manually on the FSEconomy game world website.

5.4. Ending a Flight

The current flight can be ended as soon as the following criteria are met:

- The total flight time must be longer than one minute
- The aircraft stands still on the ground
- The parking brake must be set for more than 5 seconds (ignored for aircraft with skids instead of wheels), or the engine(s) must be shut down.

The client monitors these conditions and will allow ending a flight once these criteria are met.

5.4.1. Manual Ending

As soon as the above-mentioned criteria are met, you can manually end the flight through the *End Flight* command in the *Action* menu. Ending a flight manually is only required if

- you chose to not use the auto-end option or
- you need to end the flight before having left the ground

The latter may occur if you have to taxi your aircraft from one airport recognition zone to another airport recognition zone, or if you need to burn off some fuel (dry rental only!).

5.4.2. Automatic Ending

If the global [auto-end](#) option or the corresponding [per-aircraft override](#) is enabled, XFSE will automatically end and report a flight to the server as soon as the above-mentioned criteria are met, as long as the aircraft has previously left the ground.

The client will send an in-simulator message upon auto-ending a flight, informing you about the success or failure of the operation (exception: you deactivate the [corresponding notification settings](#)), but will not raise a message box in the application, unless it needs to report a critical error.

TIP

If the client fails to report a flight to the server due to server downtime or maintenance, just keep the client and the simulator open and wait patiently for the server to become available again — XFSE will automatically complete the report as soon as the server starts responding.

Chapter 6. Troubleshooting

6.1. XFSE Detected as Malware

We're aware that some virus scanners falsely detect XFSE as malware — depending on the scanner, the results vary among different trojans (**Win32/Wacatac.B!ml** is a pretty popular theory of Microsoft Defender Antivirus). The reason this happens is the way XFSE is packaged into an executable. Unfortunately, changing this is not an option in the near future. We have submitted XFSE's executable multiple times to various security teams among the different antivirus companies; the answer has always been "false positive".

Therefore, if you encounter this situation, please submit the client as suspected false positive to the security team of your antivirus solution (some companies don't accept anonymous submissions, so we can do this only for the small selection of antivirus products the development team members have a license for).

Here is a list of links to submit false positives for several popular antivirus solutions:

- Microsoft Defender Antivirus: <https://www.microsoft.com/en-us/wdsi/filesubmission>
- BitDefender: <https://www.bitdefender.com/consumer/support/answer/29358/>
- TotalAV: <https://www.totalav.com/submit-file>
- Norton: <https://symsubmit.symantec.com/> and <https://submit.norton.com/>
- McAfee: <https://www.mcafee.com/en-us/consumer-support/dispute-detection-allowlisting.html>

6.1.1. Microsoft Defender Antivirus

On several occasions we found that outdated virus definitions and a stale detection history kept Defender blocking XFSE's executable, even if the file had been cleared by Microsoft's security team. Therefore, please try this before reporting an issue to the FSEconomy development team: Open a command line window with administrator privileges and issue these commands:

```
MpCmdRun.exe -removedefinitions -dynamicsignatures
MpCmdRun.exe -SignatureUpdate
del /f "C:\ProgramData\Microsoft\Windows Defender\Scans\History\*
```

After completing the steps above, open the Windows Defender settings and toggle the Real-time protection and Cloud-delivered protection services off and on.

6.2. Log Files

6.2.1. XFSE Application Log

The XFSE application keeps record of any mishaps and also execution of some routine tasks in its log file. The log file is the first place to look at when troubleshooting a problem. You may find the XFSE log file under

- `%LOCALAPPDATA%\FSEconomy\xfse\Logs\xfse_log.txt` (Windows)
- `~/Library/Logs/xmse/xmse_log.txt` (macOS)
- `~/.cache/xmse/log/xmse_log.txt` (Linux)

If you feel the default log level isn't verbose enough to diagnose your problem, you can enable debug-mode logging by passing `--debug` as command line argument to the XFSE executable.

NOTE

We strongly recommend to limit using the `--debug` flag to troubleshooting — XFSE becomes very talkative and logs huge amounts of data when run with this flag. Consequently, the log file can quickly become quite large.

6.2.2. Connector Log

XfseConnect

The XfseConnect plugin writes its log entries to X-Plane's default `log.txt` log file. All log entries written by XfseConnect follow this pattern:

```
1:44:49.246 [XFSE-CONNECT]: INFO: Shut down UDP server
```

The first part is the timestamp (based on X-Plane's uptime), followed by an identifier, the log entry's criticality and the message itself.

By default, XfseConnect only logs messages with a criticality of **INFO** or higher, and suppresses **DEBUG** level messages. For troubleshooting purposes, you can increase the plugin's verbosity by modifying the **Output/XfseConnect/conf.txt** file (path relative to X-Plane's root directory).

WARNING

Never run XfseConnect in debug mode except for troubleshooting purposes. Verbose logging has a significant impact on X-Plane's runtime performance; i.e. it will seriously hurt X-Plane's FPS.

XPlaneConnect

The XPlaneConnect plugin writes a dedicated log file called **XPCLog.txt**, which can be found in the X-Plane root folder. XPlaneConnect does not allow you to reduce or increase the amount of information logged.

NOTE

Like X-Plane, the XPlaneConnect plugin starts a fresh log file each time you start X-Plane, so if you need it to report an issue, make sure to copy the file to a safe location before starting X-Plane again.

6.3. Configuration Files

The XFSE application stores its configuration information in four files:

- `config.sec` is a binary format and contains encrypted information. **Never share this file**, you will never need to reveal its content when asking for support.
- `config.json` holds the global application configuration
- `aircraft.json` holds all the aircraft configurations you created (aliases and fuel tank configurations).
- `wx_ui.json` holds window position and size information, restoring the XFSE main window at the same place where it was when the application was shut down the last time.

While it is not recommended to manually edit any of the `.json` files, particularly the content of the `aircraft.json` file might be requested when reporting a problem related to a specific aircraft. Removing (or renaming) all four configuration files will reset the XFSE application to its original state and enforce the creation of three fresh, blank configuration files.

You can find the configuration files in these directories:

- `%LOCALAPPDATA%\FSEconomy\xfse\` (Windows)
- `~/Library/Application Support/xmse/` (macOS)
- `~/.config/xmse/` (Linux)

NOTE

If one of the `.json` files gets damaged, XFSE might not start correctly. In this case you can try to rescue your configuration data by manually editing the files (XFSE will tell you which one is causing the issue).

Chapter 7. Support Requests

7.1. Community Support

One of FSEconomy's greatest strengths is its community. You will most likely find a friendly soul trying to help you solve the issue you experience. You can ask for help (player to player support) either on the forums, or on the FSEconomy Discord server.

7.2. Official Support

The official support (through the FSEconomy staff) is mainly directed at solving technical issues, i.e. bugs or unexpected behavior of the software. Support for the XFSE app is exclusively provided through the FSEconomy forums. To open a support request, please create a new post in the [Client Support](#) sub-forum.

NOTE

Please do not post your own issue within another user's support request thread. Even if you feel the topic could be related, please create your own dedicated topic. You can refer to the other topic by including a link.

When seeking support in the forums, please follow the instructions below:

In the topic subject, please indicate the client type and simulator, e.g.

XFSE v1.0 / X-Plane 11.55 — <short problem tagline>

This ensures the right staff members can directly pick the right topics to look at.

In your problem report, include the following information:

- A description of the issue you encountered. Be as precise and descriptive as possible, and if possible add a screenshot to illustrate the issue. Express any error messages you received literally, exactly as they were phrased by the client or simulator. If you're describing an unexpected behavior of the software, also explain what behavior you would have expected instead.
- The steps that lead to the issue you're reporting. Again try to be as precise and descriptive as possible. This might not always seem to be relevant to you, but if the staff can't reproduce an issue, they will hardly be able to fix it.
- Include as much supporting information as possible. Logs and configuration files, screenshots or a short video recording can be a tremendous help for identifying and clarifying an issue.

A. Appendix

A.1. XFSE Changelog

All notable changes to XFSE are documented in this file. XFSE's versioning scheme adheres to [PEP 440](#).

A.1.1. Version 1.1.1 (2023-05-01)

Changed

- [XFSE-209] Update bundled connector to v1.1.1

Fixed

- [XFSE-206] Prevent auto-start in replay mode
- [XFSE-207] Auto-end hangs when server responds with error message
- [XFSE-208] Engine monitoring produces NaN values for CHT

Security

- [XFSE-210] Update requests to 2.29.0
- [XFSE-211] Update cryptography to 40.0.2

A.1.2. Version 1.1 (2023-04-08)

Added

- [XFSE-127] Override global auto-end setting per aircraft
- [XFSE-130] Notify other plugins of FSE client status
- [XFSE-135] Options to manage fuel & payload manually
- [XFSE-157] Auto-start flights
- [XFSE-160] Install connector plugin through XFSE application
- [XFSE-178] Better in-sim notifications for VR users
- [XFSE-182] Access to user manual and FSE operations guide from Help menu

Changed

- [XFSE-145] Show fuel flow alongside with fuel type on My Flight page
- [XFSE-158] Native Apple Silicon compatibility (Rosetta no longer needed on M1/M2)

systems)

- [XFSE-159] Show start location and great circle distance flown on My Flight page
- [XFSE-168] Include engine information in aircraft data dump
- [XFSE-170] Warn about unusually long flight time deltas in log file
- [XFSE-171] Enable debug logging via command line flag

Deprecated

- [XFSE-187] Support for macOS 10.15 “Catalina” will be removed in version 1.2
- [XFSE-188] Support for Windows 8 / 8.1 will be removed in version 1.2

Removed

- [XFSE-165] Support for Windows 7 has been removed

Fixed

- [XFSE-92] Blurry illustrations in User Manual
- [XFSE-141] Incompatibility with Xchecklist plugin
- [XFSE-142] App window doesn’t restore on macOS
- [XFSE-167] App keeps displaying “Ready for departure” status when connection to simulator is lost
- [XFSE-169] Damage calculation incorrect for new engine model in X-Plane 12
- [XFSE-180] App freezes when starting a flight and aircraft detection fails
- [XFSE-190] Incorrect speed thresholds (comparing knots to meters per second)
- [XFSE-203] App crashes when checking for updates and server not available

Security

- [XFSE-165] Update Python run time to 3.10
- [XFSE-173] Update cryptography to 40.0.1

A.1.3. Version 1.0.3 (2023-02-20)

Fixed

- [XFSE-186] Application crashes on Windows 10/11 builds > 22621

A.1.4. Version 1.0.2 (2022-12-30)

Added

- [XFSE-149] Include OS information in logon handshake
- [XFSE-152] Dump aircraft configuration data to ease data collection for aircraft nominations
- [XFSE-156] Support Ubuntu 22.04 “Jammy Jellyfish”

Fixed

- [XFSE-143] Do not block application UI while ending a flight
- [XFSE-144] Fix application crash if one of the configuration files got corrupted
- [XFSE-146] Fix server-side exception due to invalid heat damage values reported (NaN)
- [XFSE-151] Fix damage calculation for non-piston aircraft (mixture vs. condition)
- [XFSE-155] Fix too tight timeout conditions when ending/auto-ending a flight

Security

- [XFSE-150] Update third-party libraries and dependencies to the latest versions

A.1.5. Version 1.0.1 (2022-04-02)

Added

- [XFSE-140] Check for available updates when starting the application

Fixed

- [XFSE-137] Fix mac build to also run on macOS Catalina
- [XFSE-138] Fix incorrect flight time calculation when using time compression (Ctrl+T)
- [XFSE-139] Adjust UI element spacing on macOS and Linux

A.1.6. Version 1.0 (2022-03-19)

Deprecated

- [XFSE-165] Support for Windows 7 will be removed in v1.1