



FlyLive Studio

User Manual

Contents

Preface	4
Prerequisites	5
Installation	6
a) Installation process if you have downloaded from the FlyApps Website	6
b) Installation Process if you have downloaded from the AVSIM Library or any other third party	7
The Dashboard	8
The Flight Dispatcher	9
The Overlay Editor.....	10
Creating a new overlay	11
The editor	12
Starting your overlay / Creating New Elements	14
Adding Flight Data to Text.....	15
Navigation around the sidebar	16
Changing the size & position of elements	16
Element Properties.....	18
Descriptions of elements & properties	19
Global Properties	19
Text Element	20
Image Element	22
BrowserSource Element	23
Landing Rate Element	24
Progress Bar Element	26
Example Overlay	27
Exporting the Overlay	28
Adding the overlay to OBS	29
Saving the Overlay as a Template	30
The FlyLive Studio Chatbot.....	31
Setup	31

Login	31
Password	33
Commands	35
Song Requests	38
Landing Rate	40
Variables	43
Global Variables	43
Notification Variables	45
The Settings Page	47
Conclusion	49
LEGAL	50

Preface

Thank you for downloading FlyLive Studio. Your interest in this app is appreciated – I hope you enjoy using it!

- Mark Ng – Developer of FlyLive Studio

Prerequisites

The following software **must** be installed for FlyLive Studio to function:

- Microsoft Flight Simulator 2004, Microsoft Flight Simulator X, Prepar3D v2.x – 4.x or X-Plane 9/10/11
- FSUIPC 2.x, 3.x, 4.x or 5.x depending on your flight simulator
- A streaming client such as OBS Studio that supports the adding of HTML files
- Visual C++ 2015 Redistributable (or newer)

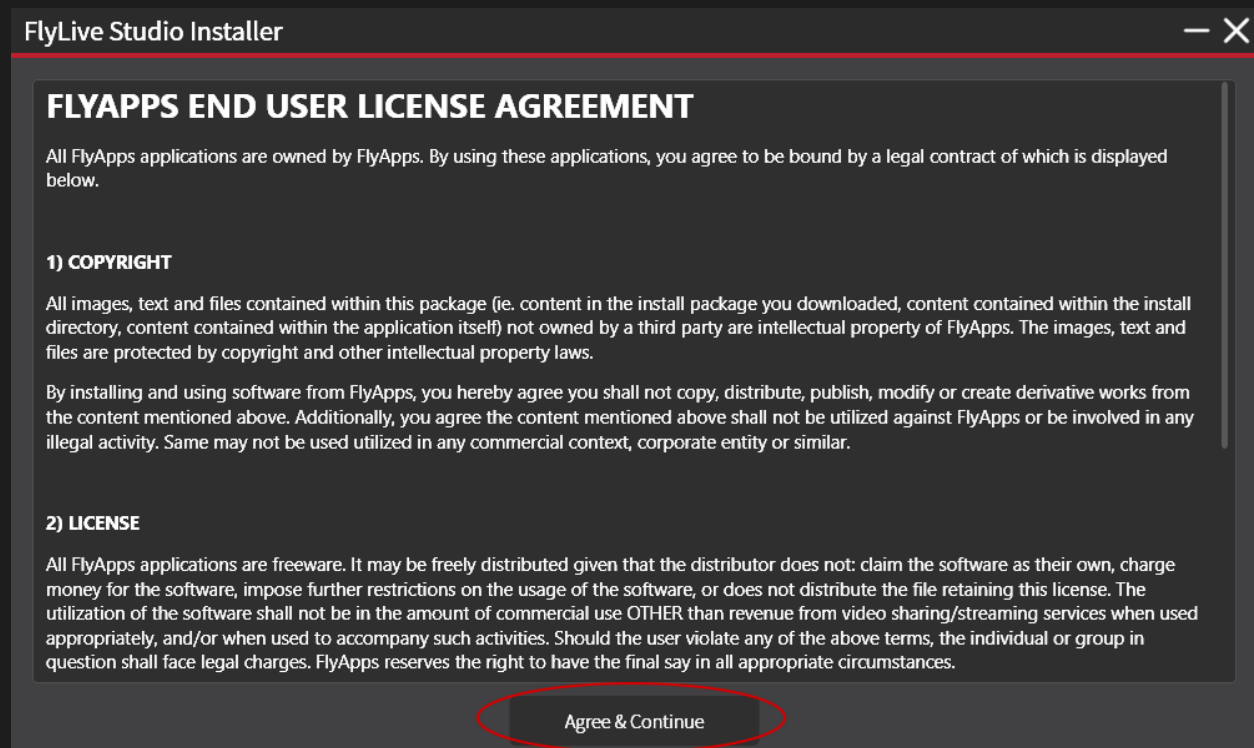
Installation

There are two ways you may have downloaded FlyLive Studio:

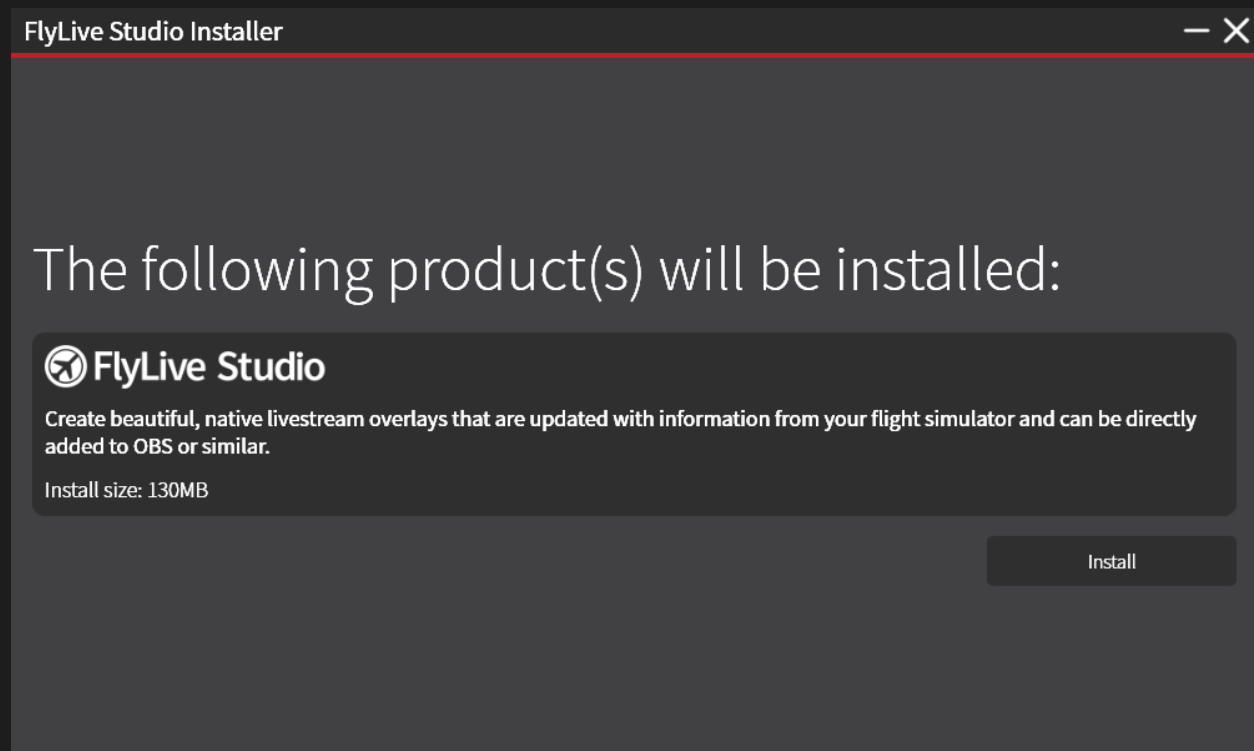
- a) Directly through the FlyApps website
- b) From the AVSIM Library

a) Installation process if you have downloaded from the FlyApps Website

Once you have downloaded the installer from the FlyApps website and opened the file, you will see this page:



Please carefully read the terms and conditions of the application. If you agree, press 'Agree & Continue'. You will then reach this page:



If you have at least 220MB on your system, click 'Install'. (The reason for this is because the download size of the libraries is about 68MB while the extracted size is about 130MB, and in order to complete the installation, you need the storage that amounts to their combined values.)

The download will then initialize and begin. Once the installation is finished, you may click 'Launch' to start FlyLive Studio or 'Close' to simply close the installer.

b) Installation Process if you have downloaded from the AVSIM Library or any other third party

If you have downloaded from the AVSIM Library or another third party, you may be presented with a slightly different window when you open the installer.

If the terms and conditions are the first thing you see after opening the file you downloaded, please refer to a)

If a page saying 'The following product(s) installers will be downloaded', then please click 'Download' then refer to a).

The Dashboard

When you open FlyLive Studio, the first thing you will see is the **Dashboard**.

The **Dashboard** displays all your simulator's information. This includes your airplane's current altitude, heading and speed and your simulator's framerate, time and VAS.

The screenshot shows the FlyLive Studio Dashboard interface. At the top, there is a navigation bar with 'FlyLive', 'Dashboard', 'Overlay', 'Chatbot', and 'Settings'. A status bar indicates 'Not Connected to FSUIPC/XPUIPC' and 'Change simulator / data source'. The main content is divided into several sections:

- AIRPLANE INFORMATION:** A table with columns for AIRCRAFT, ALTITUDE, HEADING, GROUND SPEED, IAS, and V/S, all showing 'n/a'.
- SIMULATOR INFORMATION:** A table with columns for SIM ZULU, REAL ZULU, FRAMERATE, and VAS. Values are 'n/a', '07:18z', 'n/a', and 'n/a' respectively.
- FLIGHT DISPATCHER:** A section with a progress bar at 0%, 'METAR NOT AVAILABLE', and 'AIRPLANE COORDINATES' (LATITUDE and LONGITUDE, both 'n/a').
- FLIGHT STATUS:** A table with columns for FLIGHT, DTG, ETA, and TTG, all showing 'n/a'.

The dashboard

Here is what each piece of information means:

- Altitude: Your simulator's airplane's altitude in feet.
- Heading: Your simulator's airplane's heading in degrees.
- Ground speed: Your simulator's airplane's ground speed in knots.
- IAS: Your simulator's airplane's indicated airspeed (IAS) in knots.
- V/S: Your simulator's airplane's vertical speed/velocity in feet per minute. Negative values mean the plane is descending; positive values mean the plane is ascending.
- Sim Zulu: Your simulator's current Zulu (GMT) time.
- Real Zulu: The real-world Zulu (GMT) time based on your computer's time.
- Framerate: The framerate of your simulator measured in frames per second.
- VAS: The remaining VAS available to your simulator measured in GB – only for 32 bit.
- Latitude: Your simulator's airplane's latitude in degrees.

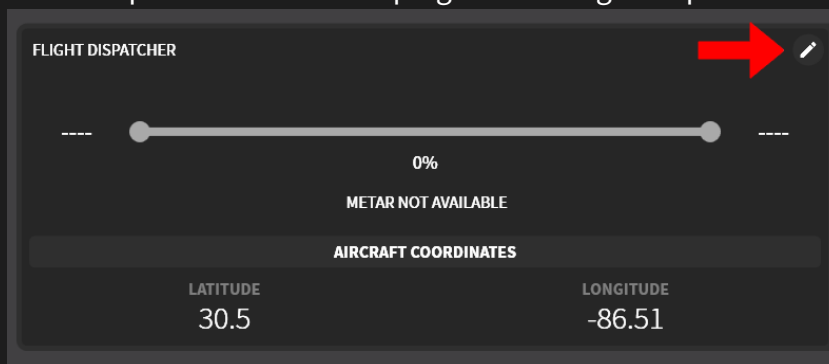
- Longitude: Your simulator's airplane's longitude in degrees.
- Flight: The current flight's distance in nautical miles.
- DTG: The distance to the current flight's destination in nautical miles.
- ETA: The estimated Zulu time of when you will arrive at your destination airport relative to the real world Zulu time.
- TTG: The estimated time remaining before you arrive at your destination.

The Flight Dispatcher

The flight dispatcher is used to give FlyLive Studio information regarding your current flight, so it can calculate information such as your distance to go and the estimated time of arrival.

To set up a flight, please follow these steps:

- 1) Click the pencil icon at the top right of the flight dispatcher:



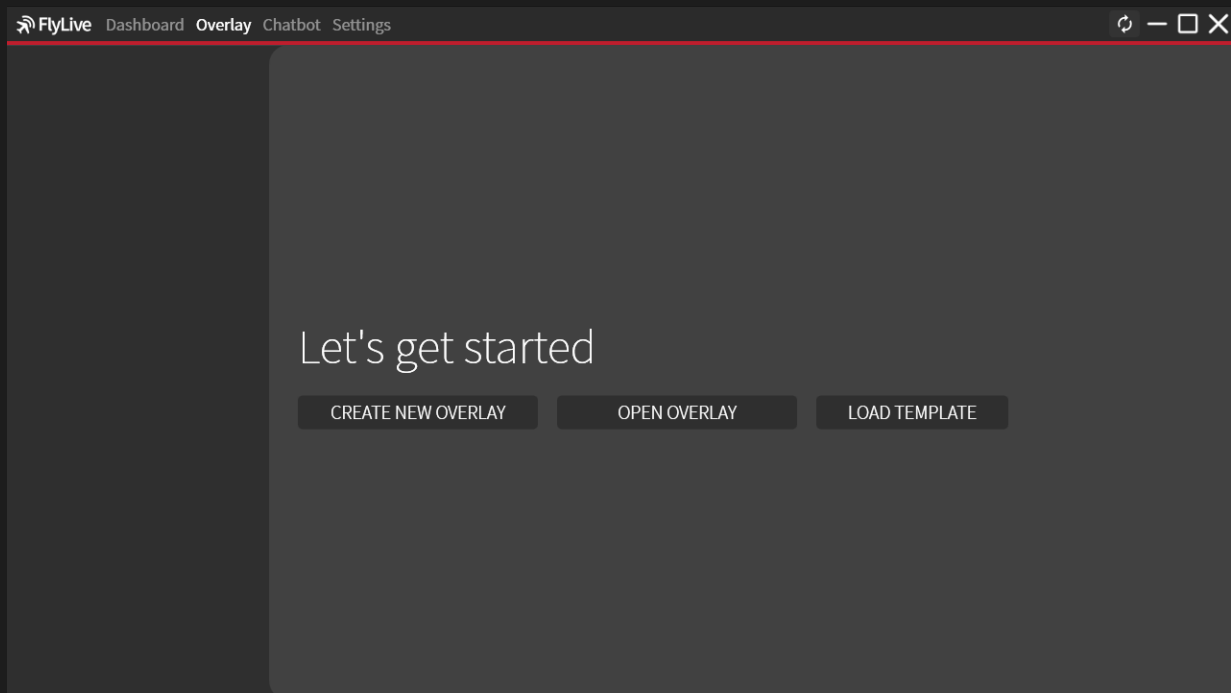
- 2) Fill in the departure and arrival ICAO's.
- 3) Enter the altitude at when you want the ETA to begin calculating. **It is recommended that you set this value to your cruising altitude.** The ETA will begin calculating 2000 feet below the entered value and will only calculate when you are airborne.
- 4) Enter your callsign.
- 5) Press 'OK'.

Your flight details have now been received by FlyLive Studio and it will begin calculating the relevant information from it.

The Overlay Editor

To enter the Overlay Editor, press 'Overlay' at the top of the application. The 'Overlay' button is sandwiched between 'Dashboard' and 'Settings'.

The **Overlay Editor** is the sole purpose of FlyLive Studio. It is used to create rich livestream overlays filled with flight information and widgets that can be easily imported into your streaming client.



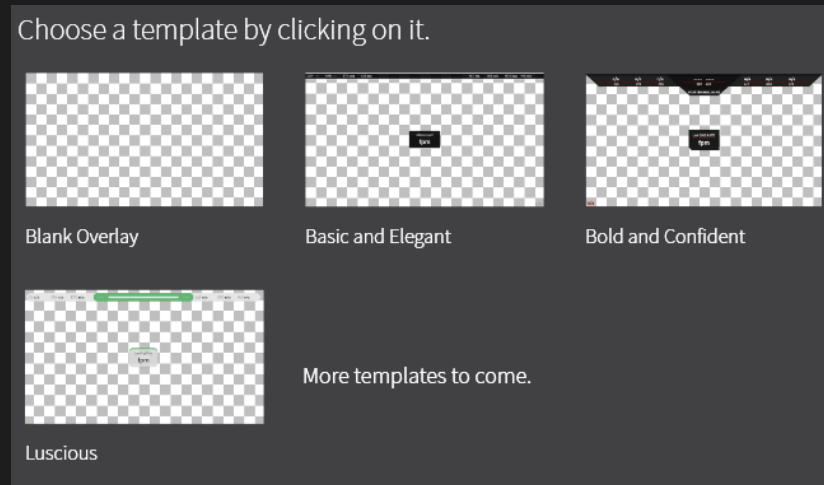
To create a **new** overlay, press 'CREATE NEW OVERLAY'.

To open an **existing** overlay, press 'OPEN OVERLAY'.

To open a **template**, press 'LOAD TEMPLATE'.

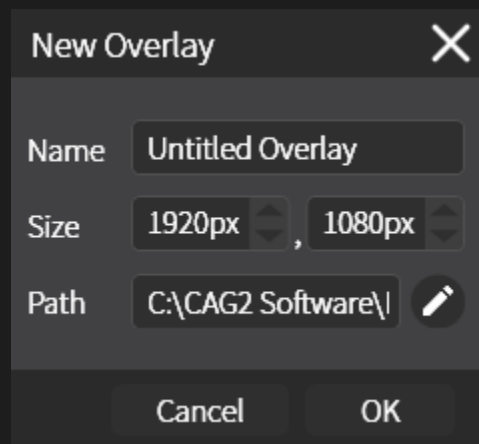
Creating a new overlay

To create a **new** overlay, press 'CREATE NEW OVERLAY'. You should then see this popup:



To create your **own** overlay, please select 'Blank Overlay' and read on.

If you would like to use one of the provided templates, select one of them then skip to **Exporting the Overlay**.



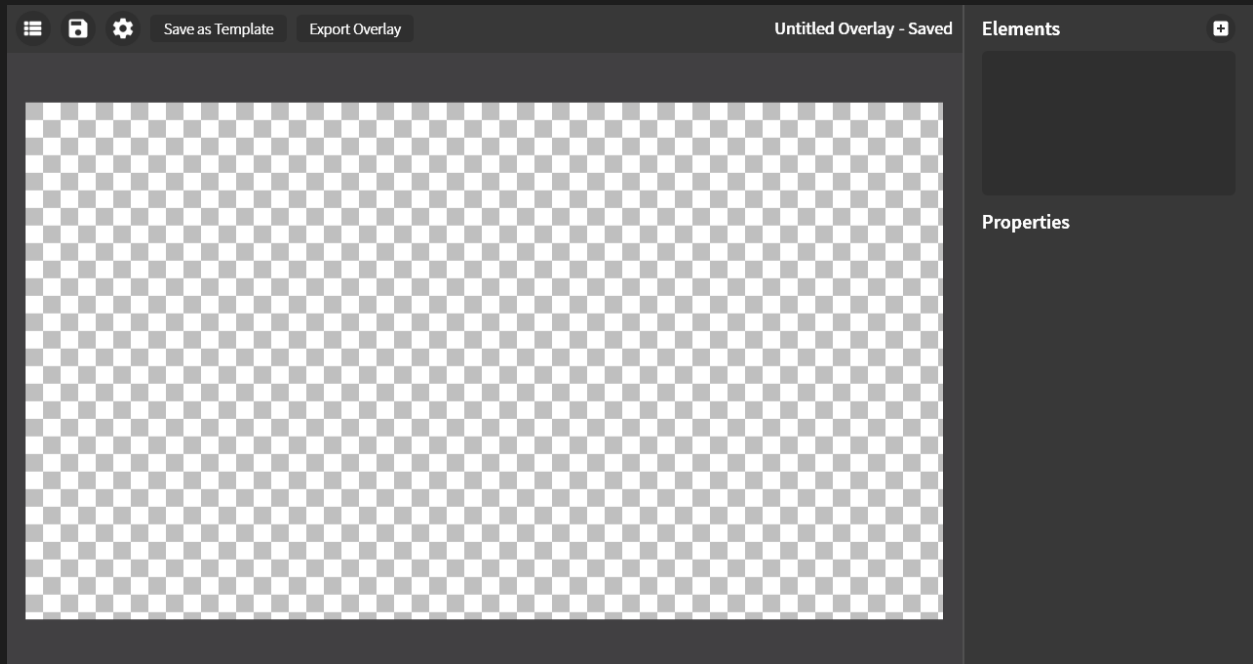
Here are what the different fields mean:

- Name: This is what your overlay will be called. It is purely for your own reference.
- Size: This is the canvas size of the overlay. The default value of 1920x1080px will be suitable for most users.
- Path: This is the save location for your overlay.

All the fields can be changed at any time.

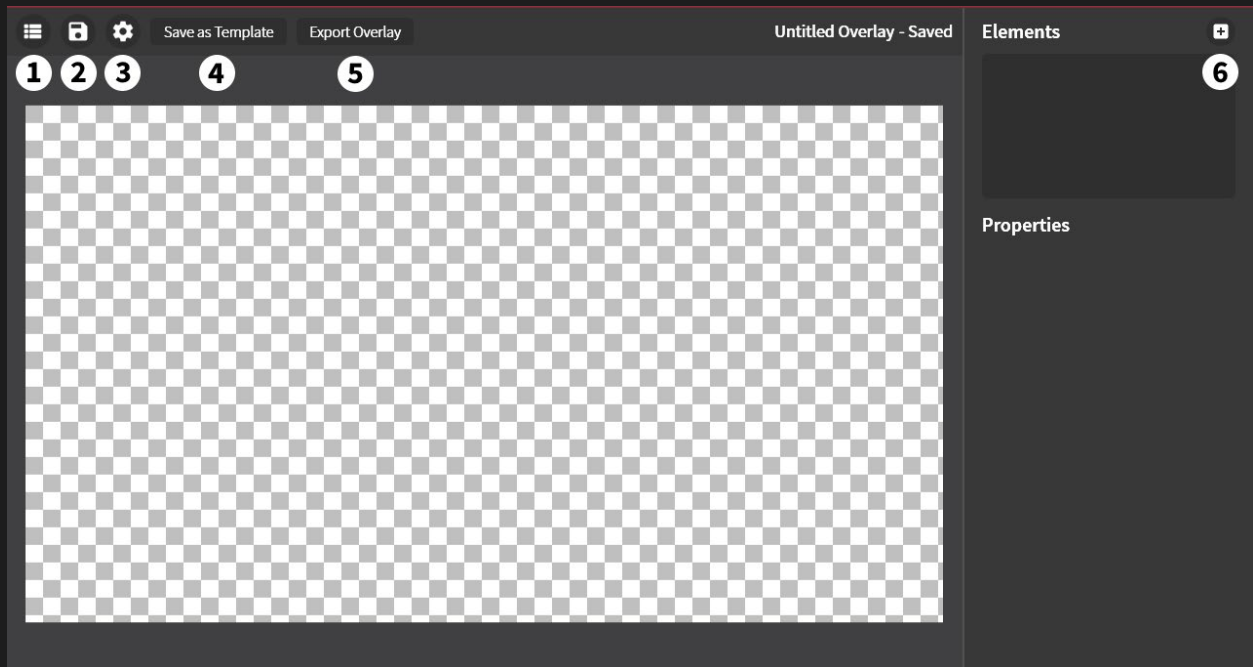
Fill in the fields and press 'OK' to finish creating the new overlay.

The editor



This is what you will see when you create a new overlay.

Below is a breakdown of what each button does:

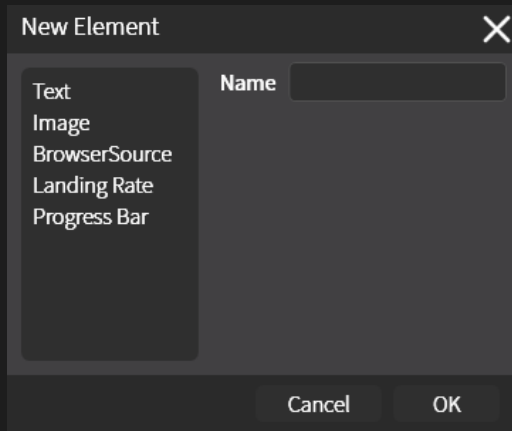


No.	Button	Description
1	Menu	Returns you back to the menu.
2	Save	Saves the current overlay.
3	Settings	Modify the overlay's properties, such as its size and its name.
4	Save as Template	Saves the overlay as a template which can be shared with other people.
5	Export	Export the overlay as an HTML in preparation for importing it into your steaming client.
6	New Element	Adds a new element to the overlay.

The canvas at the left is the **Overlay Preview** pane.

Starting your overlay / **Creating New Elements**

To get started with your overlay, click the + button at the top right (labelled **5** on the diagram). You should then see this popup:



This window allows you to add **objects and widgets** to your overlay. These are called **elements**.

- The **text** element displays text. You can set the text element to contain certain flight data, such as the altitude. You can also set the text element to read from a local .txt file.
- The **image** element displays a given image. You can select an image from your file system to be displayed on the overlay.
- The **BrowserSource** element displays a webpage. This can be used for alert boxes or donation goals from third party services and more.
- The **Landing Rate** element is invisible for most of the time; it only pops up for a few seconds when you touch down in the simulator to display the landing rate. The landing rate is measured in feet per minute.
- The **Progress Bar** element displays the current progress of your flight in a bar. The closer you are to your destination, the more the bar fills up. When you reach your destination, the bar will be filled.

Tip: You should **name** your element something meaningful, so you don't forget what it is later down the road.

When you add a new element, two things will happen:

- The sidebar will fill up with properties which you can change to customize the appearance of the element.
- The element will appear on the preview pane and will be outlined in red.

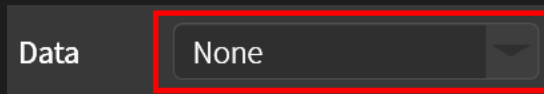
Adding Flight Data to Text

To add flight data to text, create a **text** element. Then, in the **properties pane**, scroll down until you see this:



Click it to uncheck it (unless it is already unchecked).

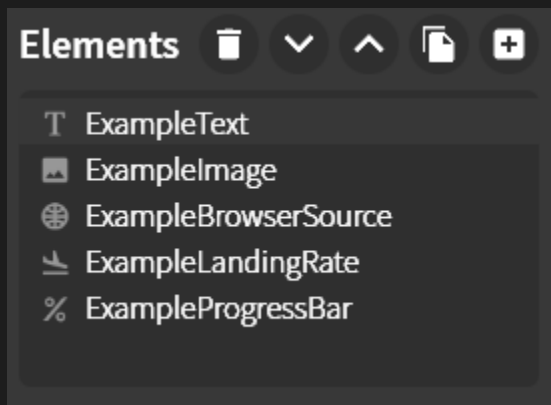
Next, click the box to the right of **data**:



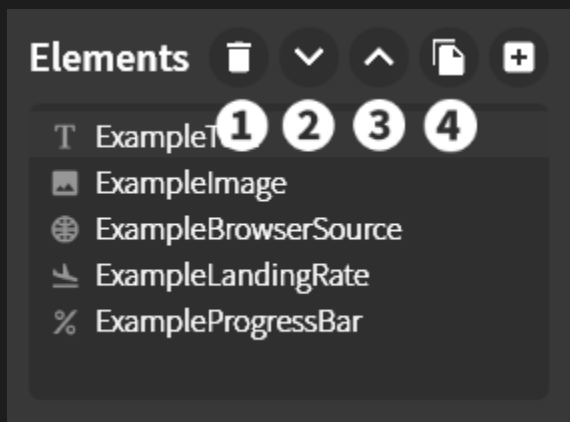
Select which type of flight data you want to add to the text.

Please note that doing this will override the text element's custom text. It will also override the path you've selected in 'text path' if you have selected it.

Navigation around the sidebar



As you add more and more elements to your overlay, the **Elements Box** will begin to fill up like the one shown above. Below is a labelled version of the above diagram and an explanation of how to use it:



To **select** an element you have created previously, click the element in the elements box.

To **delete** an element, press the **button that is labelled 1** on the above diagram.

To **move an element up or down**, click the up and down arrows above the elements box. Button **2** moves the selected element down; button **3** moves the selected element up.

Elements that are higher in the elements box will appear in front of elements below it on the overlay.

To **duplicate** an element (to create a copy of an element), press the button labelled **4** on the above diagram.

Changing the size & position of elements

There are **two** appropriate ways of changing the size and position of elements:

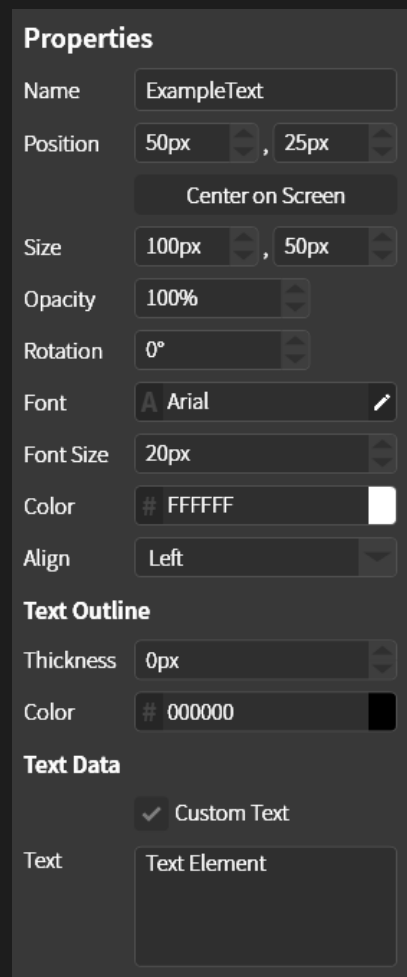
- You can change the size and position of an element by editing the **Position** and **Size** fields in the sidebar.
- You can change the size and position of an element by clicking on it on the preview, then dragging it around and/or dragging the **small, filled red squares (handles)** that reside on the square outline.

Element Properties

Element properties dictate the appearance of an element on your overlay. These are edited on the **Properties Pane** below the **Element Box**. These are examples of **Element Properties**:

- Position
- Size
- Opacity
- Font Family
- Font Size
- Text Color
- Image path
- Etc.

Below is what the properties pane looks like when a **text element** is selected:



The screenshot shows a 'Properties' pane for a text element. It is organized into several sections:

- Basic Properties:**
 - Name: ExampleText
 - Position: 50px, 25px (with a 'Center on Screen' button)
 - Size: 100px, 50px
 - Opacity: 100%
 - Rotation: 0°
 - Font: Arial
 - Font Size: 20px
 - Color: #FFFFFF
 - Align: Left
- Text Outline:**
 - Thickness: 0px
 - Color: #000000
- Text Data:**
 - Custom Text
 - Text: Text Element

Each row houses a specific property. The left column describes the property. The property itself is set in the right column.

Descriptions of elements & properties

Global Properties

These properties apply to all elements. Generally, these affect the transformation of the element (ie. position and size).

Property	Description
Name	This is name of the element.
Position	This is the position of the element in pixels. An element is positioned by its center point. Default value: <i>Variable</i>
Size	This is the size of the element in pixels. Default value: <i>Variable</i>
Opacity	This is the opacity of the element. (Opacity is the measure of how transparent something is) Default value: <i>100%</i>
Rotation	This is the rotation of an element in degrees, Default value: <i>0°</i>

Text Element

The text element displays text within a given boundary. The text can be given by the user and can also be read from a local text file or be given by flight data.


Text element in depth

If the size of the element isn't large enough to house all the given text, then the last word that can fit on a line will stay and the next word and everything subsequent will wrap to the next line.

If the text is read from a text file or is binded to flight data, then it will be dynamically updated in the exported overlay. Custom text will not update in the exported overlay when changed (unless the overlay is re-exported), however.

Properties

These properties apply to the text element only.

Property	Description
Font	This is the font of the text. This property controls the following: <ul style="list-style-type: none">- The font family of the element- The font style of the element- The font weight of the element It does not control the font size. Default value: <i>Arial</i>
Font Size	This is the font size of the text in pixels. Default value: <i>20px</i>
Color	This is the text's color. To edit the color, press the colored portion of the text box:  Default value: #FFFFFF (White)
Align	This is the text alignment of the text. The text will align based on the size of the element in question. Default value: <i>Left</i>
Text Outline Thickness	This is the thickness of the text outline. Default value: <i>0px</i>

Text Color	Outline	This is the color of the text outline. Default value: #000000
Custom Text		This property decides whether the text is set by the user, or will be filled with data from a local text file or flight information. Default value: Checked (true)
Text		This is the text/content of the text element. This is only visible when the 'Custom Text' checkbox is checked. Default value: <i>Text Element</i>
Data		This is what flight data the text element shall be populated with. This overrides the 'Text Path' property. Default value: <i>None</i>
Text Path		The text element's text will be set to the text file selected. Default value: None

Image Element

The image element displays an image within a given boundary.

Properties

These properties apply to the image element only.

Property	Description
Path	This is the path of the image. Default value: None

BrowserSource Element

The BrowserSource element displays a webpage within a given boundary.

BrowserSource element in depth

Only webpages that don't have the **X-Frame-Options** HTTP header set can be used with the BrowserSource element.

Properties

These properties apply to the BrowserSource element only.

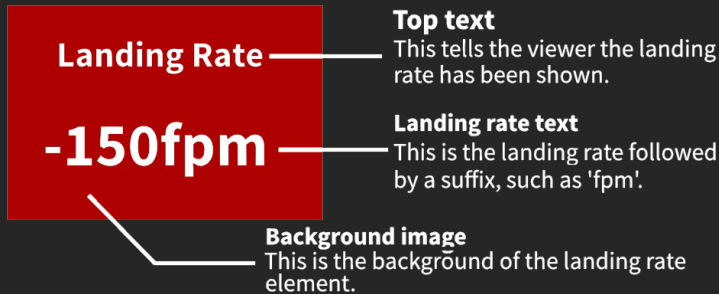
Property	Description
URL	This is the URL of the website that the BrowserSource will display. It is not recommended to force-display a local HTML file. Default value: None

Landing Rate Element

This element displays the landing rate for a specified length of time when you have touched down.

Landing Rate element in depth

This is the anatomy of the Landing Rate element:



Properties

These properties apply to the Landing Rate element only.

Property	Description
Background	This is the background image of the landing rate element. For example, a solid red background image will make the landing rate element's background fully red overall. Default value: C:\CAG2 Software\FlyLive Studio\landing.png
Duration	This is how long the landing rate will be displayed for (in seconds) after the plane has touched down. Default value: <i>5 seconds</i>
TOP TEXT PROPERTIES	These properties affect the appearance of the top text. More information is available at the 'Landing Rate element in depth' box above.
Top Text	This is the text that is displayed at the top of the landing rate element. Default value: <i>Landing Rate</i>
Top Margin	This property corrects the y-position of the top text. Default value: <i>3%</i>
Font	This is the font family, weight and style of the top text. Default value: <i>Arial</i>
Font Size	This is the font size of the text in pixels. Default value: <i>20px</i>
Color	This is the text's color. Default value: #FFFFFF (White)

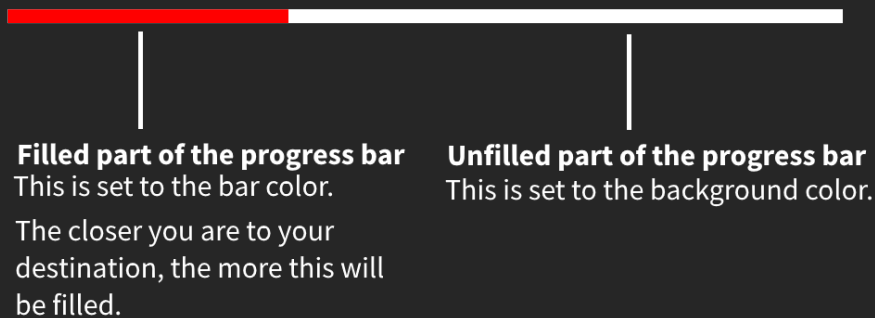
LANDING RATE TEXT PROPERTIES	These properties affect the appearance of the landing rate text. More information is available at the 'Landing Rate element in depth' box above.
Suffix	This is the text that is displayed to the right of the landing rate. For example, if the suffix is 'fpm' and the landing rate was -300, then the final text displayed will be '-300fpm', Default value: <i>fpm</i>
Top Margin	This property corrects the y-position of the landing rate text. Default value: <i>7%</i>
Font	This is the font family, weight and style of the landing rate text. Default value: <i>Arial</i>
Font Size	This is the font size of the text in pixels. Default value: <i>50px</i>
Color	This is the text's color. Default value: <i>#FFFFFF (White)</i>

Progress Bar Element

The Progress Bar element displays the current progress of your flight in a bar. The closer you are to your destination, the more the bar fills up. When you reach your destination, the bar will be filled.

Progress Bar element in depth

This is how the Progress Bar element is structured:



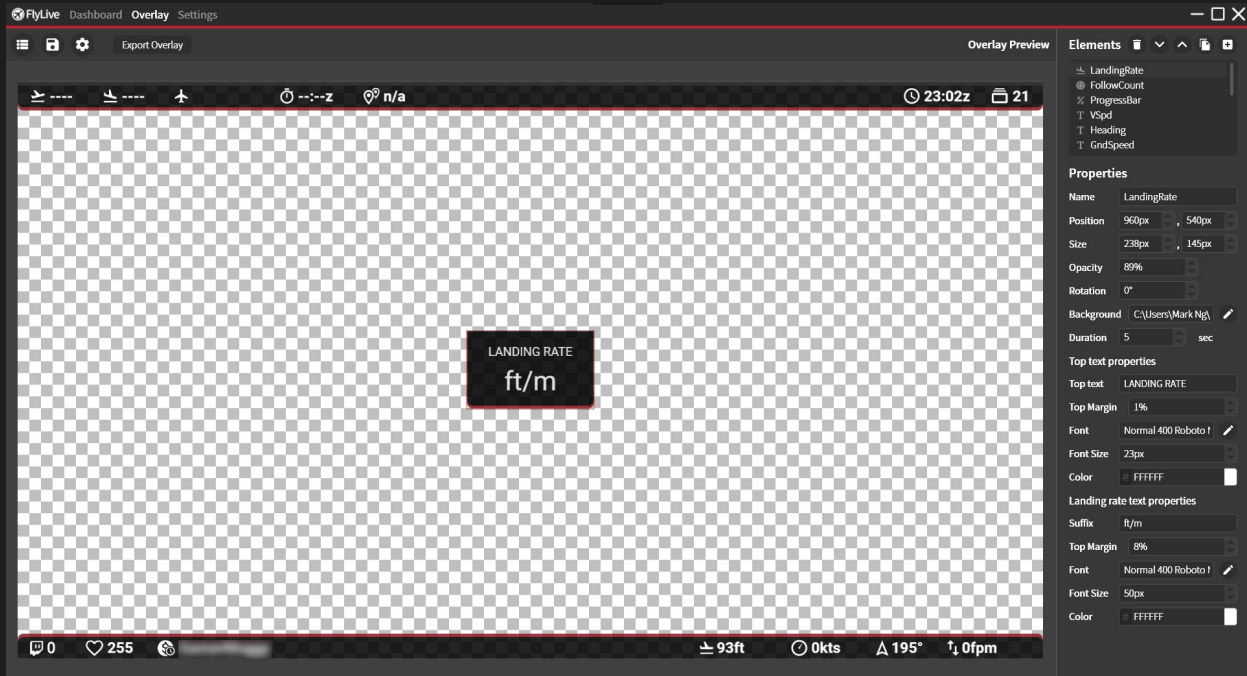
Properties

These properties apply to the Progress Bar element only.

Property	Description
Bar Color	This sets the color of the filled part of the progress bar. Default value: <code>#FFFFFF</code> (<i>White</i>)
Background Color	This sets the color of the unfilled part of the progress bar. Default value: <code>#000000</code> (<i>Black</i>)

Example Overlay

This is an example overlay that will hopefully inspire you in your journey to creating a great overlay. This is in fact the overlay I personally use when livestreaming!



(Some information at the bottom has been blurred out)

Exporting the Overlay

Now that you've created your overlay, it's time to export it. To do this, simply press the 'Export Overlay' button on the top bar:



It is **highly recommended** that you create a new folder to put your overlay in. This is because FlyLive Studio will copy all of the resources that is in your overlay into a new folder called 'Assets' in the directory of your overlay.

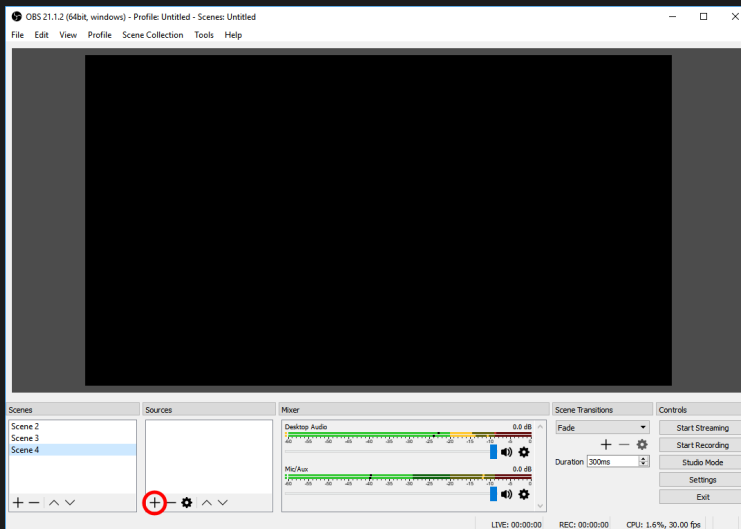
(This must be done due to the way the OBS BrowserSource treats local image files)

Adding the overlay to OBS

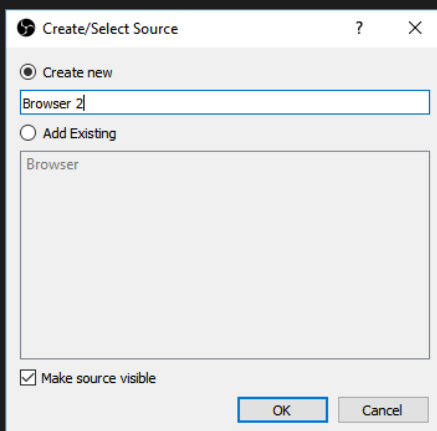
There will not be instructions on how to add the overlay to other streaming clients. These instructions also assume that you know how to show your simulator in OBS as well.

To add the overlay to OBS:

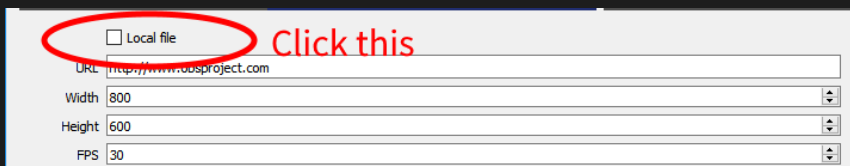
- 1) Open OBS.
- 2) Click the circled button:



- 3) Click 'Browser'.
- 4) Click 'OK':



- 5) Check 'Local File':



- 6) Click 'Browse' and select your exported HTML file.
- 7) Resize the Browser to the overlay size you set in FlyLive Studio.
- 8) Click 'OK'.

Saving the Overlay as a Template

In FlyLive Studio, you can export your overlay as a template .zip file and share it with other people for them to use.

To get started, click 'Save as Template' in the overlay editor. You should now see this popup:

Export Template

Template Name

Template Author

Export to

Preserved:

- Text and flight data sources
- Media, such as images
- All widgets (ie. landing rate)

Not Preserved:

- Text that reads from a file
- Fonts not installed on the user's system (the user will need to install the fonts used)

Cancel OK

To export the overlay as a template, fill in the required fields (you do not need to use your real name for 'Template Author').

Export to is where the exported .zip file will be located.

Note: The person you are sharing the template with must have **all fonts used in the overlay** installed on their system. Consider restricting yourself to only using the default Windows fonts if you plan on sharing your overlay to the general public. You could also distribute the template with the necessary font files.

The FlyLive Studio Chatbot

The FlyLive Studio chatbot links your Twitch chat with your simulator. It allows viewers to interact with you, the stream and your simulator.

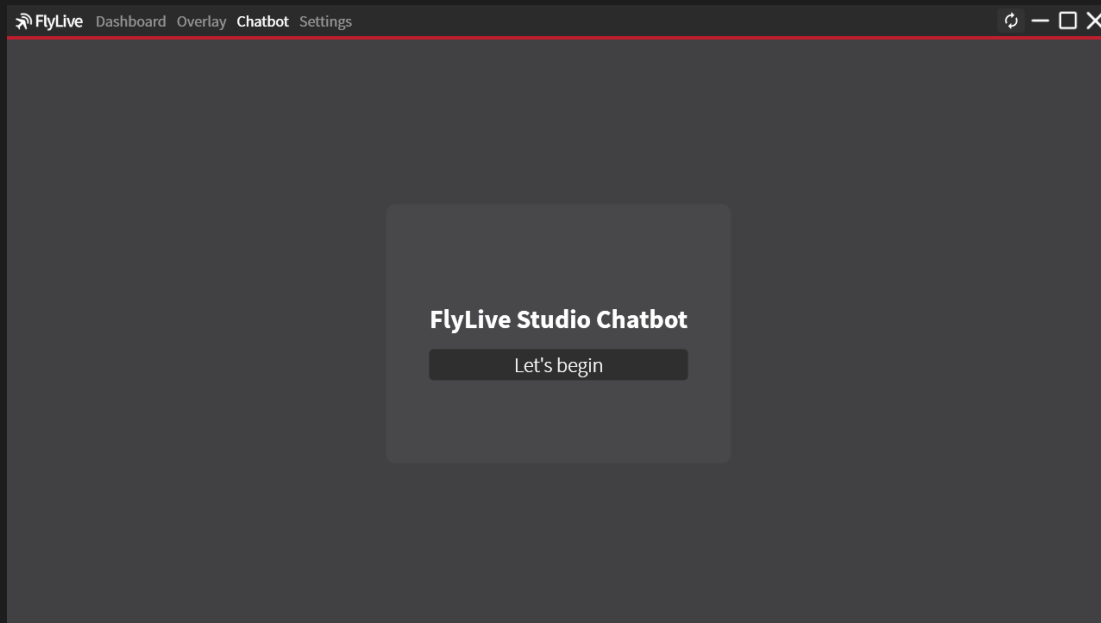
Setup

For the chatbot, we **highly recommend** that you create an account **separate from your streaming account** for the chatbot.

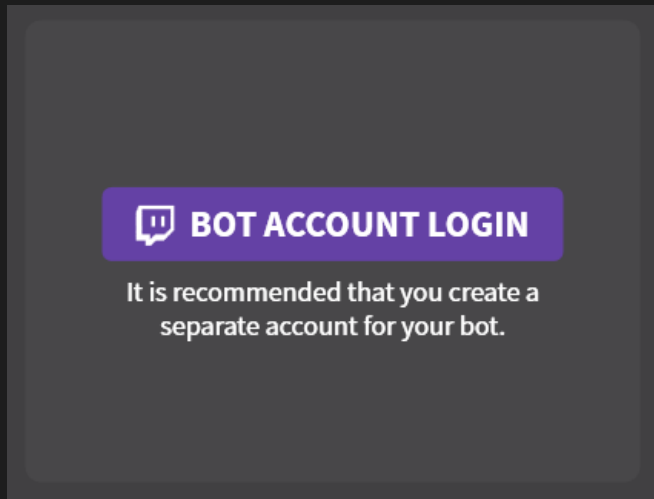
For example, if your username is *john*, you would create a **new Twitch account** named *john_chatbot* or similar.

Login

Start by going to the **Chatbot** page in FlyLive Studio.

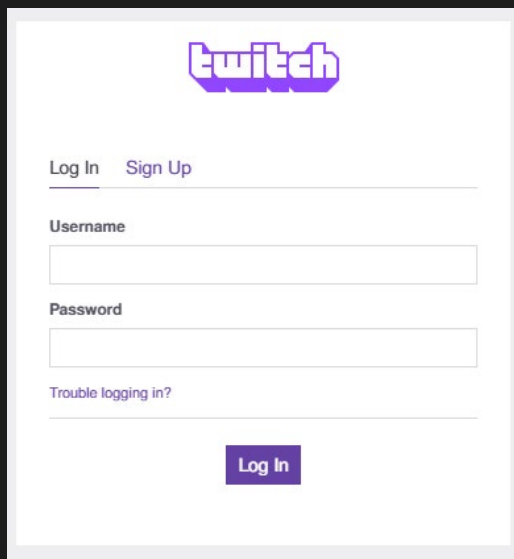


Click **Let's begin**.

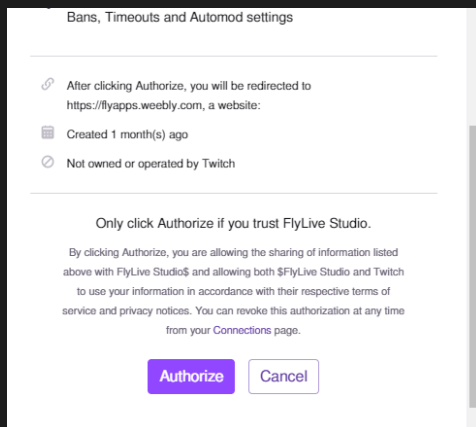


Click **BOT ACCOUNT LOGIN**.

You will then be directed to this page:



Login to Twitch as you normally would, but with your **bot account** and not your **channel's account**.



Click **Authorize**.

Now, click **CHANNEL LOGIN**. This time, login with your **channel's account**.

Password

After you have successfully logged in to your streamer and channel account, you will be given the option to set a **password**.

The password is used to further encrypt your authentication details (your OAuth Tokens if you're interested). This is to protect you from viruses stealing said details when FlyLive Studio is not running.

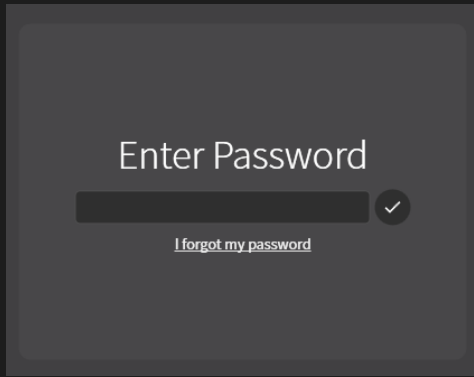
*Please note that FlyLive Studio does **not** store your Twitch password **or** the password set in FlyLive Studio.*



Set a password by typing it in then clicking the tick button.

If you don't want a password, leave the password box blank then click the tick button.

If you decided to set a password, whenever you open FlyLive Studio, you should go to the **chatbot** tab and enter your password. The chatbot is **unable to and will not function** until you enter the password.



Commands

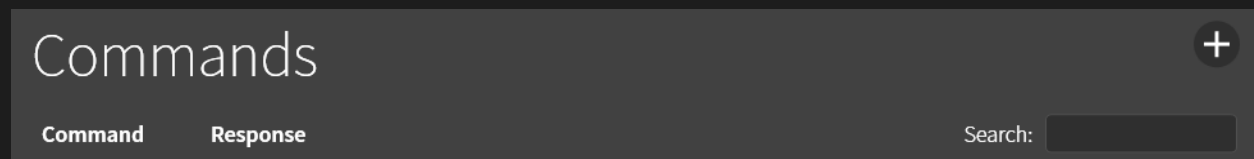
Commands are a way for users to interact with the chatbot.

For example, a user types **!website** in the chat, and the chatbot would respond with your website.

Commands can also contain **variables** to make the command responses more relevant and useful. For example, **!flight** could be responded to with:

EGLL – KSFO | Time until landing is 4:32 | Distance to KSFO is 2302nm

To create a command, go to the **Commands** page in FlyLive Studio.



Click the + button at the top right-hand corner.

A new command will then be created:



You can then freely modify the command.

If **Reply through whisper** is enabled, then the bot will send a **whisper** to the user instead of sending the response through chat.

The **Scope** defines who can use the command.

The **Cooldown** is the minimum amount of time between usages of the command. If someone uses the command, the next person must wait the cooldown time before they can then use it.

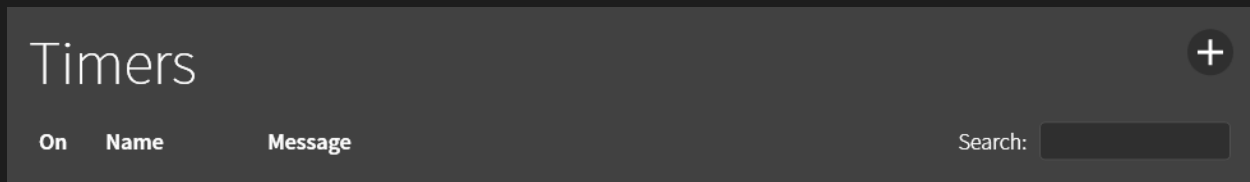
To see all the possible variables, click [here](#).

Timers

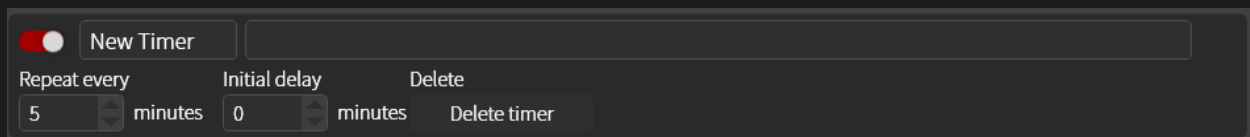
You can add **timers** to the chatbot. Timers repeatedly send messages to the chat over a fixed interval.

For example, you can add a timer that sends a link to your website every 5 minutes.

To add a timer, go to the **Timers** page.



Click the + button at the top right-hand corner.



You may now freely customize the timer.

The **initial delay** is how long the timer should wait before it starts counting **x** minutes.

Notifications

Notifications are messages that fire when something happens on your stream, such as when you land, a user subscribes, etc.

There are currently 5 notifications available:

- Landing
- Subscription
- Resubscription
- Raid
- Host

To set these, go to the **Notifications** page.

Notifications

On Landing

\$(channel) just landed with a landing rate of \$(landingrate)fpm!

Subscription

\$(username) just subscribed with a tier \$(subtier) sub!

Resubscriptions

\$(username) just resubscribed for a total of \$(submonths)!

Raid Min viewers:

\$(username) is raiding with \$(viewercount) viewers! PogChamp

Host Min viewers:

\$(username) just hosted with \$(viewercount) viewers!

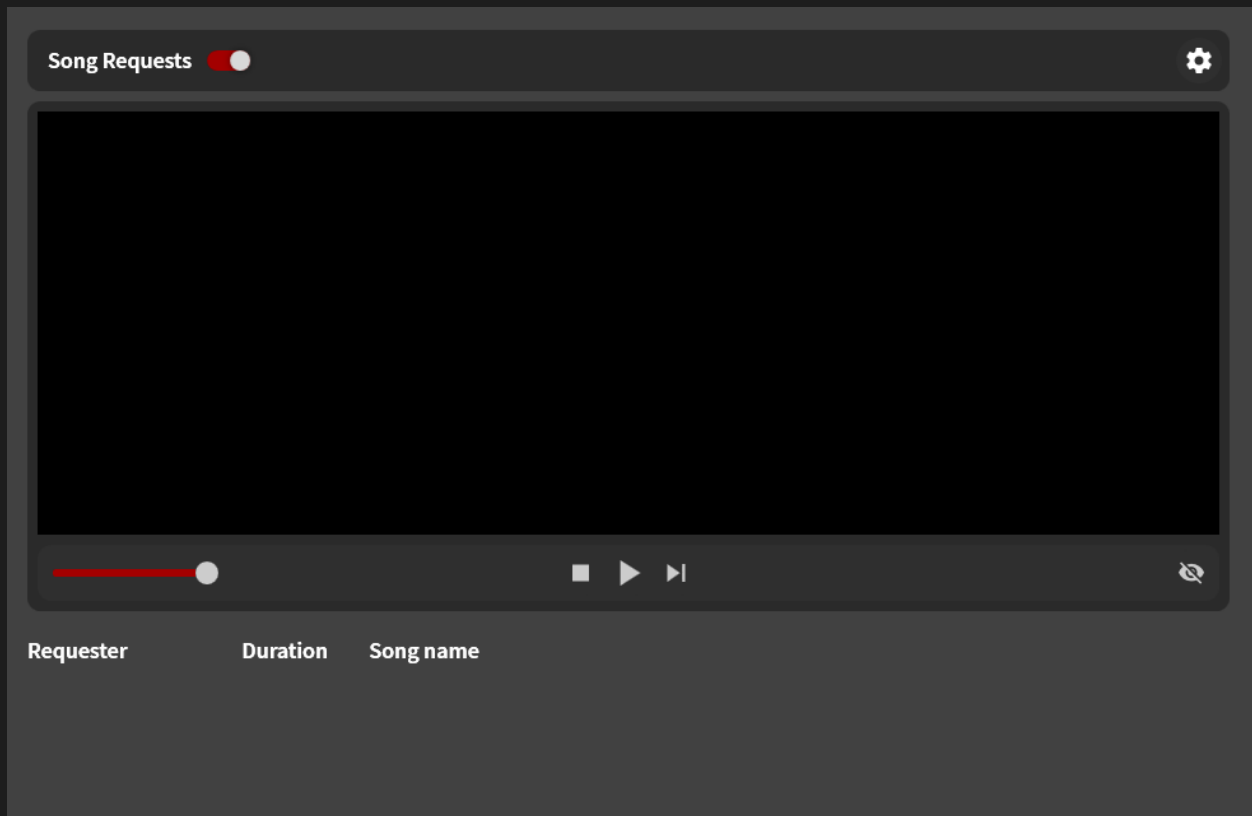
Each notification has their own special variables. To see the specifics, please click [here](#).

Song Requests

Song requests allow users to request songs that are played on your stream.

Enable this at your own risk. FlyLive Studio has moderation settings built-in but is ultimately unable to completely filter out inappropriate requests.

To configure song requests, go to the **Song Requests** page.



By default, FlyLive Studio has moderately strict moderation settings. To make them stricter or less strict, click the **cog** at the top right-hand corner.

Maximum queue length:	<input type="text" value="25"/>	Maximum song duration:	<input type="text" value="8"/> min
Req. viewer level:	<input type="text" value="Everyone"/>	Minimum song views:	<input type="text" value="10000"/> views
Max requests per user:	<input type="text" value="5"/> requests	<input checked="" type="checkbox"/>	Song must have more likes than dislikes
Cooldown:	<input type="text" value="2"/> sec	<input type="checkbox"/>	Song must be in YouTube's 'Music' category

Your viewers can request songs with !songrequest or !sr, and cancel their request with !wrongsong.

Setting	Description
Max queue length	The maximum amount of songs that can be in the queue at once.
Req. viewer level	Who can use the song request feature.
Max requests per user	The maximum amount of requests a single user can have in the queue.
Cooldown	The cooldown of !sr or !songrequest.
Max song duration	The maximum duration of a song to be added to the playlist.
Min song views	The minimum amount of views a song must have to be added to the playlist.

Users can add song requests with one of these commands:

!songrequest <search term>

!sr <search term>

FlyLive Studio will then search **YouTube** for that song then add it to the playlist if it adheres to your moderation settings.

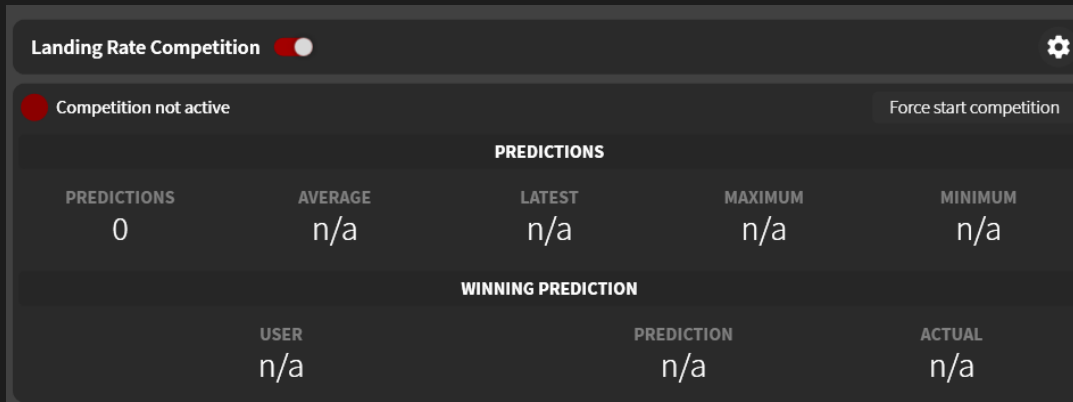
Your users can cancel their latest song request using **!wrongsong**.

Landing Rate Competition

The **landing rate competition** is a fun game your viewers can participate in whenever you are about to land.

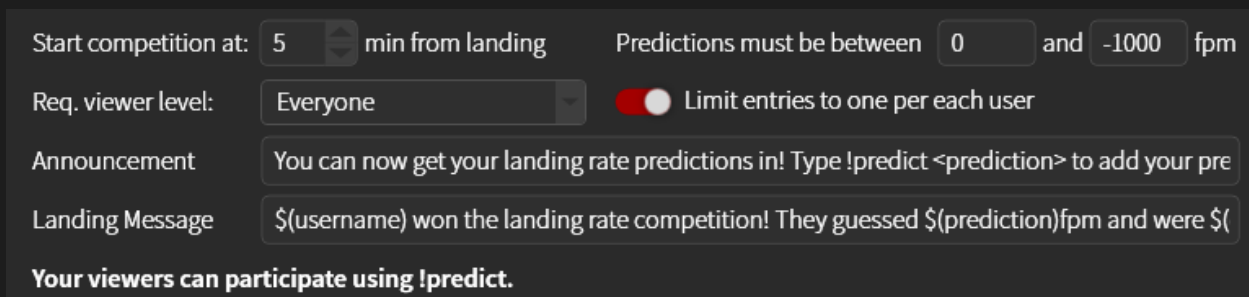
Users will be allowed to **predict** your landing rate, and when you land, the user with the prediction closest to your actual landing rate wins.

To configure the landing rate competition, go to the **Landing Rate Competition** page.



Landing Rate Competition				
Competition not active Force start competition				
PREDICTIONS				
PREDICTIONS	AVERAGE	LATEST	MAXIMUM	MINIMUM
0	n/a	n/a	n/a	n/a
WINNING PREDICTION				
USER	PREDICTION	ACTUAL		
n/a	n/a	n/a		

To configure the settings, click the **cog** at the top right-hand corner.



Start competition at: 5 min from landing Predictions must be between 0 and -1000 fpm

Req. viewer level: Everyone Limit entries to one per each user

Announcement: You can now get your landing rate predictions in! Type !predict <prediction> to add your pre

Landing Message: \$(username) won the landing rate competition! They guessed \$(prediction)fpm and were \$(

Your viewers can participate using !predict.

Setting	Description
Start competition at	When the competition automatically starts.
Req. viewer level	Who can participate in the competition.
Predictions must be between	The range of allowed predictions (to prevent trolling).
Limit entries to one per each user	Whether the users are only allowed one prediction. If this is enabled, when a user types !predict, their previous prediction will be overridden.
Announcement	The chatbot will send this in the chat when the competition activates.
Landing Message	The chatbot will send this in the chat when the competition is over and announce the winner.

Users can predict the landing rate with the following command:

!predict <landing rate>

For example, if the user types **!predict -301**, they've entered their prediction of **-301fpm** into the system.

Note: FlyLive Studio will **always** negate the landing rate, so **!predict 301** will result in **-301fpm** being added to the system.

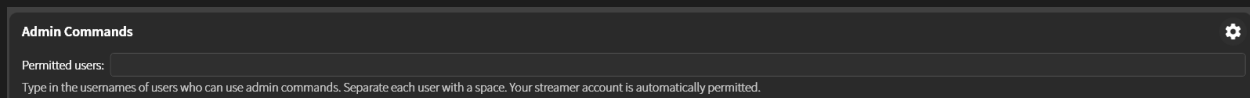
Admin Commands


Admin commands allow you and your moderators to control FlyLive Studio from the chat. For example, they can type:

```
!setflight EGLL CYYZ ACA123
```

... to set a flight from **London Heathrow (EGLL)** to **Toronto (CYYZ)** with callsign **ACA123**.

To add permit users to use admin commands, go to the **Admin Commands** page. Click the gear at the top right of the screen to show this box:



Admin Commands 

Permitted users:

Type in the usernames of users who can use admin commands. Separate each user with a space. Your streamer account is automatically permitted.

Now, type in the list of users you give permission to use the commands.

All admin commands can be enabled or disabled, and disabling one will prevent anyone, including you, from using it.

All admin commands also have an **alias**, which is a shortened version of the command for the ease of typing. For example, instead of typing:

```
!setflight EGLL CYYZ ACA123
```

... one could instead type:

```
!sf EGLL CYYZ ACA123
```

Both above commands do the same thing, with the only difference being one is shortened and easier to type.

Variables

Variables are objects you can place inside of your command response to add extra functionality to your chat bot.

The syntax for any command is:

\$(<command invoker> [parameters])

Global Variables

\$(n), 0 ≤ n ≤ 9

Selects the command's zero-indexed *n*th parameter.

Example:

!greet	Hello @\$\$(0)! Have some \$(1)!
Input	!greet John cookies
Output	Hello @John! Have some cookies!

\$(query)

Selects the command's first query. Has the same behavior as `$(0)`.

Example:

!thank	Thank you @\$\$(query)!
Input	!thank Bob
Output	Thank you @Bob!

\$(querystring)

Selects the command's first query, then returns an HTML-encoded value. Useful for `$(urlfetch)`.

Example:

!encodehtml	\$(querystring)
Input	!encodehtml https://www.google.com/
Output	https%3A%2F%2Fwww.google.com%2F

\$(user)

The username of the person calling the command.

Example: User 'Joe' types !hello into chat:

!hello	HeyGuys @\$ (user)
Input	!hello
Output	HeyGuys @Joe

\$(channel)

The name of the streamer's channel.

\$(urlfetch)

Downloads the data from a website.

Example:

!metar	\$(urlfetch http://metar.vatsim.net/?id=\$(query))
Input	!metar KSFO
Output	KSFO 180256Z 04011KT 10SM FEW170 SCT200 12/06 A3005 RMK AO2 SLP174 T01220056 56012

Flight Data Variables

Variable	Description
<i>\$(altitude)</i>	The current altitude of the plane <i>without a suffix</i> .
<i>\$(heading)</i>	The current heading of the plane <i>without a suffix</i> .
<i>\$(groundspeed)</i>	The current ground speed of the plane <i>without a suffix</i> .
<i>\$(airspeed)</i>	The current indicated airspeed of the plane <i>without a suffix</i> .
<i>\$(verticalspeed)</i>	The current vertical speed of the plane <i>without a suffix</i> .
<i>\$(latitude)</i>	The current latitude of the plane <i>without a suffix</i> .
<i>\$(longitude)</i>	The current longitude of the plane <i>without a suffix</i> .

<i>\$(departure)</i>	The ICAO of the departure airport.
<i>\$(departurename)</i>	The name of the departure airport.
<i>\$(arrival)</i>	The ICAO of the arrival airport.
<i>\$(arrivalname)</i>	The name of the arrival airport.
<i>\$(callsign)</i>	The callsign of the current flight.
<i>\$(dtg)</i>	How far the plane is to the destination in nautical miles.
<i>\$(flightdistance)</i>	The distance between the departure and arrival airport in nautical miles.
<i>\$(flightpercent)</i>	The % completion of the flight.
<i>\$(eta)</i>	The ETA of the flight in Zulu time.
<i>\$(ttg)</i>	How long it will be until you reach your destination.
<i>\$(simzulu)</i>	Your simulator's current Zulu time.
<i>\$(realzulu)</i>	The real-world current Zulu time.

Notification Variables

Landing Variables

\$(landingrate) The landing rate without a suffix.

Subscription Variables

\$(username) The user who subscribed.

\$(subtier) The tier of the subscribing user from 1 to 3.

\$(message) The user's sub message.

Resubscription Variables

\$(username) The user who subscribed.

\$(subtier) The tier of the subscribing user from 1 to 3.

\$(submonths) How many months the user has been subscribed for.

\$(submessage) The user's sub message.

Raid Variables

\$(username) The username of the raider.

\$(viewercount) The amount of raiders incoming.

Host Variables

\$(username) The username of the person hosting.

\$(viewercount) The number of viewers of the hosting person's stream.

Landing Rate Competition Variables

\$(username) The username of the person who won the competition.

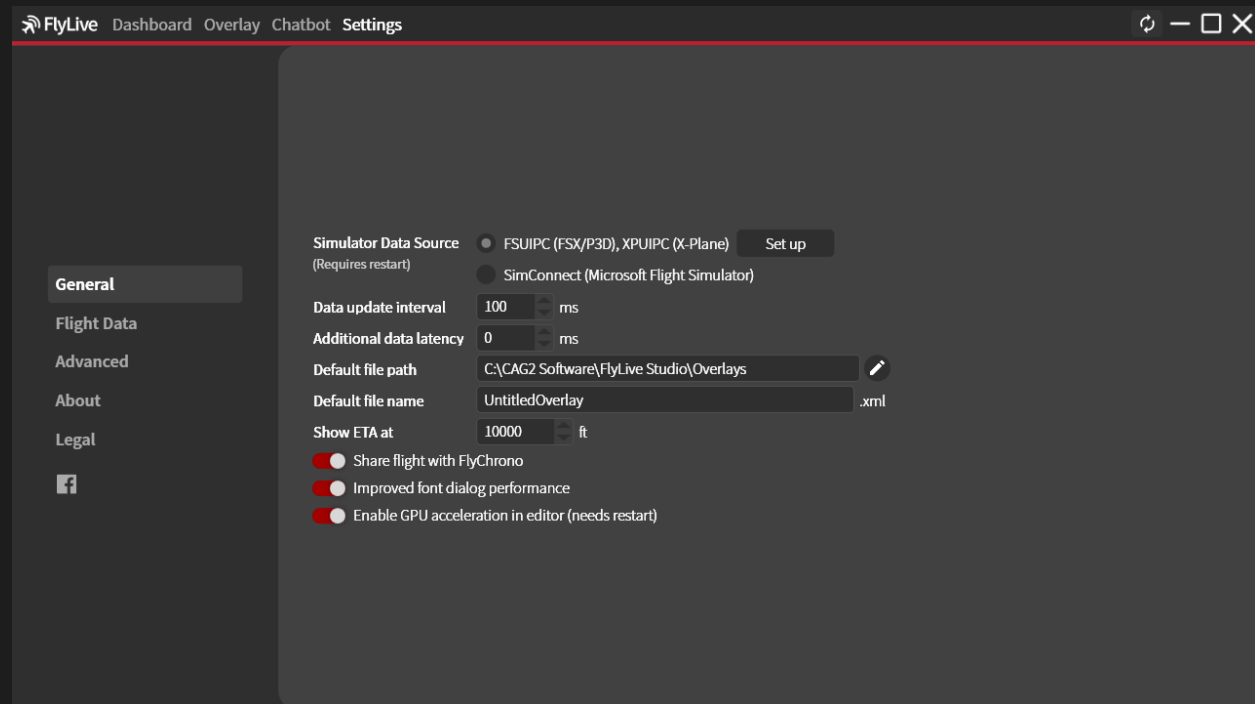
\$(landingrate) The landing rate without a suffix.

\$(prediction) The user's landing rate prediction.

\$(difference) The absolute difference between the prediction and the actual landing rate.

The Settings Page

To get to the settings page, click 'Settings' on the top bar.



This is a breakdown of what each setting does:

- **Simulator Data Source:** This is how FlyLive Studio gets the data from the simulator.
 - o For FSX/P3D, you need FSUIPC installed (the free version will suffice). For X-Plane, you need XPUIPC installed. You should choose FSUIPC/XPUIPC if you use these simulators.
 - o For Microsoft Flight Simulator, you should choose SimConnect, though FSUIPC works with it as well, albeit not as well.
- **Data update interval:** This is how often the flight data is updated in milliseconds. The minimum value is 50ms and the maximum is 1000ms. Lower values may decrease performance!
- **Data latency:** The added delay to the data on your overlay being updated. This can be useful if you use NDI to stream and you want to synchronize your simulator and overlay on the stream.
- **Default file path:** This is the default folder for where new overlays are saved.
- **Default file name:** This is the default file name for new overlays.
- **Show ETA at:** When FlyLive Studio should begin calculating your ETA.
- **Share flight with FlyChrono:** Whether FlyLive should share the current flight with FlyChrono. *For this feature to work, you must have this setting enabled in FlyChrono as well.*

- **Improved font dialog performance:** The font dialog is less informative but is much faster.
- **Enable GPU acceleration in editor:** Whether GPU acceleration should be used.
- **X suffix:** Sets the suffix that X flight data precedes. For example, an altitude suffix of 'ft' and your altitude was 36000, the final outputted text would be '36000ft'.
- **Show colon:** If this setting is checked, then a colon will be shown in between hours and minutes in all time-related data. For example, 16:00z vs 1600z.
- **Disconnected text:** This text replaces all flight data when the simulator is disconnected and replaces flight data that isn't available.
- **Hide FPS when not used:** Whether the FPS should be invisible instead of being displayed as n/a on your overlay.
- **Use simulator time for ETA:** Whether the simulator time should be used instead of the real zulu time for ETA calculations.
-

The following settings are **advanced** and should only be changed **only if necessary**:

- **Server Port:** This is the `http://localhost` port of the data transfer system which the server host will share with the AJAX client for necessary communications. An application restart and an overlay re-export **must** be preceded by the changing of this setting.
 - o You **should** change this setting if you usually have another application running on `http://localhost:8080/`.
- **Disable automatic version verification:** FlyLive Studio will check for discrepancies between the locally stored application version and the version given by FlyApps's servers through a **WebClient**, and it will take appropriate action upon discovering the aforementioned discrepancies. Turning this setting on is not recommended.
 - o You **should** change this setting if you are unable to access the website <https://flyapps.weebly.com> without a VPN.
 - o Weebly has blocked all incoming traffic from **Russia** and few other countries. If you are using FlyLive Studio in one of those countries, then you should turn this setting on.
- **Use legacy app updater:** Forces FlyLive Studio to use the legacy app updater.

Conclusion

I hope you found this documentation helpful. Please enjoy using FlyLive Studio to create your own stunning overlays!

Should you have any questions, please contact me here: <https://flyapps.weebly.com/>

You may also contact me by e-mail: markng989@gmail.com

LEGAL

All FlyApps applications are owned by FlyApps. By using these applications, you agree to be bound by a legal contract of which is displayed below.

1) COPYRIGHT

All images, text and files contained within this package (ie. content in the install package you downloaded, content contained within the install directory, content contained within the application itself) not owned by a third party are intellectual property of FlyApps. The images, text and files are protected by copyright and other intellectual property laws.

By installing and using software from FlyApps, you hereby agree you shall not copy, distribute, publish, modify or create derivative works from the content mentioned above. Additionally, you agree the content mentioned above shall not be utilized against FlyApps or be involved in any illegal activity. Same may not be used utilized in any commercial context, corporate entity or similar.

2) LICENSE

All FlyApps applications are freeware. It may be freely distributed given that the distributor does not: claim the software as their own, charge money for the software, impose further restrictions on the usage of the software, or does not distribute the file retaining this license. The utilization of the software shall not be in the amount of commercial use OTHER than revenue from video sharing/streaming services when used appropriately, and/or when used to accompany such activities. Should the user violate any of the above terms, the individual or group in question shall face legal charges. FlyApps reserves the right to have the final say in all appropriate circumstances.

3) WARRANTY, LIMTATIONS & DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Installation and usage of the applications is completely at your own risk. In circumstances, whereat damage, loss or charges are the indirect or direct consequence of your use of the software, the author shall not be liable for it.

This application uses the FSUIPC Client DLL for .NET by Paul Henty.