**This is Ivy, your new co-pilot.**

Gone are the days, when you could fly without transponder, enlighten the stratosphere with your landing lights and transform the passenger's meal in pottery to pottery in passenger meal upon landing, without anyone complaining.

Now that your airline has assigned your new co-pilot no mistake will go unnoticed. So make sure that you fly properly!

**Installation:**

You need the following to run Ivy:

* Python 2.7
* Sandy Barbour's Python Interface
* Pygame (for audio support)
* Copy the contents of this folder to X-Plane 11\Resources\plugins\PythonScripts

Python 2.7:

https://www.python.org/downloads/  
When given the possibility in the installation, enable adding Python 2.7 to the path environment variable.

Sandy Barbour's Python Interface

http://www.xpluginsdk.org/python\_interface.htm

Pygame can be installed via Python:

https://www.pygame.org/wiki/GettingStarted

Personally, I use the following commands to install pygame. Remember to set your Path environment variables first, so that python can be started from anywhere. Otherwise you must do this in your Python 2.7 folder:

*python -m ensurepip*

*python -m ensurepip --upgrade*

*pip install wheel*

*pip install wheel --upgrade*

*python -m pip install --upgrade pip*

*python -m pip install -U pygame --user*

If you need support installing pygame or python interface, please refer to the pygame or x-plane community. Do not PM me, I have rather high response times and I only know the issues I had myself.

**Soundpacks:**Just copy the Audio Folder + the new Ivy.ini file into your X-Plane 11\Resources\plugins\PythonScripts

If you do not want to lose your custom settings, you may skip copying the Ivy.ini file and adjust the mp3\_dir folder accordingly. This allows you also to swap between sound packs.

**Implemented Failure Detections:**

1. Bump on the ground
2. Tire blown
3. Hard braking
4. Transponder not active when airborne
5. Landing lights not on when close to the ground in the night
6. Landing lights not off on high altitude
7. Beacon lights not on when taxiing
8. Nav lights not on when airborne
9. Strobes not on when airborne
10. Battery low
11. Engine fire
12. Engine flameout
13. Engine ground failure
14. Engine airborne failure
15. Engine hot start
16. Battery not on
17. Aircraft Overspeed
18. Flaps Overspeed
19. Gear Overspeed
20. Stall
21. Cabin pressure raising too fast
22. Cabin pressure raising extremely rapidly
23. Bank angle pre-warning
24. Bank angle too high
25. Bank angle extremely high
26. Pitch down pre-warning
27. Pitch too low
28. Vertical G Force high
29. Vertical G Force very high
30. Vertical G Force very, very high
31. Vertical G Force too low
32. Vertical G Force negative
33. Barometric pressure not set accordingly while close to ground or taxiing (within tolerance)
34. Barometric pressure not set to standard above transition altitude
35. Ice airframe low
36. Ice airframe high
37. Ice pitot low
38. Ice pitot high
39. Ice propeller low
40. Ice propeller high
41. Ice cockpit window low
42. Ice cockpit window high
43. Cabin pressure low
44. Cabin pressure too low to breath
45. Birdstrike

Most variables needed to configure the tolerances of failure detection are editable in the **Ivy.ini** file.

**Landing evaluation:**

Most people only think about sink rate upon landing, however, your passengers will not fly with you again,

if the g-forces upon landing are too high. No matter what your vertical speed was.

Or to put it in other words:

A friend of my father was once happy like a little kid, because upon short runway on a greek island (I think it was Mykonos), he put his MD-80 with force to the ground (to ensure a no-flare situation), which resulted in serious pain in his back, broken ceramics of the passenger meals they had for their return flight, but the technician said after checking his data: *"No, this was not a hard landing."*

Well, the passengers might have other constraints than your technicians. Hence, the rating is the following:

*Rating A:*

Sink rate < 100 ft/min

Vertical forces < 1.5g

*Rating B:*

Sink rate < 250 ft/min

Vertical forces < 2g

*Rating C:*

Sink rate < 400 ft/min

Vertical forces < 3g

*Rating D:*

Sink rate < 500 ft/min

Vertical forces < 4g

*Rating F:*

Everything else that did not trigger the X-Plane crash detection

A proper landing requires you to **touch down and stop the aircraft** for more than **5 seconds**. An aircraft is considered stopped if the ground speed is below taxi speed (default: 5 kt). The rating includes **all bounces** within a 10 seconds window before your final touchdown.

Rating of your flight, depending on the errors you made:

0 Errors : Excellent

<5 Errors : Good (nice)

<10 Errors : Bad (not nice)

>=10 Errors : Horrible

Details of your highest sink rate and vertical g-forces are spoken upon landing (including all bounces in the evaluation)

As aircrafts are sometimes placed above the ground on loading and it is pretty annoying to get a landing evaluation in such an event, you need to be airborne at least 100ft while having a climb rate of >100 ft/min. This means that just hovering a bit and putting a helicopter back to the ground, also does not trigger a landing evaluation.

Every landing is stored in your IvyLogbook. The landing is **only stored**, if you wait for the landing evaluation. If you exit X-Plane before, it will not be saved.

**Implemented callouts for all aircraft:**

1. Gear down callout
2. Gear up callout
3. 60/80/100 knots callouts (setting via menu)
4. Positive rate of climb (default: 100ft/min)
5. Approaching Minimums (default: DH+100, DH must not be zero)
6. Fasten Seatbelts
7. Take Off Announcement on Non-Smoking Toogle or Commmand
8. Landing Announcement on Non-Smoking Toogle or Commmand

Remember that Ivy is a Union member and will only perform one take off and one landing announcement per flight. However, she might consider doing it on multi-leg flights. Unfortunately, some airplanes do not put seat belts or non smoking sign switches on the corresponding X-Plane Datarefs.

**Implemented callouts for specific aircrafts:**

* V-Speeds:
  + V1
  + VR
  + V2
  + V2 not achieved within 5 seconds after take off
* Flaps settings
* Slats settings

I supply multiple aircraft configuration files, but I can only implement and test them for aircraft I own. Auto V-Speeds are currently available for CL 300, Rotate MD-80 and the Freeware Do-328. Static V-Speeds are set in the Do-228 as I found an old manual on the web.

* Standard MD-80
* Baron B58
* Cessna 172 SP Skyhawk
* Cirrus personal jet
* King Air C90
* Stinson L5 Sentinel
* Bombardier Challenger 300 for XP 11
* Rotate MD-80
* Jetstream 32
* CRJ-200 (no v-speeds supported - not in datarefs)
* Twin Otter Version 2
* Freeware Douglas C-47
* VSKYLABS DC-3
* Standard B747-400
* ERJ-140
* Carenado Do-228
* Freeware Do-328
* Let-410

You can open the data for slats and flaps positions via menu or command and create your own configuration file if you like

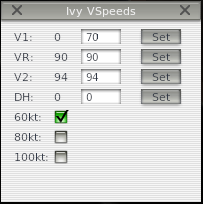
**Commands**

The following commands can be bound to your keyboard:

* *Ivy/cabin\_announcement*: Ivy will make a Take-Off or Landing announcement.
* *Ivy/say\_baro:* Say the current barometric pressure
* *Ivy/say\_wind*: Say wind direction and speed
* *Ivy/show\_output*: Show the flaps/slats position for creating *IvyAircraft\_X.ini*
* *Ivy/reset\_ivy*:Resets Ivy. Recommended for multi leg flights.

**V-Speed Callouts**

If your aircraft does not have datarefs for the v-speeds configured, you can access the v-speed settings via the V-Speed Dialog: Menu/Plugins/Ivy/Set V-Speeds



Here, you can set the individual v-speeds for your session. Zero values disable the affected callout. You can also enable/disable the callouts for 60/80/100 knots or set your decision height in case your aircraft does not support this. The 60/80/100 knots callouts are saved when you close X-Plane, as these callouts are not necessarily aircraft specific.

The definition of dataref based v-speeds or static v-speeds for individual aircraft is described in the Appendix.

**Loogbook**

Ivy remembers everything! She keeps precise tracking of all your mistakes and landings, noting every detail in your logbook. At least, most of it. You can open your loogbook in the plugins menu. You can also find the IvyLogbook.txt in your PythonScripts folder. In case you want to cancel a flight, you can simply edit the text file.

Arrival and departure airports are simply the next Airport Refs from your take-off/landing. There are certain runways in x-plane close to another airport, which might cause wrong airport names. This is a well known limitation of X-Plane and there is nothing I can do about it.

The landing is **only stored**, if you wait **20 seconds** after touchdown for the landing evaluation. If you exit X-Plane before, it will **not** be saved.

**What else is there to say?**

Remember that fun is subjective. If you don't like certain call outs, you can simply remove the individual audio file, if the end on "\_#.ogg". If you don't like the plug-in at all, go write your own.

All speech was generated using the Amazon Polly Text-to-Speech synthesis engine. You may generate your own sound files, if you want more proper call outs. You just need an AWS account, which is currently free of charge or any other speech synthesis software. However, I hereby deny the use of any speech that contains sexism, racism or fascism (there are always some idiots out there).   
  
Amazon offers a variety of voices and it is definitely on my ToDo list to generate different voice packs. Ivy is just the most funny voice that I decided to start with.

This software is published under the GNU General Public License v3. Remember that this gives you no warranty for functionality and by using this software, you yourself take the full responsibility for any fatalities caused by any bugs.

This software was **not** written by a professional pilot. It does **not** follow any real life procedures and is **not** safe for flight training. If you cause a fatal crash, because you followed Ivy's suggestions, we might consider your nomination for the Darwin Award.

Many animals were hurt during the creation of this product. Deere were hit on the runway, birds were soaked into the engine. Most of them are better now. Even though the turkey was too well done. Not all virtual pilots yet recovered from the injuries of countless crashes that were used to train Ivy's supervision talents. As freeware does not produce any income, we cannot afford to pay them a doctor. Yes, you should feel bad about that!

**References**

The passenger screaming was taken from freesound.org under the attribution license:  
https://freesound.org/people/InspectorJ/sounds/421852/

The other sounds were also taken from freesound.org, where all chosen sounds were using the creative commons 0 license.

**Appendix**

**How to create an IvyAircraft.ini:**

The aircraft definition is still very basic. There is no automatic type detection for datarefs. Flaps/Slats datarefs are floats, so is their value. Positions must be integer (Ivy say numbers does not support floats). V-Speed datarefs must be integer. There is a workaround in place for the Rotate MD-80, which uses an array of floats for the v-speeds.

Everything that is not defined in your .ini file will be filled with default settings, i.e., aircraft specific settings are all disabled by default. Hence, you only need to enter the stuff you need. So for example if your aircraft has no datarefs for V-Speeds, just skip it. You do not even need to disable it.

There are two ways to find the configuration parameters that are necessary for individual aircraft configuration. The first one is to use the x-plane dataref editor plug-in (recommended). However, if you do not want to bother with additional plug-ins, Ivy should give you the tools you need for flaps and slats. There are no defaults for V-Speeds, so you really have to search for them in the dataref editor or ask the author of the aircraft.

If you enable the Ivy Output from the plugins menu, the first row gives you the aircraft name defined in the .acf file plus the tail number. If no name or tail number is defined, you might want to edit your .acf file or use plane maker to enter a valid aircraft name. The second row gives you the aircraft name of the .ini file loaded. If everything works well, this is the string you entered in your .ini file. If no .ini file is loaded, "UnconfiguredIvyAircraft" will appear.

The data entries following will only show data, if they are configured and enabled in the .ini file. So if your flaps dataref is wrong, no data will appear in the output.

Having to restart X-Plane after each .ini file change can be pretty cumbersome. Consider reloading the python scripts using the python interface control window that can be found in the plug ins menu. However, **certain things like NOAA weather plugin or having numpy installed might crash x-plane when reloading scripts**.

**Flaps Settings**

1. You need to create an IvyAircraft\_#.ini file or copy an existing one. Any number between 1 and 99 is sufficient, no leading zeros.

2. The first line always needs to be "[IVY\_AIRCRAFT]".

3. You need to define the aircraft name. It is not necessary to enter the full name, a part that is unique is fully sufficient. If there is no aircraft name, use the tail number. I look in both, as some aircraft authors tend to enter just one of them.

4. Enable the flaps datarefs by entering flaps\_enabled = true

5. Define the dataref that tells you the flaps settings. A good starting point is the flaps handle or the flaps deploy ratio

The following is an example of the most basic aircraft config:

[IVY\_AIRCRAFT]

aircraft\_name = Bombardier Challenger 300

flaps\_enabled = true

flaps\_data\_ref = sim/flightmodel2/controls/flap1\_deploy\_ratio

Alternatively, you might also want to consider:

flaps\_data\_ref = sim/cockpit2/controls/flap\_handle\_deploy\_ratio

Some more complex aircraft use multiple flaps, then you have no other choice than using the flaps handle position.

Now load the aircraft having the name you entered in aircraft\_name and enable the Ivy output. If the IvyAircraft is not called "UnconfiguredIvyAircraft", an .ini file has been loaded correctly. Now, the flaps output will show you the dataref value that corresponds with your flaps position (if you have enabled them). Note the positions down, 2 digits after decimal point should be sufficient. You also need to define a tolerance, here I would suggest "0.03" as a starting point.

Now, the final flap settings file might look like this:

[IVY\_AIRCRAFT]

aircraft\_name = Bombardier Challenger 300

flaps\_enabled = true

flaps\_data\_ref = sim/flightmodel2/controls/flap1\_deploy\_ratio

flaps\_tolerance = 0.03

flaps\_value\_1 = 0.3333

flaps\_position\_1 = 10

flaps\_value\_2 = 0.6666

flaps\_position\_2 = 20

flaps\_value\_3 = 1.0

flaps\_position\_3 = 30

Slats work the same way. If you use the flaps handle position, just enter the same dataref for slats and flaps. Enter the value for the slats positions only for the slats, otherwise you will get two annoucements for flaps and slats.

You can define up to 10 flaps and 10 slats positions.

**V-Speeds**

**Constant V-Speeds**

You can simply enter a static V-Speed using the following entries.

v1\_static = 110

vr\_static = 120

v2\_static = 130

**No need to enable anything**, the enable entry only enables dataref scanning.

**V-Speeds via Datarefs**

V-Speeds are limited to integer datarefs at the moment. You need to enable v-speed datarefs and define the v-speed dataref for your aircraft:

vspeeds\_enabled = true

v1\_data\_ref = cl300/refspds\_v1

vr\_data\_ref = cl300/refspds\_vr

v2\_data\_ref = cl300/refspds\_v2

**Ivy.ini Explained:**

Your Ivy.ini files always has to start with [IVY\_SETTINGS] as the first line. Python Configparser requires you to define a section for your config, even if you just have a single one. There are no comments within the file, because then every string is considered to contain spaces until the '#'.

[IVY\_SETTINGS]

mp3\_dir = IvyMP3s # Name of your MP3 Folder. May differ for different Soundpacks

pos\_rate\_climb = 100.0 # Minimum climb rate for the positive rate call out

ivy\_ouch\_g = 2.0 # Minimum vertical g-force that will trigger ivy to call ouch when

# hitting a bump on the ground

brake\_max\_forward\_g = 0.5 # Maximum forward g-force that your passengers consider comfortable

alt\_landing\_lights\_low = 1000.0 # During the night, landing lights must be on below this altitude

# (radar above ground)

alt\_landing\_lights\_high = 10000.0 # Landing lights must be off, above this barometric altitude. 3000ft

# radio alt are considered, if you start from > 7000ft (e.g. VILH)

night\_world\_light\_precent = 0.5 # What is a night? When the street lights are above this threshold

taxi\_ground\_speed\_min = 5.0 # Minimum speed that is considered taxiing

vis\_is\_fog = 5000.0 # Below this sight, we have fog

cab\_rate\_low = -1500.0 # Cabin pressure change rate that is not comfortable anymore

cab\_rate\_high = -2500.0 # Cabin pressure change rate that causes pain

cab\_rate\_reset\_hysteresis = 100.0 # Hysteresis for error reset

bank\_reset\_low = 15.0 # Bank angle that will cause a low error to reset

bank\_low = 32.0 # Bank angle that will give a warning

bank\_high = 37.0 # Bank angle that is considered an error

bank\_xhigh = 45.0 # Bank angle that is way off limits

pitch\_reset\_low = -5.0 # Pitch angle to reset lower error

pitch\_low = -10.0 # Pitch angle for pre-warning

pitch\_high = -20.0 # Pitch angle that is considered an error

max\_g\_down\_low\_reset = 1.5 # Vertical g to reset lower error

max\_g\_down\_low = 2.0 # Vertical g too high

max\_g\_down\_high = 3.0 # Veritcal g very high

max\_g\_down\_xhigh = 5.0 # Vertical g extremely high

trans\_alt = 18000.0 # Transition altitude (for setting standard pressure)

trans\_hysteresis = 1000.0 # Hysteresis for barometric error

baro\_tolerance = 3.0 # Tolerance for barometric barometric settings (+/-)

baro\_alt\_low = 3000.0 # Radio altitude, below you are required to set correct pressure

ice\_low = 0.05 # Icing first error threshold

ice\_high = 0.2 # Icing second error threshold

cab\_press\_low = 13000.0 # Cabin altitude for first error

# (afaik, this is the rl limit for safe operation)

cab\_press\_high = 20000.0 # Cabin altitude where it becomes dangerous

non\_smoking\_annoucetime = 3.0 # Time for toggling non-smoking to signal an annoucement

decition\_height\_arm = 500.0 # Radio alt that must be achieved to arm the decision height call out

decition\_height\_plus = 100.0 # Offset for the decision height call out (typically DH+100ft)

log\_window\_pos\_x = 300.0 # X Position for the Logbook Window

log\_window\_pos\_y = 550.0 # Y Position for the Logbook Window

log\_window\_height = 350.0 # Height of the Logbook Window

log\_window\_width = 1000.0 # Width of the Logbook Window

log\_window\_entries = 16.0 # Number of entries that fit into the Logbook Window

log\_afc\_name\_length = 40.0 # Limit for the aircraft name length