

FSUIPC 4.959 changes since 4.95

1. FSUIPC now automatically adds comments into the settings file (FSUIPC4.INI) to show the decoded control numbers and key presses used there. This can be turned off by changing the [General] parameter Annotate=Yes to No.

The action is applied on loading to each [Axes], [Buttons] and [Keys] sections, and those in Profiles too when they are used. The annotation is generated from the encoding currently in the file and will reflect changes made manually as well as via the Settings dialogue.

The text added is enclosed with -{ and }-, and follows any user comments already there, so do not use these in your own comments. The opening -{ is preceded by a tab character to make tidy formatting to view and review easier to achieve in an editor.

NOTE: A bug in this facility, whereby annotations made for buttons would be deleted if the relevant joystick controller is disconnected, is fully fixed in 4.958.

2. A bug is fixed in the UseProfiles=Files facility when the filed settings for a Profile have not yet been created for Axis assignments and/or Calibrations.

Without the separate files option it is perfectly possible to have a Profile for an aircraft or aircraft group which only had separate Button and Key assignments, with Axis and Calibration settings being generic. However, because of the bug, now fixed, if this setup applied, the generic settings for Axes and Calibrations would not be read and any changes made would be added to the generic settings *unless* the Use Profile option was selected first.

This arose in part because the Axis and Calibration settings for an aircraft are either only generic or only specific. They cannot be mixed in the way that button and Key assignments can be.

The corrected version works as it should, with no difference in action for UseProfiles=Files mode (excepting, of course, that any specific changes go to the separate files).

3. The programming facilities added recently to allow L:vars (local panel variables) to be read and written from an external program interfacing to FSUIPC are extended as follows:

The offset providing or receiving the value of the L:Var is placed into offset 0D6C, currently defined as a DWORD (4 bytes, or 32-bits). The offset only needs the lowest (first) 16 bits (2 bytes). The "high" word is now used to provide the TYPE of the variable supplied or required after reading. FSUIPC will do any conversions needed.

The values which can be written to 0D6C, providing both the offset ("nnnn") and the type are:

- 0x0nnnn for 64-bit double (as before)
- 0x1nnnn for 32-bit float (FLT)
- 0x2nnnn for 32-bit signed integer (SD)
- 0x3nnnn for 32-bit unsigned integer (UD)
- 0x4nnnn for 16-bit signed integer (SW)
- 0x5nnnn for 16-bit unsigned integer (UW)
- 0x6nnnn for 8-bit signed integer (SB)
- 0x7nnnn for 8-bit unsigned integer (UB)

With reads into a fixed point value (the last 6 above), the floating point value provided from the Gauge system is rounded to the nearest integer (up for positive numbers, down for negative).

This change allows more scope for reading multiple values into the limited offset space allowed (66C0—66FF only allows up to 8 doubles at a time. Now you can read up to 64 one byte values, or any mix between them.

As a clarification, remember that the read/write operation is immediate, not asynchronous, so that in one Process call a series of writes to 0D6C and 0D70 can be made followed by one read of the whole area, (66C0-max used) to get the results.

Note that if the offset provided is outside the range allowed (66C0-66FF), or so near the end of the range that the value would not fit, the request is simply ignored.

4. The installer now places the FSUIPC4_Loader into the FSUIPC Documents subfolder, where it can be copied into the main modules folder if really necessary. It is not packaged in the download ZIP so that users are not encouraged to automatically install it into Modules. It is hardly ever needed, and can give problems of its own.
5. The installer places the FSUIPC4 entry in the DLL.XML file before any as_btstrp.dll entry (for ASN), which should always load after FSUIPC4.
6. It appears that certain programs, interfacing to FSUIPC, write to the offset 3400, which controls logging options. This can cause lots of unwanted log entries, and create very large log files. One of the culprits is the driver for Flight Deck Solutions hardware – “interfacelT MCS”.

This FSUIPC update stops this unless the User allows it to happen. This is via the new [General] section parameter:

LogOptionsProtect =Yes

If you need to allow programs to change the logging, change this to ‘No’ before running FS.

Note: Version 4.954c corrected the logging which incorrectly said the options were “BLOCKED” even when legitimately changed by the user in the Settings Tab.

7. The lua library function **com.gethidbuttoncount** had a bug in it which caused it to always return zero instead of the correct number of buttons.
8. The Installer now provides tidier options for dealing with cases where a version of FS or P3D is installed according to the Registry, but isn’t found where it says it should be. The options are:
ByPass: simply ignore the fact that the Registry gave this information,
Fix: remove the invalid registry entries so that it doesn’t happen again, or
Find: ask the user to find the program in an Explorer Window. This is the option which always applied before this release. There’s an option then given to change the path given by the Registry, to mach. This choice can be aborted by pressing “Cancel” in the Explorer window, which makes it like “ByPass”.
9. Version 3.3 of Prepar3D is now supported. The installer recognises versions up to 3.3.5.
10. A bug is fixed in the new INI comment-adding facility which could cause a crash early during the loading process. This was incorrect handling of missing Macro files or entries therein.
11. Macro assignments to Axes were being incorrectly labelled as “missing” in the INI annotations because the macros hadn’t been loaded beforehand. That is now fixed.
12. Fixed the old FS98-style weather offset access (0F1C-0F8C and 04D2-04D4) so that writes to them always generate a GLOBal mode weather setting action into FSX / P3D. Previously this only occurred if a different value was written from a previous write.

13. Offset 0245 (Day of Month) was incorrectly being set as "Day of Week". This is a very old error, dating back almost to the start of FSUIPC4, but is now fixed.
14. A diagnostic facility to make FSUIPC avoid all weather reading and writing activities has been added. This is invoked by adding "**NoWeatherAtAll=Yes**" to the [General] section of the FSUIPC4.INI file.
15. Axis range assignments to compound key presses (i.e. shift+ etc) were getting altered to just the main active key when the INI file was saved, as when exiting FS or making other changes. Fixed in 4.955h.
16. In P3D version 3, the controls ("events") lists provided for assignments in FSUIPC were previously missing quite a number, due to the re-organisation of the CONTROLS.DLL tables from one integrated table into three separate tables. This also applied, to a lesser extent, to P3D version 2.5, but both are fixed now in FSUIPC 4.955i.

A fully revised controls list showing differences in the controls on offer in FSX, P3Dv2 and P3Dv3 is in preparation and will be included in the next full Installer Release.
17. The controls listed in the assignments dropdown were missing quite a few when FSUIPC is used with P3D versions 2 or 3. The improved lists now cover a much larger range of assignable FS controls. A revised Controls List is now supplied which lists every one, with colour highlighting to show differences between FSX and P3D versions.
18. A new option:

TimeForLuaClosing=1

Is available in the FSUIPC4.INI [General] section. This sets the time allowed for Lua plug-ins to close correctly before FS is closed. The time starts as soon as the **event.terminate** function has been called, assuming the plug-in uses it.

The value is in seconds and can be set from 1 to 20. A longer period might help those more complex plug-ins, like LINDA, to tidy up properly and this in turn might help avoid some of the reported crashes of FS on closedown.

With effect from version 4.959, this parameter can be set to 0 in order to automatically and immediately forcibly terminate all running Lua plugins at the end of the session. This is to cope with the odd case where some action has been started by the plug-in, whether Network, Internet or device related, which won't terminate easily or tidily.

19. Some initialisation messages in the log were inadvertently omitted in some recent incremental releases. They are now reinstated.
20. An offset method of setting friction values, similar to that offered by the **ipc.SetFriction** function for Lua plug-ins, is added. This uses a 12-byte array of offset spaces, as follows (tables below give the values to be used):

7360	4 bytes	32-bit float value (FLT) to be written*
7364	1 byte	The "class" value, 0-6
7365	1 byte	The surface type, 0-24
7366	1 byte	The direction, 0 or 1
7367	1 byte	The condition, 0-3
7368	4 bytes	32-bit float value (FLT) giving the previous value (read only)

* The complete 8 bytes, 7360-7367 must either be written at once, i.e. as a single structure, or the FLT value must be written to 7360 last. It is that write which triggers the action.

The saved original frictions can be restored by setting the Class value in 7364 to 255 (0xFF) then just writing anything to 7360.

The 4 single byte values are as follows:

Class:	0	BRAKE		4	LOAT	
	1	WHEEL		5	WRUDDER	
	2	SCRAPE		6	SKI	
	3	SKID				
Surface:	0	CONCRETE	13		CORAL	
	1	GRASS	14		GRAVEL	
	2	WATER	15		OIL_TREATED	
	3	GRASS_BUMPY	16		STEEL_MATS	
	4	ASPHALT	17		BITUMINUS	
	5	SHORT_GRASS	18		BRICK	
	6	LONG_GRASS	19		MACADAM	
	7	HARD_TURF	20		PLANKS	
	8	SNOW	21		SAND	
	9	ICE	22		SHALE	
	10	URBAN	23		TARMAC	
	11	FOREST	24		WRIGHT_FLYER_TRA	
	12	DIRT				
Direction:	0	ROLLING	Condition:	0	DRY	
	1	SLIDING		1	RAIN	
				2	ICE	
				3	SNOW	

21. Traffic scanning for the FSUIPC TCAS tables (and Traffic Zapper) is now performed in a separate SimConnect Client connection, which hopefully will be more reliable.

When this change was first implemented, it prevented FSUIPC client programs deleting AI traffic by ID. That problem is fixed in 4.958.

22. Version 3.4 of Prepar3D is now supported.

23. The Lua + WideClient facility to divert SimConnect text menus from the FS screen to a Client computer did not work correctly with P3D version because of a change in the name used for the Window being used. This is now fixed. However, because P3D now treats the screen display in much the same way as FSX does, it is not possible to hide or move the sim display of the menu out of view completely. FSUIPC does its best, making the window smaller and moving it top left, hopefully out of direct sight from your flying position. This is much like it is on FSX.

24. Version 4.957b fixes an error which, in P3D 3.4 only, caused a crash if the "Fix control acceleration" option is enabled.

25. Lua global variables, set and read by **ipc.set** and **ipc.get** library functions, are now shared over the Network between all connected WideFS clients and FSUIPC, providing all the PCs involved are in the same Windows Workgroup. This needs the WideClient version in use on the client PCs to be updated to at least version **6.999z1**.
26. Prepar3D version **3.4.18** is supported by this release.
27. Two parameters are provided to help overcome some problems, especially on some Prepar3D releases, where reading AI data for TCAS displays can upset other add-ons such as sophisticated GPS based gauges. The Mindstar Garmin emulations are a particular example.

These are:

ProvideAldata=Yes (default). Set to No to stop FSUIPC reading any AI traffic data.

ProvideAlairports=Yes (default). Set to No to just stop FSUIPC reading departure and destination airport data.

28. A **Traffic Limiter** option is now provided. This is enabled by parameters in the INI file, or more easily, by setting the desired limit in the edit box for this on the lower right corner of the Miscellaneous options tab.

Traffic is limited by removing AI aircraft, from whatever source, as they are added, once the limit is reached. The aircraft being deleted is not necessarily the last one added. It will usually be the furthest one -- Traffic is only spawned within the "reality bubble" (about 80 nm radius) so that's as far as it might get.

The behaviour is modified by parameters in the INI file. There will be a new section, probably added at the end of the file, containing, by default:

```
[Traffic Limiter]
TrafficLimit=0
GroundPreference=50
AirportPreference=50
NearerPreference=50
```

The limit set to 0 switches the facility off. The other parameters determine how to decide which ones to delete when the limit is breached, These are:

GroundPreference: This is the % probability of nearby (0.5 to 20nm) ground traffic being deleted provided they do not already have clearance, or are shutting down after arrival in any case. Increasing this will give better frame rates and more free parking spaces at the local airports, but can of course ultimately limit the number of aircraft taking off. Decreasing it will delete fewer local ground aircraft, with none being deleted if this is set to 0. (I increase it to about 70 which still gives me enough local traffic and allows me to decrease the next parameter without spoiling the performance).

AirportPreference: The gives the % probability of aircraft destined for one of the six nearest airports (including the one you may be on) being deleted if it is a candidate (i.e. furthest, or as determined by the other parameters). The default of 50% makes this an even chance, other factors allowing. Increasing this will give less arrivals locally, decreasing it will give more. At 0 no local arrivals will be deleted unless there's no other choice to keep to the limit. (I like seeing the arrivals so I take this down to about 20).

NearerPreference: This is the probability of deleting aircraft which are not necessarily the furthest away. With it set to anything above 0 the aircraft chosen for deletion might be anything from 20nm away. The idea of this is to give newly added aircraft in flight, spawned as usual at the far reaches of that “reality bubble”, a chance to follow their plans. Otherwise what is likely to happen, after the initial period where the aircraft are spawned all over the place (according to their schedules), almost every scheduled flight from anywhere outside that bubble would probably be deleted as soon as it was added. (I have this set to 70 as I like to see the greater variety of airlines and departure airports I get when newly spawned flights are not deleted so often).

The defaults are set at 50% so you can see the “average” results, and then increase or decrease them to meet your preferences. To allow experimentation here, this complete section of the INI file is re-read whenever you enter the FSUIPC Options screen and OK out (even if you don’t visit any tabs).

If they are all set to 0 (the original defaults in a pre-release) then the only criteria used when deciding which to delete is the distance – furthest only, every time.

29. The recently added mechanism to broadcast Lua global variables to WideFS clients has been made optional, but enabled by default. To turn it off (perhaps for some critical timing reasons), change the parameter

WideLuaGlobals=Yes

Which you will now find in your FSUIPC4.INI file in its [General] section, to a ‘No’.

In a future version, probably the first one in 2017, any timing concerns will be alleviated by putting the broadcast and reception actions into threads of their own.